In [2]:

```
import pandas as pd
import plotly.express as px
import plotly.graph_objects as go
from folium.features import Choropleth
import folium
from folium.features import Tooltip
import seaborn as sns
```

```
/opt/conda/lib/python3.10/site-packages/scipy/__init__.py:146: UserWarning: A NumPy version >=1.1
6.5 and <1.23.0 is required for this version of SciPy (detected version 1.23.5
  warnings.warn(f"A NumPy version >={np_minversion} and <{np_maxversion}"
```

In [3]:

```
df = pd.read_csv("/kaggle/input/covid-world-vaccination-progress/country_vaccinations_by_manufacture
r.csv")
```

In [4]:

```
df.head(10)
```

Out[4]:

Out[4]:

	location	date	vaccine	total_vaccinations
0	Argentina	2020-12-29	Moderna	2
1	Argentina	2020-12-29	Oxford/AstraZeneca	3
2	Argentina	2020-12-29	Sinopharm/Beijing	1
3	Argentina	2020-12-29	Sputnik V	20481
4	Argentina	2020-12-30	Moderna	2
5	Argentina	2020-12-30	Oxford/AstraZeneca	3
6	Argentina	2020-12-30	Sinopharm/Beijing	1
7	Argentina	2020-12-30	Sputnik V	40583
8	Argentina	2020-12-31	Moderna	2
9	Argentina	2020-12-31	Oxford/AstraZeneca	3

In [5]:

```
df["location"].nunique()
```

Out[5]:

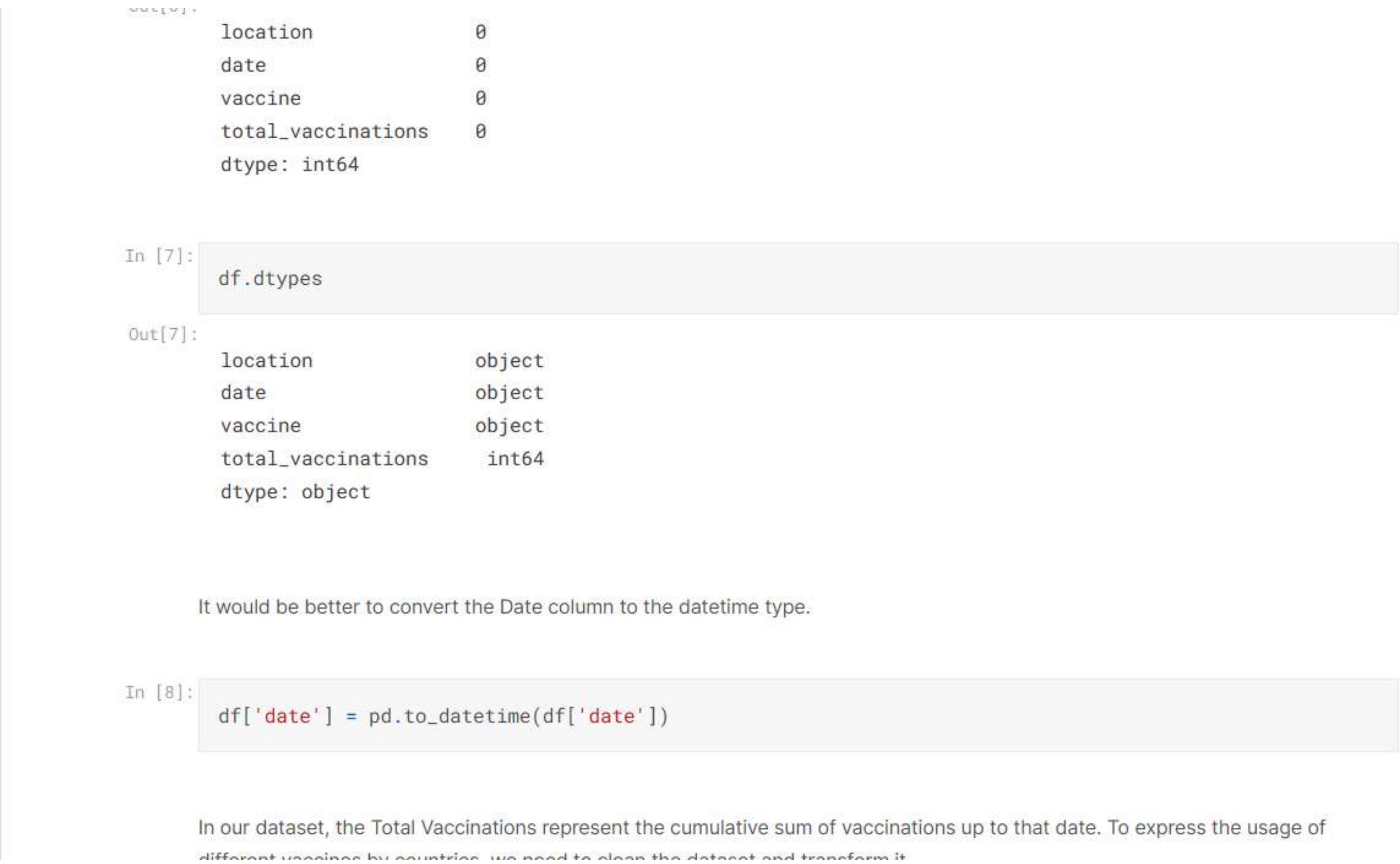
```
43
```

In [6]:

```
df.isnull().sum()
```

Out[6]:

```
location              0
date                  0
```



Out[6]:

```
location                0
date                    0
vaccine                 0
total_vaccinations      0
dtype: int64
```

In [7]:

```
df.dtypes
```

Out[7]:

```
location                object
date                    object
vaccine                 object
total_vaccinations       int64
dtype: object
```

It would be better to convert the Date column to the datetime type.

In [8]:

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

In our dataset, the Total Vaccinations represent the cumulative sum of vaccinations up to that date. To express the usage of different vaccines by countries, we need to clean the dataset and transform it

In [9]:

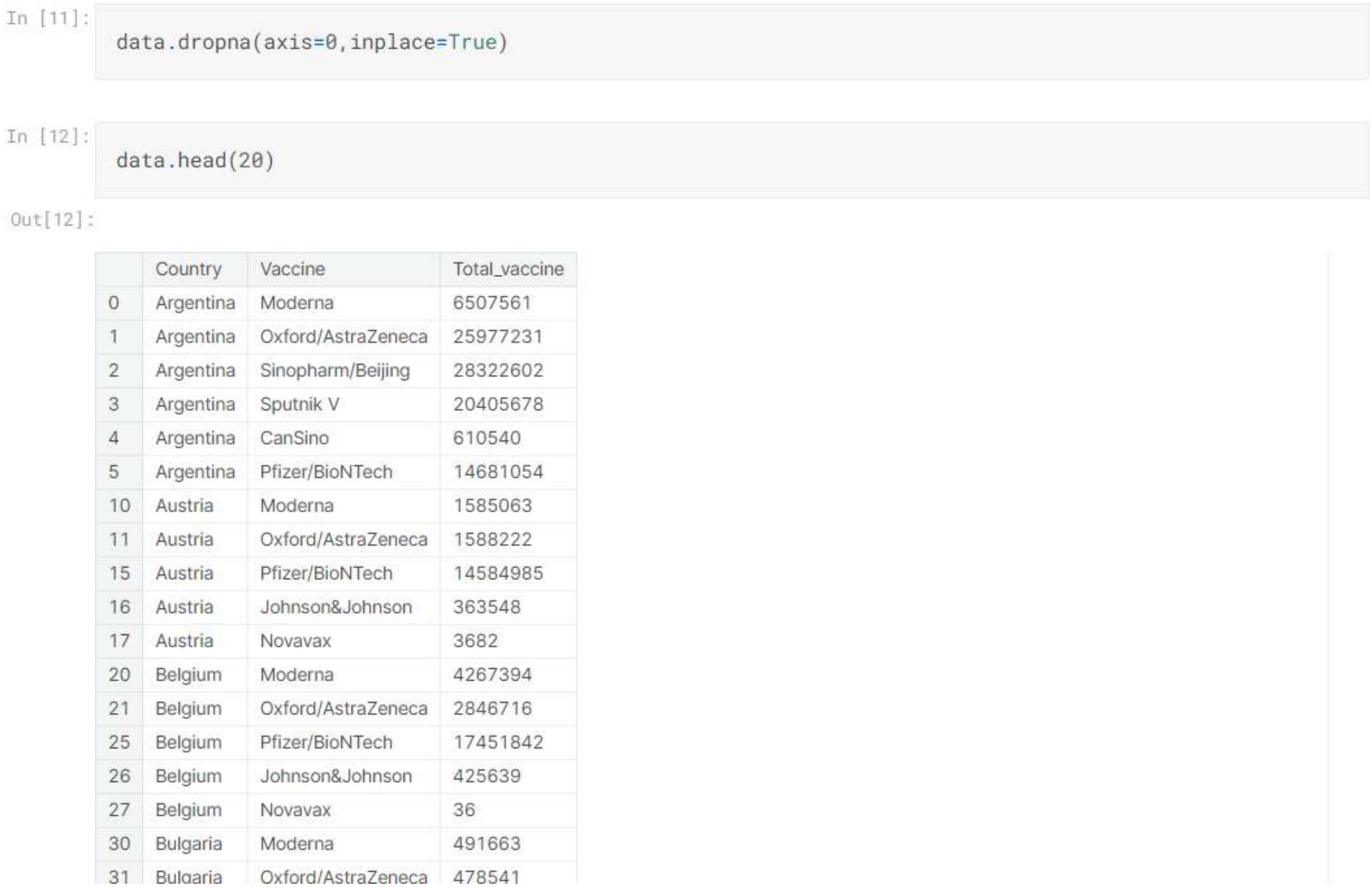
```
data=pd.DataFrame(columns=['Country', 'Vaccine', 'Total_vaccine'])
for country in df["location"].unique():
    for vaccine in df["vaccine"].unique():
        filtered_data = df[(df['location'] == country) & (df['vaccine'] == vaccine)]
        total_count = filtered_data['total_vaccinations'].max()
        data = pd.concat([data, pd.DataFrame({'Country': [country], 'Vaccine': [vaccine], 'Total_vac
cine': [total_count]})], ignore_index=True)
```

In [10]:

```
data.head(10)
```

Out[10]:

	Country	Vaccine	Total_vaccine
0	Argentina	Moderna	6507561
1	Argentina	Oxford/AstraZeneca	25977231
2	Argentina	Sinopharm/Beijing	28322602
3	Argentina	Sputnik V	20405678
4	Argentina	CanSino	610540
5	Argentina	Pfizer/BioNTech	14681054
6	Argentina	Johnson&Johnson	NaN
7	Argentina	Novavax	NaN
8	Argentina	Sinovac	NaN
9	Argentina	Covaxin	NaN



In [11]:

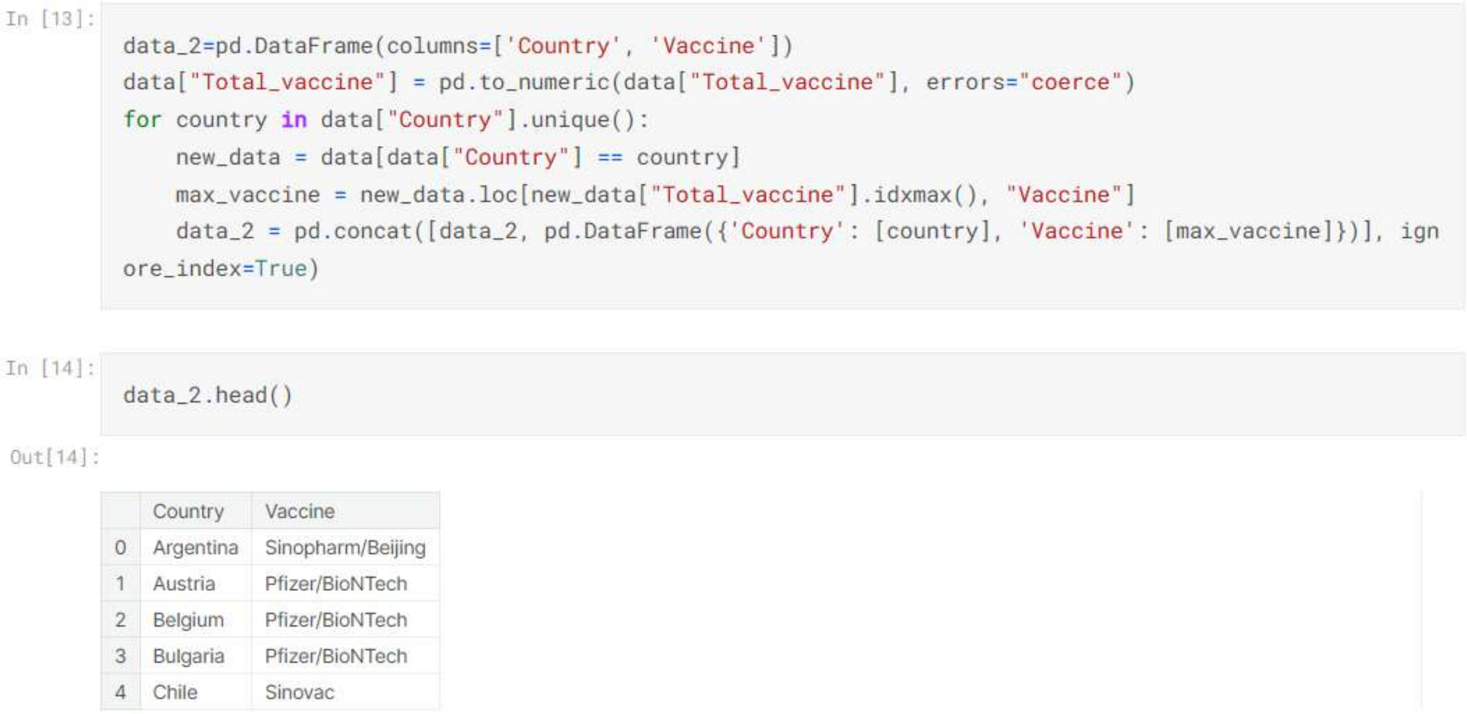
```
data.dropna(axis=0,inplace=True)
```

In [12]:

```
data.head(20)
```

Out[12]:

	Country	Vaccine	Total_vaccine
0	Argentina	Moderna	6507561
1	Argentina	Oxford/AstraZeneca	25977231
2	Argentina	Sinopharm/Beijing	28322602
3	Argentina	Sputnik V	20405678
4	Argentina	CanSino	610540
5	Argentina	Pfizer/BioNTech	14681054
10	Austria	Moderna	1585063
11	Austria	Oxford/AstraZeneca	1588222
15	Austria	Pfizer/BioNTech	14584985
16	Austria	Johnson&Johnson	363548
17	Austria	Novavax	3682
20	Belgium	Moderna	4267394
21	Belgium	Oxford/AstraZeneca	2846716
25	Belgium	Pfizer/BioNTech	17451842
26	Belgium	Johnson&Johnson	425639
27	Belgium	Novavax	36
30	Bulgaria	Moderna	491663
31	Bulgaria	Oxford/AstraZeneca	478541



In [13]:

```
data_2=pd.DataFrame(columns=['Country', 'Vaccine'])
data["Total_vaccine"] = pd.to_numeric(data["Total_vaccine"], errors="coerce")
for country in data["Country"].unique():
    new_data = data[data["Country"] == country]
    max_vaccine = new_data.loc[new_data["Total_vaccine"].idxmax(), "Vaccine"]
    data_2 = pd.concat([data_2, pd.DataFrame({'Country': [country], 'Vaccine': [max_vaccine]})], ign
ore_index=True)
```

In [14]:

```
data_2.head()
```

Out[14]:

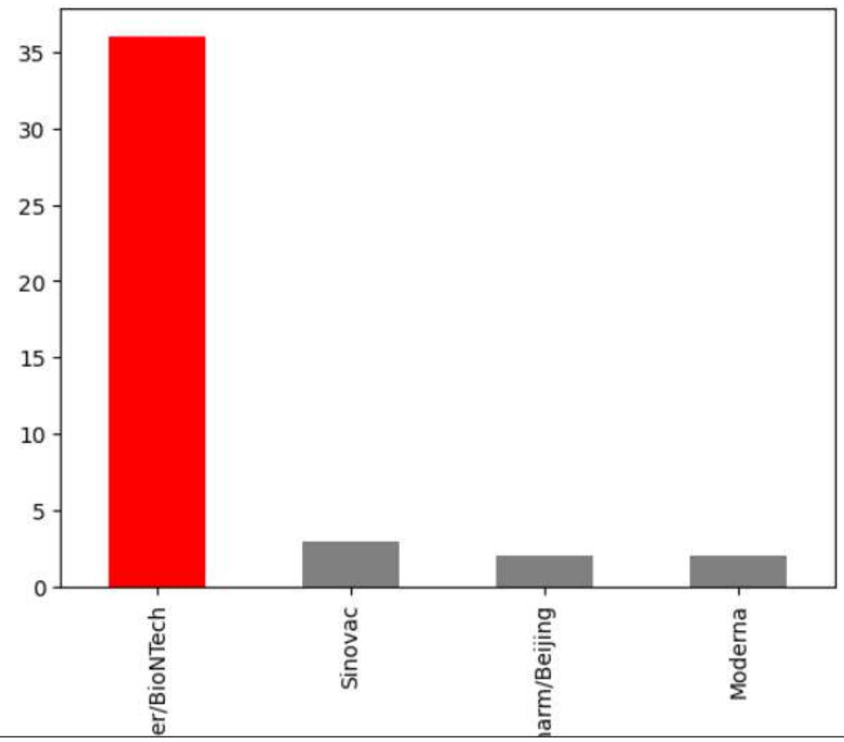
	Country	Vaccine
0	Argentina	Sinopharm/Beijing
1	Austria	Pfizer/BioNTech
2	Belgium	Pfizer/BioNTech
3	Bulgaria	Pfizer/BioNTech
4	Chile	Sinovac

In [15]:

```
data_2["Vaccine"].value_counts().plot(kind="bar",
                                      color=["Red","Gray","Gray","Gray"])
```

Out[15]:

```
<Axes: >
```



In [16]:

```
number_of_days = (df["date"].max() -df["date"].min() ).days
```

In [17]:

```
dtfrm=data[data["Vaccine"]=="Pfizer/BioNTech"]
dtfrm = dtfrm.drop(dtfrm[dtfrm['Country'] == 'European Union'].index)
```

In [18]:

```
dtfrm.head(10)
```

Out[18]:

	Country	Vaccine	Total_vaccine
5	Argentina	Pfizer/BioNTech	14681054
15	Austria	Pfizer/BioNTech	14584985
25	Belgium	Pfizer/BioNTech	17451842
35	Bulgaria	Pfizer/BioNTech	2852218
45	Chile	Pfizer/BioNTech	7910264
55	Croatia	Pfizer/BioNTech	3921503
65	Cyprus	Pfizer/BioNTech	1188656
75	Czechia	Pfizer/BioNTech	14604323
85	Denmark	Pfizer/BioNTech	10259219
95	Ecuador	Pfizer/BioNTech	8552679

In [19]:

```
dtfrm["average_vaccination_count"] = dtfrm["Total_vaccine"] / number_of_days
dtfrm["average_vaccination_count"] =dtfrm["average_vaccination_count"].astype(int)
```

In [20]:

```
dtfrm.head(15)
```

Out[20]:

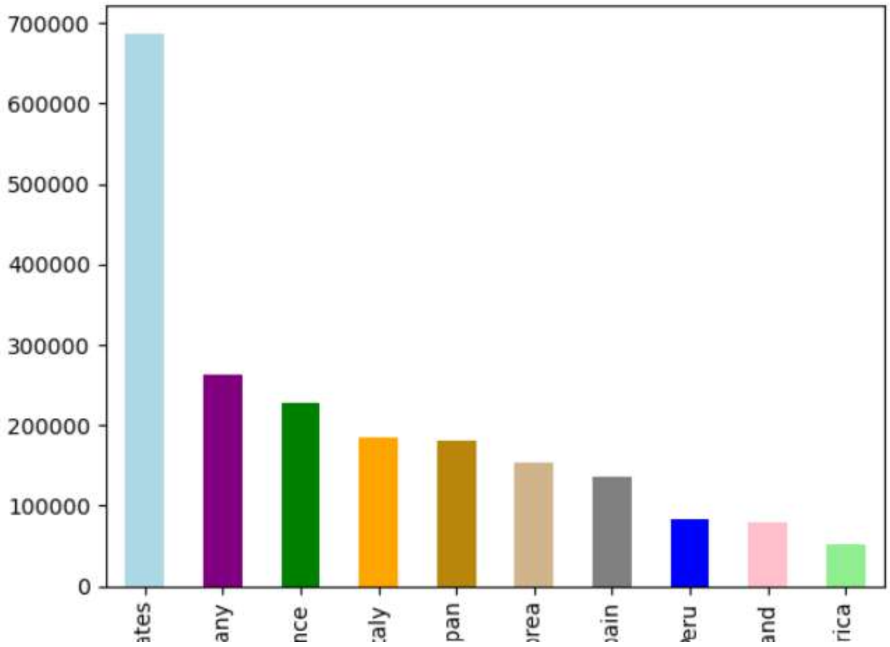
	Country	Vaccine	Total_vaccine	average_vaccination_count
5	Argentina	Pfizer/BioNTech	14681054	30521
15	Austria	Pfizer/BioNTech	14584985	30322
25	Belgium	Pfizer/BioNTech	17451842	36282
35	Bulgaria	Pfizer/BioNTech	2852218	5929
45	Chile	Pfizer/BioNTech	7910264	16445
55	Croatia	Pfizer/BioNTech	3921503	8152
65	Cyprus	Pfizer/BioNTech	1188656	2471
75	Czechia	Pfizer/BioNTech	14604323	30362
85	Denmark	Pfizer/BioNTech	10259219	21328
95	Ecuador	Pfizer/BioNTech	8552679	17781
105	Estonia	Pfizer/BioNTech	1488804	3095
115	Finland	Pfizer/BioNTech	9235420	19200
125	France	Pfizer/BioNTech	109187212	227000
135	Germany	Pfizer/BioNTech	126041243	262040
145	Hong Kong	Pfizer/BioNTech	8879482	18460

In [22]:

```
color=["Lightblue","Purple","Green","Orange","darkgoldenrod","tan","Gray","Blue","Pink","Lightgreen"]
dtfrm["average_vaccination_count"].sort_values(ascending=False).head(10).plot(kind="bar",color=color)
```

Out[22]:

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



In [23]:

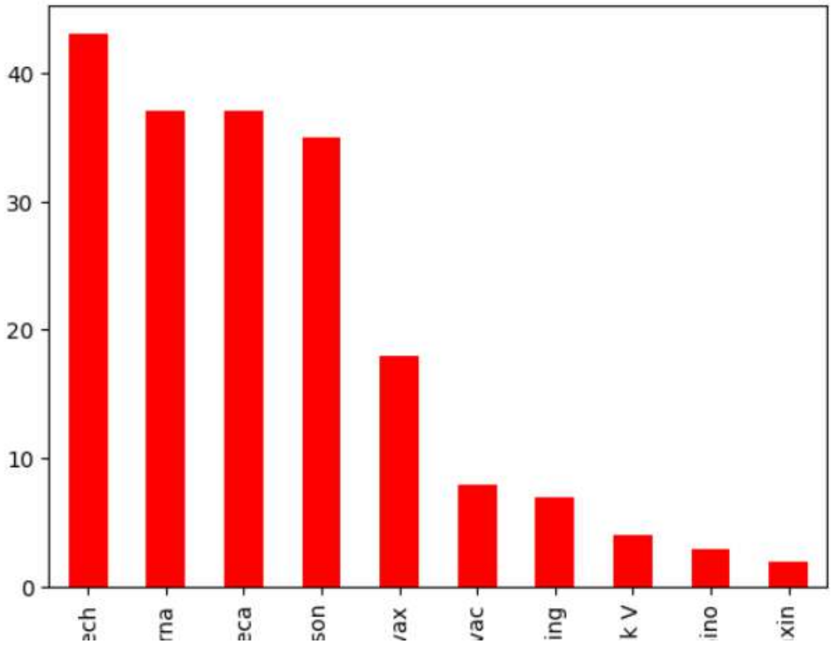
```
number_of_vaccines = data.groupby('Vaccine')['Country'].nunique()
```

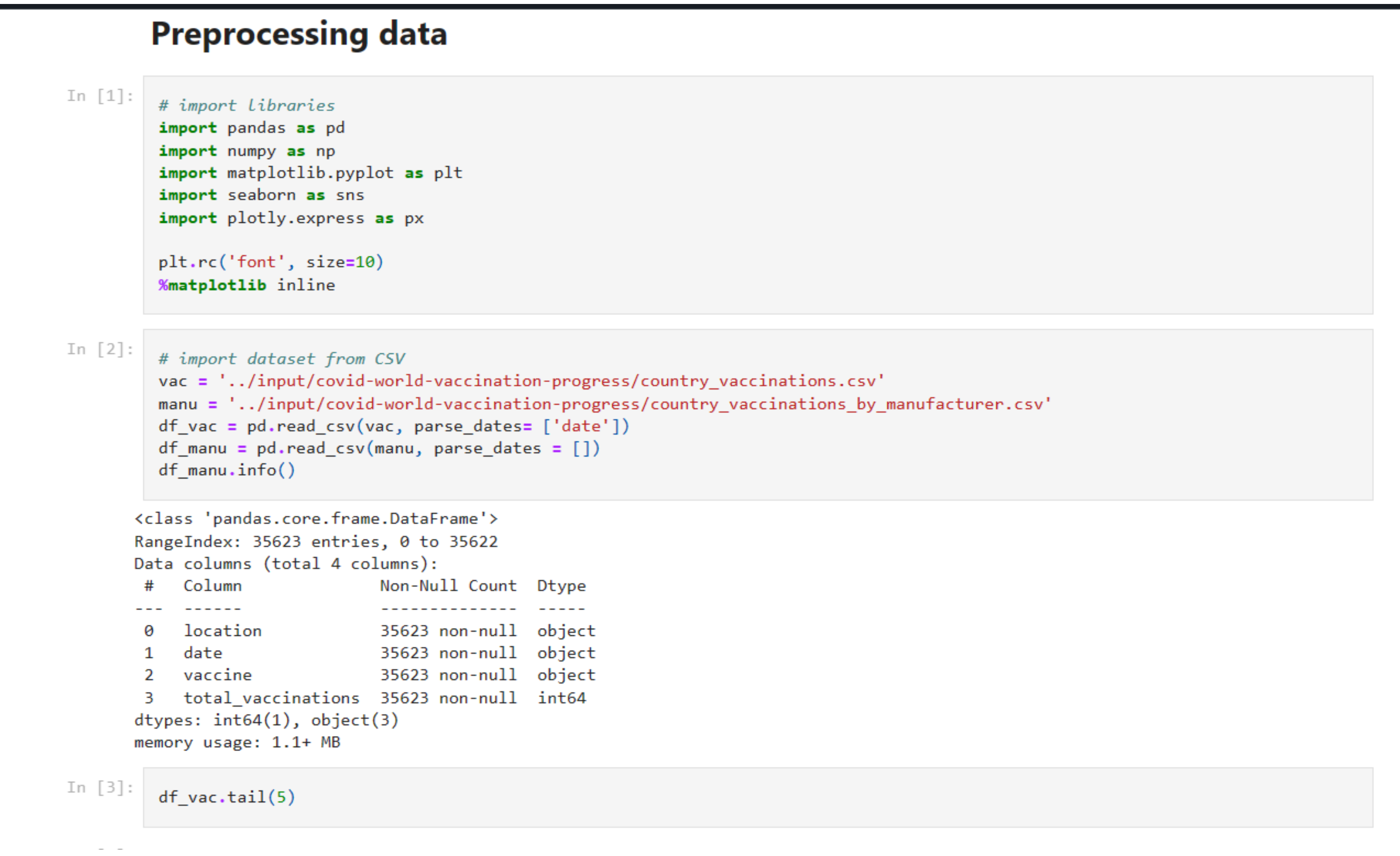
In [24]:

```
number_of_vaccines.sort_values(ascending=False).plot(kind="bar",color="r")
```

Out[24]:

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





Preprocessing data

In [1]:

```
# import libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px

plt.rc('font', size=10)
%matplotlib inline
```

In [2]:

```
# import dataset from CSV
vac = '../input/covid-world-vaccination-progress/country_vaccinations.csv'
manu = '../input/covid-world-vaccination-progress/country_vaccinations_by_manufacturer.csv'
df_vac = pd.read_csv(vac, parse_dates= ['date'])
df_manu = pd.read_csv(manu, parse_dates = [])
df_manu.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 35623 entries, 0 to 35622
Data columns (total 4 columns):
 #   Column              Non-Null Count  Dtype 
---  ------              --------------  ----- 
 0   location            35623 non-null  object
 1   date                35623 non-null  object
 2   vaccine             35623 non-null  object
 3   total_vaccinations  35623 non-null  int64 
dtypes: int64(1), object(3)
memory usage: 1.1+ MB
```

In [3]:

```
df_vac.tail(5)
```

Out[3]:

	country	iso_code	date	total_vaccinations	people_vaccinated	people_fully_vaccinated	daily_vaccinations_raw	daily_vaccinatio
86507	Zimbabwe	ZWE	2022-03-25	8691642.0	4814582.0	3473523.0	139213.0	6957
86508	Zimbabwe	ZWE	2022-03-26	8791728.0	4886242.0	3487962.0	100086.0	8342
86509	Zimbabwe	ZWE	2022-03-27	8845039.0	4918147.0	3493763.0	53311.0	9062
86510	Zimbabwe	ZWE	2022-03-28	8934360.0	4975433.0	3501493.0	89321.0	10061
86511	Zimbabwe	ZWE	2022-03-29	9039729.0	5053114.0	3510256.0	105369.0	10375

In [4]:

```
df_manu.head()
```

Out[4]:

	location	date	vaccine	total_vaccinations
0	Argentina	2020-12-29	Moderna	2
1	Argentina	2020-12-29	Oxford/AstraZeneca	3
2	Argentina	2020-12-29	Sinopharm/Beijing	1
3	Argentina	2020-12-29	Sputnik V	20481

1	Argentina	2020-12-29	Oxford/AstraZeneca	3
2	Argentina	2020-12-29	Sinopharm/Beijing	1
3	Argentina	2020-12-29	Sputnik V	20481
4	Argentina	2020-12-30	Moderna	2

Close

In [5]:

```
# Check how many SEA countries in dataset
sea = ['Brunei', 'cambodia', 'India', 'Indonesia', 'Laos', 'Malaysia', 'Myanmar', 'Philippines', 'Singapore', 'Thailand', 'V
df_vac[df_vac['country'].isin(sea)]['country'].unique() # 10 countries
```

Out[5]:

```
array(['Brunei', 'India', 'Indonesia', 'Laos', 'Malaysia', 'Myanmar',
       'Philippines', 'Singapore', 'Thailand', 'Vietnam'], dtype=object)
```