Mini Project: Dataset: "COVID-19 World Vaccination Progress"

In [1]:

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

In [2]:

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

Out[2]:

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

In [3]:

```
df.describe()
```

Out[3]:

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

In [4]:

```
df.dtypes
```

Out[4]:

```
country                                 object
iso_code                                object
date                                    object
total_vaccinations                     float64
people_vaccinated                      float64
people_fully_vaccinated                float64
daily_vaccinations_raw                 float64
daily_vaccinations                     float64
total_vaccinations_per_hundred         float64
people_vaccinated_per_hundred          float64
people_fully_vaccinated_per_hundred    float64
daily_vaccinations_per_million         float64
vaccines                                object
source_name                             object
source_website                          object
dtype: object
```

In [5]:

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

In [6]:

```
df['Total_vaccinations(count)'] = df.groupby("country")['total_vaccinations'].tail(1)
df
```

Out[6]:

	country	iso_code	date	total_vaccinations	people_vaccinated	people_fully_vaccinated	daily_vaccinations_raw	daily_vaccinations	total_vaccin
0	Afghanistan	AFG	2021-02-22	0.0	0.0	NaN	NaN	NaN	
1	Afghanistan	AFG	2021-02-23	NaN	NaN	NaN	NaN	1367.0	
2	Afghanistan	AFG	2021-02-24	NaN	NaN	NaN	NaN	1367.0	
3	Afghanistan	AFG	2021-02-25	NaN	NaN	NaN	NaN	1367.0	
4	Afghanistan	AFG	2021-02-26	NaN	NaN	NaN	NaN	1367.0	
...	...	...	...	...	...	...	...	...	
86507	Zimbabwe	ZWE	2022-03-25	8691642.0	4814582.0	3473523.0	139213.0	69579.0	
86508	Zimbabwe	ZWE	2022-03-26	8791728.0	4886242.0	3487962.0	100086.0	83429.0	
86509	Zimbabwe	ZWE	2022-03-27	8845039.0	4918147.0	3493763.0	53311.0	90629.0	
86510	Zimbabwe	ZWE	2022-03-28	8934360.0	4975433.0	3501493.0	89321.0	100614.0	
86511	Zimbabwe	ZWE	2022-03-29	9039729.0	5053114.0	3510256.0	105369.0	103751.0	

86512 rows × 16 columns

In [7]:

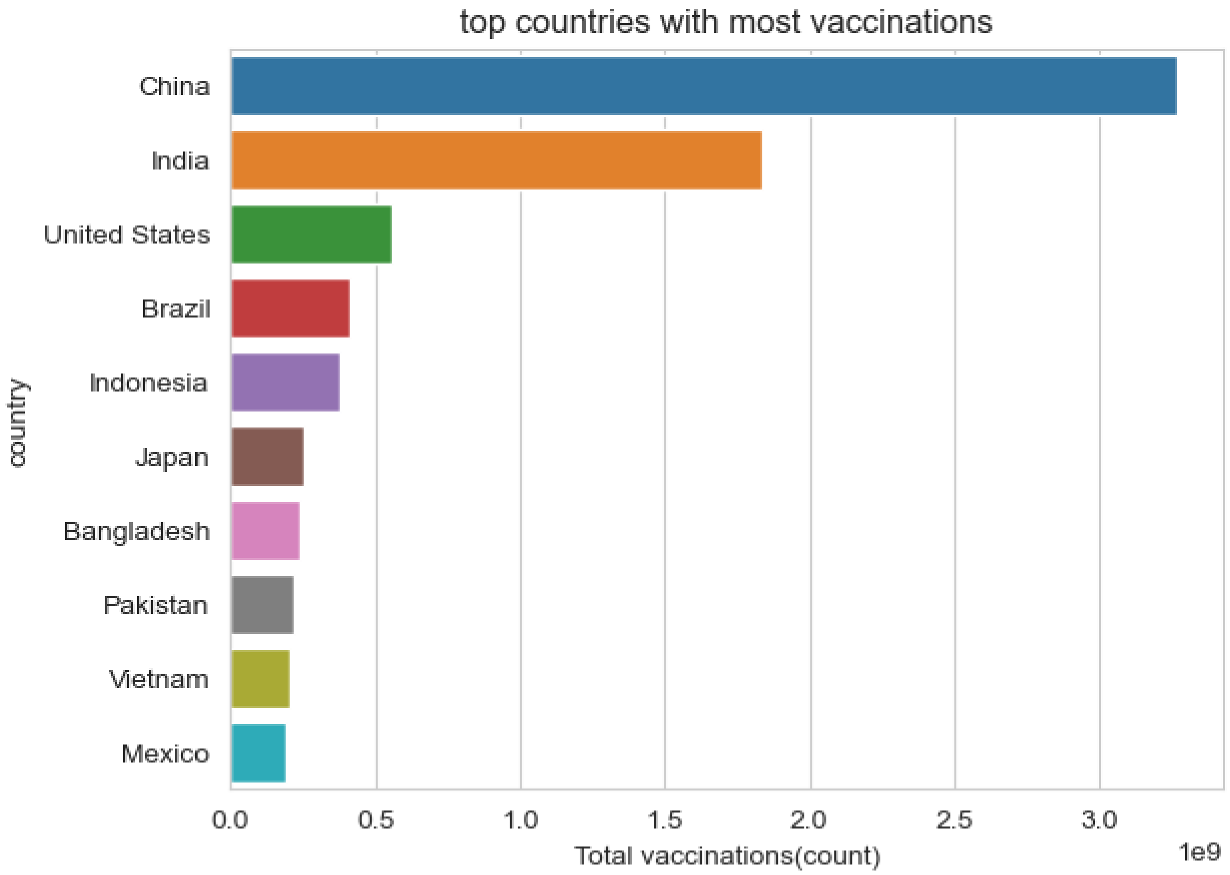
```
# top countries with most vaccines
df.groupby('country')['Total_vaccinations(count)'].mean().sort_values(ascending=False).head(10)
```

Out[7]:

```
country
China            3.263129e+09
India            1.834501e+09
United States    5.601818e+08
Brazil           4.135596e+08
Indonesia        3.771089e+08
Japan            2.543456e+08
Bangladesh       2.436427e+08
Pakistan         2.193686e+08
Vietnam          2.031444e+08
Mexico           1.919079e+08
Name: Total_vaccinations(count), dtype: float64
```

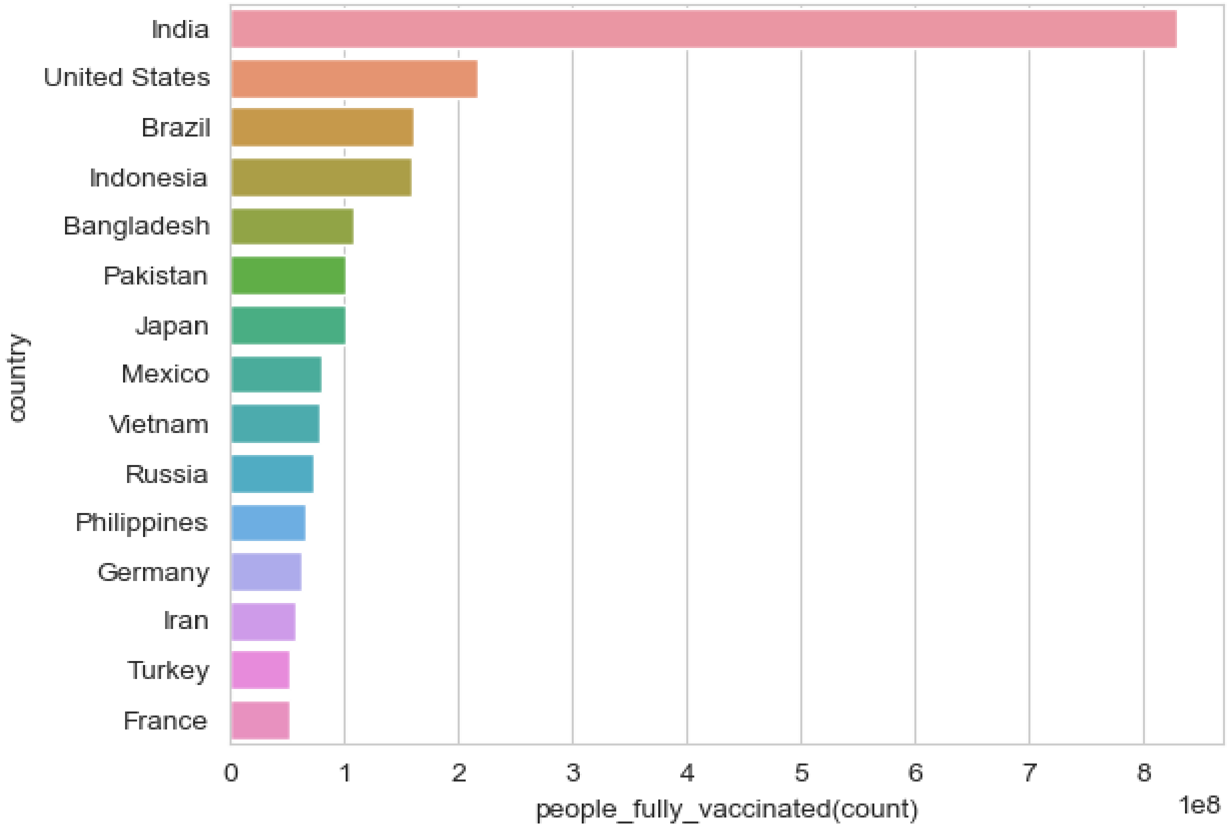
In [8]:

```
#barplot visualization of top countries with most vaccinations
data = df.groupby('country')['Total_vaccinations(count)'].mean().sort_values(ascending=False).head(10)

sns.set_style("whitegrid")
ax = sns.barplot(y = data.index, x = data.values)
ax.set_xlabel('Total vaccinations(count)')
ax.set_title("top countries with most vaccinations")
plt.show()
```



In [9]:

```
#Top countries with fully  vaccinated peoples
df['people_fully_vaccinated(count)'] = df.groupby('country')['people_fully_vaccinated'].tail(1)

data = df.groupby('country')['people_fully_vaccinated(count)'].mean().sort_values(ascending=False).head(15)

ax = sns.barplot(x=data.values, y=data.index)
ax.set_xlabel('people_fully_vaccinated(count)')
plt.show()
```



In [10]:

```
vaccines = df.vaccines.unique()
list(vaccines)
```

```
Out[10]: ['Johnson&Johnson, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing',
 'Oxford/AstraZeneca, Pfizer/BioNTech, Sinovac, Sputnik V',
 'Oxford/AstraZeneca, Sinopharm/Beijing, Sinovac, Sputnik V',
 'Moderna, Oxford/AstraZeneca, Pfizer/BioNTech',
 'Oxford/AstraZeneca',
 'Oxford/AstraZeneca, Pfizer/BioNTech',
 'Oxford/AstraZeneca, Pfizer/BioNTech, Sputnik V',
 'CanSino, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sputnik V',
 'Moderna, Oxford/AstraZeneca, Sinopharm/Beijing, Sinovac, Sputnik V',
 'Pfizer/BioNTech',
 'Johnson&Johnson, Moderna, Novavax, Oxford/AstraZeneca, Pfizer/BioNTech',
 'Johnson&Johnson, Oxford/AstraZeneca, Pfizer/BioNTech',
 'Johnson&Johnson, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sputnik Light, Sputnik V',
 'Johnson&Johnson, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sinovac',
 'Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing',
 'Sinopharm/Beijing, Sputnik V',
 'Johnson&Johnson, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech',
 'Johnson&Johnson, Oxford/AstraZeneca, Pfizer/BioNTech, Sinovac',
 'Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing',
 'Johnson&Johnson, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sputnik V',
 'Moderna, Pfizer/BioNTech',
 'Covaxin, Johnson&Johnson, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinovac',
 'Johnson&Johnson, Oxford/AstraZeneca',
 'Johnson&Johnson, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing',
 'Johnson&Johnson, Oxford/AstraZeneca, Sinopharm/Beijing',
 'Sinopharm/Beijing',
 'Johnson&Johnson, Oxford/AstraZeneca, Sinopharm/Beijing, Sinovac',
 'Covaxin, Oxford/AstraZeneca',
 'CanSino, Oxford/AstraZeneca, Pfizer/BioNTech, Sinovac',
 'CanSino, Sinopharm/Beijing, Sinopharm/Wuhan, Sinovac, ZF2001',
 'Johnson&Johnson, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinovac',
 'Covaxin, Oxford/AstraZeneca, Sinopharm/Beijing',
 'Moderna, Oxford/AstraZeneca, Sinopharm/Beijing, Sputnik V',
 'Abdala, Soberana Plus, Soberana02',
 'Johnson&Johnson, Moderna, Pfizer/BioNTech',
 'Johnson&Johnson, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sinovac, Sputnik V',
 'Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sinovac',
 'Covaxin, Johnson&Johnson, Oxford/AstraZeneca, Sinopharm/Beijing, Sinovac',
 'Johnson&Johnson, Pfizer/BioNTech',
 'Pfizer/BioNTech, Sinopharm/Beijing, Sputnik V',
 'Oxford/AstraZeneca, Sputnik V',
 'Moderna',
 'Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sputnik V',
 'Oxford/AstraZeneca, Sinopharm/Beijing',
 'Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sputnik V',
 'Johnson&Johnson, Moderna',
 'Johnson&Johnson, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sputnik V',
 'Pfizer/BioNTech, Sinovac',
 'Johnson&Johnson, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sputnik V',
 'Covaxin, Oxford/AstraZeneca, Sputnik V',
 'Johnson&Johnson, Moderna, Novavax, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sinovac',
 'COVIran Barekat, Covaxin, FAKHRAVAC, Oxford/AstraZeneca, Razi Cov Pars, Sinopharm/Beijing, Soberana02, SpikoGen, Sputnik V',
 'Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sputnik V',
 'QazVac, Sinopharm/Beijing, Sputnik V',
 'Johnson&Johnson, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sinovac, Sputnik Light, Sputnik V',
 'Johnson&Johnson, Moderna, Novavax, Pfizer/BioNTech',
 'Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sinovac, Sputnik V',
 'Pfizer/BioNTech, Sinopharm/Beijing',
 'CanSino, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sinovac',
 'CanSino, Johnson&Johnson, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinovac, Sputnik V',
 'Abdala, Johnson&Johnson, Oxford/AstraZeneca, Pfizer/BioNTech, Soberana02, Sputnik Light, Sputnik V',
 'Oxford/AstraZeneca, Pfizer/BioNTech, Sinovac',
 'CanSino, Covaxin, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sinovac, Sputnik V',
 'Johnson&Johnson, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sinovac, Sputnik Light, Sputnik V',
 'Covaxin, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sinovac, Sputnik V',
 'EpiVacCorona, Sputnik V',
 'Johnson&Johnson, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sinovac, Sputnik V',
 'Pfizer/BioNTech, Sputnik V',
 'Oxford/AstraZeneca, Sinopharm/Beijing, Sputnik V',
 'Moderna, Pfizer/BioNTech, Sinopharm/Beijing, Sinovac',
 'Johnson&Johnson, Moderna, Novavax, Oxford/AstraZeneca, Pfizer/BioNTech, Sputnik V',
 'Johnson&Johnson, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sinovac',
 'Johnson&Johnson, Oxford/AstraZeneca, Sinopharm/Beijing, Sinovac, Sputnik Light, Sputnik V',
 'Medigen, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech',
 'Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinovac, Sputnik V',
 'Johnson&Johnson, Pfizer/BioNTech, Sinopharm/Beijing',
 'Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sinovac',
 'Pfizer/BioNTech, Sinovac, Turkovac',
 'EpiVacCorona, Oxford/AstraZeneca, QazVac, Sinopharm/Beijing, Sputnik V, ZF2001',
 'Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sinopharm/Wuhan, Sputnik V',
 'Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinovac, Sputnik Light, Sputnik V, ZF2001',
 'Abdala, Sinopharm/Beijing, Sinovac, Soberana02, Sputnik Light, Sputnik V',
 'Abdala, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sputnik V',
 'Johnson&Johnson, Oxford/AstraZeneca, Sinovac']
```

In [11]:

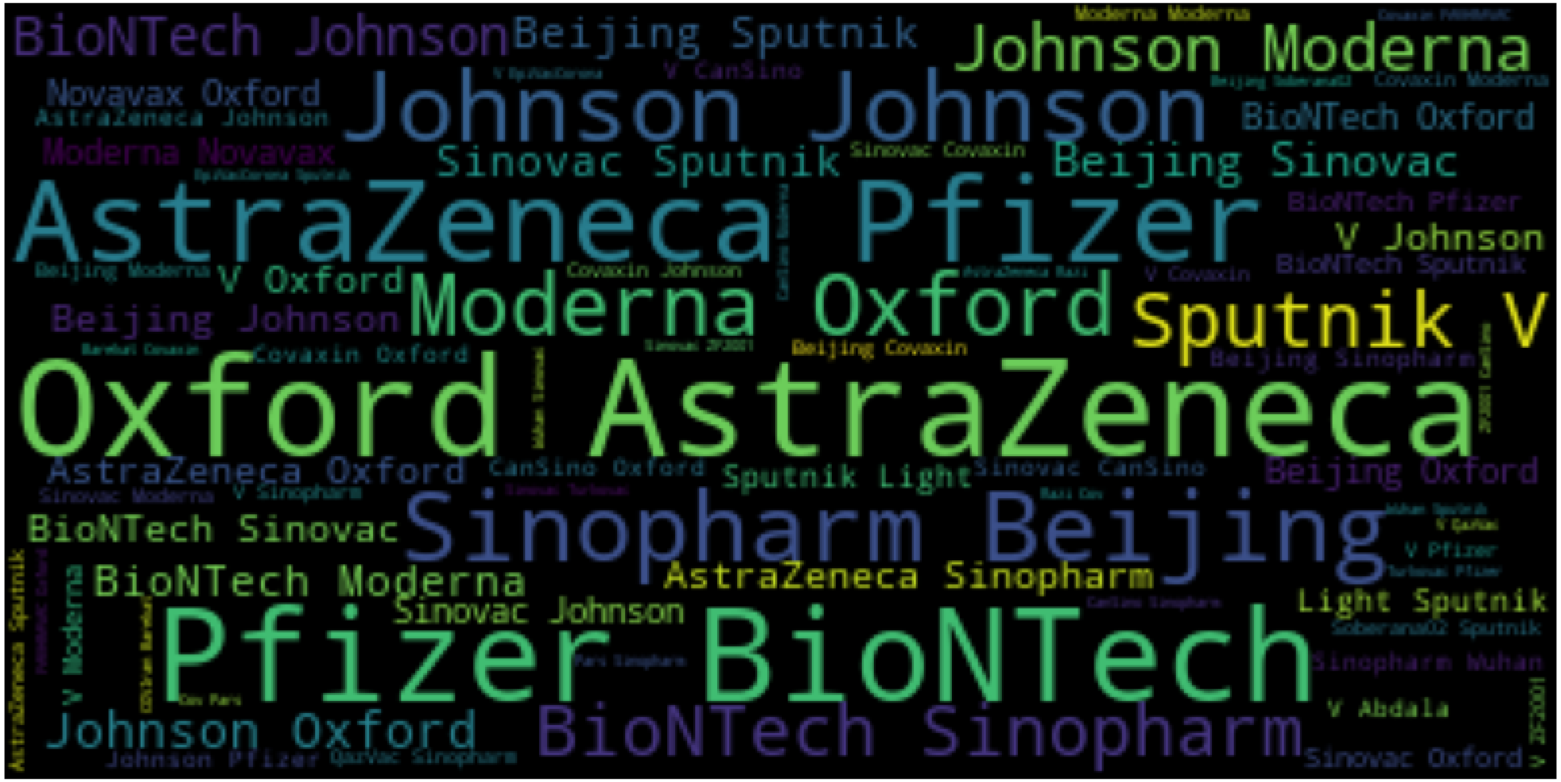
```
#most common vaccines
df.vaccines.value_counts(ascending=False).head(10)
```

Out[11]:

```
Johnson&Johnson, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech             7608
Moderna, Oxford/AstraZeneca, Pfizer/BioNTech                              6263
Oxford/AstraZeneca                                                        6022
Oxford/AstraZeneca, Pfizer/BioNTech                                       4629
Johnson&Johnson, Moderna, Novavax, Oxford/AstraZeneca, Pfizer/BioNTech    3564
Johnson&Johnson, Oxford/AstraZeneca, Sinopharm/Beijing                    2484
Moderna, Pfizer/BioNTech                                                  2309
Pfizer/BioNTech                                                           2271
Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sputnik V         2041
Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing                    2030
Name: vaccines, dtype: int64
```

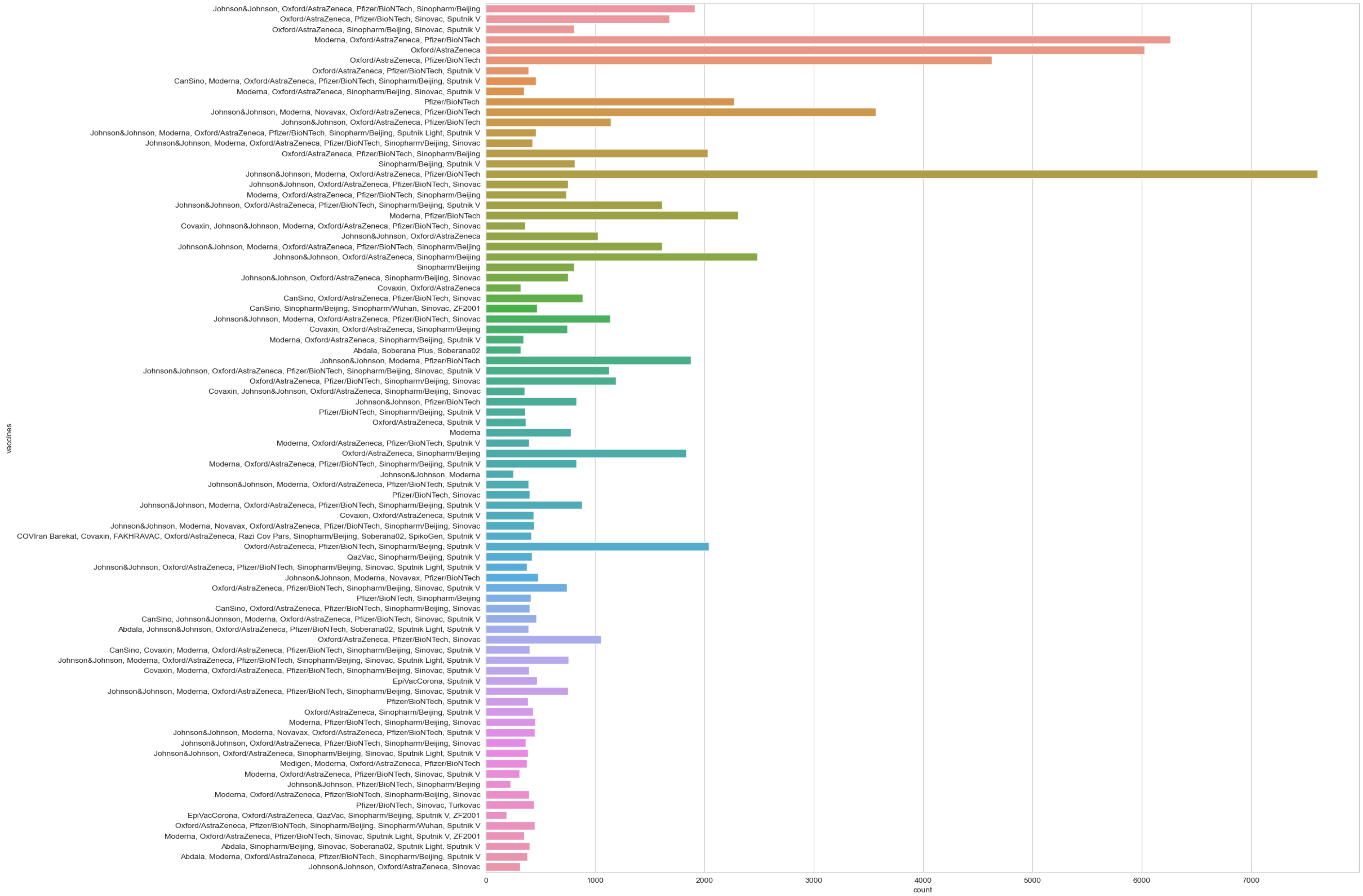
In [12]:

```python
from wordcloud import WordCloud

plt.figure(figsize=(10, 10))
words = " ".join(df.vaccines)

image = WordCloud().generate_from_text(words)
plt.imshow(image)
plt.axis("off")
plt.show()
```



```
In [13]: plt.figure(figsize=(20,20))
         sns.countplot(data=df, y='vaccines')
```

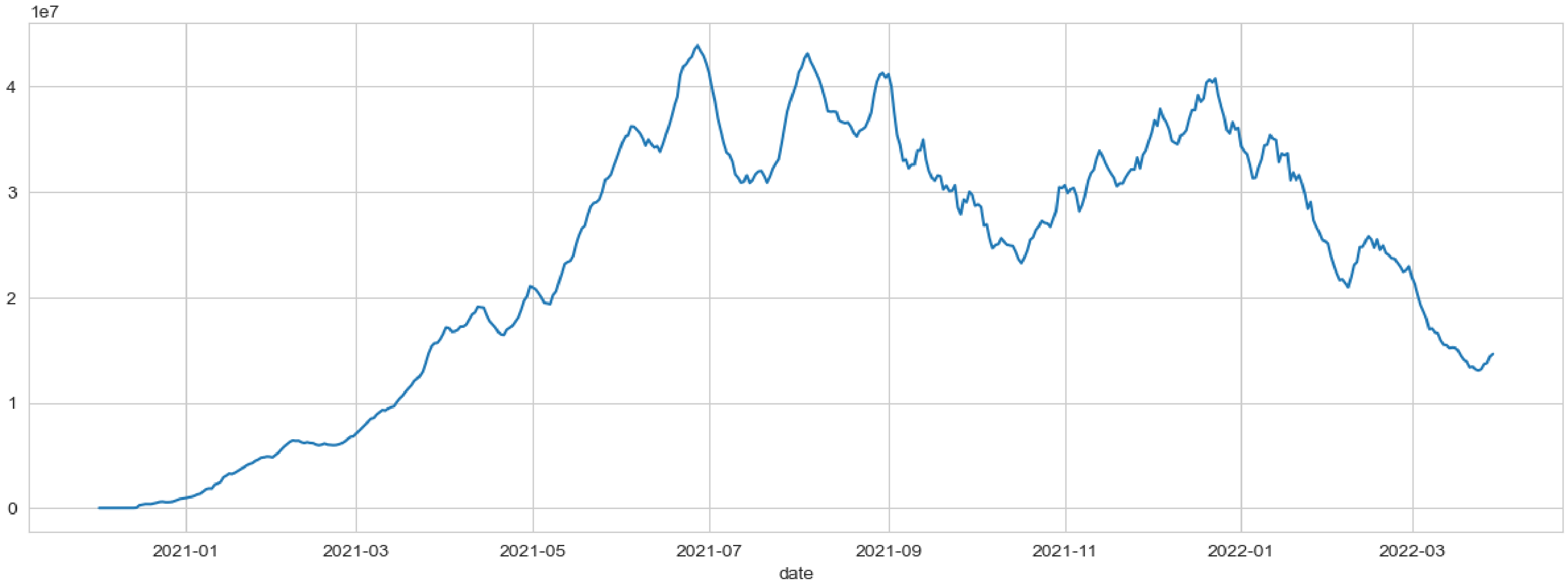
```
Out[13]: <Axes: xlabel='count', ylabel='vaccines'>
```



In [14]:
```
#daily vaccinations
data = df.groupby('date')['daily_vaccinations'].sum()

plt.figure(figsize=(15, 5))
sns.lineplot(data=data, x=data.index, y=data.values)
```

Out[14]:

```
<Axes: xlabel='date'>
```

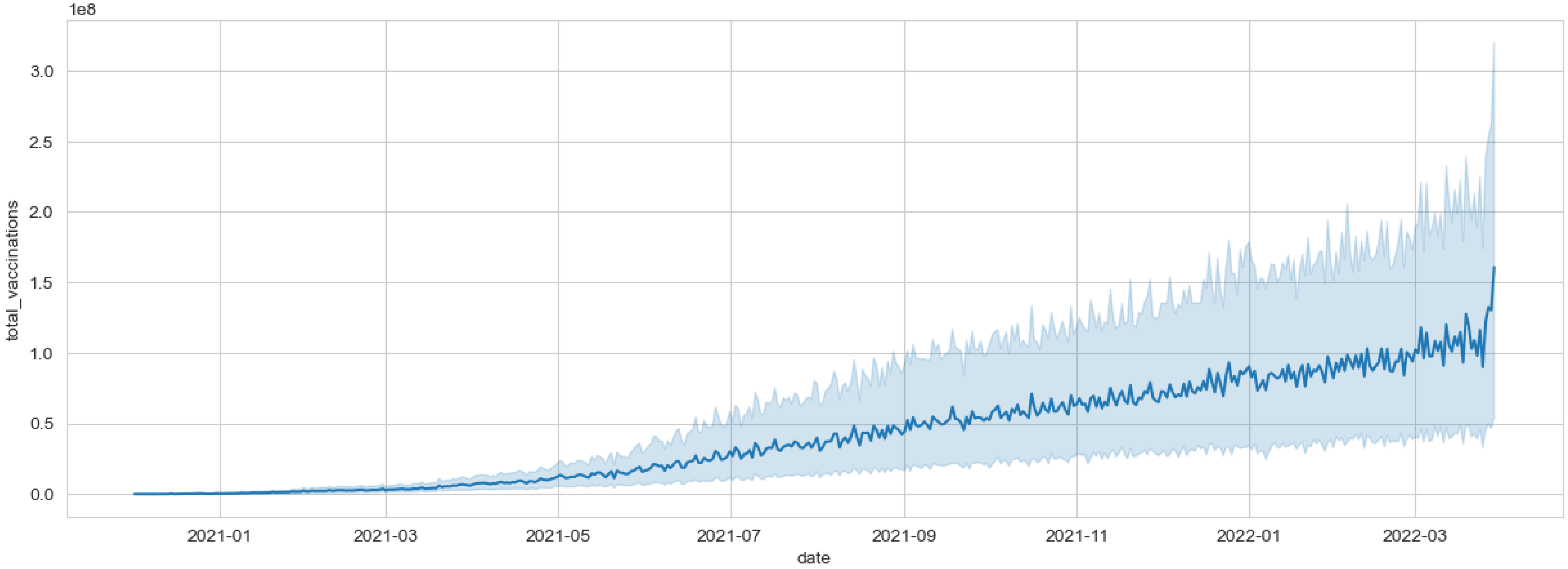


In [15]:

```
#total vaccinations
plt.figure(figsize=(15,5))
sns.lineplot(data=df, x='date', y='total_vaccinations')
```

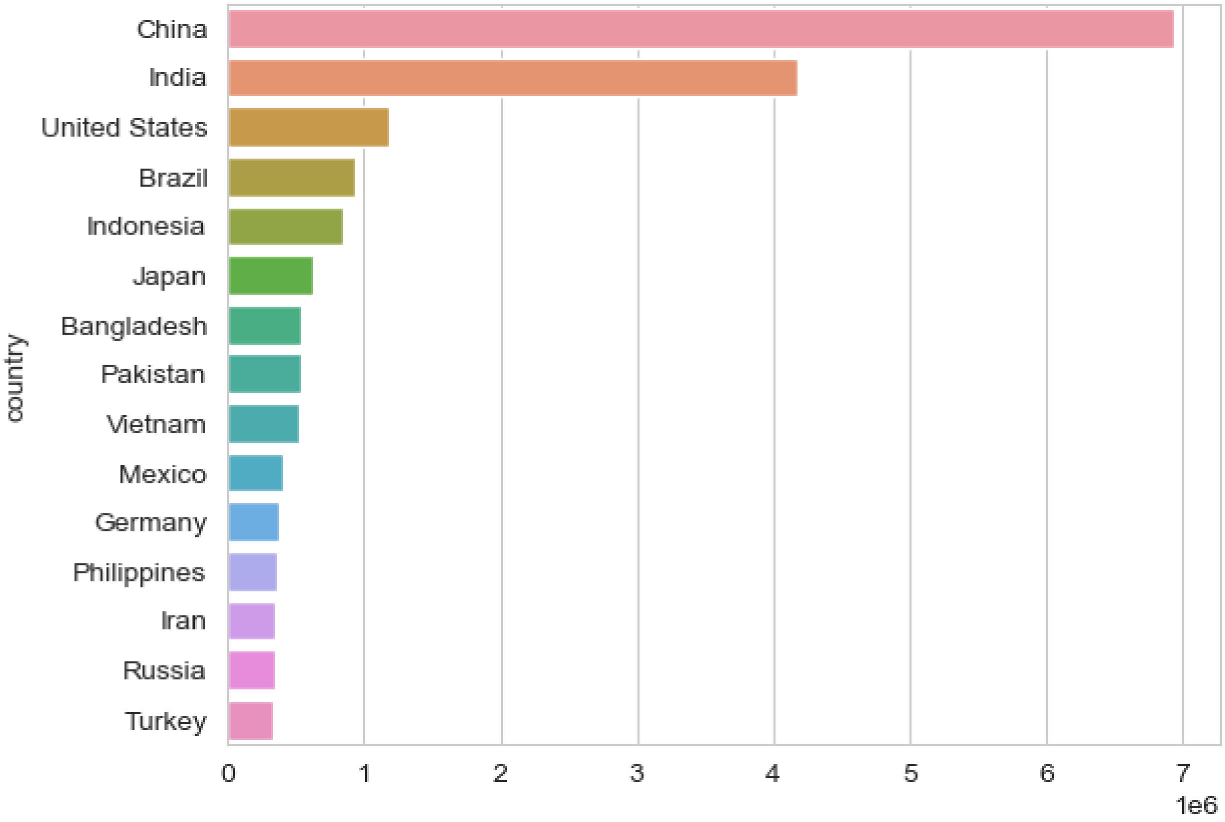
Out[15]:

```
<Axes: xlabel='date', ylabel='total_vaccinations'>
```



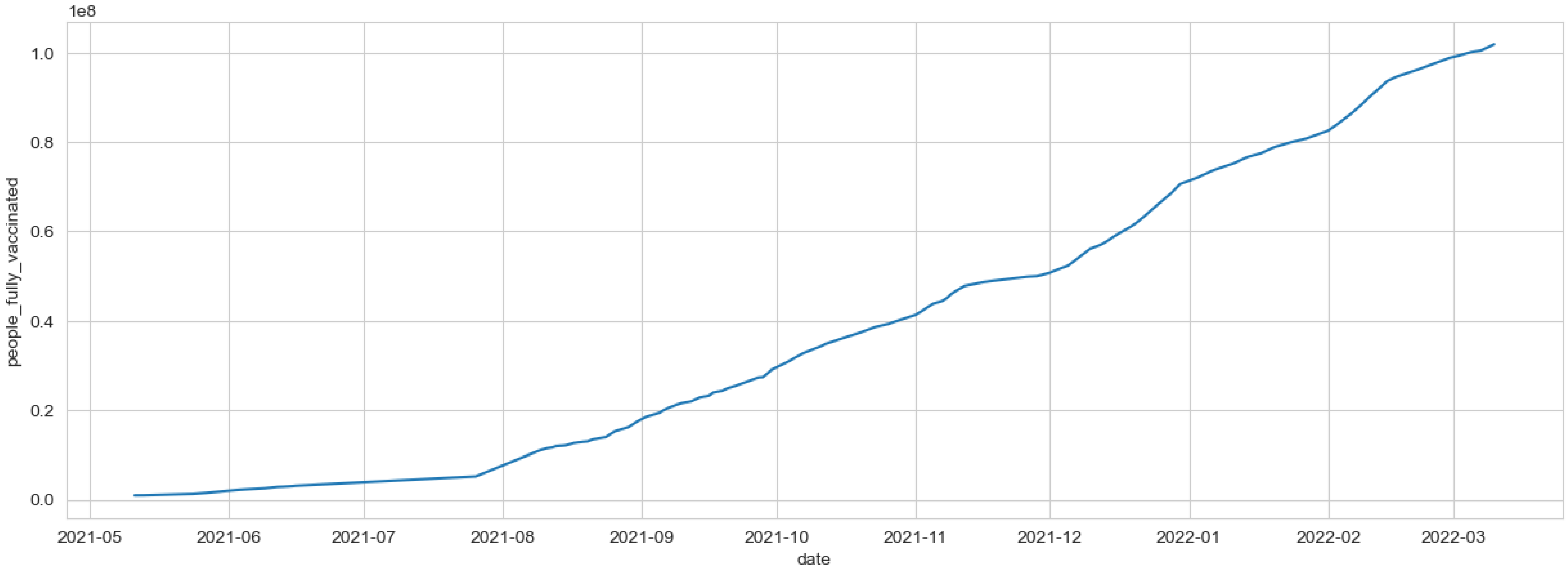
In [16]:

```
#Countries with best daily average vaccinations
data = df.groupby('country')['daily_vaccinations'].mean().sort_values(ascending=False).head(15)

sns.barplot(y=data.index, x=data.values)
plt.show()
```



In [17]:
```
# vaccinations in Pakistan
plt.figure(figsize=(15, 5))
sns.lineplot(x='date', y='people_fully_vaccinated', data=df[df.country=='Pakistan'])
```

Out[17]:
```
<Axes: xlabel='date', ylabel='people_fully_vaccinated'>
```

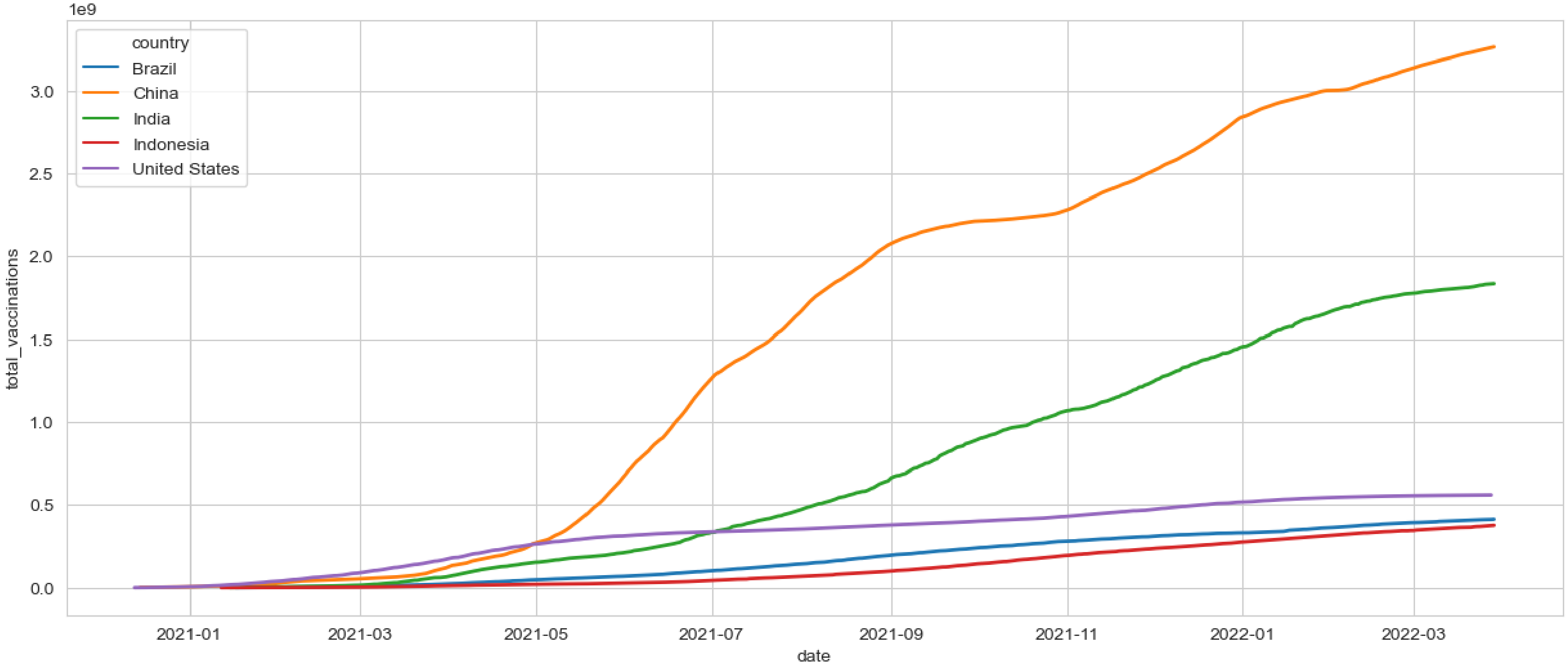


In [18]:
```python
# top preferred vaccines in pakistan
df['vaccines'].value_counts(ascending=False).head(1).index.to_list()
```

Out[18]:
```
['Johnson&Johnson, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech']
```

In [19]:
```python
#COMPARING TOP 5 COUNTRIES WITH MOST VACCINATIONS
top_countries = list(df.groupby('country')['total_vaccinations'].mean().sort_values(ascending=False).head().index)
data = df.loc[df.country.apply(lambda x: x in top_countries)]

plt.figure(figsize=(15, 6))
sns.lineplot(x='date', y='total_vaccinations', data=data, hue='country', lw=2)
```

Out[19]:
```
<Axes: xlabel='date', ylabel='total_vaccinations'>
```

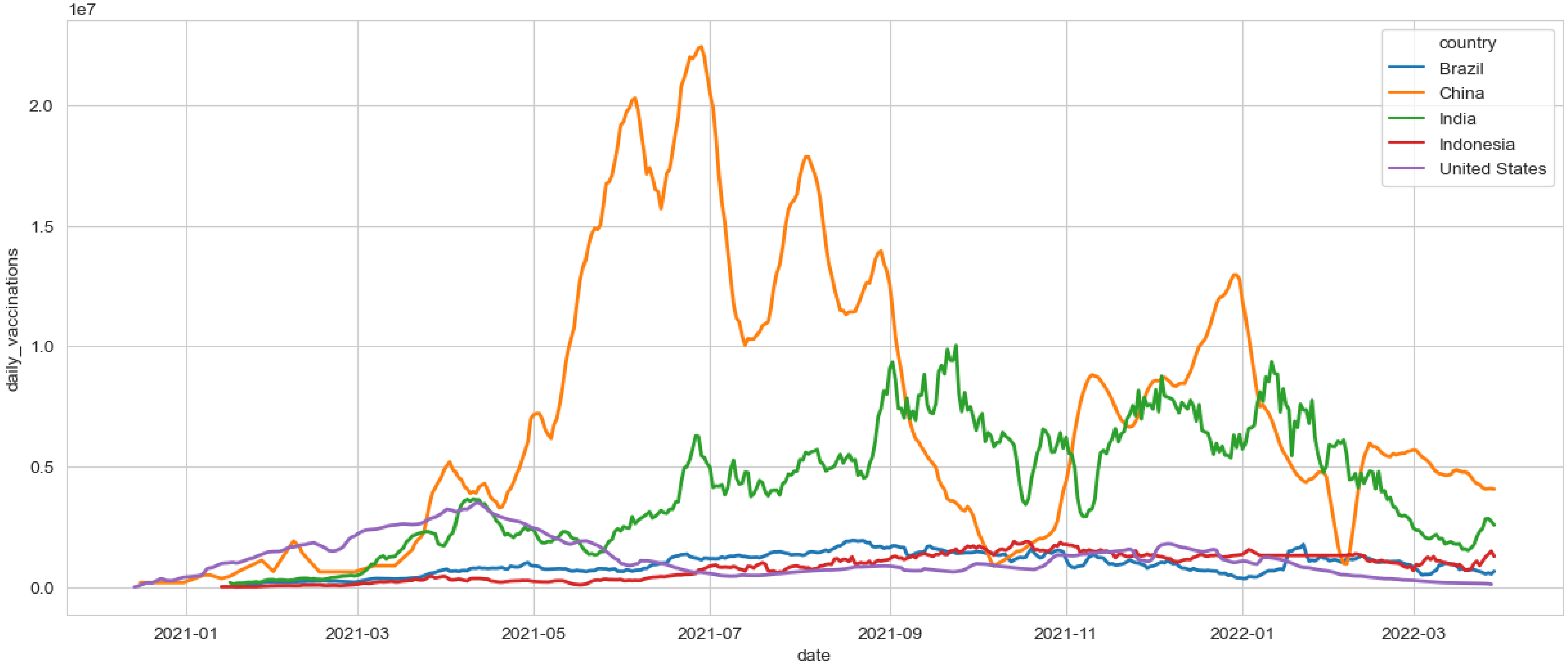


In [20]:

```
#daily vaccination comparison
plt.figure(figsize=(15, 6))
sns.lineplot(x='date', y='daily_vaccinations', data=data, hue='country', lw=2)
```

Out[20]:

```
<Axes: xlabel='date', ylabel='daily_vaccinations'>
```



In [21]:

```
from plotly.offline import init_notebook_mode, iplot
import plotly.express as px
import plotly as py
import plotly.graph_objs as go

init_notebook_mode(connected=True)
```

In [22]:

```
df['date'] = df['date'].dt.strftime("%Y-%m-%d")
df
```

Out[22]:

	country	iso_code	date	total_vaccinations	people_vaccinated	people_fully_vaccinated	daily_vaccinations_raw	daily_vaccinations	total_vaccin
0	Afghanistan	AFG	2021-02-22	0.0	0.0	NaN	NaN	NaN	
1	Afghanistan	AFG	2021-02-23	NaN	NaN	NaN	NaN	1367.0	
2	Afghanistan	AFG	2021-02-24	NaN	NaN	NaN	NaN	1367.0	
3	Afghanistan	AFG	2021-02-25	NaN	NaN	NaN	NaN	1367.0	
4	Afghanistan	AFG	2021-02-26	NaN	NaN	NaN	NaN	1367.0	
...	...	...	...	...	...	...	...	...	
86507	Zimbabwe	ZWE	2022-03-25	8691642.0	4814582.0	3473523.0	139213.0	69579.0	
86508	Zimbabwe	ZWE	2022-03-26	8791728.0	4886242.0	3487962.0	100086.0	83429.0	
86509	Zimbabwe	ZWE	2022-03-27	8845039.0	4918147.0	3493763.0	53311.0	90629.0	
86510	Zimbabwe	ZWE	2022-03-28	8934360.0	4975433.0	3501493.0	89321.0	100614.0	
86511	Zimbabwe	ZWE	2022-03-29	9039729.0	5053114.0	3510256.0	105369.0	103751.0	

86512 rows × 17 columns

In [23]:
```
df
```

Out[23]:

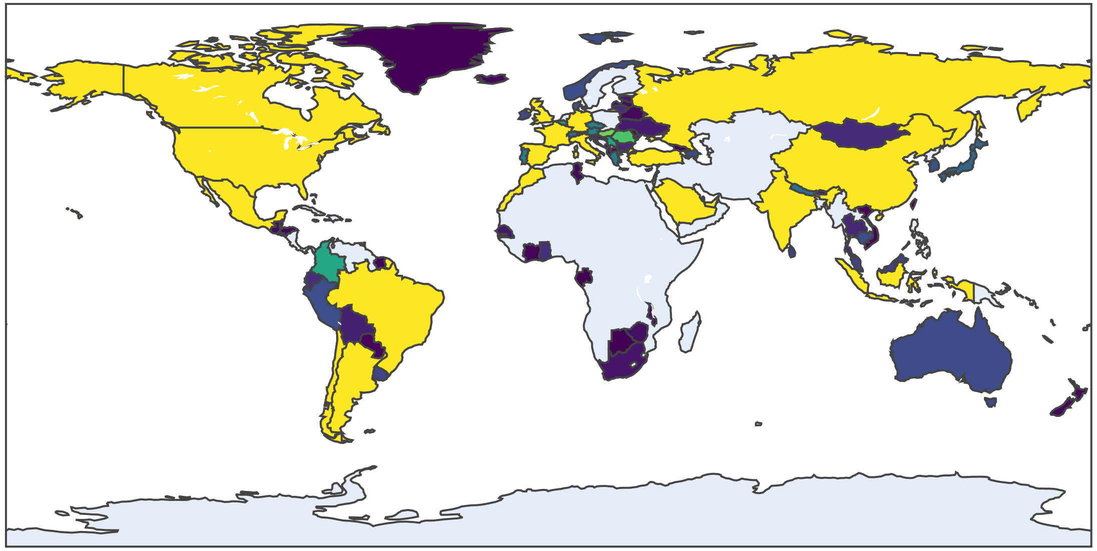
	country	iso_code	date	total_vaccinations	people_vaccinated	people_fully_vaccinated	daily_vaccinations_raw	daily_vaccinations	total_vaccin
0	Afghanistan	AFG	2021-02-22	0.0	0.0	NaN	NaN	NaN	
1	Afghanistan	AFG	2021-02-23	NaN	NaN	NaN	NaN	1367.0	
2	Afghanistan	AFG	2021-02-24	NaN	NaN	NaN	NaN	1367.0	
3	Afghanistan	AFG	2021-02-25	NaN	NaN	NaN	NaN	1367.0	
4	Afghanistan	AFG	2021-02-26	NaN	NaN	NaN	NaN	1367.0	
...	...	...	...	...	...	...	...	...	
86507	Zimbabwe	ZWE	2022-03-25	8691642.0	4814582.0	3473523.0	139213.0	69579.0	
86508	Zimbabwe	ZWE	2022-03-26	8791728.0	4886242.0	3487962.0	100086.0	83429.0	
86509	Zimbabwe	ZWE	2022-03-27	8845039.0	4918147.0	3493763.0	53311.0	90629.0	
86510	Zimbabwe	ZWE	2022-03-28	8934360.0	4975433.0	3501493.0	89321.0	100614.0	
86511	Zimbabwe	ZWE	2022-03-29	9039729.0	5053114.0	3510256.0	105369.0	103751.0	

86512 rows × 17 columns

In [24]:

```
fig = px.choropleth(
    df.sort_values(by='date', ascending=True),
    locations='iso_code',
    color='total_vaccinations',
    title='Number of people vaccinated',
    color_continuous_scale='viridis',
    animation_frame='date',
    range_color=[0,5000000]
)

fig.show()
```

Number of people vaccinated

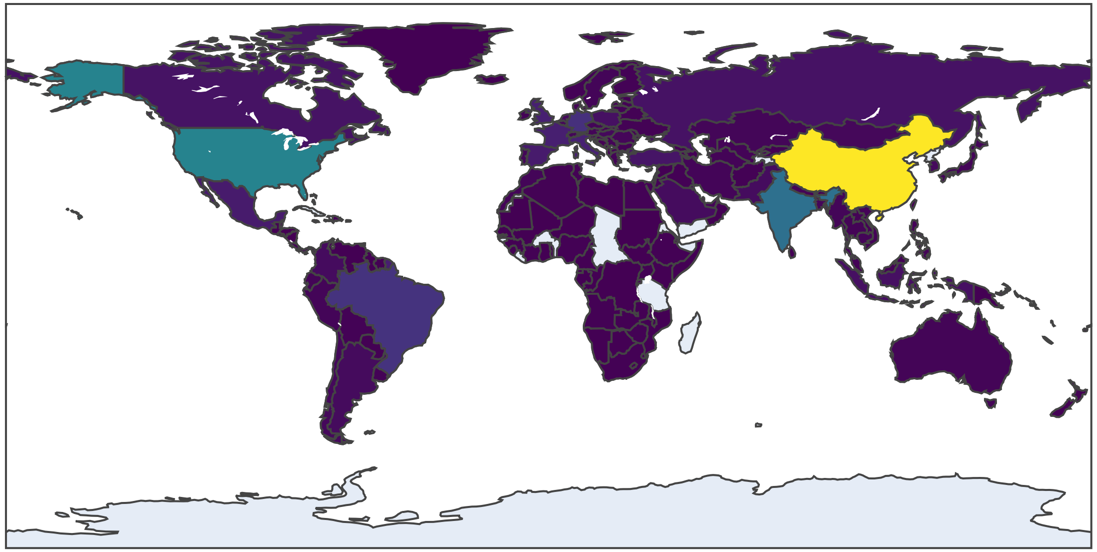


In [25]:

```
fig = px.choropleth(
    data_frame=df.sort_values(by='date', ascending=True),
    locations='iso_code',
    color='daily_vaccinations',
    title='Number of daily vaccinations',
    color_continuous_scale='viridis',
    animation_frame='date',
    range_color=[0,5000000]
)
```

```
fig.show()
```

Number of daily vaccinations



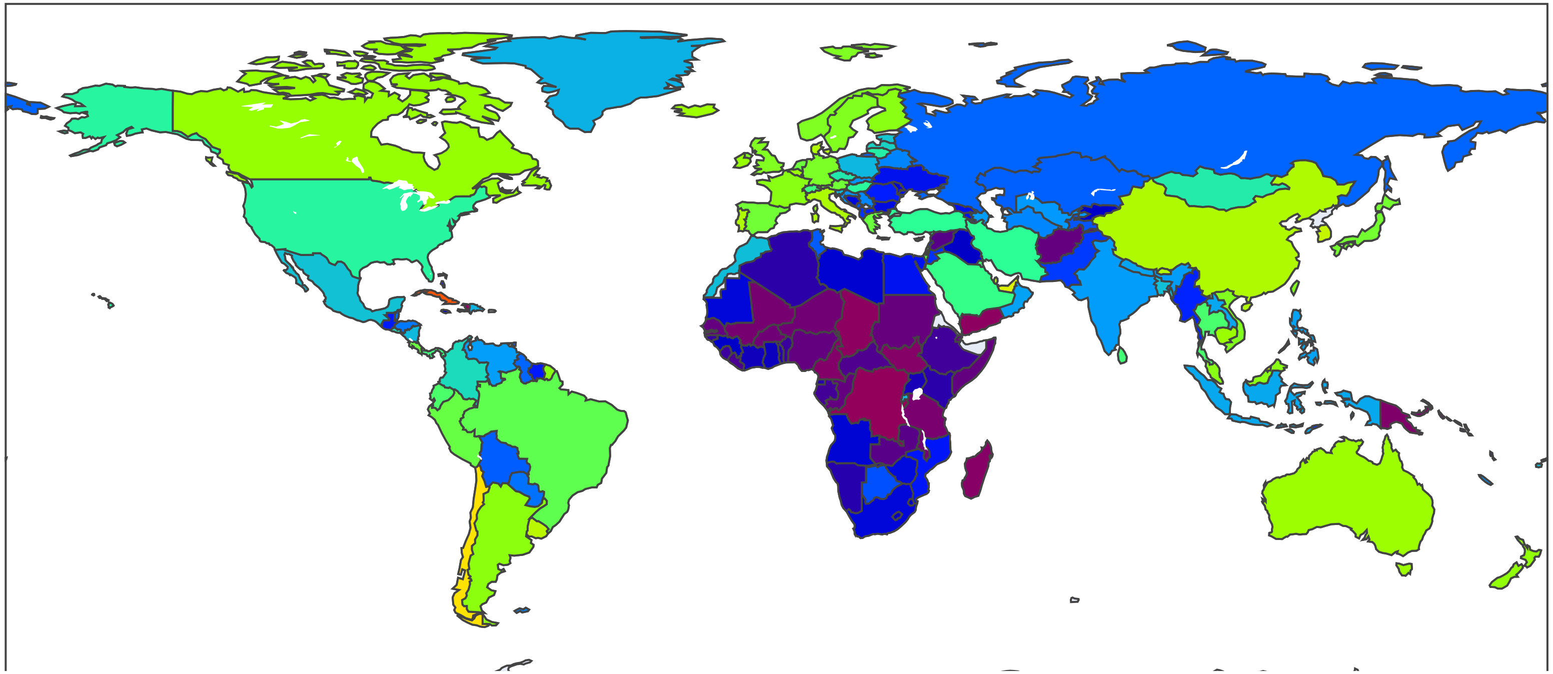
In [32]:
```
vacc_data = df[["country","iso_code","total_vaccinations","people_vaccinated","total_vaccinations_per_hundred","vaccines",
                "daily_vaccinations"]].groupby("country").max().sort_values(by='total_vaccinations', ascending=True)

vacc_data.head()
```

Out[32]:

	iso_code	total_vaccinations	people_vaccinated	total_vaccinations_per_hundred	vaccines	daily_vaccinations
country						
Pitcairn	PCN	94.0	47.0	200.00	Oxford/AstraZeneca	1.0
Tokelau	TKL	1936.0	968.0	141.52	Pfizer/BioNTech	23.0
Niue	NIU	4161.0	1650.0	257.81	Pfizer/BioNTech	87.0
Montserrat	MSR	4211.0	1897.0	84.54	Oxford/AstraZeneca	53.0
Falkland Islands	FLK	4407.0	2632.0	124.91	Oxford/AstraZeneca	189.0

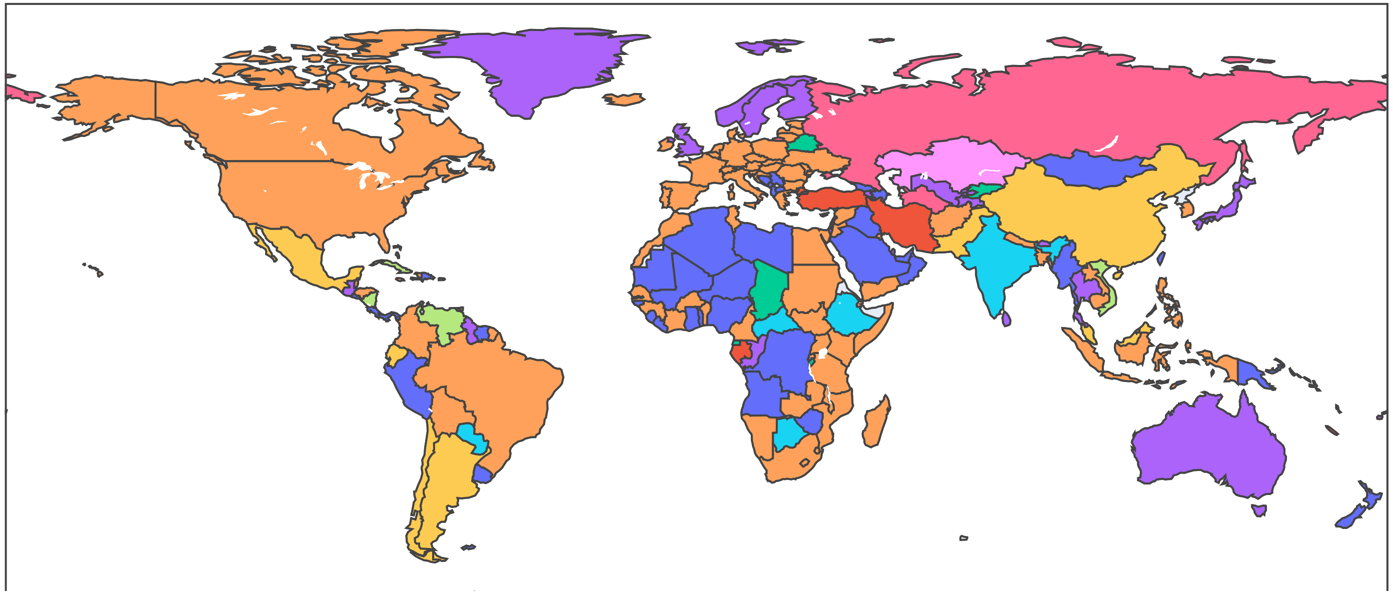
In [35]:

```
fig = px.choropleth(
    vacc_data,
    locations='iso_code',
    color='total_vaccinations_per_hundred',
    color_continuous_scale='rainbow'
)
fig.show()
```



```
In [39]: fig = px.choropleth(
             locations=vacc_data.iso_code,
             color=vacc_data.vaccines.apply(lambda x: x.split(",")[0]),
             title='Name of Vaccine'
         )

         fig.show()
```

Name of Vaccine



```
In [59]: data = df.groupby(["country", "iso_code", "vaccines"])['total_vaccinations'].max().reset_index()
         data
```

Out[59]:

	country	iso_code	vaccines	total_vaccinations
0	Afghanistan	AFG	Johnson&Johnson, Oxford/AstraZeneca, Pfizer/Bi...	5751015.0
1	Albania	ALB	Oxford/AstraZeneca, Pfizer/BioNTech, Sinovac, ...	2754244.0
2	Algeria	DZA	Oxford/AstraZeneca, Sinopharm/Beijing, Sinovac...	13704895.0
3	Andorra	AND	Moderna, Oxford/AstraZeneca, Pfizer/BioNTech	151997.0
4	Angola	AGO	Oxford/AstraZeneca	17535411.0
...	...	...	...	...
218	Wales	OWID_WLS	Moderna, Oxford/AstraZeneca, Pfizer/BioNTech	6927437.0
219	Wallis and Futuna	WLF	Moderna	13073.0
220	Yemen	YEM	Johnson&Johnson, Oxford/AstraZeneca, Sinovac	807502.0
221	Zambia	ZMB	Johnson&Johnson, Oxford/AstraZeneca, Sinopharm...	3402612.0
222	Zimbabwe	ZWE	Oxford/AstraZeneca, Sinopharm/Beijing, Sinovac...	9039729.0

223 rows × 4 columns

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: