Introduction

Problem Description

The new pandemic of COVID-19 was the most discussed matter of the last 2 years. As such, it is important to analyze and predict the trends of the vaccination process at the level of the whole world, but also at the country-level.

The data collected is meant to answer questions such as:

1. What kind of vaccine is used by which country?
2. Which country has the most advanced vaccine program?
3. Where are more people vaccinated daily?
4. How is the vaccination rate evolving, with regard to the population already vaccinated?

The data set

The data set is updated daily and cotains the following collumns:

- **Country** - this is the country for which the vaccination information is provided;
- **Country ISO Code** - ISO code for the country;
- **Date** - date for the data entry; for some of the dates we have only the daily vaccinations, for others, only the (cumulative) total;
- **Total number of vaccinations** - this is the absolute number of total immunizations in the country;
- **Total number of people vaccinated** - a person, depending on the immunization scheme, will receive one or more (typically 2) vaccines; at a certain moment, the number of vaccination might be larger than the number of people;
- **Total number of people fully vaccinated** - this is the number of people that received the entire set of immunization according to the immunization scheme (typically 2); at a certain moment in time, there might be a certain number of people that received one vaccine and another number (smaller) of people that received all vaccines in the scheme;
- **Daily vaccinations (raw)** - for a certain data entry, the number of vaccination for that date/country;

- **Daily vaccinations** - for a certain data entry, the number of vaccination for that date/country;
- **Total vaccinations per hundred** - ratio (in percent) between vaccination number and total population up to the date in the country;
- **Total number of people vaccinated per hundred** - ratio (in percent) between population immunized and total population up to the date in the country;
- **Total number of people fully vaccinated per hundred** - ratio (in percent) between population fully immunized and total population up to the date in the country;
- **Number of vaccinations per day** - number of daily vaccination for that day and country;
- **Daily vaccinations per million** - ratio (in ppm) between vaccination number and total population for the current date in the country;
- **Vaccines used in the country** - total number of vaccines used in the country (up to date);

Note: For this first deliverable, we also sorted the countries based on some of these collumns in order to obtain a top of them.

The task

The task for today *(25.03.2021)* was to familiarize ourselves with the data from the dataset and also present the data under different forms, using different types of data visualization techniques.

Subtask 1 - Fetching the data

For this subtask, we fetched the data directly from the Kaggle website by making use of the kaggle API.

Subtask 2 - Familiarizing ourselves with the data

This subtask was completed by understanding the meaning of each column in the table and thinking about ways this data can provide us important information. As such, we present vaccination details about Romania, but also the top 10 countries in terms of percentage of vaccination as well as total number of vaccinations.

Subtask 3 - Data visualization

This subtask was completed by searching for different types of data visualization. As such, we used pie plot, scatter plots (on the whole world map), bar plots, line plots a.s.o. This gave us a general understanding of the number of vaccinations (both total and daily).

Data Fetching and opening the file as a DataFrame

In [1]:

```
import os
import zipfile
import kaggle

DATA_PATH = r"datasets\vaccines"
kaggle.api.authenticate()
kaggle.api.dataset_download_files('gpreda/covid-world-vaccination-progress')


def fetch_data(data_path=DATA_PATH):
    if not os.path.isdir(data_path):
        os.mkdir(data_path)
    zf = zipfile.ZipFile('covid-world-vaccination-progress.zip')
    zf.extractall(DATA_PATH)


fetch_data()
```

In [2]:

```
# Retreive the data from the file, fill the null values with 0 and convert date string to date

import pandas as pd
import plotly.express as px
import plotly.graph_objects as go


df = pd.read_csv(r"datasets\vaccines\country_vaccinations.csv")
df['date'] = pd.to_datetime(df['date'], format='%Y-%m-%d')
df = df.fillna(0)
df
```

Out[2]:

	country	iso_code	date	total_vaccinations	people_vaccinated	people_fully_vaccinated	daily_vaccinations_raw	daily_vaccinations	total_v
0	Afghanistan	AFG	2021-02-22	0.0	0.0	0.0	0.0	0.0	
1	Afghanistan	AFG	2021-02-23	0.0	0.0	0.0	0.0	1367.0	

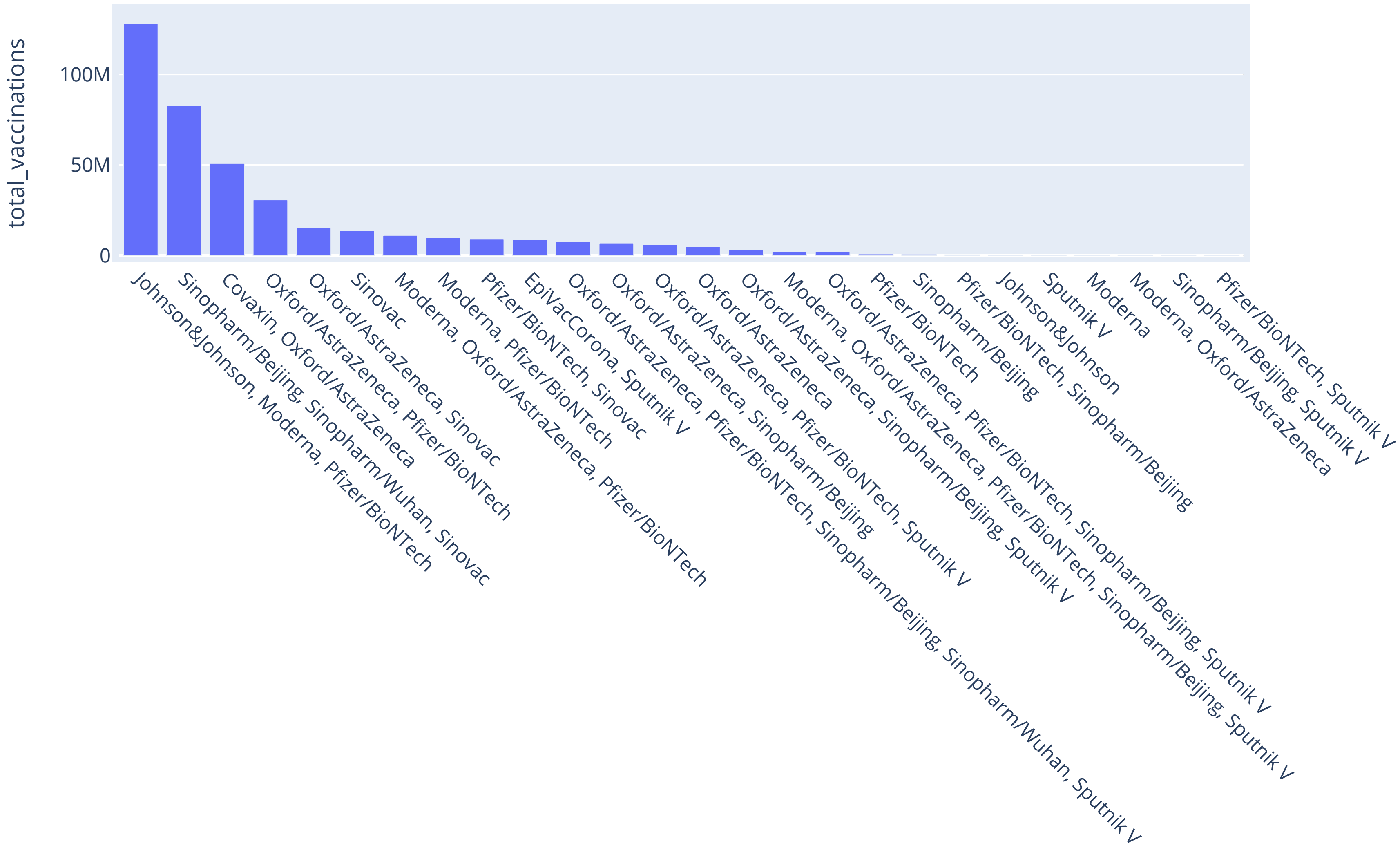
	country	iso_code	date	total_vaccinations	people_vaccinated	people_fully_vaccinated	daily_vaccinations_raw	daily_vaccinations	total_v
2	Afghanistan	AFG	2021-02-24	0.0	0.0	0.0	0.0	1367.0	
3	Afghanistan	AFG	2021-02-25	0.0	0.0	0.0	0.0	1367.0	
4	Afghanistan	AFG	2021-02-26	0.0	0.0	0.0	0.0	1367.0	
...	...	...	...	...	...	...	...	...	
7892	Zimbabwe	ZWE	2021-03-19	41687.0	41687.0	0.0	1685.0	772.0	
7893	Zimbabwe	ZWE	2021-03-20	42210.0	42210.0	0.0	523.0	836.0	
7894	Zimbabwe	ZWE	2021-03-21	42729.0	42729.0	0.0	519.0	910.0	
7895	Zimbabwe	ZWE	2021-03-22	43294.0	43294.0	0.0	565.0	805.0	
7896	Zimbabwe	ZWE	2021-03-23	44135.0	44135.0	0.0	841.0	655.0	

7897 rows × 15 columns

Most used vaccines across the world

In [3]:

```
countries = df.groupby(["country", "vaccines"]).max().reset_index()

vaccines = df.groupby(["vaccines"]).max().reset_index()
vaccines = vaccines.sort_values(["total_vaccinations"], ascending=False)
fig = px.bar(vaccines, x='vaccines', y='total_vaccinations')
fig.update_layout(barmode='group', xaxis_tickangle=45, height=600, margin=dict(l=50, r=50, b=300, t=50, pad=4))
fig.show()
```

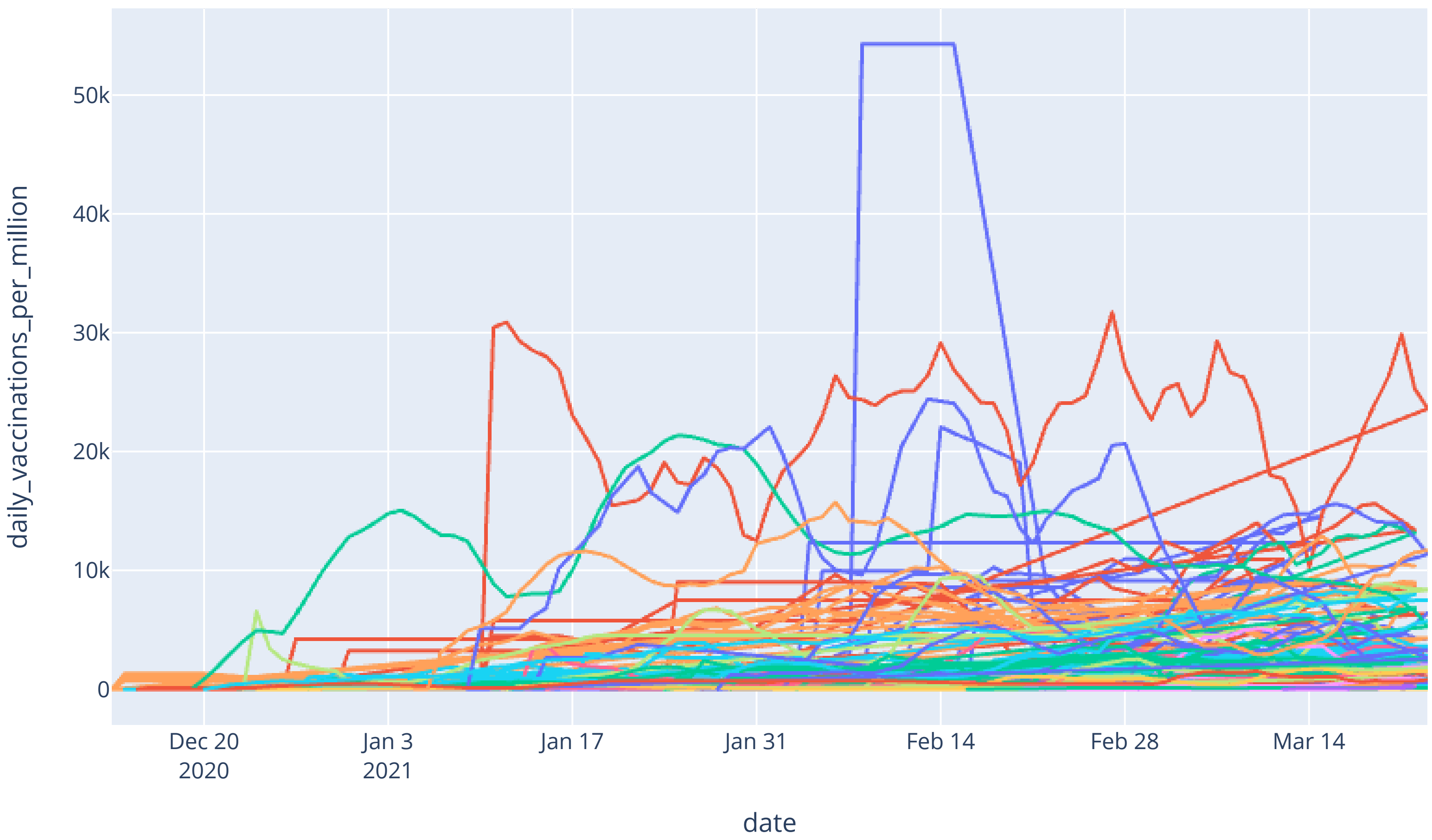


progress of vaccine usage

In [4]:

```
fig = px.line(df, x="date", y="daily_vaccinations_per_million", color="vaccines",
              hover_data=['total_vaccinations'])
```

```
fig.update_layout(showlegend=False)
fig.show()
```

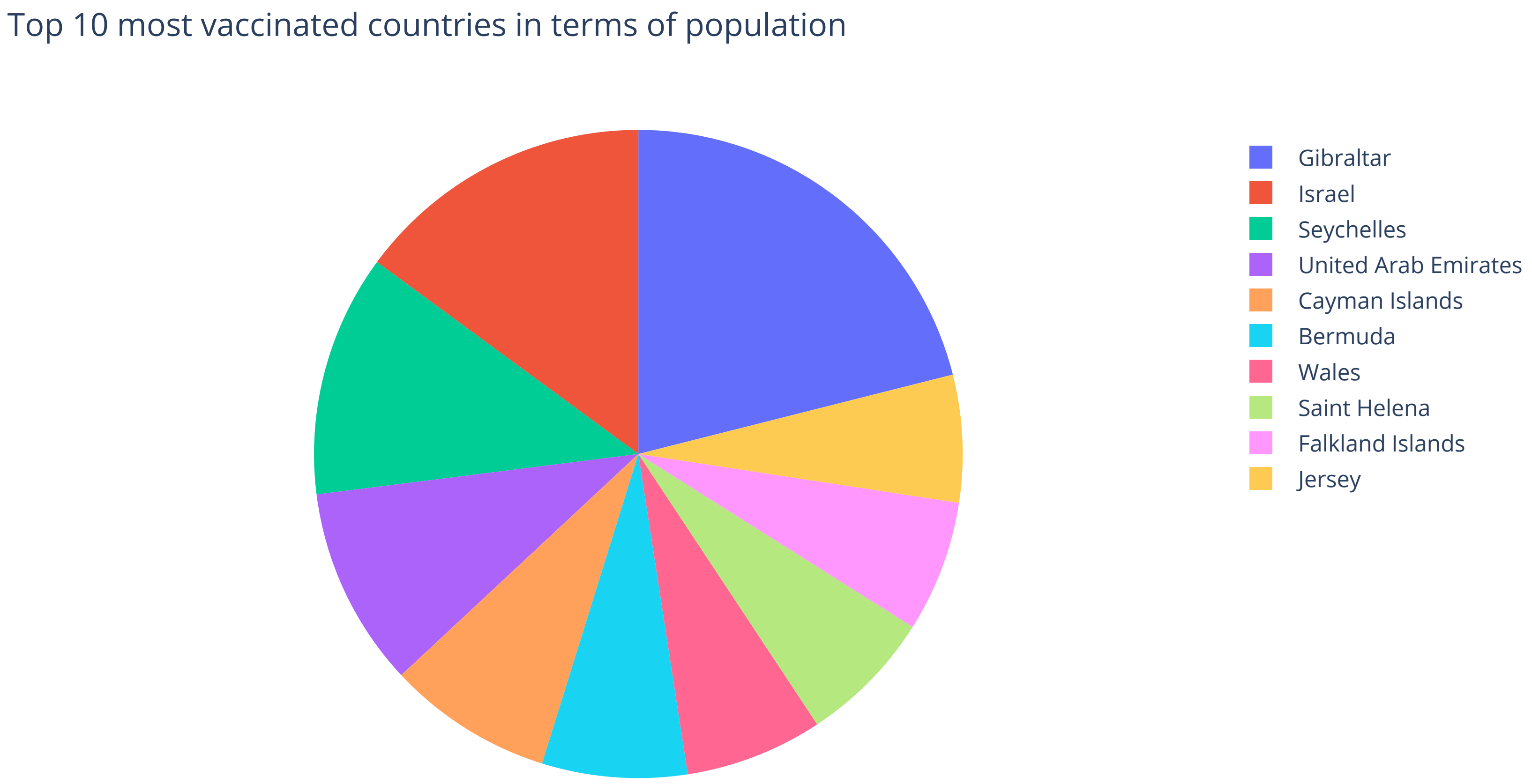


10 most and least vaccinated countries in terms of population

In [5]:

```
max_vaccines = countries.sort_values(["total_vaccinations_per_hundred"], ascending=False)
max_vaccines = max_vaccines.head(10)
```

