This Notebook aims to determine and check the details of vaccination on among countries

In [2]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
import seaborn as sns
plt.rcParams['figure.figsize'] = 10, 12
import warnings
warnings.filterwarnings('ignore')
```

Import the Dataset

In [3]:
```
df= pd.read_csv("country_vaccinations.csv")
df.head()
```

Out[3]:

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

Data Description

1. country = The name of the country
2. iso_code = ISO code for the country
3. Date = Date for the data entry
4. Total number of vaccinations = This is the absolute number of total vaccinations in the country
5. Total number of people vaccinated = The number of people vaccinated.
6. Total number of people fully vaccinated = The number of people fully vaccinated(may be 2-3 doses).
7. Daily vaccinations (raw) = For a certain data entry, the number of vaccination for that date/country
8. Daily vaccinations = for a certain data entry, the number of vaccination for that date/country;
9. Total vaccinations per hundred = ratio (in percent) between vaccination number and total population up to the date in the country;
10. Total number of people vaccinated per hundred = ratio (in percent) between population immunized and total population up to the date in the country;
11. 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;
12. Number of vaccinations per day = number of daily vaccination for that day and country
13. Daily vaccinations per million = ratio (in ppm) between vaccination number and total population for the current date in the country
14. Vaccines used in the country = total number of vaccines used in the country (up to date);

15. Source name = source of the information (national authority, international organization, local organization etc.);
16. Source website = website of the source of information;

In [4]:
```
df.columns
```

Out[4]:
```
Index(['country', 'iso_code', 'date', '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', 'daily_vaccinations_per_millio
n',
       'vaccines', 'source_name', 'source_website'],
      dtype='object')
```

In [5]:
```
# Let's check the information in this dataset
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 24620 entries, 0 to 24619
Data columns (total 15 columns):
 #   Column                               Non-Null Count  Dtype
---  ------                               --------------  -----
 0   country                              24620 non-null  object
 1   iso_code                             24620 non-null  object
 2   date                                 24620 non-null  object
 3   total_vaccinations                   13858 non-null  float64
 4   people_vaccinated                    13060 non-null  float64
 5   people_fully_vaccinated              10392 non-null  float64
 6   daily_vaccinations_raw               11461 non-null  float64
 7   daily_vaccinations                   24389 non-null  float64
 8   total_vaccinations_per_hundred       13858 non-null  float64
 9   people_vaccinated_per_hundred        13060 non-null  float64
 10  people_fully_vaccinated_per_hundred  10392 non-null  float64
 11  daily_vaccinations_per_million       24389 non-null  float64
 12  vaccines                             24620 non-null  object
 13  source_name                          24620 non-null  object
 14  source_website                       24620 non-null  object
dtypes: float64(9), object(6)
memory usage: 2.8+ MB
```

In [6]:
```
df.shape
```

Out[6]:
```
(24620, 15)
```

In [7]:
```
# Check the NULL VAlues
df.isnull().sum()
```

Out[7]:
```
country                                0
iso_code                               0
date                                   0
total_vaccinations                 10762
people_vaccinated                  11560
people_fully_vaccinated            14228
daily_vaccinations_raw             13159
daily_vaccinations                   231
total_vaccinations_per_hundred     10762
people_vaccinated_per_hundred      11560
people_fully_vaccinated_per_hundred    14228
daily_vaccinations_per_million       231
vaccines                               0
source_name                            0
source_website                         0
dtype: int64
```

In [8]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 24620 entries, 0 to 24619
Data columns (total 15 columns):
 #   Column                                Non-Null Count  Dtype  
---  ------                                --------------  -----  
 0   country                               24620 non-null  object 
 1   iso_code                              24620 non-null  object 
 2   date                                  24620 non-null  object 
 3   total_vaccinations                    13858 non-null  float64
 4   people_vaccinated                     13060 non-null  float64
 5   people_fully_vaccinated               10392 non-null  float64
 6   daily_vaccinations_raw                11461 non-null  float64
 7   daily_vaccinations                    24389 non-null  float64
 8   total_vaccinations_per_hundred        13858 non-null  float64
 9   people_vaccinated_per_hundred         13060 non-null  float64
 10  people_fully_vaccinated_per_hundred   10392 non-null  float64
 11  daily_vaccinations_per_million        24389 non-null  float64
 12  vaccines                              24620 non-null  object 
 13  source_name                           24620 non-null  object 
 14  source_website                        24620 non-null  object 
dtypes: float64(9), object(6)
memory usage: 2.8+ MB
```

In [9]:
```
#converting date to date
df['date'] = pd.to_datetime(df['date'])
df.fillna(0,inplace=True)
df.people_vaccinated =df.people_vaccinated.astype(int)
df.people_fully_vaccinated=df.people_fully_vaccinated.astype(int)
df.daily_vaccinations = df.daily_vaccinations.astype(int)
df.total_vaccinations = df.total_vaccinations.astype(int)
df.daily_vaccinations_raw = df.daily_vaccinations_raw.astype(int)
df.total_vaccinations_per_hundred=df.total_vaccinations_per_hundred.astype(in
df.people_fully_vaccinated_per_hundred=df.people_fully_vaccinated_per_hundred
df.daily_vaccinations_per_million=df.daily_vaccinations_per_million.astype(in
df.people_vaccinated_per_hundred=df.people_vaccinated_per_hundred.astype(int)
```

