Covid-19 Vaccine Analysis

Team members:

SS Ameesha

Gowri Suresh

Nima P

DeviChandhana A

Phase 2 Submission Document

Project: Covid-19 Vaccine Analysis

Introduction:

* **Covid-19 Vaccine Data Analysis** aims to provide insights into the efﬁcacy and safety of the vaccines. We will analyze data from clinical trials, real-world studies, and adverse event reports to provide a comprehensive overview of the vaccines.
* **Covid-19 Vaccines** are designed to prevent infection or reduce the severity of the disease. They work by stimulating the immune system to recognize and ﬁght the virus. There are currently several types of vaccines available, including mRNA, viral vector, and protein subunit
* Data analysis is used to evaluate the efﬁcacy of vaccines. We will examine the statistical methods used to analyze vaccine efﬁcacy data, including the calculation of vaccine efﬁcacy rates and conﬁdence intervals.
* **Future Developments** in Covid-19 vaccine research include the development of new vaccines, booster shots, and the study of long- term vaccine efﬁcacy and safety. Ongoing research is essential to ensure that we have effective tools to combat the pandemic.

Content for Project Phase 2 :

For analyzing data, we need some libraries. In this section, we are importing all the required libraries like pandas, NumPy, matplotlib, plotly, seaborn, and word cloud that are required for data analysis. Check the below code to import all the required libraries.

Data Source

A good data source for covid-19 vaccine analysis should be Accurate, Complete, Covering the geographic area of interest, Accessible.

Dataset Link: (https://www.kaggle.com/datasets/gpreda/covid-world-vaccination-progress/)

# Data Collection and Preprocessing:

* Data collection is a critical component of vaccine evaluation. We will examine the different types of data collected in clinical trials, including safety data, efﬁcacy data, and immunogenicity data.

# Data preprocessing: Analyzing the effect of data preprocessing techniques using machine learning algorithms on the diagnosis of COVID-19

# Exploratory Data Analysis ( EDA ):

# Exploratory Data Analysis (EDA) is a field of data analysis used to visually represent the knowledge embedded deep in the given data set. The technique is widely used to generate inferences from a given data set. Data set of current pandemic, the COVID-19 is widely made available by the standard dataset repository. EDA can be applied to these standard dataset to generate inferences. In this paper, data visualization technique is applied to the dataset and is used to formulate patterns for better insights on the effects of the pandemic with respect to the variables/ labels given in the dataset. A Web application tool called Jupyter Notebook is used to generate graphs using python language as it consists of libraries which are used for the process of EDA and the visualization is depicted for the attributes showing higher correlation. Based on the graphs obtained, we can draw conclusions from the current situation based on the data available, understand why a certain variable is increasing/decreasing with respect to another and what can be done to improve the drawbacks found.

# Feature Engineering:

The outbreak of the COVID-19 pandemic has also triggered a tsunami of news, instructions, and precautionary measures related to the disease on social media platforms. Despite the considerable support on social media, a large number of fake propaganda and conspiracies are also circulated. People also reacted to COVID-19 vaccination on social media and expressed their opinions, perceptions, and conceptions. The present research work aims to explore the opinion dynamics of the general public about COVID-19 vaccination to help the administration authorities to devise policies to increase vaccination acceptance. For this purpose, a framework is proposed to perform sentiment analysis of COVID-19 vaccination-related tweets. The influence of term frequency-inverse document frequency, bag of words (BoW), Word2Vec, and combination of TF-IDF and BoW are explored with classifiers including random forest, gradient boosting machine, extra tree classifier (ETC), logistic regression, Naïve Bayes, stochastic gradient descent, multilayer perceptron, convolutional neural network (CNN), bidirectional encoder representations from transformers (BERT), long short-term memory (LSTM), and recurrent neural network (RNN). Results reveal that ETC outperforms using BoW with a 92% of accuracy and is the most suitable approach for sentiment analysis of COVID-19-related tweets. Opinion dynamics show that sentiments in favor of vaccination have increased over time.

Advanced Regression Techniques:

Linear Regression: In linear regression, two variables are employed: one is the dependent variable (plotted on the y-axis) on which the prediction is based, and the other is an independent variable (plotted on the x-axis) utilized to make the prediction. Variable-based prediction might be univariate (based on one variable) or multivariate (based on several variables) ([Moore et al., 2013](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8877421/#B34)). A regression line is a straight line that explains how the dependent variable changes with the change in the independent variable. To contrast the model predictions against several sets of field data, we use vaccine dose data to calculate the number of COVID-19 cases and the number of people dying considering the above factors. We fit the model with

Regression line,y=mx + c,

(1)

where c is an intercept and m is the slope and y is the dependent variable and x is the independent (explanatory) variable **r2**is the coefficient of determination which is calculated by Karl Pearson’s coefficient (r) ([Calkins, 2005](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8877421/#B13)). This coefficient indicates how many variations are explained by the variable being predicted. The greater the slope, the greater the correlation between the variables, and the greater the ability to explain fluctuations in other variables. Linear regression improves prediction since it focuses on situations with one or more predictor variables (in our study, vaccine data for First Dose and Second Dose) and one outcome variable ([Marill, 2004](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8877421/" \l "B30)).

Multiple linear regression,y=ax+bz+c,

(2)

where a and b are coefficients of regression, c is the intercept while having x and z as multiple explanatory variables.

The outcome variable, y, is a linear function of each predictor variable, x, and z, forcing the regression model to be a straight line ([Marill, 2004](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8877421/" \l "B30)). For a good model, the r-value should lie in 0.5 to 1.0 so this score gives a good correlation and a good predictor. Regression analysis is also used to predict the p-value for significance testing. The statistical inference approach is based on a complex network that includes assumptions about how data was gathered and analyzed and how the research results were presented.

### **RMSE and r2-Value**

The RMSE is the square root mean error. This error value gives an idea about the fitness of the model, i.e., how the values deviate from the true value. RMSE is an absolute measure of fit, while R-squared is a relative measure of fit. RMSE can be interpreted as the standard deviation of the unexplained variance since it is the square root of the variance. It has the advantage of being in the same units as the answer variable. The lower the RMSE value the better will be the prediction. If the main goal of the model is prediction, the RMSE is the essential criterion for fit because it is a valid standard of how well the model predicts the response.

Mean Square Error=True value−Predicted value

(3)

Root mean square error=(True value−Predicted value)2

(4)

We square the error because the estimate can be above or below the true value, resulting in a negative or positive difference. If we didn't square the errors, the sum might fall due to negative differences rather than a strong model fit. Lower values of RMSE indicate a better fit.

* Support vector machine : Calculating the accuracy of a model-designed support vector machine (SVM) algorithm is a good measure. It is a supervised learning algorithm that classifies data into 2 classes based upon which training is done and then, using that, future learning classifications are made. These algorithms are more efficient as their performance is high. Using SVM, a hyperplane can be plotted between datasets which are called a decision boundary. Based upon that classifier, classification can be performed. This is an advanced version of the linear and polynomial regression model analyzed above. By using SVM we can make some predictions and these predictions can be compared with the actual values and in the last, the accuracy of the model can also be obtained ([Bruno, 2017](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8877421/#B10)). SVM regression analysis will complete our analysis and tell us about the efficacy of the model and decrease the error value of the model by making it more precise.Model Evaluation and Selection:

Split the dataset into training and testing sets.

Evaluate models using appropriate metrics (e.g., Mean Absolute Error, Mean Squared Error, R-squared) to assess their performance.

Use cross-validation techniques to tune hyperparameters and ensure model stability.

Compare the results with traditional linear regression models to highlight improvements.

Select the best-performing model for further analysis.

# Model Interpretability:

The coronavirus disease 2019 (COVID-19) pandemic has posed a severe threat to global health and economy while producing some of the richest data we have ever seen in terms of infectious disease tracking. The quantity and quality of data placed epidemic modeling and forecasting at the forefront of worldwide public policy making. Compared to previous infectious diseases, COVID-19 shows special transmission characteristics, yielding significant fluctuations and non-stationarity in the new COVID-19 cases. This poses grand challenges in effective prediction, and, on the other hand, draws attention of the global community to epidemic tracking and prediction..

# Deployment and Prediction:

Deploy the chosen regression model to covid-19 vaccine analysis.

Develop a user-friendly interface for users to input property features.

Program:

### Covid-19 Vaccine Analysis

### IMPORT LIBRARIES

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import plotly.express as px

import plotly.graph\_objects as go

import matplotlib.patches as mpatches

from plotly.subplots import make\_subplots

from wordcloud import WordCloud

import seaborn as sns

sns.set(color\_codes = True)

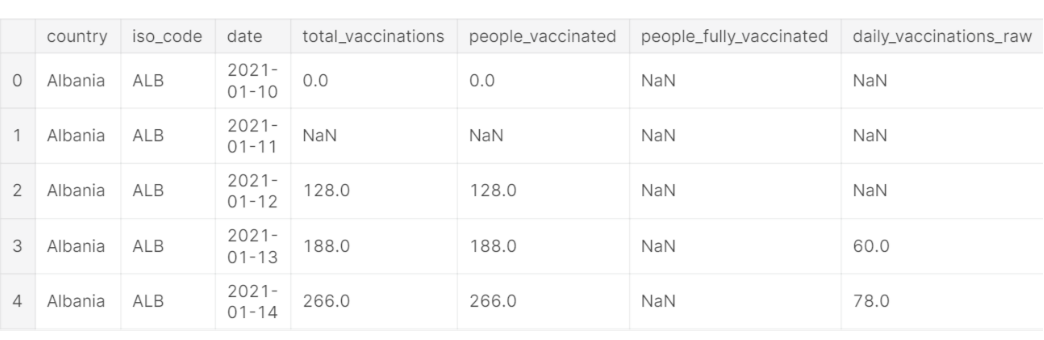
sns.set(style="whitegrid")

import plotly.figure\_factory as ff

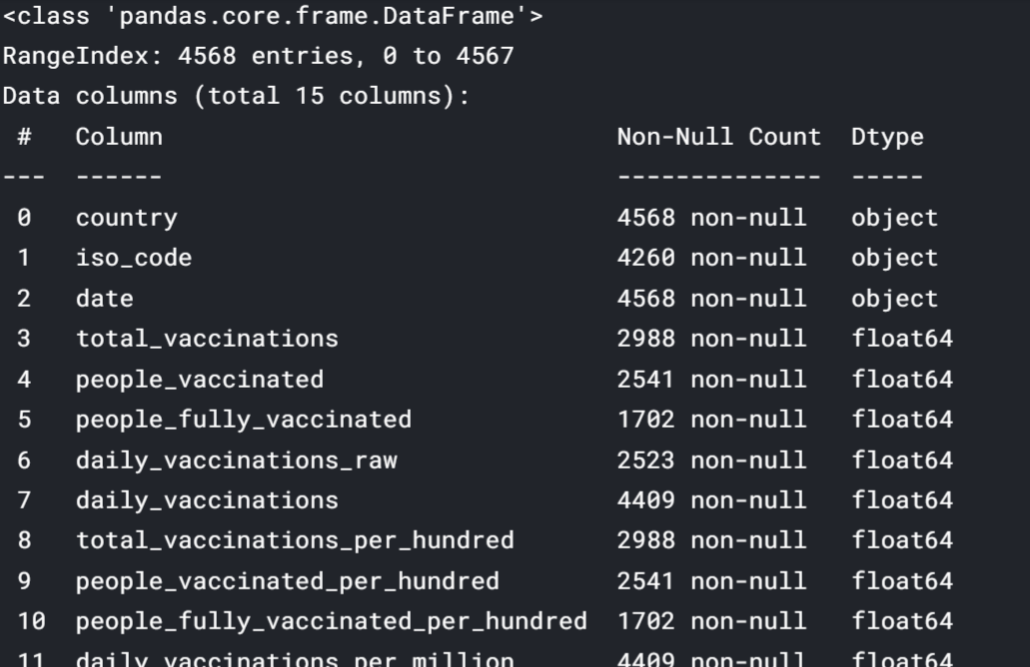
from plotly.colors import n\_colors

### READ DATA AND BASIC INFORMATION

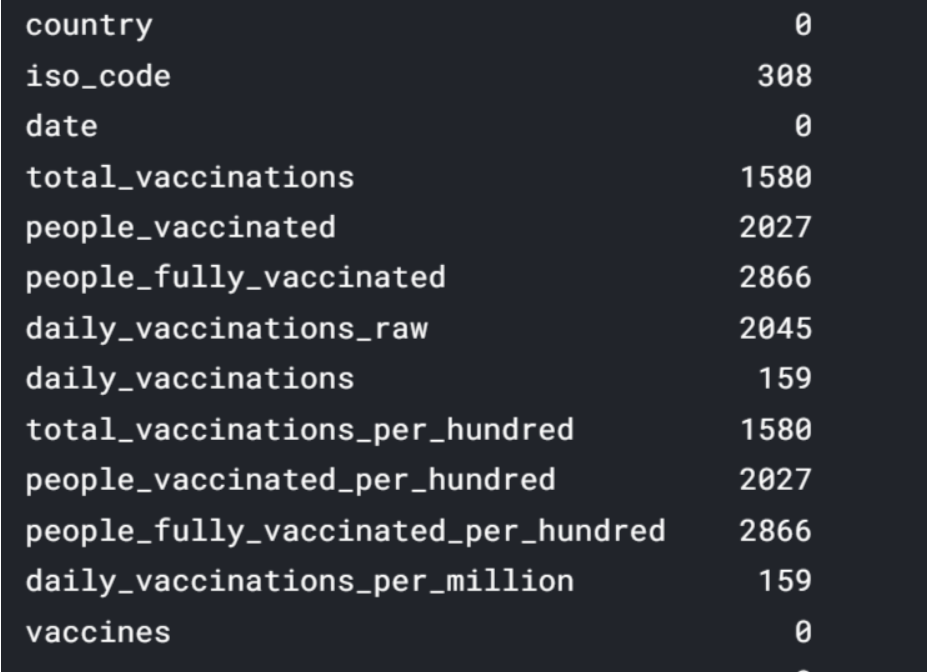
Read the CSV file using pandas read\_csv() function and show the output using head() function.



df.info()



df.isnull().sum()



### DATA CLEANING

df.fillna(value = 0, inplace = True)

df.total\_vaccinations = df.total\_vaccinations.astype(int)

df.people\_vaccinated = df.people\_vaccinated.astype(int)

df.people\_fully\_vaccinated = df.people\_fully\_vaccinated.astype(int)

df.daily\_vaccinations\_raw = df.daily\_vaccinations\_raw.astype(int)

df.daily\_vaccinations = df.daily\_vaccinations.astype(int)

df.total\_vaccinations\_per\_hundred = df.total\_vaccinations\_per\_hundred.astype(int)

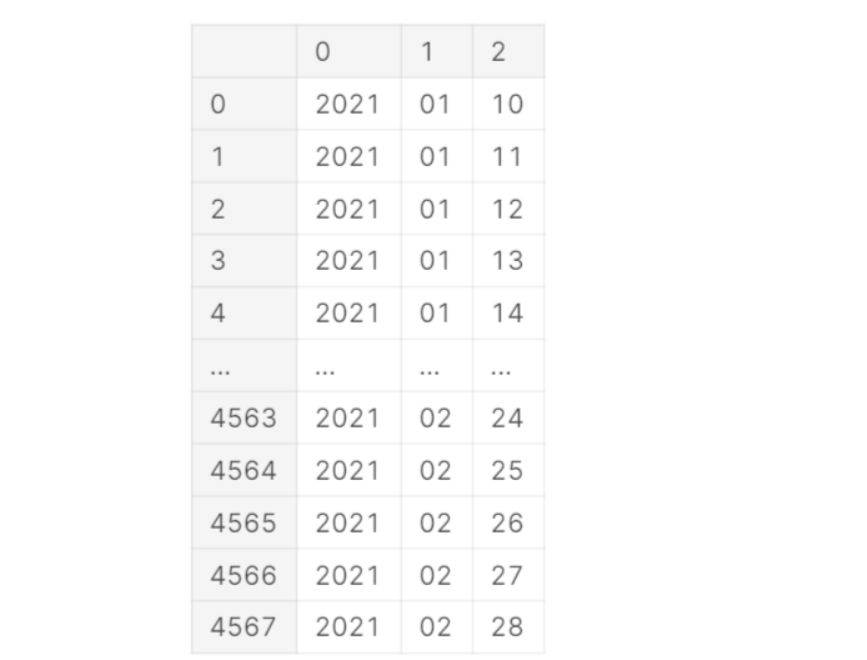
df.people\_fully\_vaccinated\_per\_hundred = df.people\_fully\_vaccinated\_per\_hundred.astype(int)

df.daily\_vaccinations\_per\_million = df.daily\_vaccinations\_per\_million.astype(int)

df.people\_vaccinated\_per\_hundred = df.people\_vaccinated\_per\_hundred.astype(int)

date = df.date.str.split('-', expand =True)

date



df['year'] = date[0]

df['month'] = date[1]

df['day'] = date[2]

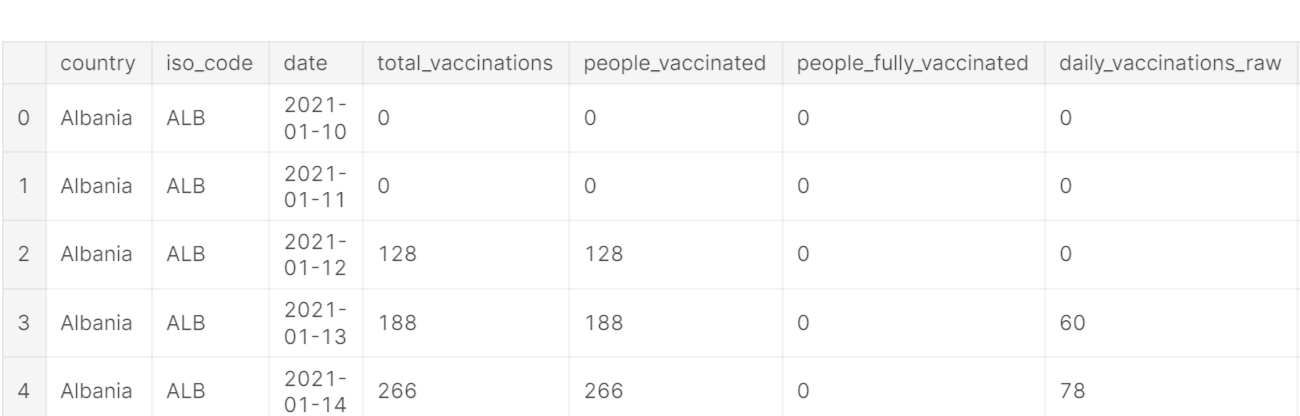
df.year = pd.to\_numeric(df.year)

df.month = pd.to\_numeric(df.month)

df.day = pd.to\_numeric(df.day)

df.date = pd.to\_datetime(df.date)

df.head()



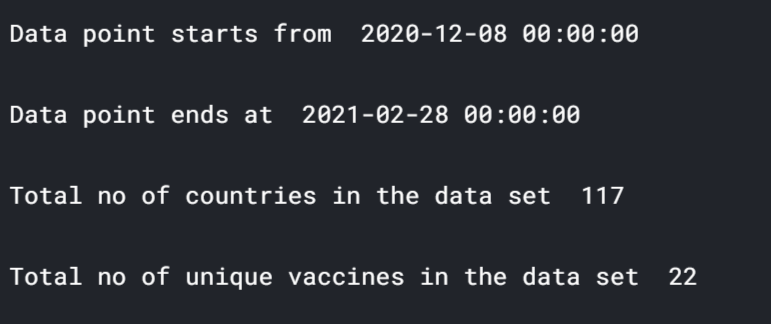
### SOME FEATURES

print('Data point starts from ',df.date.min(),'n')

print('Data point ends at ',df.date.max(),'n')

print('Total no of countries in the data set ',len(df.country.unique()),'n')

print('Total no of unique vaccines in the data set ',len(df.vaccines.unique()),'n')



**Observation**

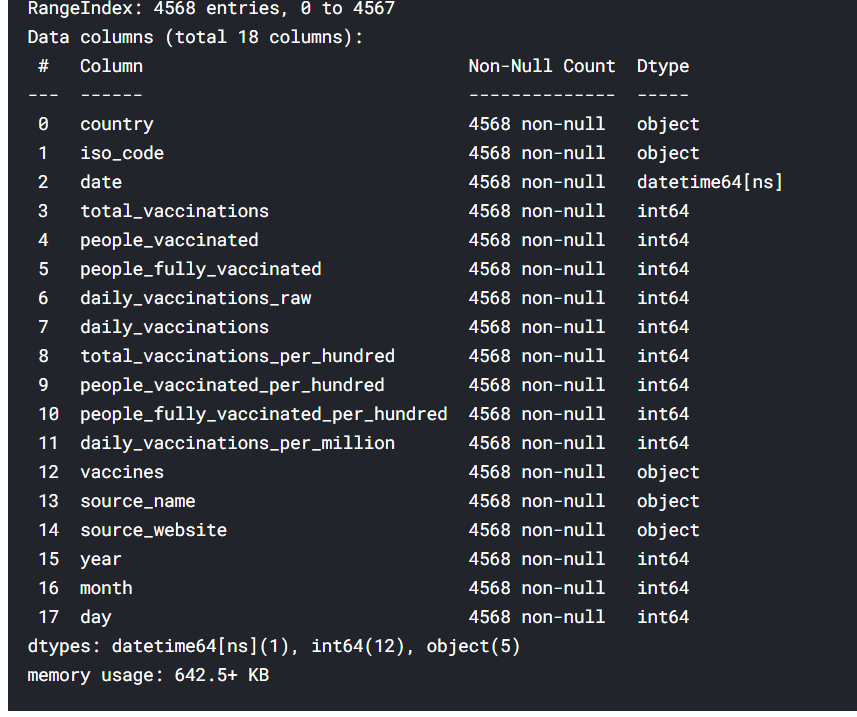
Data points start from 2020-12-08

Data points end at 2021-02-28

Total Number of countries in the data set = 117

Total Number of Unique Vaccines in the data set = 22

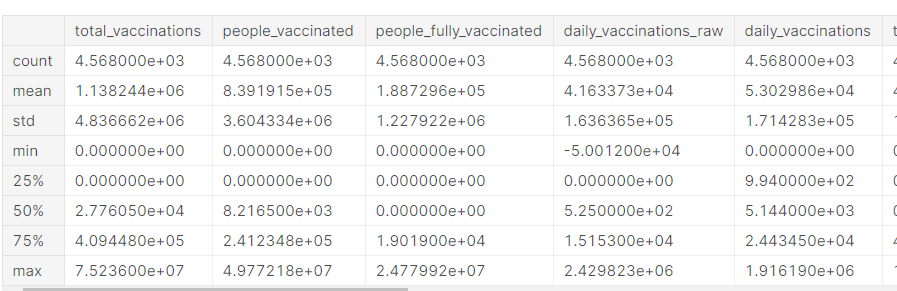
df.info()



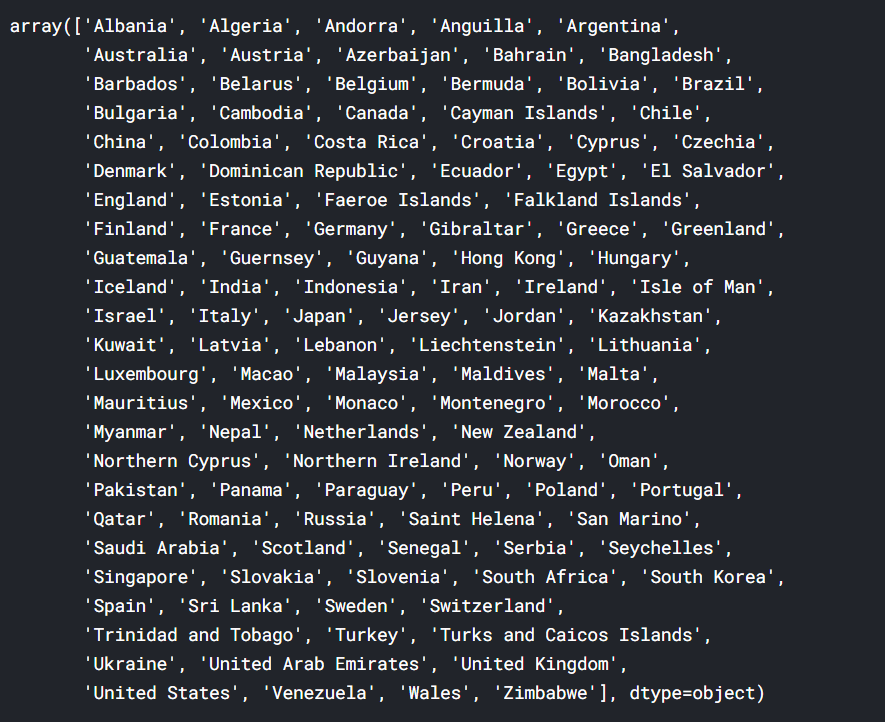
## DATA VISUALIZATION

.

df.describe()



df.country.unique()



def size(m,n):

fig = plt.gcf();

fig.set\_size\_inches(m,n);

### Word Art of Countries

wordCloud = WordCloud(

background\_color='white',

max\_font\_size = 50).generate(' '.join(df.country))

plt.figure(figsize=(15,7))

plt.axis('off')

plt.imshow(wordCloud)

plt.show()



### Total Vaccinated Till Date

country\_wise\_total\_vaccinated = {}

for country in df.country.unique() :

vaccinated = 0

for i in range(len(df)) :

if df.country[i] == country :

vaccinated += df.daily\_vaccinations[i]

country\_wise\_total\_vaccinated[country] = vaccinated

country\_wise\_total\_vaccinated\_df = pd.DataFrame.from\_dict(country\_wise\_total\_vaccinated,

orient='index',

columns = ['total\_vaccinted\_till\_date'])

country\_wise\_total\_vaccinated\_df.sort\_values(by = 'total\_vaccinted\_till\_date', ascending = False, inplace = True)

country\_wise\_total\_vaccinated\_df

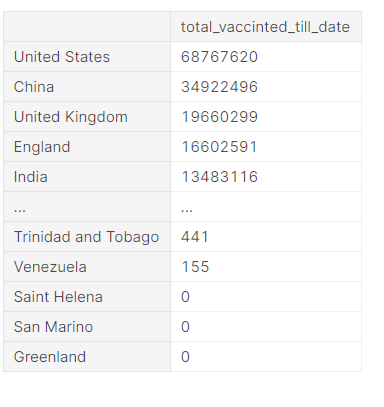


fig = px.bar(country\_wise\_total\_vaccinated\_df,

y = 'total\_vaccinted\_till\_date',

x = country\_wise\_total\_vaccinated\_df.index,

color = 'total\_vaccinted\_till\_date',

color\_discrete\_sequence= px.colors.sequential.Viridis\_r

)

fig.update\_layout(

title={

'text' : "Vaccination till date in various countries",

'y':0.95,

'x':0.5

},

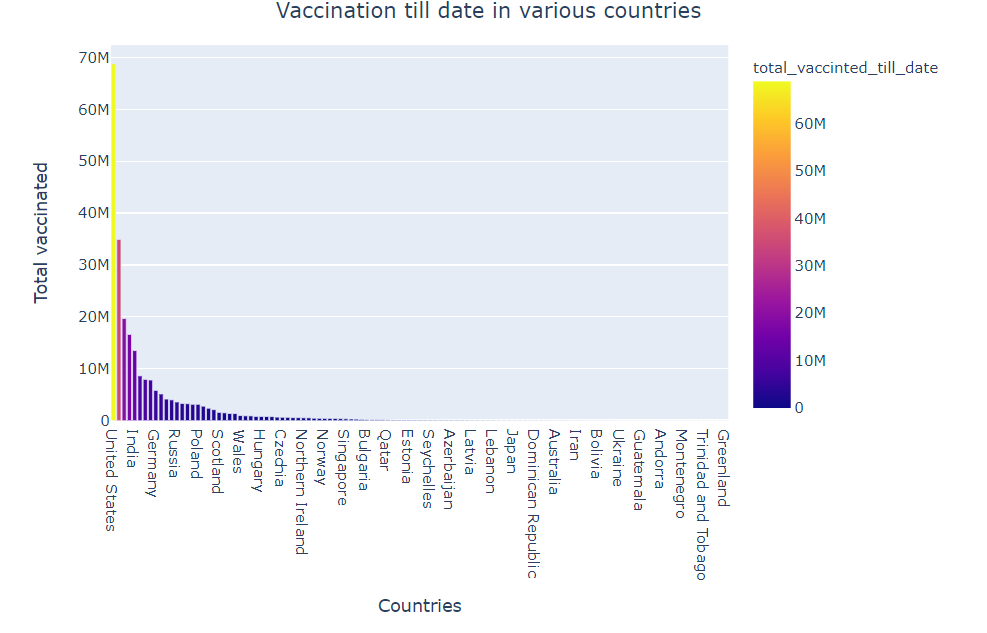
xaxis\_title="Countries",

yaxis\_title="Total vaccinated",

legend\_title="Total vaccinated"

)

fig.show()



### Country Wise Daily Vaccination

fig = px.line(df, x = 'date', y ='daily\_vaccinations', color = 'country')

fig.update\_layout(

title={

'text' : "Daily vaccination trend",

'y':0.95,

'x':0.5

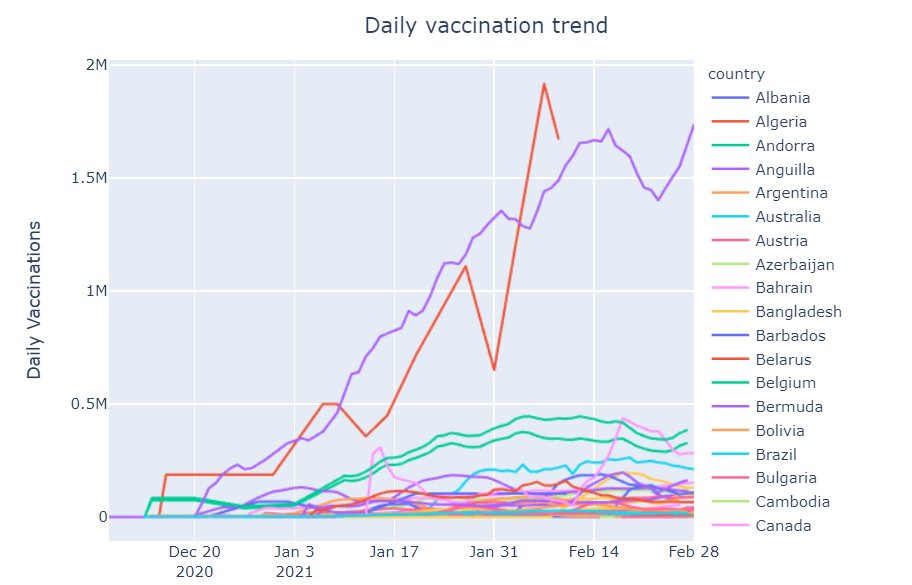
},

xaxis\_title="Date",

yaxis\_title="Daily Vaccinations"

)

fig.show()



**Plot Till Date Function**

def plot\_till\_date(value1, value2, title, color1, color2) :

so\_far\_dict = {}

for dates in df.date.unique() :

so\_far\_dict[dates], value1\_count, value2\_count = [], 0, 0

for i in range(len(df)) :

if df.date[i] == dates :

value1\_count += df[value1][i]

value2\_count += df[value2][i]

so\_far\_dict[dates].append(value1\_count)

so\_far\_dict[dates].append(value2\_count)

so\_far\_df = pd.DataFrame.from\_dict(so\_far\_dict, orient = 'index', columns=[value1, value2])

so\_far\_df.reset\_index(inplace = True)

so\_far\_df.sort\_values(by='index', inplace = True)

plot = go.Figure(data=[go.Scatter(

x = so\_far\_df['index'],

y = so\_far\_df[value1],

stackgroup='one',

name = value1,

marker\_color= color1),

go.Scatter(

x = so\_far\_df['index'],

y = so\_far\_df[value2],

stackgroup='one',

name = value2,

marker\_color= color2)

])

plot.update\_layout(

title={

'text' : title,

'y':0.95,

'x':0.5

},

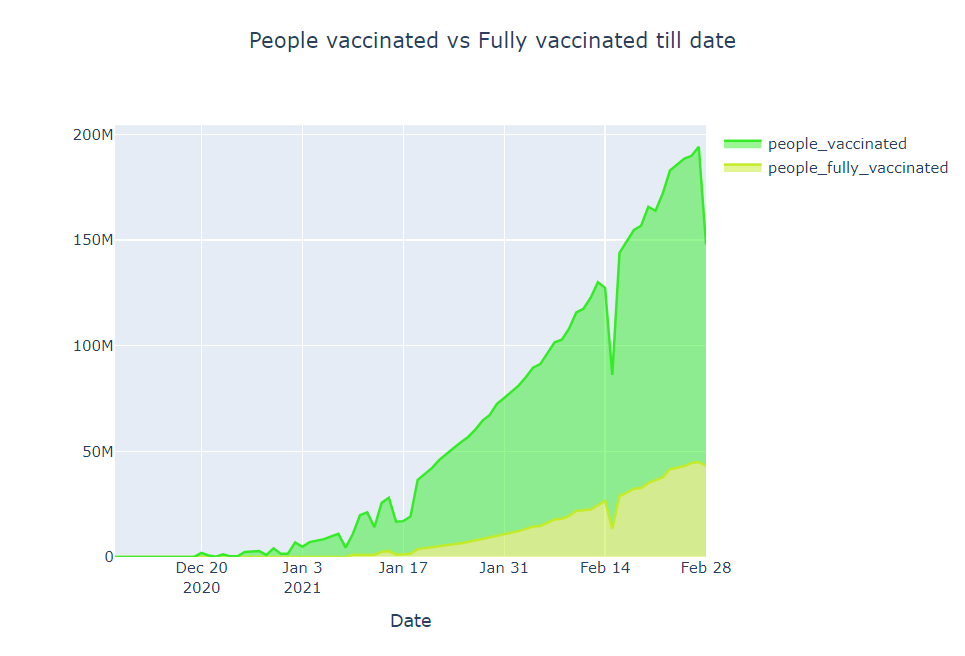
xaxis\_title="Date"

)

return plot.show()

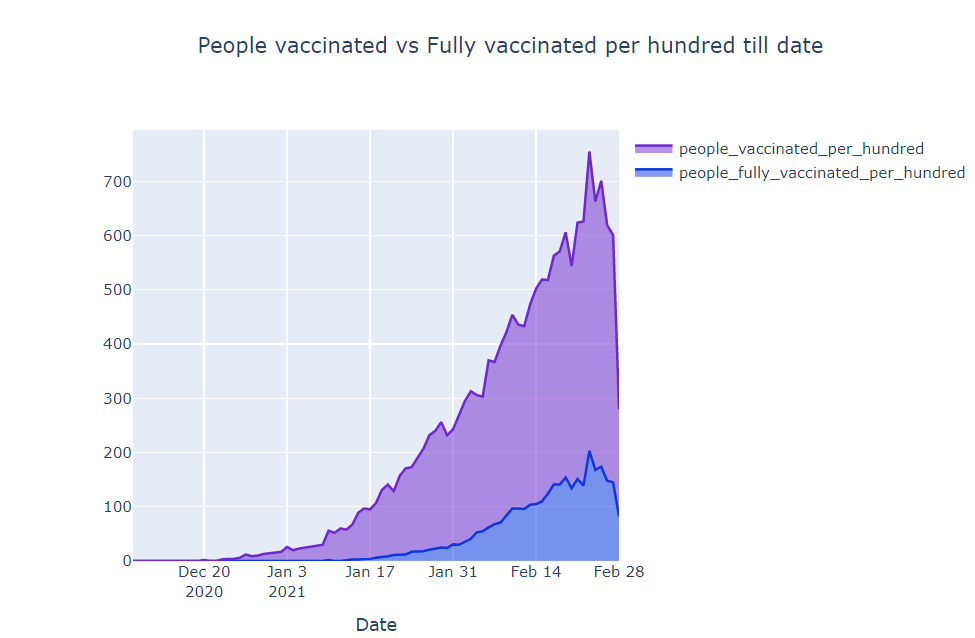
### People vaccinated vs people fully vaccinated in the world :

plot\_till\_date('people\_fully\_vaccinated', 'people\_vaccinated','People vaccinated vs Fully vaccinated till date', '#c4eb28', '#35eb28')



### The People vaccinated vs people fully vaccinated per hundred in the world

plot\_till\_date('people\_fully\_vaccinated\_per\_hundred', 'people\_vaccinated\_per\_hundred', 'People vaccinated vs Fully vaccinated per hundred till date', '#0938e3','#7127cc')



### Pie-Plot

def plot\_pie(value, title, color) :

new\_dict = {}

for v in df[value].unique() :

value\_count = 0

for i in range(len(df)) :

if df[value][i] == v :

value\_count += 1

new\_dict[v] = value\_count

new\_df = pd.DataFrame.from\_dict(new\_dict, orient = 'index', columns = ['Total'])

if color == 'plasma' :

fig = px.pie(new\_df, values= 'Total',

names = new\_df.index,

title = title,

color\_discrete\_sequence=px.colors.sequential.Plasma)

elif color == 'rainbow' :

fig = px.pie(new\_df, values= 'Total',

names = new\_df.index,

title = title,

color\_discrete\_sequence=px.colors.sequential.Rainbow)

else :

fig = px.pie(new\_df, values= 'Total',

names = new\_df.index,

title = title)

fig.update\_layout(

title={

'y':0.95,

'x':0.5

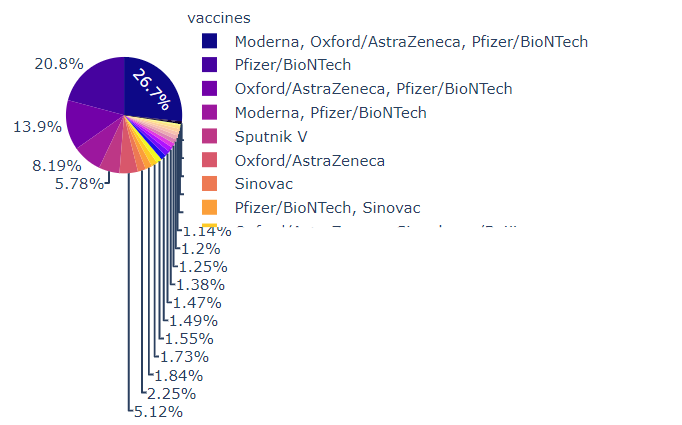
},

legend\_title = value

)

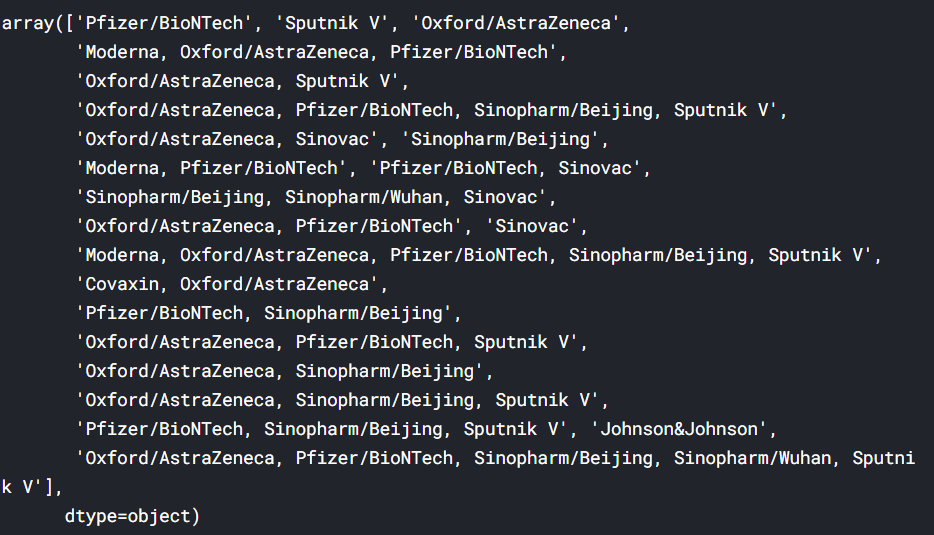
return fig.show()

plot\_pie('vaccines', 'Various vaccines and their uses', 'plasma')



### Most Used Vaccine

df.vaccines.unique()



### Word art of Vaccines

wordCloud = WordCloud(

background\_color='white',

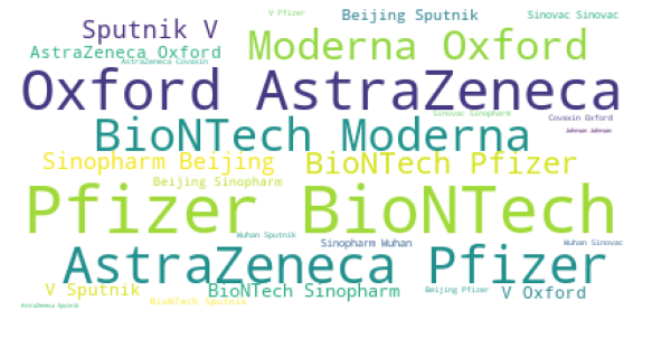
max\_font\_size = 50).generate(' '.join(df.vaccines))

plt.figure(figsize=(12,5))

plt.axis('off')

plt.imshow(wordCloud)

plt.show()



### Daily vaccination trend per million

fig = px.line(df, x = 'date', y ='daily\_vaccinations\_per\_million', color = 'country')

fig.update\_layout(

title=

'text' : "Daily vaccination trend per million",

'y':0.95,

'x':0.5

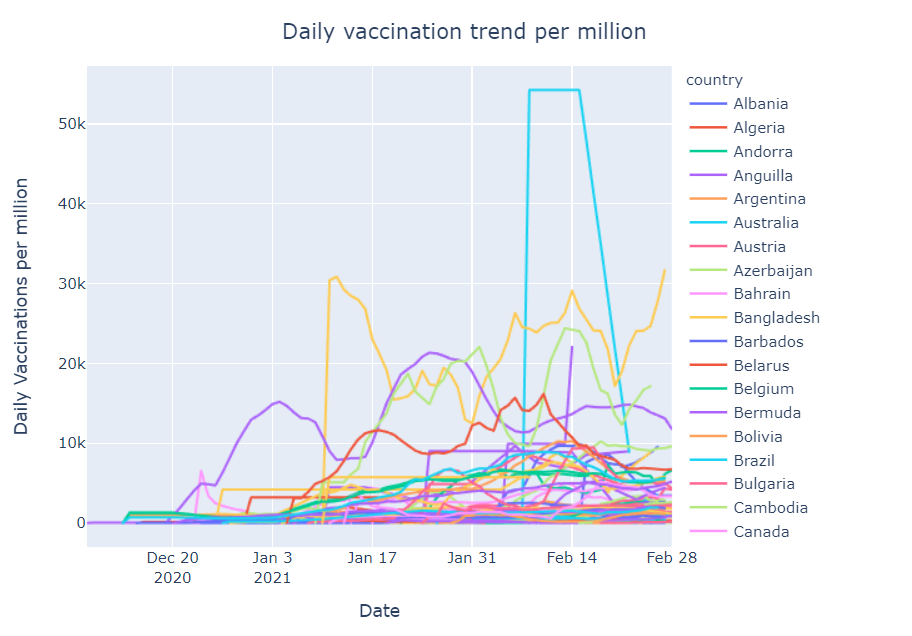
},

xaxis\_title="Date",

yaxis\_title="Daily Vaccinations per million"

)

fig.show()



### Total vaccinated – India vs the USA

india\_usa = [df[df.country == 'United States'], df[df.country == 'India']]

result = pd.concat(india\_usa)

fig = px.line(result, x = 'date', y ='total\_vaccinations', color = 'country')

fig.update\_layout(

title={

'text' : "Total vaccinated - India vs USA",

'y':0.95,

'x':0.5

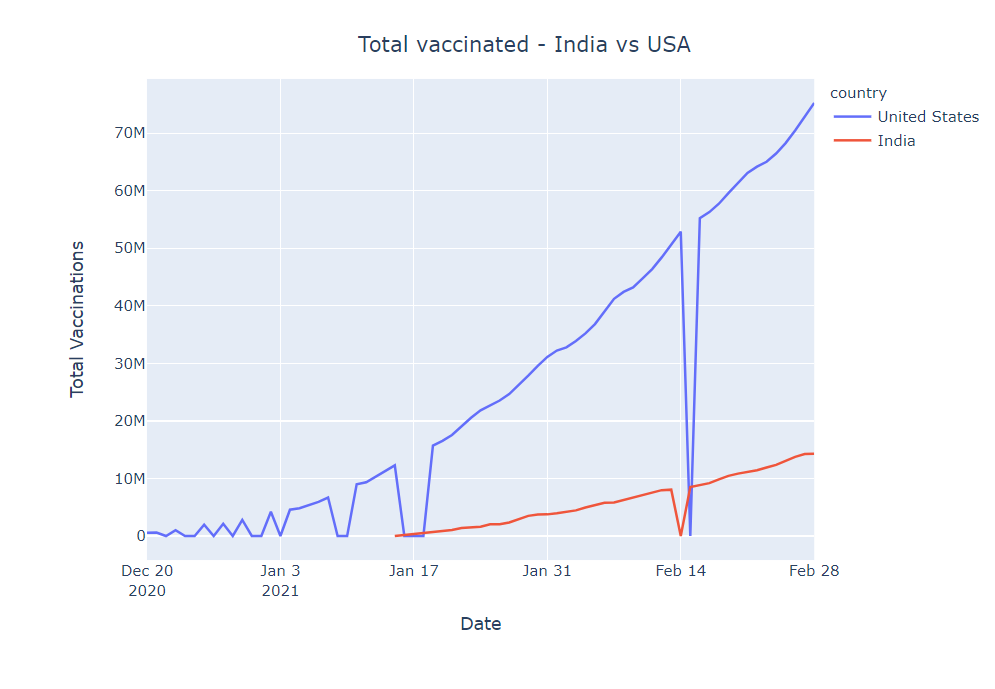
},

xaxis\_title="Date",

yaxis\_title="Total Vaccinations"

)

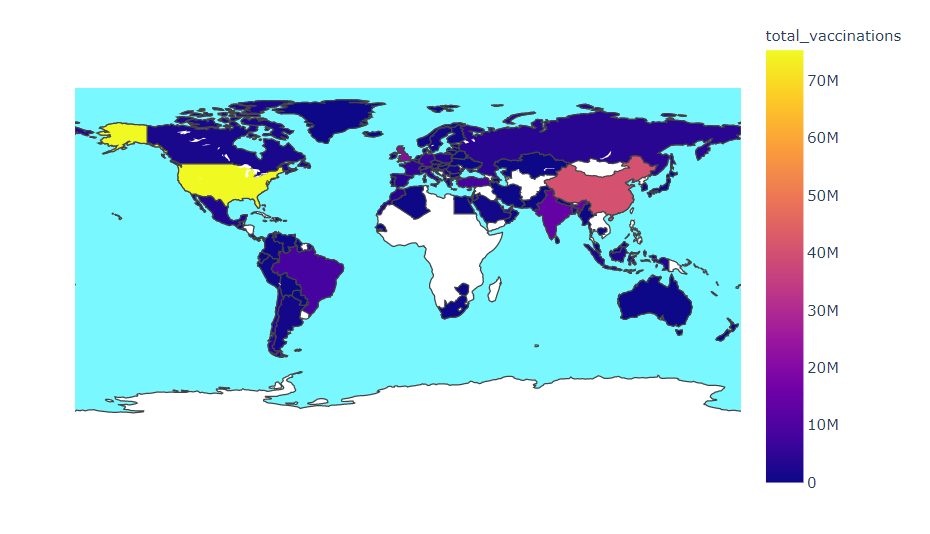
fig.show()



**MAPS**

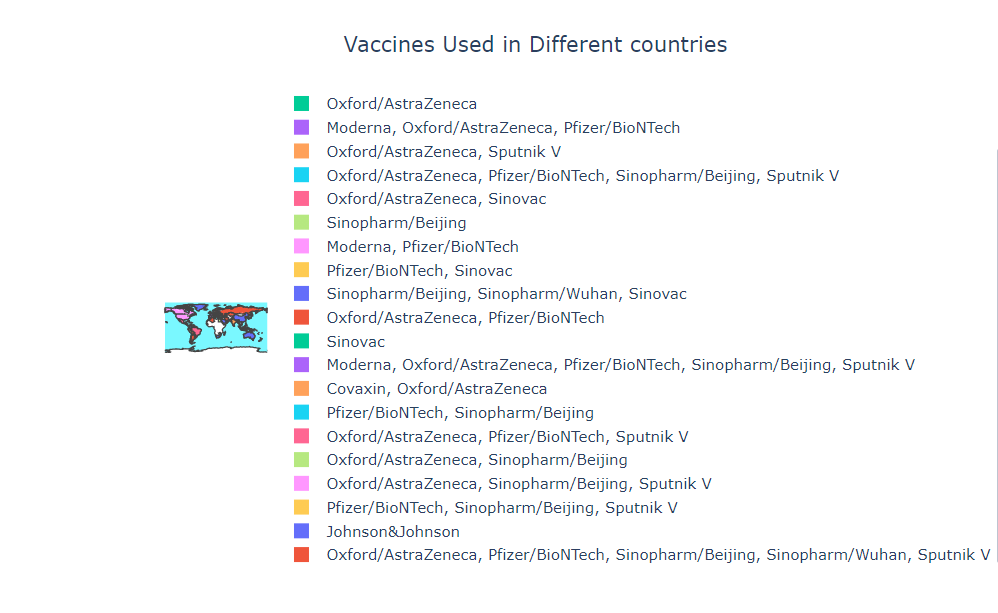
### Most vaccinated country

plot\_map('total\_vaccinations','Most vaccinated country', None)



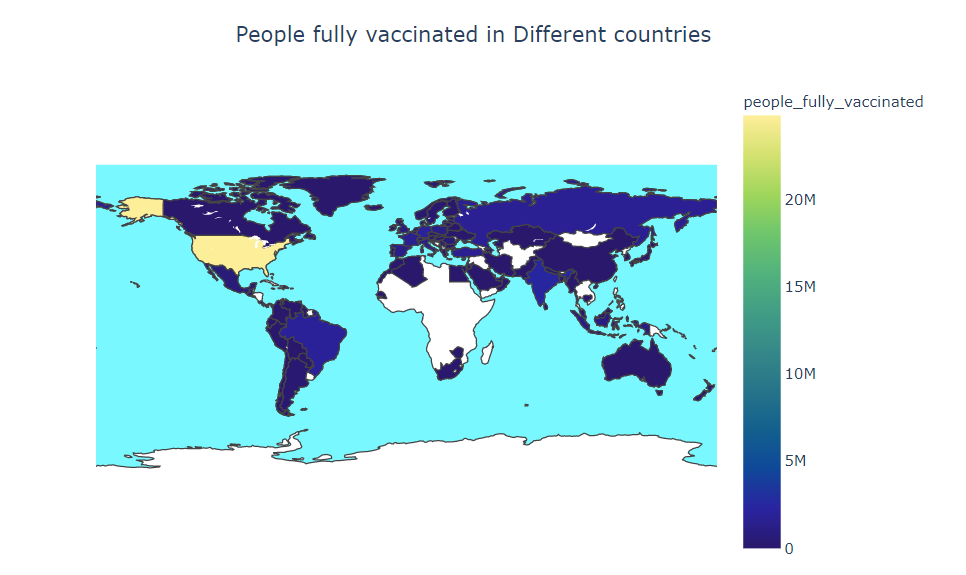
### Vaccines Used in Different countries

plot\_map('vaccines','Vaccines Used in Different countries', None)



### People fully vaccinated in Different countries

plot\_map('people\_fully\_vaccinated','People fully vaccinated in Different countries', 'haline')



Conclusion and Future Work (Phase 2)

# Project Conclusion:

In the Phase 2 conclusion, **Covid-19 Vaccine Data Analysis** provides valuable insights into the efﬁcacy and safety of the vaccines. The data obtained from clinical trials, real-world studies, and adverse event reports is essential for ensuring that the vaccines are effective and safe. Ongoing research is necessary to develop new vaccines and to study the long-term effects of the vaccines..

Future Work: **Future Developments** in Covid-19 vaccine research include the development of new vaccines, booster shots, and the study of long- term vaccine efﬁcacy and safety. Ongoing research is essential to ensure that we have effective tools to combat the pandemic.