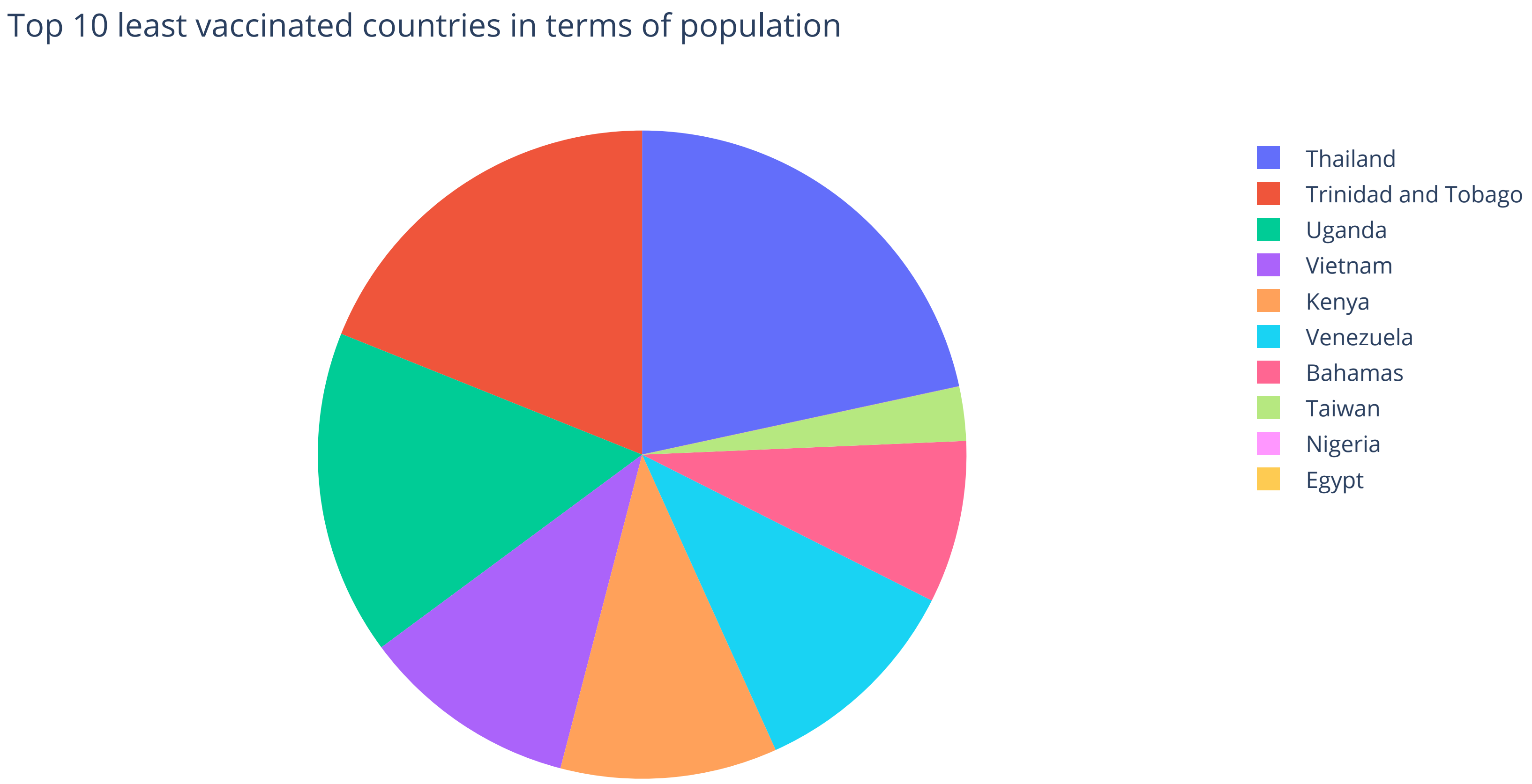
```
fig = px.pie(max_vaccines, values='total_vaccinations_per_hundred', names='country', title='Top 10 most vaccinated co
fig.update_traces(textinfo='none')
fig.show()
```



In [6]:

```
# 10 least vaccinated countries in terms of population
```

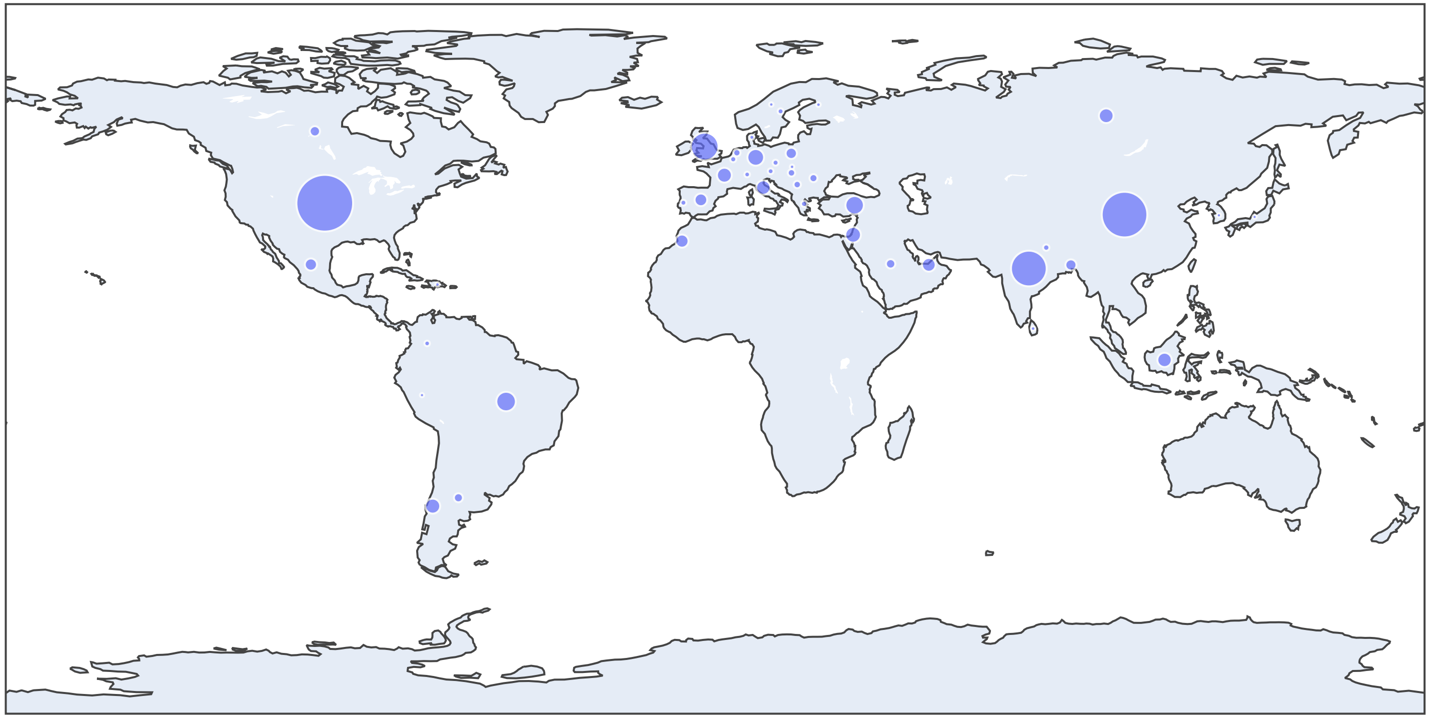
```
min_vaccines = countries.sort_values(["total_vaccinations_per_hundred"], ascending=True)
min_vaccines = min_vaccines.head(10)

fig = px.pie(min_vaccines, values='total_vaccinations_per_hundred', names='country', title='Top 10 least vaccinated c
fig.update_traces(textinfo='none')
fig.show()
```



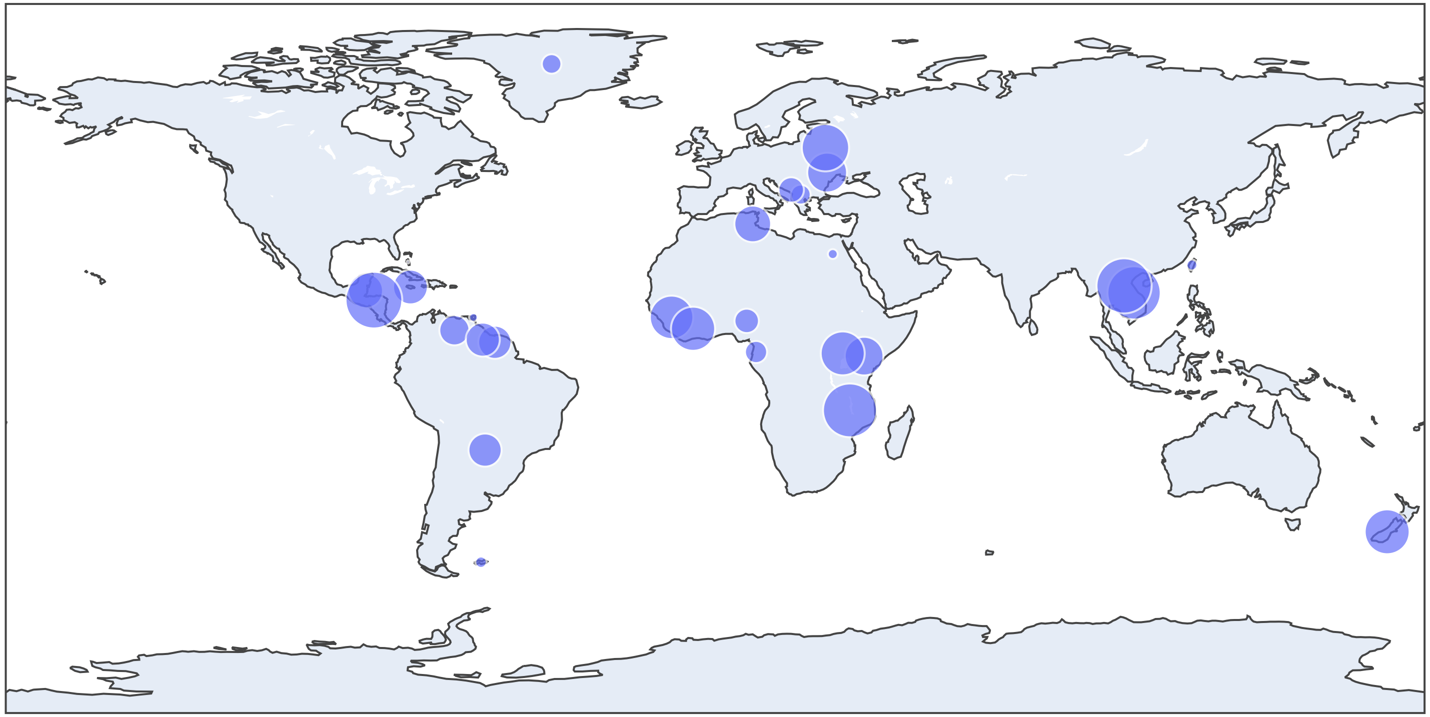
50 most and least vaccinated countries in terms of total vaccinations

In [7]:

```
## 50 most vaccinated in terms of overall number of vaccinations

max_vaccines = countries.sort_values(["total_vaccinations"], ascending=False)
max_vaccines = max_vaccines.head(50)
fig = px.scatter_geo(max_vaccines, locations="iso_code",
                     size="total_vaccinations",
                     hover_name="country"
                     )
fig.show()
```



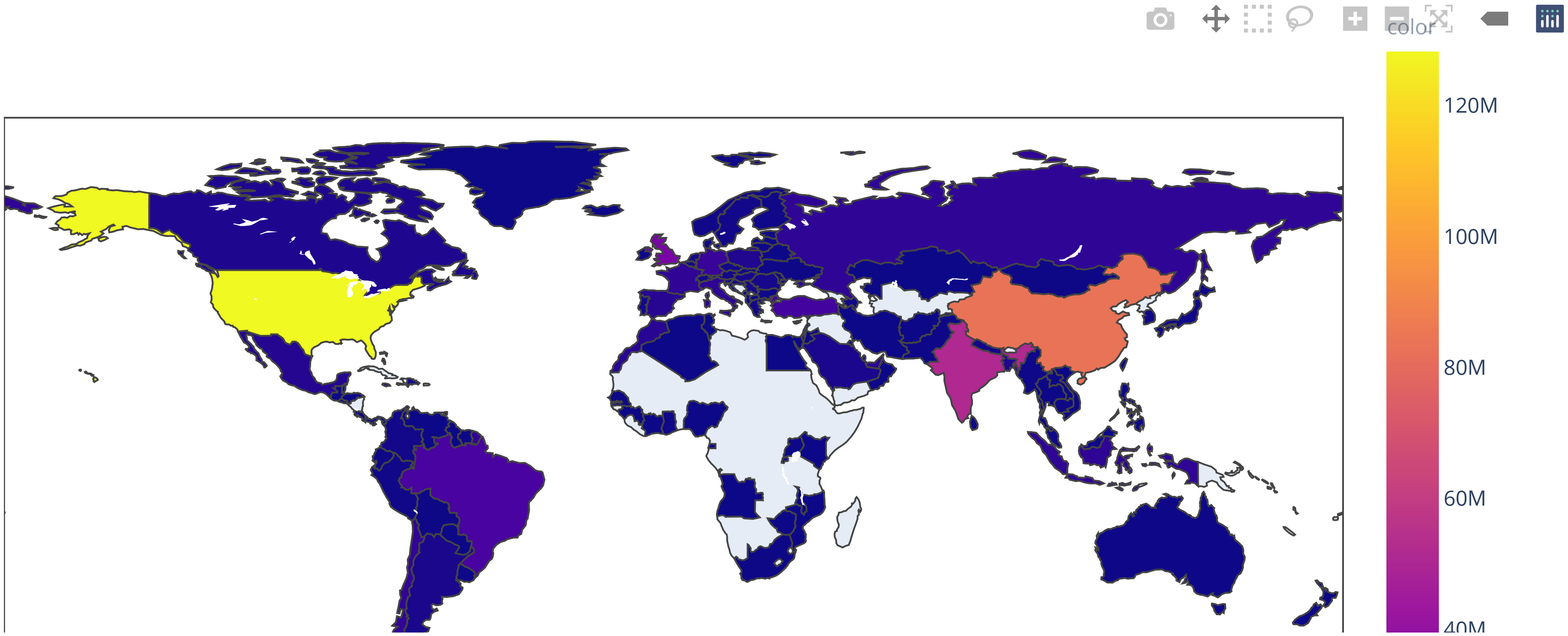
```
In [8]: #50 least vaccinated countries in terms of overall number of vaccinations

        min_vaccines = countries.sort_values(["total_vaccinations"], ascending=True)
        min_vaccines = min_vaccines.head(50)

        fig = px.scatter_geo(min_vaccines, locations="iso_code",
                             size="total_vaccinations",
                             hover_name="country"
                             )
        fig.show()
```

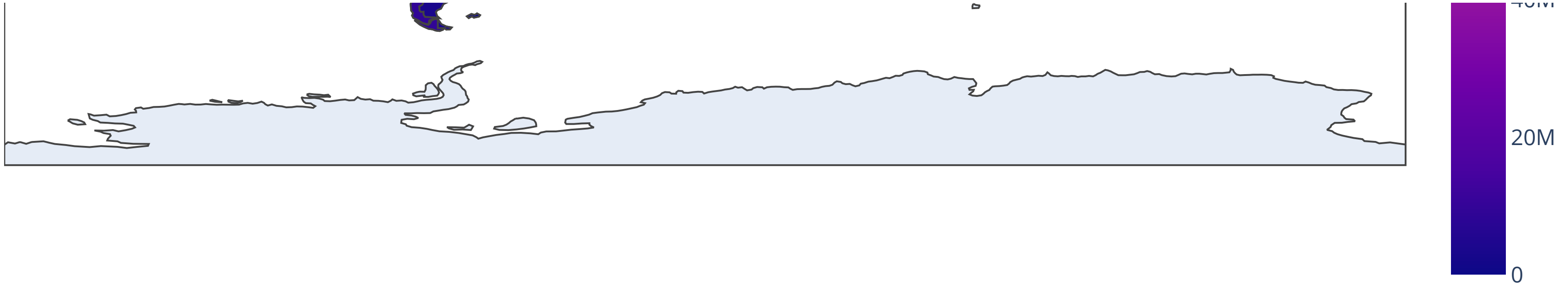


Total number of vaccinations across the world and in Europe

In [9]:

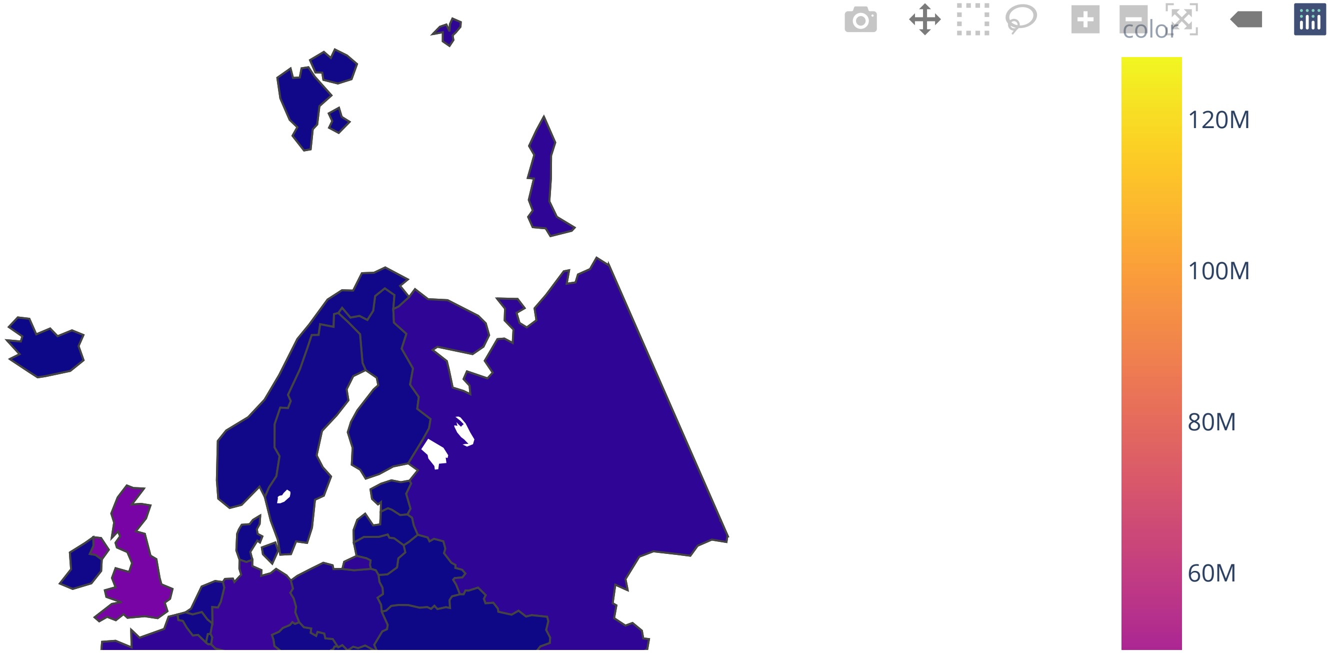
```
# total number of vaccinations across the world

fig = px.choropleth(locations = countries['country'], locationmode='country names',
            color = countries['total_vaccinations'],
            labels={'total_vaccinations':'Total Vaccinations'},
            scope = 'world')

fig.update_layout(margin={"r":0,"t":0,"l":0,"b":0})
fig.show()
```





In [10]:

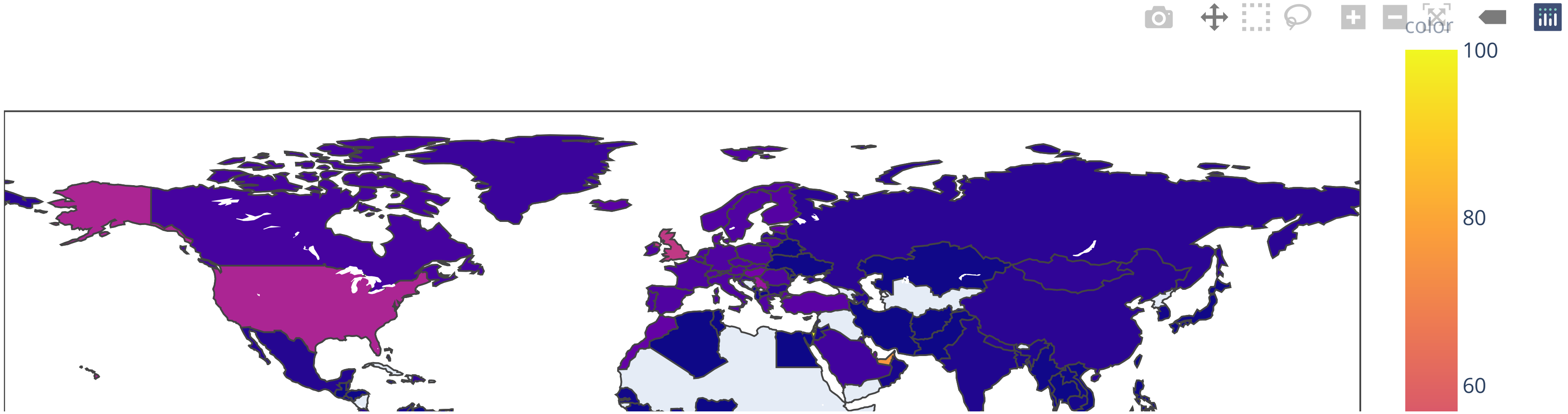
```
# number of people vaccinated in Europe

fig = px.choropleth(locations = countries['country'], locationmode='country names',
            color = countries['total_vaccinations'],
            labels={'total_vaccinations':'Total Vaccinations'},
            scope = 'europe')

fig.update_layout(margin={"r":0,"t":0,"l":0,"b":0})
fig.show()
```

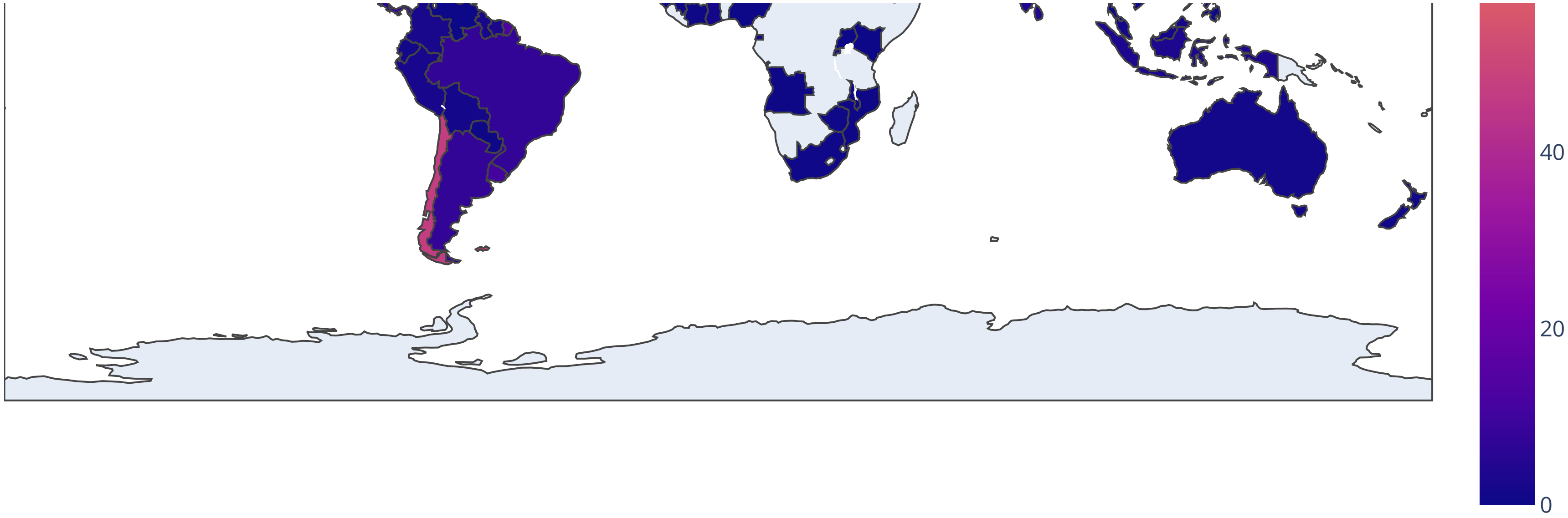


Percent of vaccinated countries in the world and in Europe

In [11]:

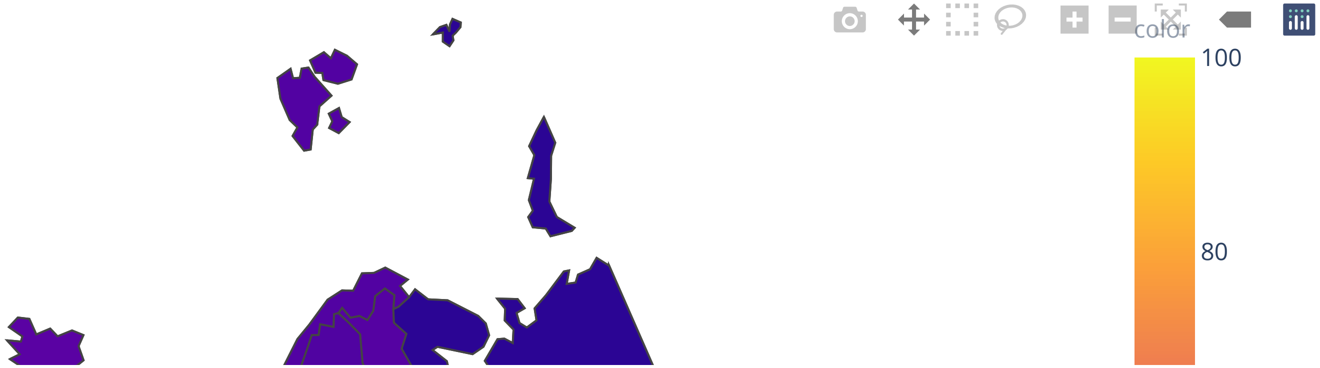
```
# percent of vaccinated countries in the world

fig = px.choropleth(locations = countries['country'], locationmode='country names',
            color = countries['total_vaccinations_per_hundred'],
            range_color=(0, 100),
            labels={'total_vaccinations_per_hundred':'Total Vaccinations in terms of population'},
            scope = 'world')

fig.update_layout(margin={"r":0,"t":0,"l":0,"b":0})
fig.show()
```

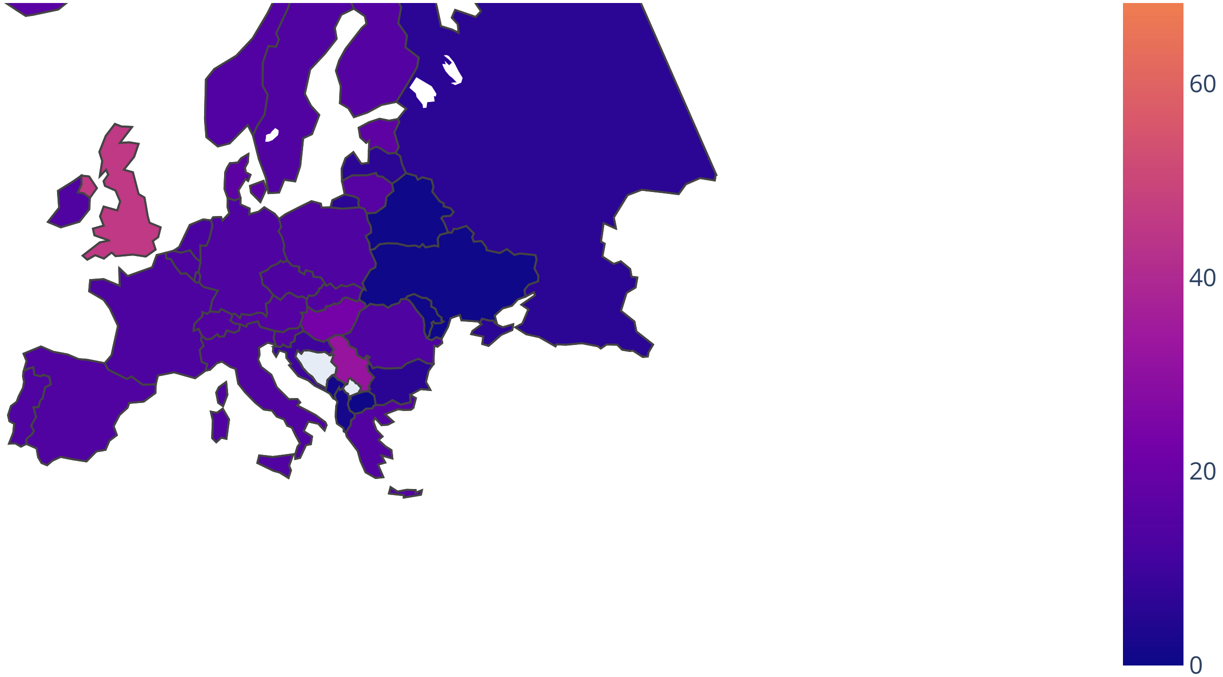




In [12]:

```
# percent of vaccinated countries in Europe

fig = px.choropleth(locations = countries['country'], locationmode='country names',
            color = countries['total_vaccinations_per_hundred'],
            range_color=(0, 100),
            labels={'total_vaccinations_per_hundred':'Total Vaccinations in terms of population'},
            scope = 'europe')

fig.update_layout(margin={"r":0,"t":0,"l":0,"b":0})
fig.show()
```

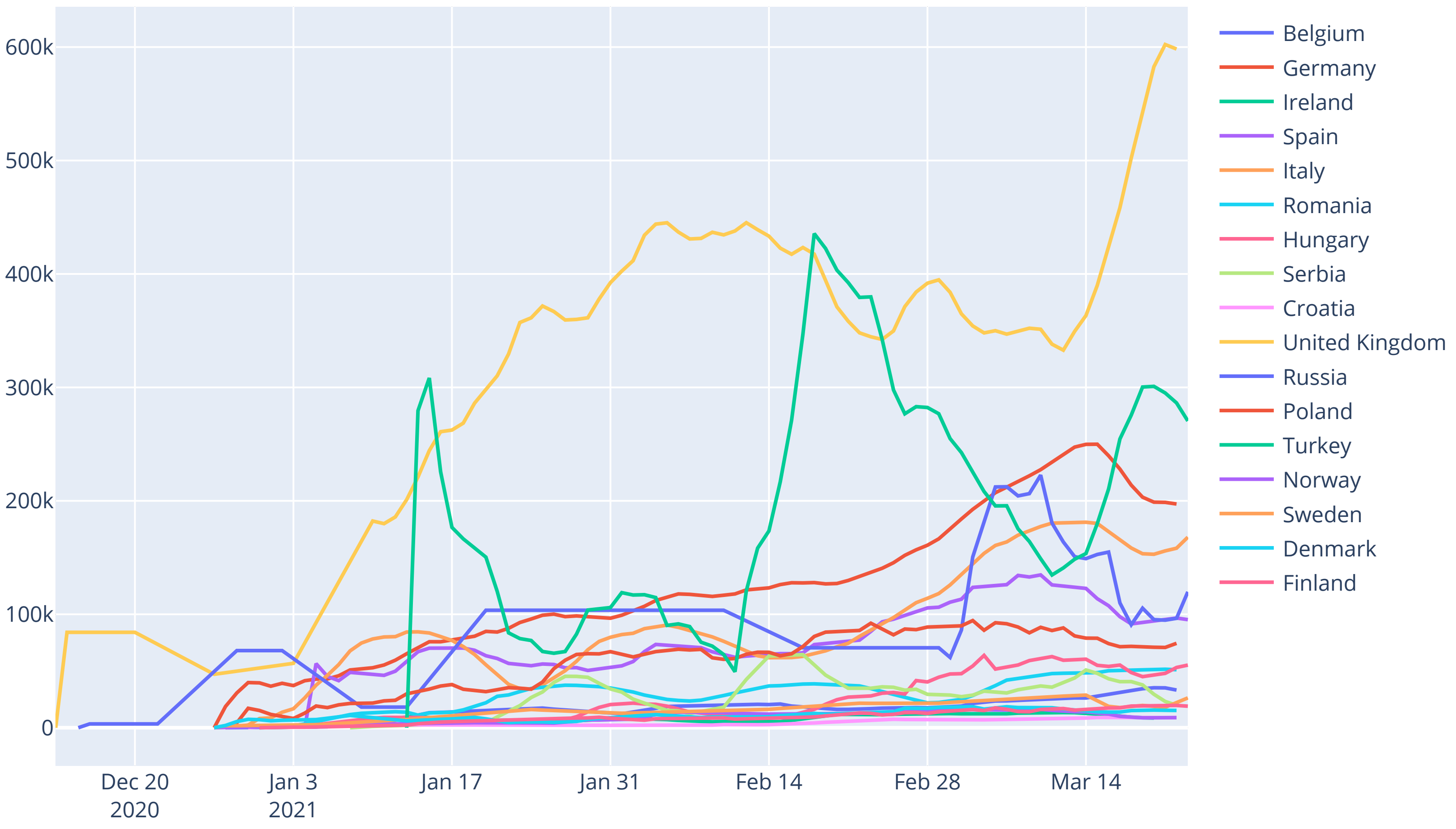




Vaccination progress in Europe based on a set of countries

In [13]:

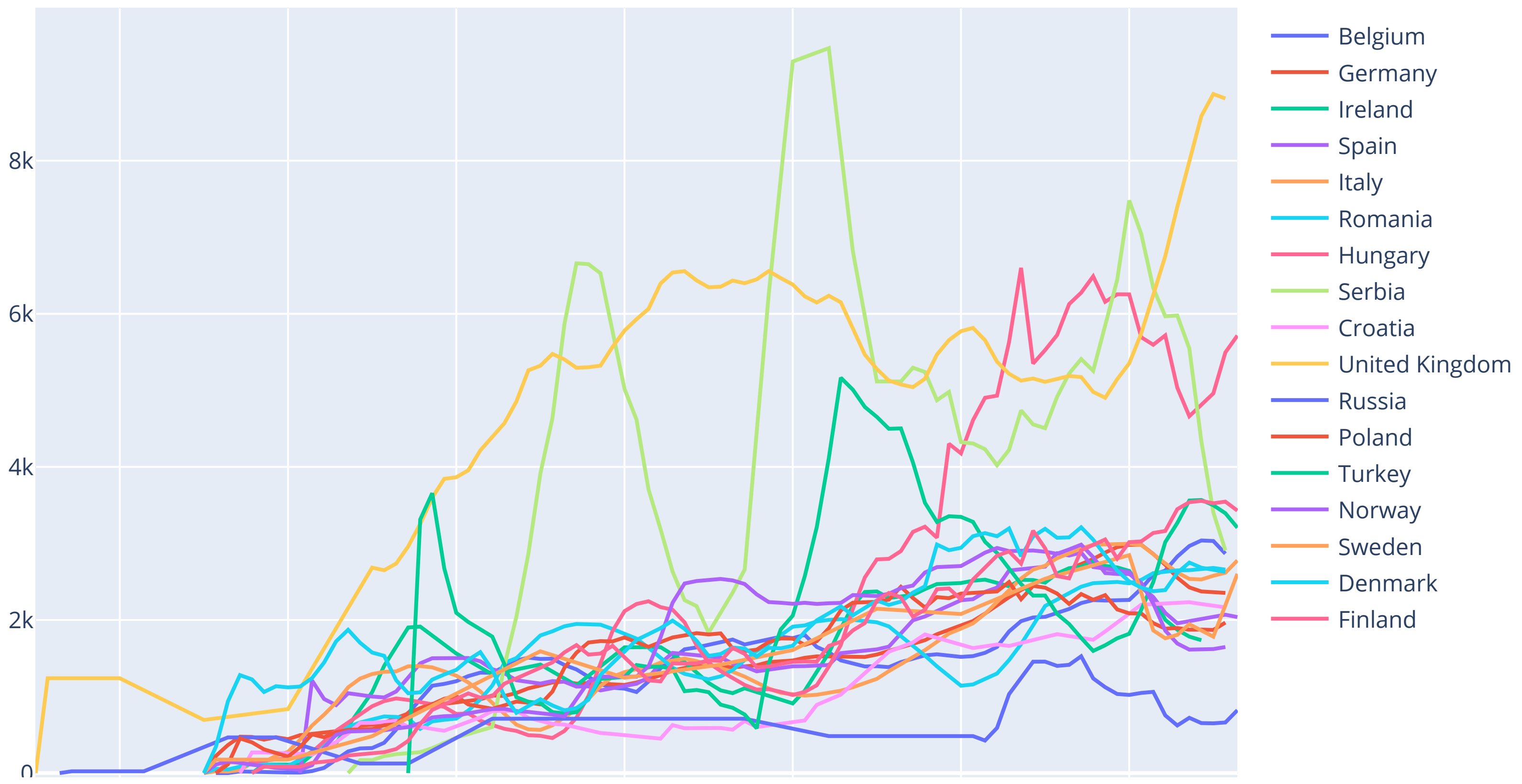
```python
# vaccination progress in europe
# Selected countries: Belgium, Germany, Ireland, Spain, Italy, Romania, Hungary, Serbia, Croatia
# United Kingdom, Russia, Poland, Moldova, Turkey, Norway, Sweden, Denmark, Finland

country_set = ['Belgium', 'Germany', 'Ireland', 'Spain', 'Italy', 'Romania', 'Hungary', 'Serbia', 'Croatia',
               'United Kingdom', 'Russia', 'Poland', 'Turkey', 'Norway', 'Sweden', 'Denmark', 'Finland']

def daily_vaccinations():
    country_fig = go.Figure()
    for country in country_set:
        country_data = df[(df.T != 0).any()]
        country_data = country_data.loc[country_data.country==country]
        country_fig.add_trace(go.Scatter(x=country_data.date, y=country_data.daily_vaccinations, name=country))
    country_fig.update_layout(height=600)
    country_fig.show()

daily_vaccinations()
```

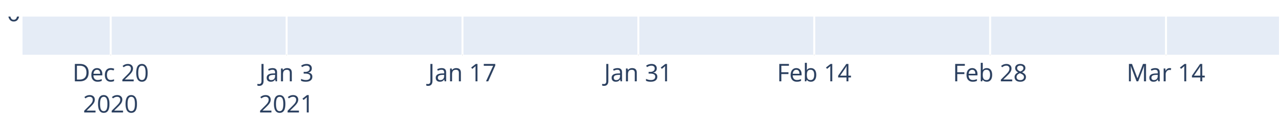


In [14]:

```
# vaccination progress in terms of population
```

```python
def population_vaccinations():
    country_fig = go.Figure()
    for country in country_set:
        country_data = df[(df.T != 0).any()]
        country_data = country_data.loc[country_data.country==country]
        country_fig.add_trace(go.Scatter(x=country_data.date, y=country_data.daily_vaccinations_per_million, name=cou
    country_fig.update_layout(height=600)
    country_fig.show()

population_vaccinations()
```

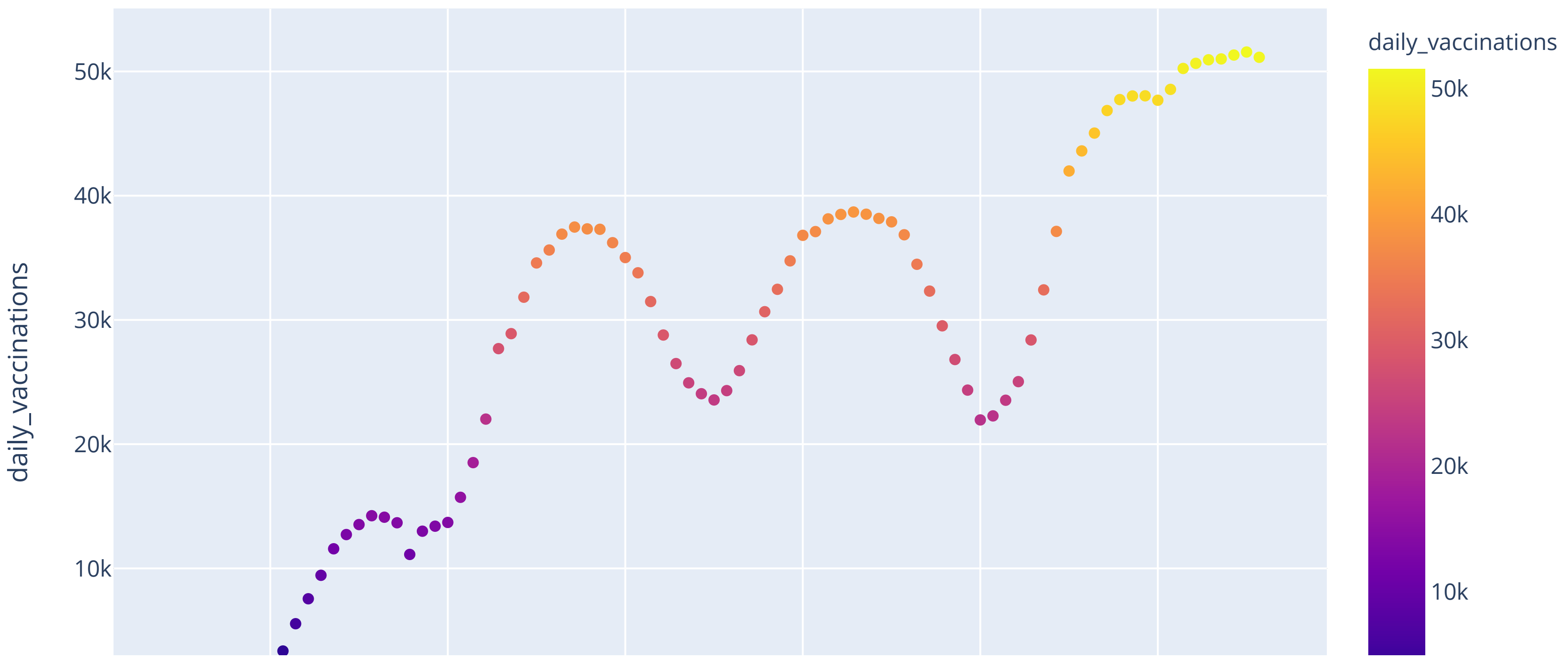


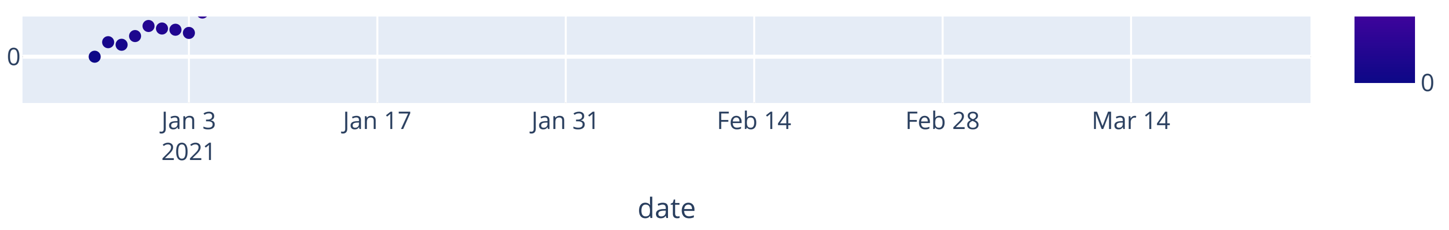


Vaccination progress in Romania

In [15]:

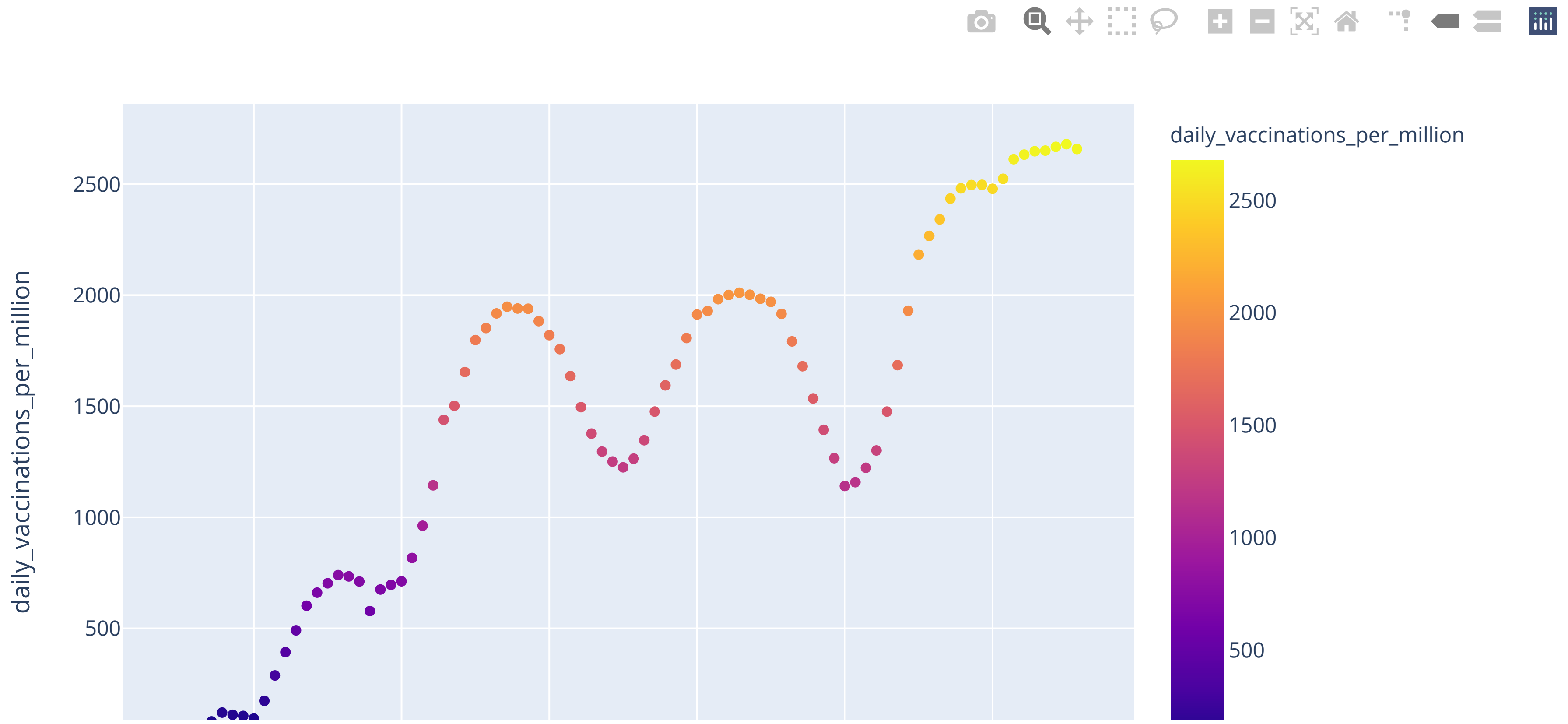
```
# vaccination progress in romania

data = df[(df.T != 0).any()]
data = data.loc[data.country=='Romania']
fig = px.scatter(data, x='date', y='daily_vaccinations', color='daily_vaccinations')
fig.show()
```

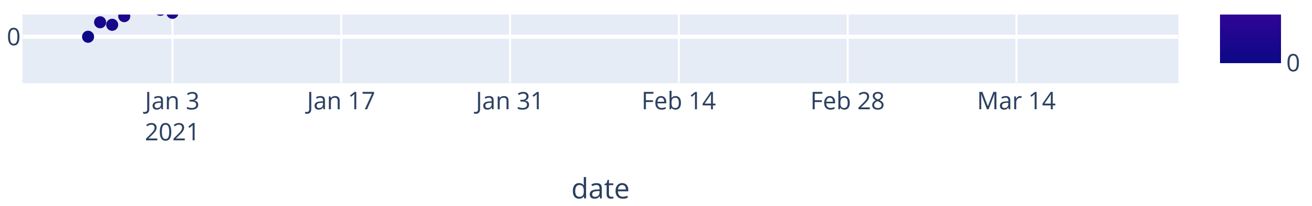




In [16]:

```
# vaccination progress in romania in terms of population

data = df[(df.T != 0).any()]
data = data.loc[data.country=='Romania']
fig = px.scatter(data, x='date', y='daily_vaccinations_per_million', color='daily_vaccinations_per_million')
fig.show()
```




0
Jan 3
2021
Jan 17
Jan 31
Feb 14
Feb 28
Mar 14
date
0