In [10]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 24620 entries, 0 to 24619
Data columns (total 15 columns):
 #   Column                               Non-Null Count  Dtype         
---  ------                               --------------  -----         
 0   country                              24620 non-null  object        
 1   iso_code                             24620 non-null  object        
 2   date                                 24620 non-null  datetime64[ns]
 3   total_vaccinations                   24620 non-null  int32         
 4   people_vaccinated                    24620 non-null  int32         
 5   people_fully_vaccinated              24620 non-null  int32         
 6   daily_vaccinations_raw               24620 non-null  int32         
 7   daily_vaccinations                   24620 non-null  int32         
 8   total_vaccinations_per_hundred       24620 non-null  int32         
 9   people_vaccinated_per_hundred        24620 non-null  int32         
 10  people_fully_vaccinated_per_hundred  24620 non-null  int32         
 11  daily_vaccinations_per_million       24620 non-null  int32         
 12  vaccines                             24620 non-null  object        
 13  source_name                          24620 non-null  object        
 14  source_website                       24620 non-null  object        
dtypes: datetime64[ns](1), int32(9), object(5)
memory usage: 2.0+ MB
```

1. Total Vaccinations among Countries? What country has vaccinated more people?

In [11]:
```
df.head()
```

Out[11]:

	country	iso_code	date	total_vaccinations	people_vaccinated	people_fully_vaccinated	daily
0	Afghanistan	AFG	2021-02-22	0	0	0	
1	Afghanistan	AFG	2021-02-23	0	0	0	
2	Afghanistan	AFG	2021-02-24	0	0	0	
3	Afghanistan	AFG	2021-02-25	0	0	0	
4	Afghanistan	AFG	2021-02-26	0	0	0	

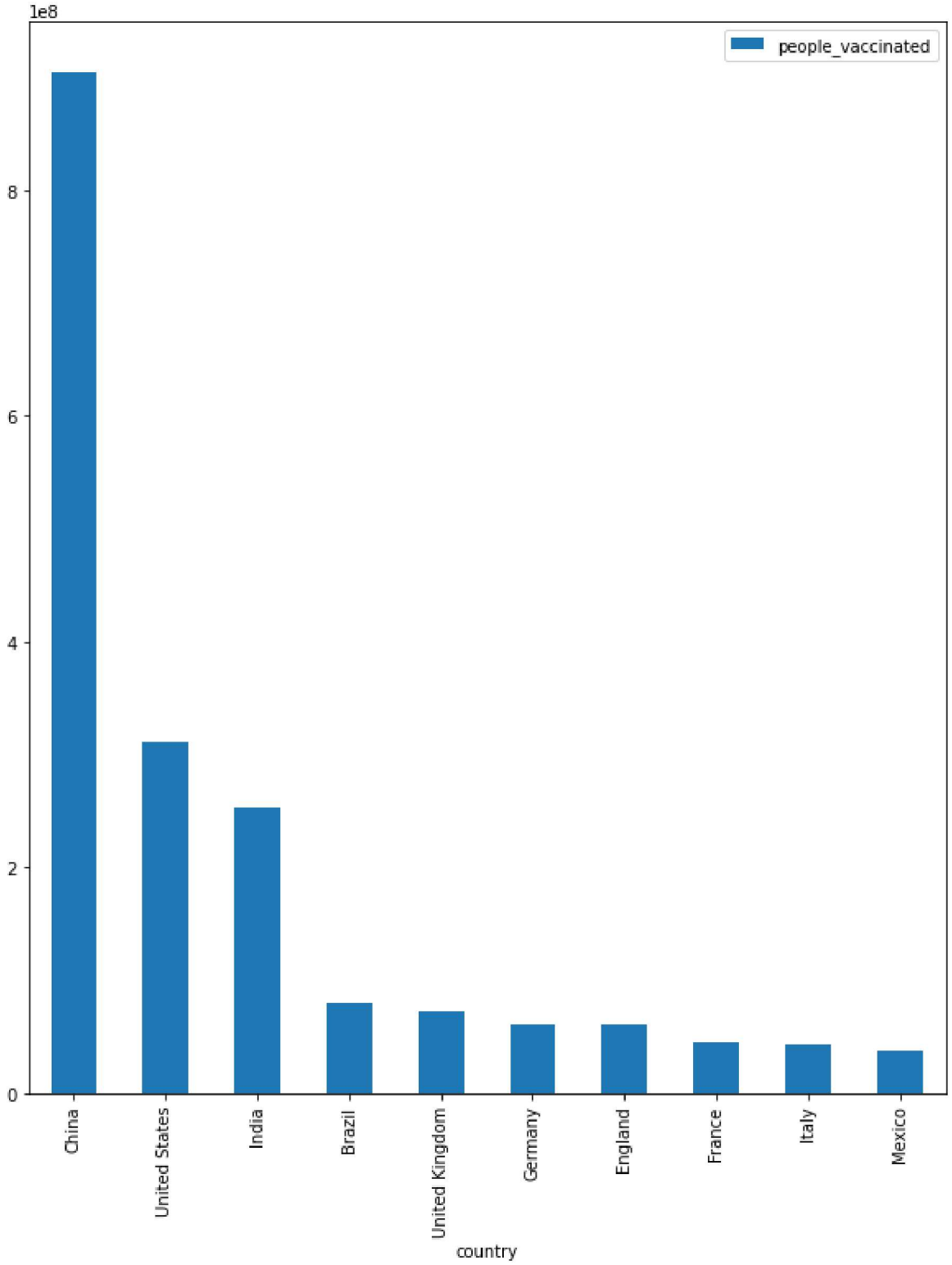
In [12]:
```
df["people_vaccinated"]= df.groupby("country").total_vaccinations.tail(1)
vaccination_country=df.groupby("country")["people_vaccinated"].mean().sort_va
```

In [13]:
```
vaccination_country
```

Out[13]:
```
country
China                904134000.0
United States        310645827.0
India                252760364.0
Brazil                78906225.0
United Kingdom        71672208.0
                           ...
Falkland Islands          4407.0
Montserrat                2584.0
Tuvalu                    2400.0
Vanuatu                    860.0
Kuwait                       0.0
Name: people_vaccinated, Length: 216, dtype: float64
```

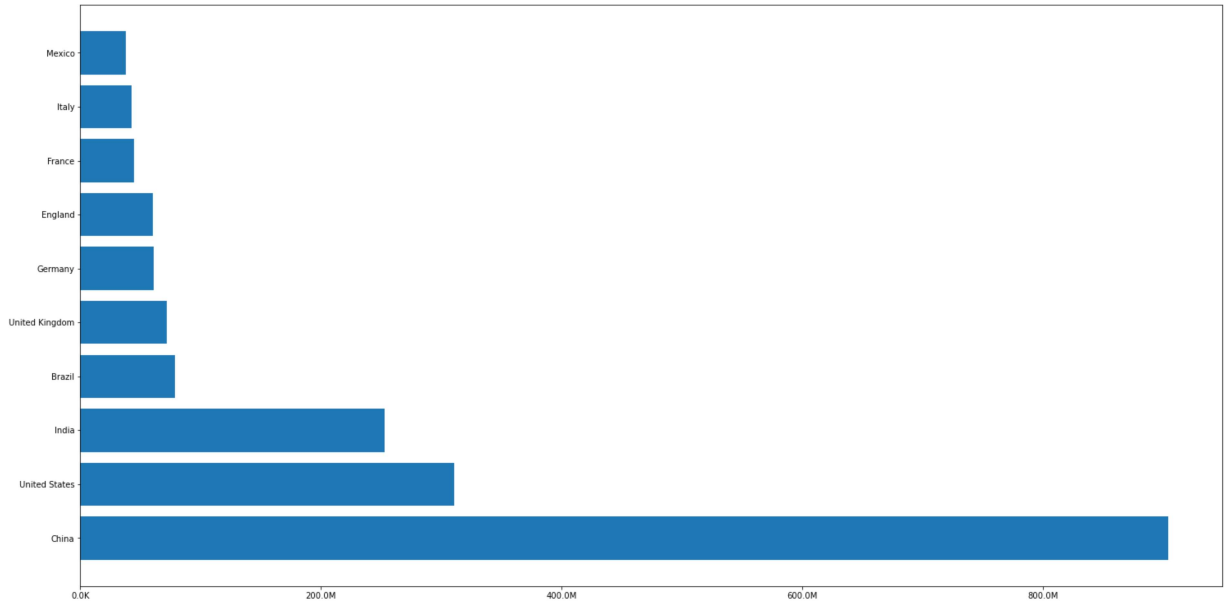
In [14]:
```
# Top 10 Countries with hishest number of total vaccination
top_10=df.groupby("country")["people_vaccinated"].mean().sort_values(ascendin
top_10=top_10.to_frame()
```

In [15]:
```
top_10.plot.bar();
```

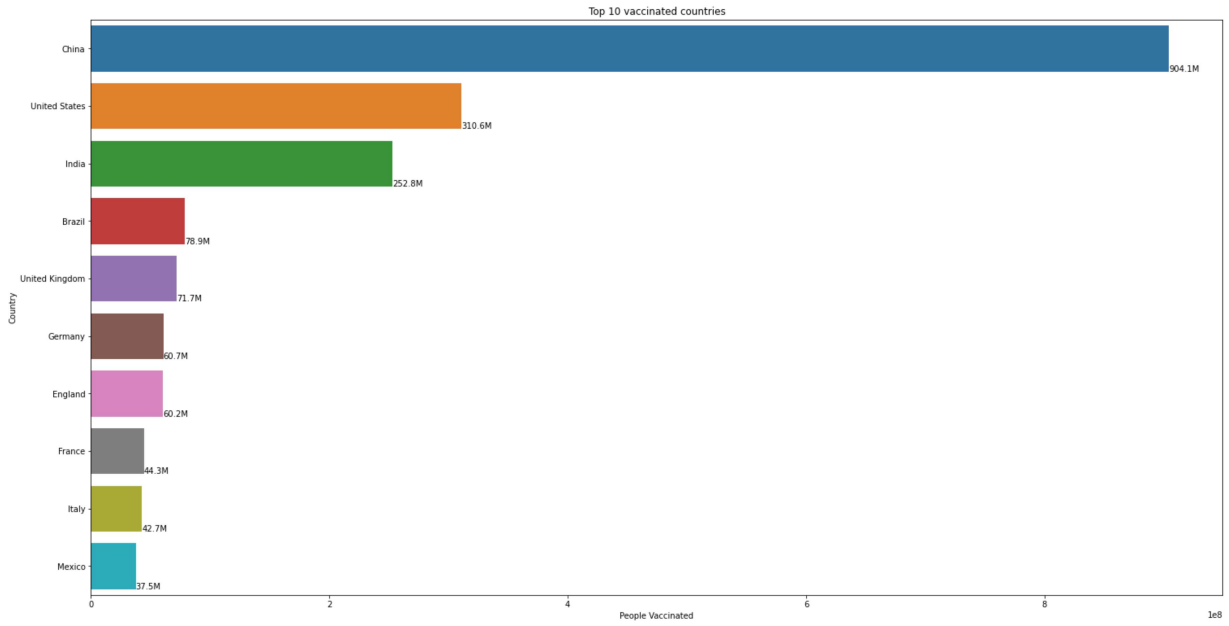


In [16]:
```
from matplotlib.ticker import FuncFormatter
def millions(data_value,index):
  if data_value >=1_000_000:
    formatte='{:1.1f}M'.format(data_value*0.000_001)
  else:
    formatte='{:1.1f}K'.format(data_value*0.001)
  return formatte

formatter = FuncFormatter(millions)
```

In [17]:
```
fig,ax=plt.subplots(figsize=(25,13))
ax.barh(top_10.index,top_10["people_vaccinated"])
ax.xaxis.set_major_formatter(formatter)
plt.show()
```



In [18]:
```
plt.figure(figsize=(25,13))
ax=sns.barplot(x=top_10["people_vaccinated"],y=top_10.index)
plt.xlabel("People Vaccinated")
plt.ylabel("Country")
plt.title("Top 10 vaccinated countries")
for patch in ax.patches:
  width=patch.get_width()
  height=patch.get_height()
  x=patch.get_x()
  y=patch.get_y()
  plt.text(width+x,height+y,'{:1.1f}M'.format(width*0.000_001))
```



2. Which country has most number of Fully Vaccinated People?

In [19]:
```
fully_vaccinated=df.groupby("country")["people_fully_vaccinated"].max().sort_
```

In [20]:
```
fully_vaccinated
```

Out[20]:
```
country
China             223299000
United States     144919339
India              47189318
United Kingdom     29973779
England            25391916
Brazil             23679758
Germany            21812429
Mexico             15062873
France             14371963
Russia             14332077
Name: people_fully_vaccinated, dtype: int32
```

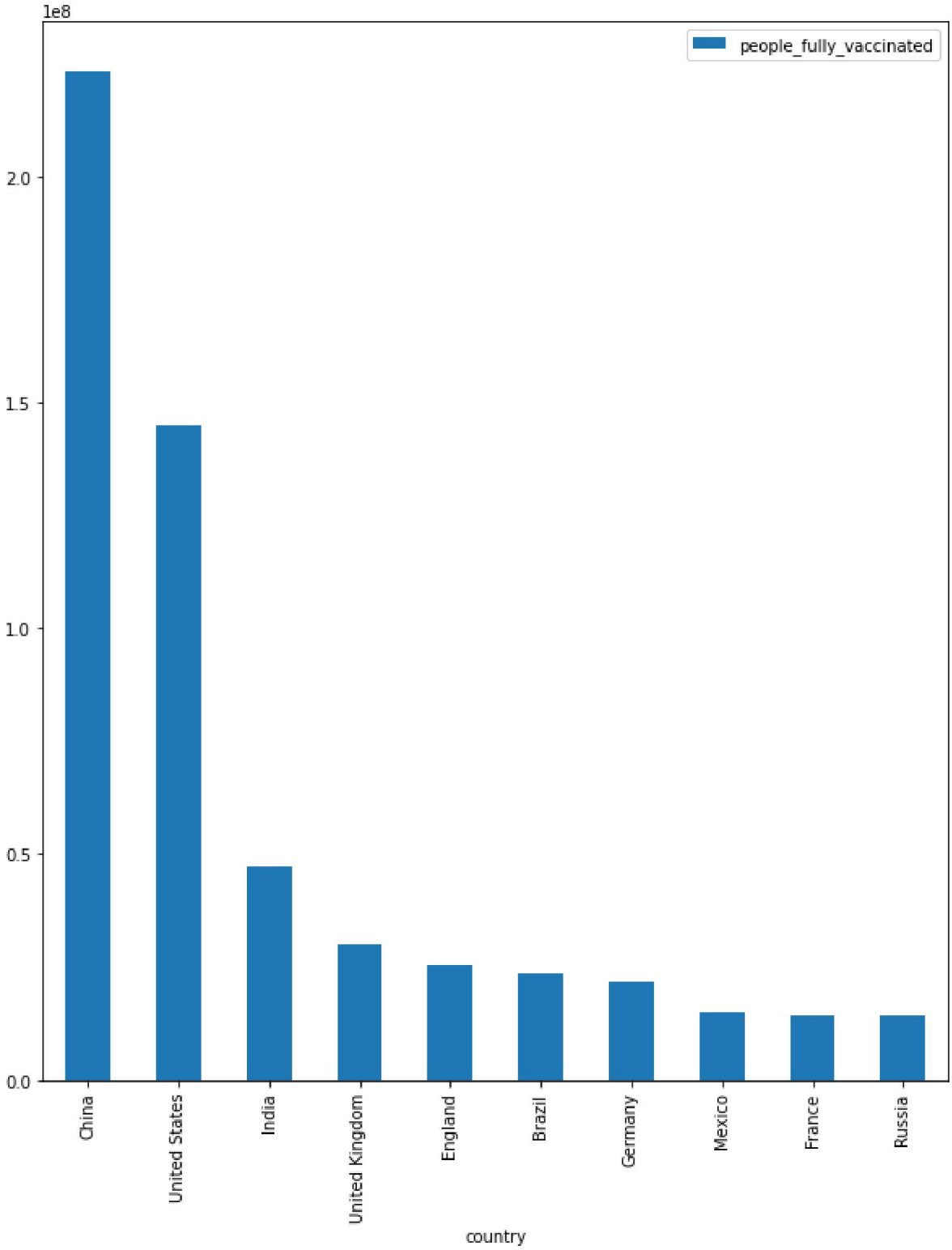
In [21]:
```
top_10=df.groupby("country")["people_fully_vaccinated"].max().sort_values(asc
top_10=top_10.to_frame()
```

In [22]:
```
top_10
```

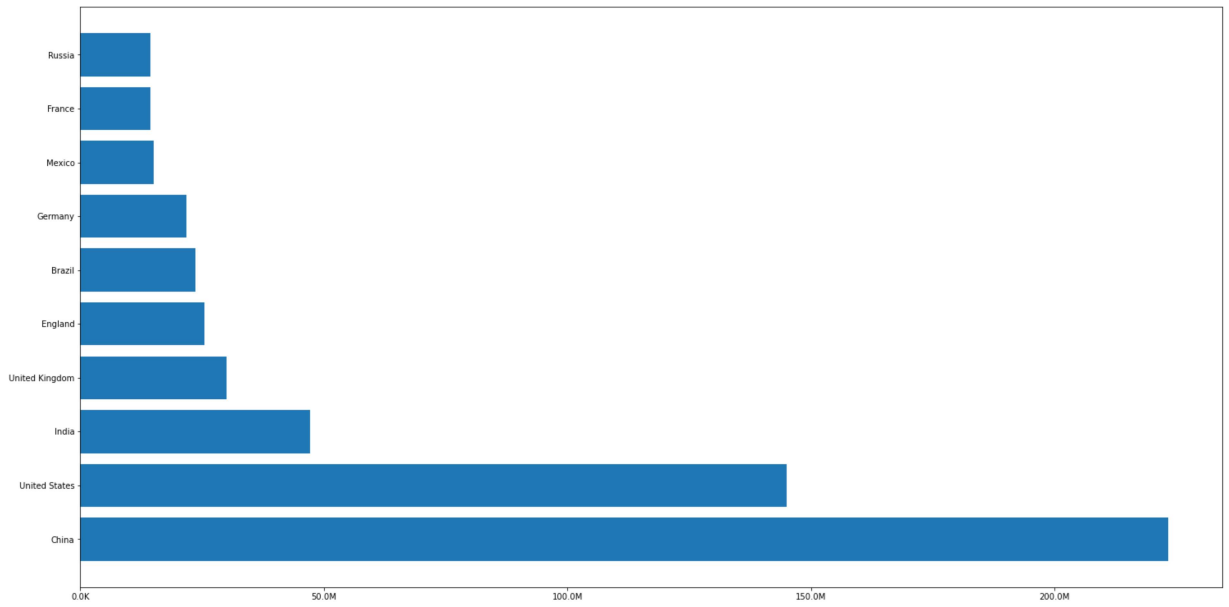
Out[22]:

country	people_fully_vaccinated
China	223299000
United States	144919339
India	47189318
United Kingdom	29973779
England	25391916
Brazil	23679758
Germany	21812429
Mexico	15062873
France	14371963
Russia	14332077

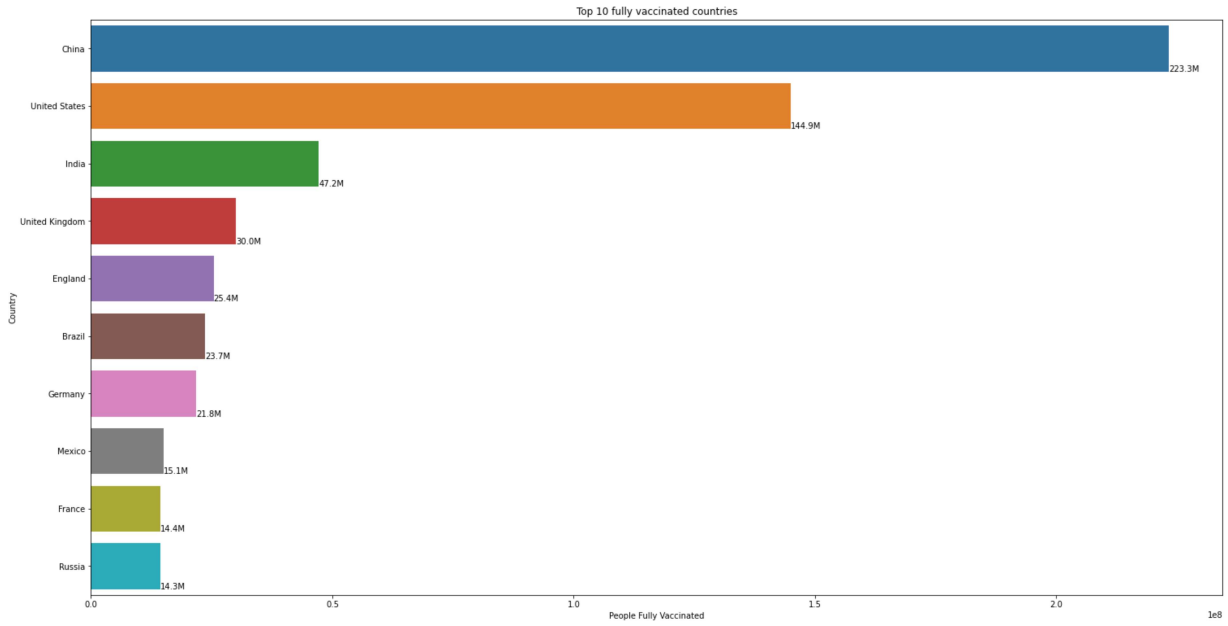
In [23]:
```
top_10.plot.bar();
```



In [24]:
```
fig,ax=plt.subplots(figsize=(25,13))
ax.barh(top_10.index,top_10["people_fully_vaccinated"])
ax.xaxis.set_major_formatter(formatter)
plt.show()
```



In [25]:
```
plt.figure(figsize=(25,13))
ax=sns.barplot(x=top_10["people_fully_vaccinated"],y=top_10.index)
plt.xlabel("People Fully Vaccinated")
plt.ylabel("Country")
plt.title("Top 10 fully vaccinated countries")
for patch in ax.patches:
  width=patch.get_width()
  height=patch.get_height()
  x=patch.get_x()
  y=patch.get_y()
  plt.text(width+x,height+y,'{:1.1f}M'.format(width*0.000_001))
```



3. Ratio between people_vaccinated vs fully vaccinated in different country

Here we have taken some specific countries like China, India, US, UK

In [26]:
```
vaccination_country=df.groupby("country")["people_vaccinated"].mean().sort_va
fully_vaccinated=df.groupby("country")["people_fully_vaccinated"].max().sort_
```

In [27]:
```
people_vaccinated=vaccination_country.query('country in ["United States","Ind
fully_vaccinated_sp=fully_vaccinated.query('country in ["United States", "Ind
```

In [28]:
```
people_vaccinated
```

Out[28]:

	country	people_vaccinated
0	China	904134000.0
1	United States	310645827.0
2	India	252760364.0
4	United Kingdom	71672208.0

In [29]:
```
fully_vaccinated_sp
```

Out[29]:

	country	people_fully_vaccinated
0	China	223299000
1	United States	144919339
2	India	47189318
3	United Kingdom	29973779

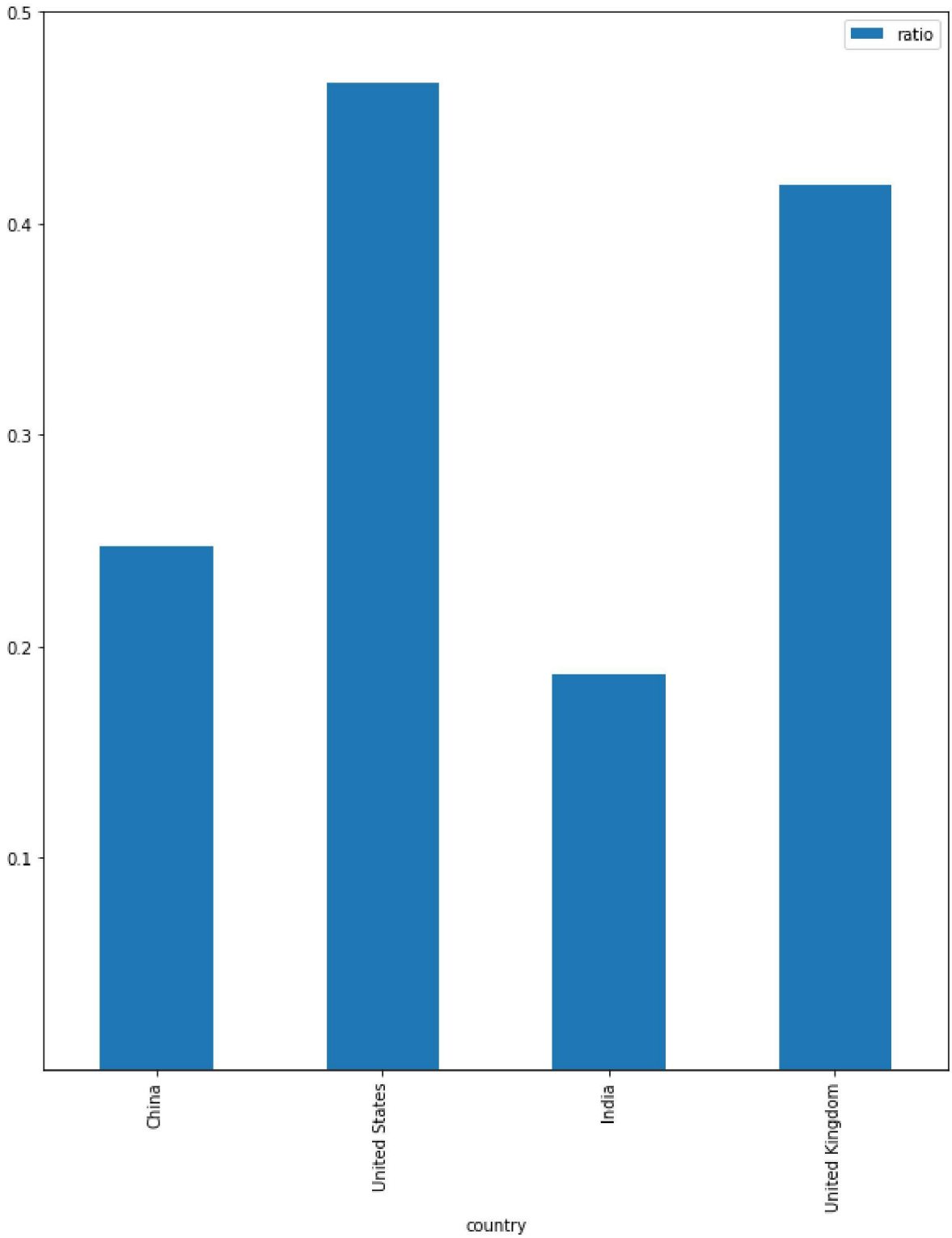
In [30]:
```
fully_vaccinated_sp['ratio']=fully_vaccinated_sp['people_fully_vaccinated'].v
```

In [31]:
```
fully_vaccinated_sp
```

Out[31]:

	country	people_fully_vaccinated	ratio
0	China	223299000	0.246976
1	United States	144919339	0.466510
2	India	47189318	0.186696
3	United Kingdom	29973779	0.418206

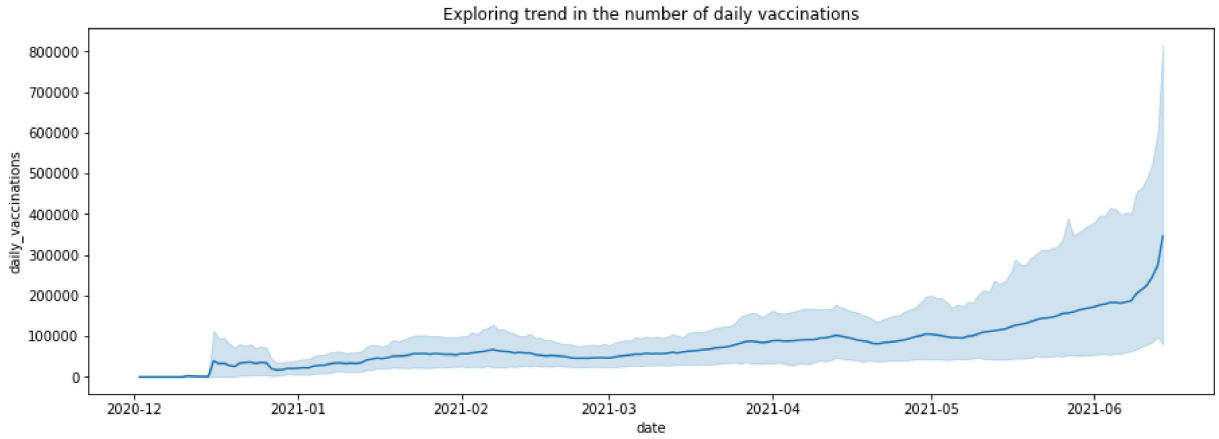
In [32]:
```
fully_vaccinated_sp.plot.bar(x='country',y='ratio');
plt.yticks([0.1,0.2,0.3,0.4,0.5]);
```



From thr above graph we can say that in US there is a good ratio between total no. of people get vaccinated vs fully vaccinated people, in India there are still many people need to be get fully vaccinated.

4. Trace the daily vaccinations dynamic?

In [33]:
```
plt.figure(figsize= (15,5))
sns.lineplot(x= "date",y= "daily_vaccinations",data= df)
plt.title("Exploring trend in the number of daily vaccinations")
plt.show()
```



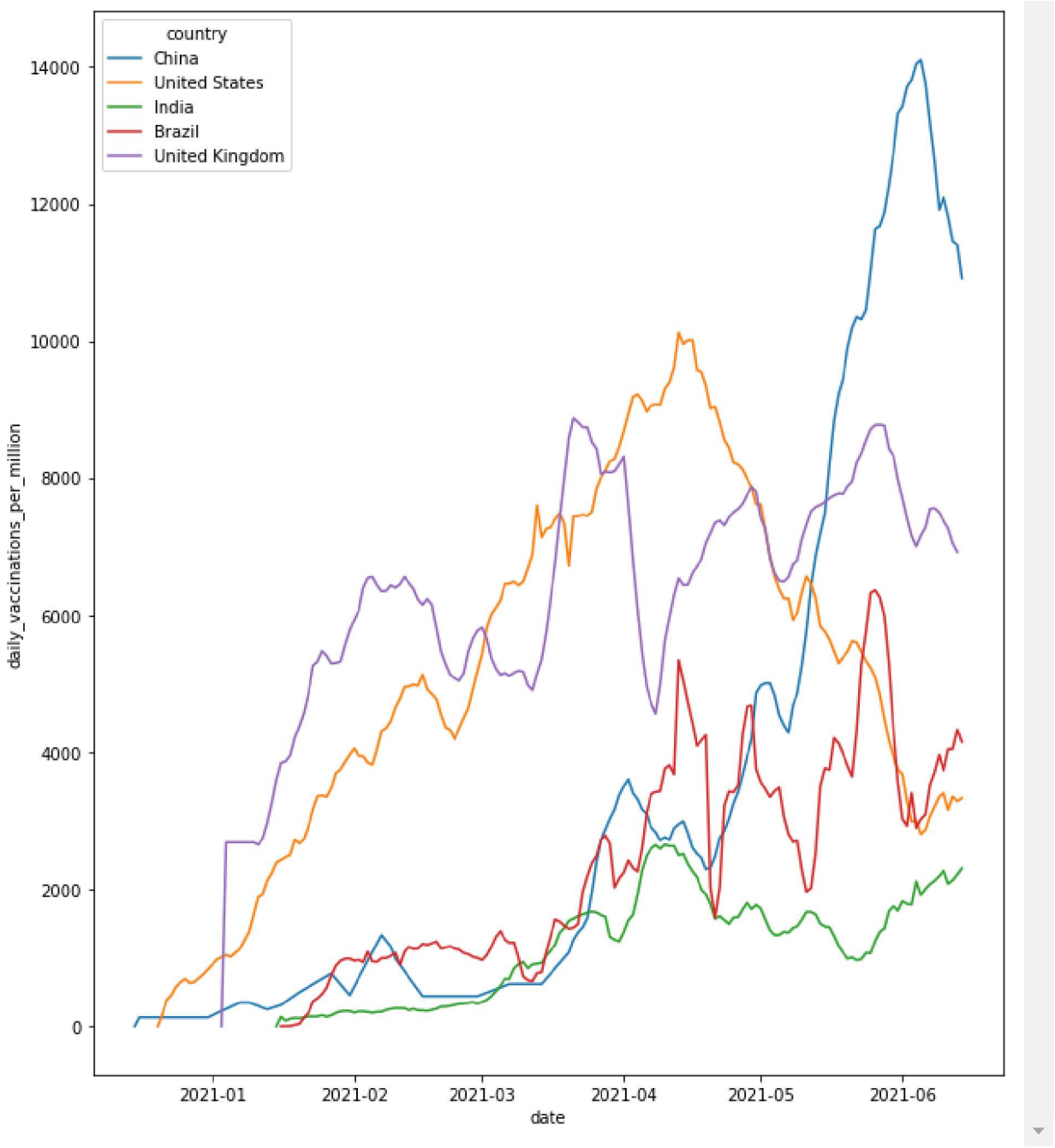
From this above figure we can sat that the vaccination starts increase in the month of february, may be caused by the second peak pf coronavirus

5. From which month based on countries vaccination procedure go on rapidly?

In [34]:
```
countries = df.groupby('country')['total_vaccinations'].max().sort_values(asc
print(countries)
top_countries = pd.DataFrame(columns= df.columns)
print(top_countries)
for country in countries:
    top_countries = top_countries.append(df.loc[df['country'] == country])
sns.lineplot(top_countries['date'],
             top_countries['daily_vaccinations_per_million'],
             hue= top_countries['country'])
```

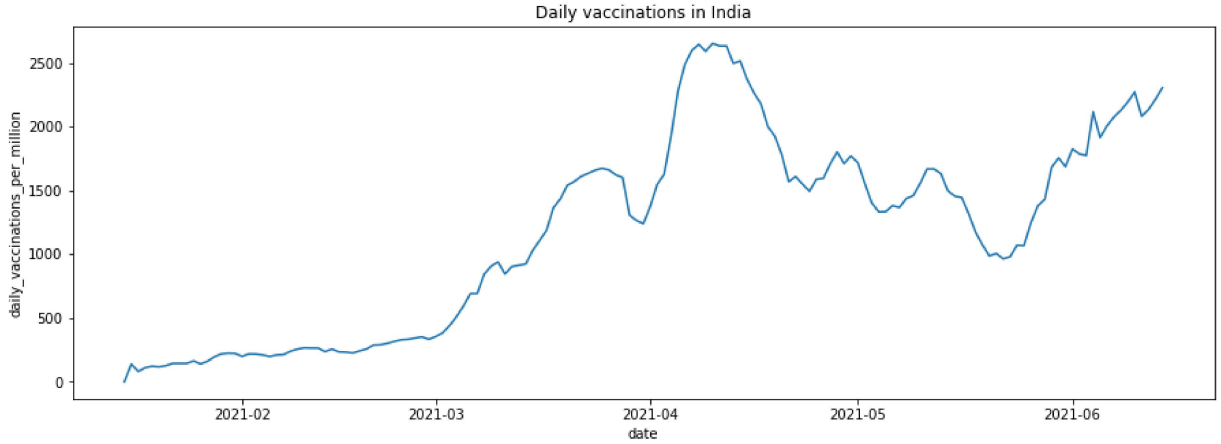
```
Index(['China', 'United States', 'India', 'Brazil', 'United Kingdom'], dtype
='object', name='country')
Empty DataFrame
Columns: [country, iso_code, date, total_vaccinations, people_vaccinated, pe
ople_fully_vaccinated, daily_vaccinations_raw, daily_vaccinations, total_vac
cinations_per_hundred, people_vaccinated_per_hundred, people_fully_vaccinate
d_per_hundred, daily_vaccinations_per_million, vaccines, source_name, source
_website]
Index: []
```

Out[34]: <AxesSubplot:xlabel='date', ylabel='daily_vaccinations_per_million'>



As we can see fromthe above graph that every country except US, increase thier vaccination procedure, some countries like India has some decrease in graph in the month of may(may be people at that time due to second wave can't able to get the vaccination also shortage of vaccination also can be a factor), but after may in the month of june onwards it's again increased

In [35]:
```
plt.figure(figsize= (15,5))
# In India
sns.lineplot(x= "date",y= "daily_vaccinations_per_million",data= df[df.countr
plt.title("Daily vaccinations in India")
plt.show()
```



6. What is the country that vaccinated completely most of the population?

In [36]:
```
population_country=df.groupby('country')['total_vaccinations_per_hundred'].ma
```

In [37]:
```
population_country
```

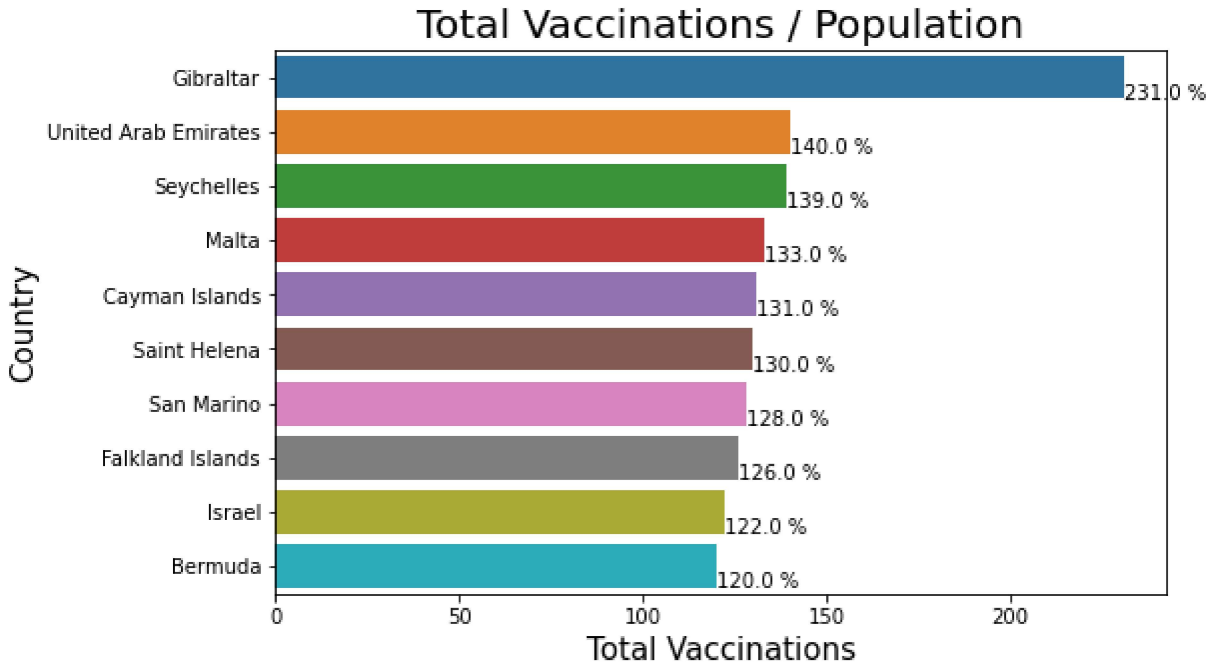
Out[37]:

	country	total_vaccinations_per_hundred
0	Gibraltar	231
1	United Arab Emirates	140
2	Seychelles	139
3	Malta	133
4	Cayman Islands	131
5	Saint Helena	130
6	San Marino	128
7	Falkland Islands	126
8	Israel	122
9	Bermuda	120

In [38]:
```
gibraltar=df.loc[df['country'] == "Gibraltar"]
gibraltar["total_vaccinations"].tail(1)
```

Out[38]:
```
8601    77831
Name: total_vaccinations, dtype: int32
```

In [39]:
```
plt.figure(figsize= (8, 5))

ax = sns.barplot(data=population_country, y= 'country', x= 'total_vaccination

plt.title('Total Vaccinations / Population', size= 20)
plt.xlabel('Total Vaccinations', size= 15)
plt.ylabel('Country', size= 15)

for patch in ax.patches:
    width = patch.get_width()
    height = patch.get_height()
    x = patch.get_x()
    y = patch.get_y()

    plt.text(width + x, height + y, '{:.1f} %'.format(width))
```



7. What country has immunized the largest percent from its population?

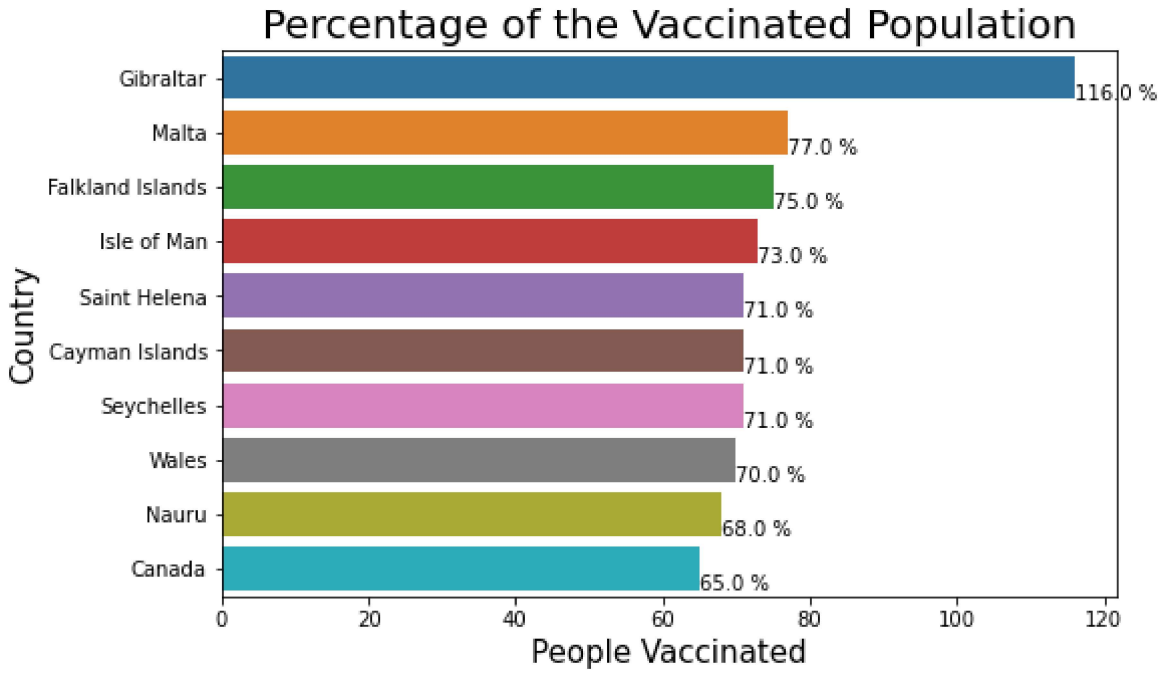
In [40]:
```
population_people_vaccination=df.groupby('country')['people_vaccinated_per_hu
```

In [41]:
```
population_people_vaccination
```

Out[41]:

	country	people_vaccinated_per_hundred
0	Gibraltar	116
1	Malta	77
2	Falkland Islands	75
3	Isle of Man	73
4	Saint Helena	71
5	Cayman Islands	71
6	Seychelles	71
7	Wales	70
8	Nauru	68
9	Canada	65

In [42]:
```
plt.figure(figsize= (8, 5))

ax = sns.barplot(data=population_people_vaccination, y= 'country', x= 'people_

plt.title('Percentage of the Vaccinated Population', size= 20)
plt.xlabel('People Vaccinated', size= 15)
plt.ylabel('Country', size= 15)

for patch in ax.patches:
    width = patch.get_width()
    height = patch.get_height()
    x = patch.get_x()
    y = patch.get_y()

    plt.text(width + x, height + y, '{:.1f} %'.format(width))
```



8. What is the vaccine used in the largest number of countries?

In [43]:
```
df['vaccines'].value_counts()
```

Out[43]:
```
Oxford/AstraZeneca
4130
Johnson&Johnson, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech
3314
Moderna, Oxford/AstraZeneca, Pfizer/BioNTech
2107
Oxford/AstraZeneca, Pfizer/BioNTech
1739
Moderna, Pfizer/BioNTech
1514
Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sputnik V
1289
Pfizer/BioNTech
1061
Oxford/AstraZeneca, Sinopharm/Beijing
1030
Oxford/AstraZeneca, Pfizer/BioNTech, Sinovac
986
Oxford/AstraZeneca, Sputnik V
527
Oxford/AstraZeneca, Pfizer/BioNTech, Sinovac, Sputnik V
479
Oxford/AstraZeneca, Sinopharm/Beijing, Sputnik V
450
BBIBP-CorV, Oxford/AstraZeneca
431
Sinopharm/Beijing, Sputnik V
324
Oxford/AstraZeneca, Sinovac
289
Sinopharm/Beijing
289
Pfizer/BioNTech, Sinovac
266
Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing
257
Oxford/AstraZeneca, Sinovac, Sputnik V
217
Moderna, Oxford/AstraZeneca
195
Covaxin, Oxford/AstraZeneca
185
Covaxin, Oxford/AstraZeneca, Sinopharm/Beijing
184
EpiVacCorona, Sputnik V
182
CanSino, Sinopharm/Beijing, Sinopharm/Wuhan, Sinovac
182
Johnson&Johnson, Moderna, Pfizer/BioNTech
177
CanSino, Oxford/AstraZeneca, Pfizer/BioNTech, Sinovac
172
CanSino, Oxford/AstraZeneca, Pfizer/BioNTech, Sinovac, Sputnik V
172
```

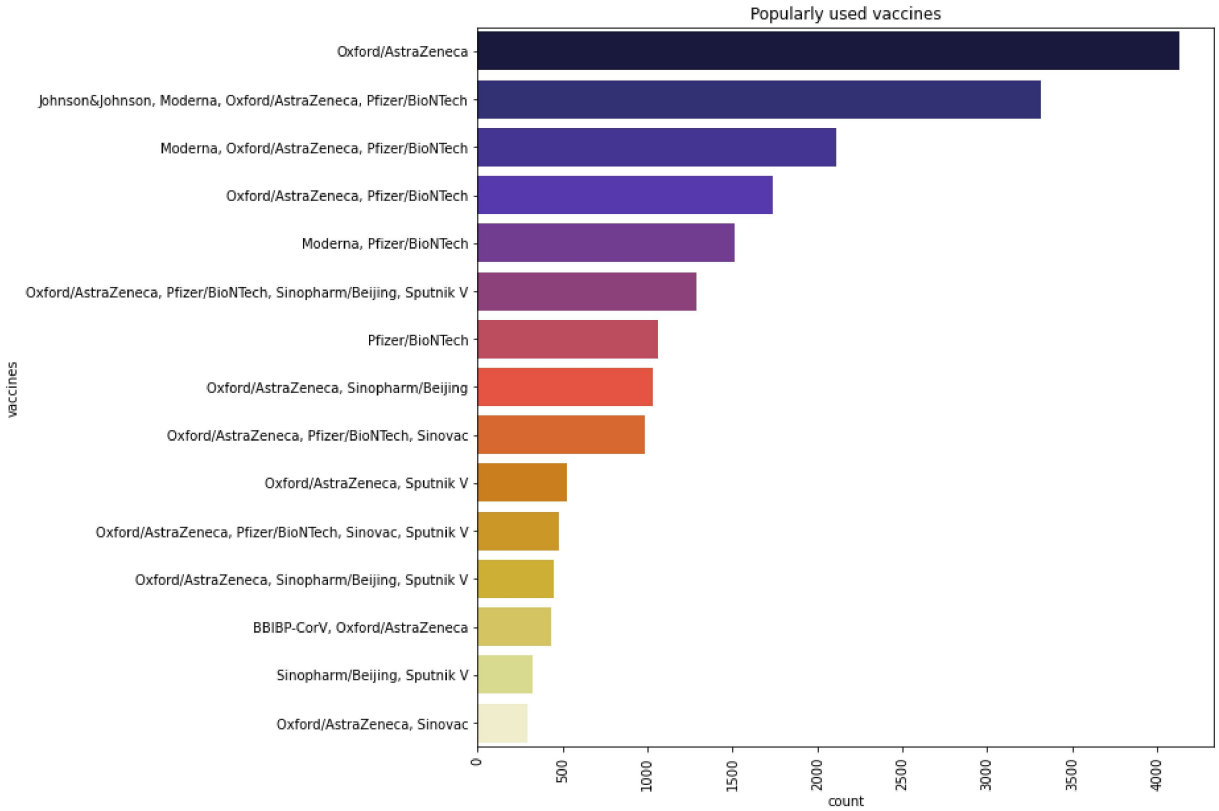
```
Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sputnik V
164
Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sinopharm/Wuhan, Spu
tnik V     161
BBIBP-CorV, Oxford/AstraZeneca, Sinovac, Sputnik V
140
Moderna
139
QazVac, Sinopharm/HayatVax, Sputnik V
135
CanSino, Oxford/AstraZeneca, Sinopharm/Beijing, Sinovac, Sputnik V
132
Pfizer/BioNTech, Sinopharm/Beijing
127
Oxford/AstraZeneca, Sinopharm/Beijing, Sinovac
126
Covaxin, Oxford/AstraZeneca, Sinopharm/Beijing, Sputnik V
121
Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sinovac
119
Johnson&Johnson, Pfizer/BioNTech
119
BBIBP-CorV, Sputnik V
115
BBIBP-CorV, Oxford/AstraZeneca, Pfizer/BioNTech
113
Covaxin, Oxford/AstraZeneca, Sinopharm/Beijing, Sinovac, Sputnik V
110
Pfizer/BioNTech, Sputnik V
107
Johnson&Johnson, Moderna, Oxford/AstraZeneca, Sputnik V
106
BBIBP-CorV, Oxford/AstraZeneca, Pfizer/BioNTech, Sinovac
97
Sputnik V
75
Oxford/AstraZeneca, RBD-Dimer, Sputnik V
63
BBIBP-CorV, Moderna, Oxford/AstraZeneca, Sputnik V
62
Johnson&Johnson, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinovac, Sput
nik V       57
BBIBP-CorV, Oxford/AstraZeneca, Sinovac
57
Abdala, Soberana02
25
EpiVacCorona, Oxford/AstraZeneca, Sinopharm/Beijing, Sputnik V
1
BBIBP-CorV
1
Name: vaccines, dtype: int64
```

In [44]:
```
pip install wordcloud
```

```
Requirement already satisfied: wordcloud in c:\users\admin\anaconda3\lib\sit
e-packages (1.8.1)
Requirement already satisfied: numpy>=1.6.1 in c:\users\admin\anaconda3\lib
\site-packages (from wordcloud) (1.20.1)
Requirement already satisfied: matplotlib in c:\users\admin\anaconda3\lib\si
te-packages (from wordcloud) (3.3.4)
Requirement already satisfied: pillow in c:\users\admin\anaconda3\lib\site-p
ackages (from wordcloud) (8.2.0)
Requirement already satisfied: python-dateutil>=2.1 in c:\users\admin\anacon
da3\lib\site-packages (from matplotlib->wordcloud) (2.8.1)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\admin\anaconda3
\lib\site-packages (from matplotlib->wordcloud) (1.3.1)
Requirement already satisfied: cycler>=0.10 in c:\users\admin\anaconda3\lib
\site-packages (from matplotlib->wordcloud) (0.10.0)
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.3 in
c:\users\admin\anaconda3\lib\site-packages (from matplotlib->wordcloud) (2.
4.7)
Requirement already satisfied: six in c:\users\admin\anaconda3\lib\site-pack
ages (from cycler>=0.10->matplotlib->wordcloud) (1.15.0)
Note: you may need to restart the kernel to use updated packages.
```

In [45]:
```
from wordcloud import WordCloud, STOPWORDS
plt.figure(figsize= (20,10))
all_words = "".join(df["vaccines"])
wc = WordCloud(stopwords=STOPWORDS,
                 background_color="black",
                max_words=2000, max_font_size=280,
                random_state=42, width=1000,
                height=800).generate(all_words)

plt.imshow(wc, interpolation="bilinear")
plt.axis('off')
plt.show()
```

In [46]:
```
plt.figure(figsize=[10,10])
sns.countplot(y=df.vaccines,palette='CMRmap',
              order=df['vaccines'].value_counts().head(15).index)
plt.title("Popularly used vaccines")
plt.xticks(rotation=90);
```



From this above graph we can get that the most used vaccines are oxford/AstraZeneca and Pfizer

9. What vaccination schemes (combination of vaccines) are used and in which countries?

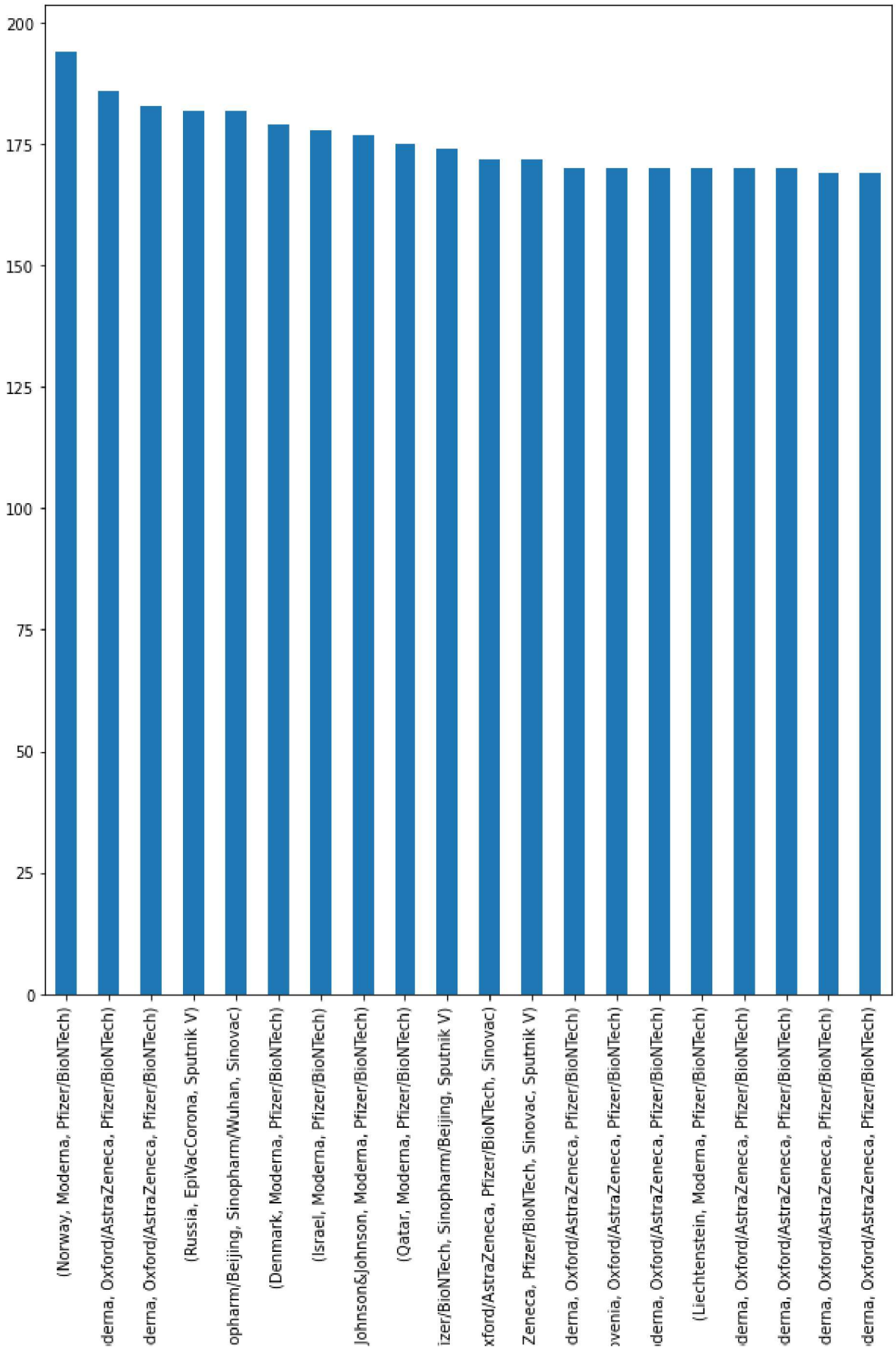
In [47]:
```
df.groupby('country')['vaccines'].value_counts().sort_values(ascending=False)
```

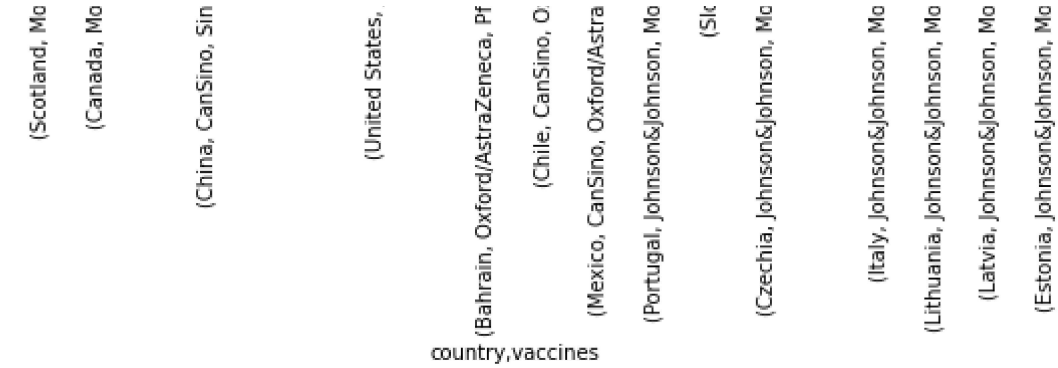
Out[47]:
```
country                          vaccines
Norway                           Moderna, Pfizer/BioNTech
194
Scotland                         Moderna, Oxford/AstraZeneca, Pfizer/BioNTec
h                          186
Canada                           Moderna, Oxford/AstraZeneca, Pfizer/BioNTec
h                          183
Russia                           EpiVacCorona, Sputnik V
182
China                            CanSino, Sinopharm/Beijing, Sinopharm/Wuha
n, Sinovac                  182

...
Burkina Faso                     Oxford/AstraZeneca
13
Vanuatu                          Oxford/AstraZeneca
8
Bonaire Sint Eustatius and Saba  Moderna, Pfizer/BioNTech
1
Turkmenistan                     EpiVacCorona, Oxford/AstraZeneca, Sinophar
m/Beijing, Sputnik V        1
Chad                             BBIBP-CorV
1
Name: vaccines, Length: 216, dtype: int64
```

so here we can see some countries using some of the mixed vaccines, like norway is using modrena as well as pfizer

In [48]:
```
df.groupby('country')['vaccines'].value_counts().sort_values(ascending=False)
```

Out[48]: <AxesSubplot:xlabel='country,vaccines'>





10. Which country prefers what vaccine?

In [49]:
```
vaccines = df.groupby('vaccines')['country'].unique()
vaccines = pd.DataFrame(vaccines).reset_index()
vaccines
```

Out[49]:

	vaccines	country
0	Abdala, Soberana02	[Cuba]
1	BBIBP-CorV	[Chad]
2	BBIBP-CorV, Moderna, Oxford/AstraZeneca, Sputn...	[Congo]
3	BBIBP-CorV, Oxford/AstraZeneca	[Cameroon, Mauritania, Niger, Senegal, Sierra ...
4	BBIBP-CorV, Oxford/AstraZeneca, Pfizer/BioNTech	[Afghanistan]
5	BBIBP-CorV, Oxford/AstraZeneca, Pfizer/BioNTec...	[Sudan]
6	BBIBP-CorV, Oxford/AstraZeneca, Sinovac	[Somalia]
7	BBIBP-CorV, Oxford/AstraZeneca, Sinovac, Sputn...	[Egypt]
8	BBIBP-CorV, Sputnik V	[Venezuela]
9	CanSino, Oxford/AstraZeneca, Pfizer/BioNTech, ...	[Chile]
10	CanSino, Oxford/AstraZeneca, Pfizer/BioNTech, ...	[Mexico]
11	CanSino, Oxford/AstraZeneca, Sinopharm/Beijing...	[Pakistan]
12	CanSino, Sinopharm/Beijing, Sinopharm/Wuhan, S...	[China]
13	Covaxin, Oxford/AstraZeneca	[Central African Republic, India]
14	Covaxin, Oxford/AstraZeneca, Sinopharm/Beijing	[Comoros, Mauritius]
15	Covaxin, Oxford/AstraZeneca, Sinopharm/Beijing...	[Paraguay]
16	Covaxin, Oxford/AstraZeneca, Sinopharm/Beijing...	[Iran]
17	EpiVacCorona, Oxford/AstraZeneca, Sinopharm/Be...	[Turkmenistan]
18	EpiVacCorona, Sputnik V	[Russia]
19	Johnson&Johnson, Moderna, Oxford/AstraZeneca, ...	[Austria, Belgium, Bulgaria, Cyprus, Czechia, ...
20	Johnson&Johnson, Moderna, Oxford/AstraZeneca, ...	[Libya]
21	Johnson&Johnson, Moderna, Oxford/AstraZeneca, ...	[Honduras]
22	Johnson&Johnson, Moderna, Pfizer/BioNTech	[United States]
23	Johnson&Johnson, Pfizer/BioNTech	[South Africa]
24	Moderna	[Greenland]
25	Moderna, Oxford/AstraZeneca	[Guatemala, Taiwan]
26	Moderna, Oxford/AstraZeneca, Pfizer/BioNTech	[Canada, Croatia, England, Finland, Guernsey, ...
27	Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, ...	[Hungary]
28	Moderna, Pfizer/BioNTech	[Bonaire Sint Eustatius and Saba, Curacao, Den...
29	Oxford/AstraZeneca	[Angola, Anguilla, Antigua and Barbuda, Bahama...
30	Oxford/AstraZeneca, Pfizer/BioNTech	[Andorra, Australia, Cape Verde, Cayman Island...

	vaccines	country
31	Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm...	[Maldives, Peru]
32	Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm...	[United Arab Emirates]
33	Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm...	[Dominican Republic]
34	Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm...	[Bahrain, Bolivia, Iraq, Jordan, Lebanon, Mold...
35	Oxford/AstraZeneca, Pfizer/BioNTech, Sinovac	[Brazil, Colombia, Ecuador, El Salvador, Malay...
36	Oxford/AstraZeneca, Pfizer/BioNTech, Sinovac, ...	[Albania, Bosnia and Herzegovina, Philippines,...
37	Oxford/AstraZeneca, RBD-Dimer, Sputnik V	[Uzbekistan]
38	Oxford/AstraZeneca, Sinopharm/Beijing	[Belize, Brunei, Gambia, Guinea-Bissau, Morocc...
39	Oxford/AstraZeneca, Sinopharm/Beijing, Sinovac	[Cambodia]
40	Oxford/AstraZeneca, Sinopharm/Beijing, Sputnik V	[Argentina, Djibouti, Sri Lanka, Syria]
41	Oxford/AstraZeneca, Sinovac	[Benin, Indonesia, Thailand]
42	Oxford/AstraZeneca, Sinovac, Sputnik V	[Armenia, Azerbaijan]
43	Oxford/AstraZeneca, Sputnik V	[Algeria, Ghana, Guyana, Kenya, Nicaragua]
44	Pfizer/BioNTech	[Aruba, Bermuda, Gibraltar, Kuwait, Monaco, Ne...
45	Pfizer/BioNTech, Sinopharm/Beijing	[Macao]
46	Pfizer/BioNTech, Sinovac	[Hong Kong, Turkey]
47	Pfizer/BioNTech, Sputnik V	[San Marino]
48	QazVac, Sinopharm/HayatVax, Sputnik V	[Kazakhstan]
49	Sinopharm/Beijing	[Equatorial Guinea, Gabon, Zimbabwe]
50	Sinopharm/Beijing, Sputnik V	[Belarus, Kyrgyzstan, Laos]
51	Sputnik V	[Guinea]

In [ ]:

In [ ]: