

In [1]:

```
import numpy as np
import pandas as pd
import seaborn as sb
import matplotlib.pyplot as plt
from sklearn.preprocessing import LabelEncoder, StandardScaler
from sklearn.decomposition import TruncatedSVD
from sklearn.metrics import confusion_matrix, accuracy_score
from sklearn.linear_model import LogisticRegression
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.cluster import KMeans
import plotly.express as px
```

In [2]:

```
data = pd.read_csv(r'C:\Users\Mannahil Miftah\Downloads\country_vaccinations.csv')
data.head(10)
```

Out[2]:

	country	iso_code	date	total_vaccinations	people_vaccinated	people_fully_vaccinated	daily_vaccinations_raw	daily_v
0	Afghanistan	AFG	2021-02-22	0.0	0.0	NaN	NaN	
1	Afghanistan	AFG	2021-02-23	NaN	NaN	NaN	NaN	
2	Afghanistan	AFG	2021-02-24	NaN	NaN	NaN	NaN	
3	Afghanistan	AFG	2021-02-25	NaN	NaN	NaN	NaN	
4	Afghanistan	AFG	2021-02-26	NaN	NaN	NaN	NaN	
5	Afghanistan	AFG	2021-02-27	NaN	NaN	NaN	NaN	
6	Afghanistan	AFG	2021-02-28	8200.0	8200.0	NaN	NaN	
7	Afghanistan	AFG	2021-03-01	NaN	NaN	NaN	NaN	
8	Afghanistan	AFG	2021-03-02	NaN	NaN	NaN	NaN	
9	Afghanistan	AFG	2021-03-03	NaN	NaN	NaN	NaN	

In [3]:

```
data.shape
```

Out[3]:

(86512, 15)

In [4]:

```
data.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_million',  
      'vaccines', 'source_name', 'source_website'],  
      dtype='object')
```

In [5]:

```
data.dtypes
```

Out[5]:

```
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 [6]:

```
data.info
```

Out[6]:

```
<bound method DataFrame.info of          country iso_code          date  total_vaccinatio  
ns \  
0      Afghanistan      AFG  2021-02-22          0.0  
1      Afghanistan      AFG  2021-02-23          NaN  
2      Afghanistan      AFG  2021-02-24          NaN  
3      Afghanistan      AFG  2021-02-25          NaN  
4      Afghanistan      AFG  2021-02-26          NaN  
...      ...      ...      ...      ...  
86507      Zimbabwe      ZWE  2022-03-25      8691642.0  
86508      Zimbabwe      ZWE  2022-03-26      8791728.0  
86509      Zimbabwe      ZWE  2022-03-27      8845039.0  
86510      Zimbabwe      ZWE  2022-03-28      8934360.0  
86511      Zimbabwe      ZWE  2022-03-29      9039729.0  
  
      people_vaccinated  people_fully_vaccinated  daily_vaccinations_raw  \  
0          0.0          NaN          NaN  
1          NaN          NaN          NaN  
2          NaN          NaN          NaN  
3          NaN          NaN          NaN  
4          NaN          NaN          NaN  
...      ...      ...      ...  
86507      4814582.0      3473523.0      139213.0  
86508      4886242.0      3487962.0      100086.0  
86509      4918147.0      3493763.0       53311.0  
86510      4975433.0      3501493.0       89321.0  
86511      5053114.0      3510256.0      105369.0  
  
      daily_vaccinations  total_vaccinations_per_hundred  \  
0          NaN          0.00
```

1	1367.0	NaN
2	1367.0	NaN
3	1367.0	NaN
4	1367.0	NaN
...	...	...
86507	69579.0	57.59
86508	83429.0	58.25
86509	90629.0	58.61
86510	100614.0	59.20
86511	103751.0	59.90

	people_vaccinated_per_hundred	people_fully_vaccinated_per_hundred	\
0	0.00	NaN	
1	NaN	NaN	
2	NaN	NaN	
3	NaN	NaN	
4	NaN	NaN	
...	...	...	
86507	31.90	23.02	
86508	32.38	23.11	
86509	32.59	23.15	
86510	32.97	23.20	
86511	33.48	23.26	

	daily_vaccinations_per_million	\
0	NaN	
1	34.0	
2	34.0	
3	34.0	
4	34.0	
...	...	
86507	4610.0	
86508	5528.0	
86509	6005.0	
86510	6667.0	
86511	6874.0	

	vaccines	\
0	Johnson&Johnson, Oxford/AstraZeneca, Pfizer/Bi...	
1	Johnson&Johnson, Oxford/AstraZeneca, Pfizer/Bi...	
2	Johnson&Johnson, Oxford/AstraZeneca, Pfizer/Bi...	
3	Johnson&Johnson, Oxford/AstraZeneca, Pfizer/Bi...	
4	Johnson&Johnson, Oxford/AstraZeneca, Pfizer/Bi...	
...	...	
86507	Oxford/AstraZeneca, Sinopharm/Beijing, Sinovac...	
86508	Oxford/AstraZeneca, Sinopharm/Beijing, Sinovac...	
86509	Oxford/AstraZeneca, Sinopharm/Beijing, Sinovac...	
86510	Oxford/AstraZeneca, Sinopharm/Beijing, Sinovac...	
86511	Oxford/AstraZeneca, Sinopharm/Beijing, Sinovac...	

	source_name	\
0	World Health Organization	
1	World Health Organization	
2	World Health Organization	
3	World Health Organization	
4	World Health Organization	
...	...	
86507	Ministry of Health	
86508	Ministry of Health	
86509	Ministry of Health	
86510	Ministry of Health	
86511	Ministry of Health	

	source_website
0	<a href="https://covid19.who.int/">https://covid19.who.int/</a>
1	<a href="https://covid19.who.int/">https://covid19.who.int/</a>
2	<a href="https://covid19.who.int/">https://covid19.who.int/</a>
3	<a href="https://covid19.who.int/">https://covid19.who.int/</a>
4	<a href="https://covid19.who.int/">https://covid19.who.int/</a>
...	...
86507	<a href="https://www.arcgis.com/home/webmap/viewer.html...">https://www.arcgis.com/home/webmap/viewer.html...</a>
86508	<a href="https://www.arcgis.com/home/webmap/viewer.html...">https://www.arcgis.com/home/webmap/viewer.html...</a>

```
86509 https://www.arcgis.com/home/webmap/viewer.html...
86510 https://www.arcgis.com/home/webmap/viewer.html...
86511 https://www.arcgis.com/home/webmap/viewer.html...
```

```
[86512 rows x 15 columns]>
```

```
In [7]:
```

```
data.describe()
```

```
Out[7]:
```

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

```
In [8]:
```

```
data.isna().sum()
```

```
Out[8]:
```

```
country          0
iso_code         0
date            0
total_vaccinations    42905
people_vaccinated    45218
people_fully_vaccinated 47710
daily_vaccinations_raw 51150
daily_vaccinations    299
total_vaccinations_per_hundred    42905
people_vaccinated_per_hundred    45218
people_fully_vaccinated_per_hundred 47710
daily_vaccinations_per_million    299
vaccines          0
source_name       0
source_website    0
dtype: int64
```

```
In [9]:
```

```
data.dropna(inplace = True)
```

```
In [10]:
```

```
data.shape
```

```
Out[10]:
```

```
(30847, 15)
```

```
In [11]:
```

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

```
In [16]:
```

```
top_10_countries = pd.DataFrame(data.groupby('country')['total_vaccinations'].sum()).reset_index()
```

```
top_10_countries = top_10_countries.sort_values(by = 'total_vaccinations', ascending=False)
top_10_countries
```

Out[16]:

	country	total_vaccinations
72	India	3.200418e+11
162	United States	1.550138e+11
19	Brazil	7.637582e+10
29	China	4.858702e+10
73	Indonesia	4.272175e+10
...	...	...
53	Gabon	3.359100e+04
61	Grenada	2.433600e+04
52	French Polynesia	1.625200e+04
3	Andorra	1.526900e+04
22	Burundi	6.340000e+03

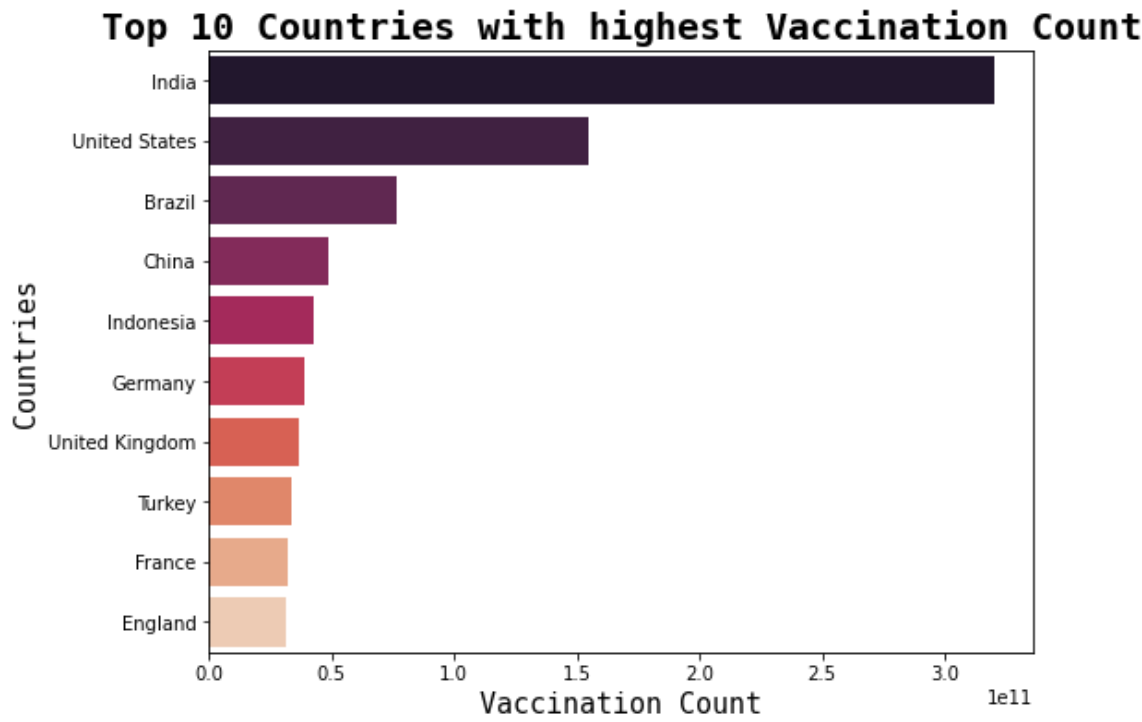
169 rows x 2 columns

In [17]:

```
plt.figure(figsize=[8,6])
sb.barplot(data = top_10_countries.head(10), x = 'total_vaccinations', y = "country", palette = 'rocket')
plt.title('Top 10 Countries with highest Vaccination Count', fontdict={'fontname': 'Monospace', 'fontsize': 20, 'fontweight': 'bold'})
plt.xlabel('Vaccination Count', fontdict={'fontname': 'Monospace', 'fontsize': 15,})
plt.ylabel('Countries', fontdict={'fontname': 'Monospace', 'fontsize': 15,})
```

Out[17]:

Text(0, 0.5, 'Countries')



In [18]:

```
top_10_countries = top_10_countries.sort_values(by = 'total_vaccinations', ascending = True)
```

```
top_10_countries
```

```
Out[18]:
```

	country	total_vaccinations
22	Burundi	6.340000e+03
3	Andorra	1.526900e+04
52	French Polynesia	1.625200e+04
61	Grenada	2.433600e+04
53	Gabon	3.359100e+04
...	...	...
73	Indonesia	4.272175e+10
29	China	4.858702e+10
19	Brazil	7.637582e+10
162	United States	1.550138e+11
72	India	3.200418e+11

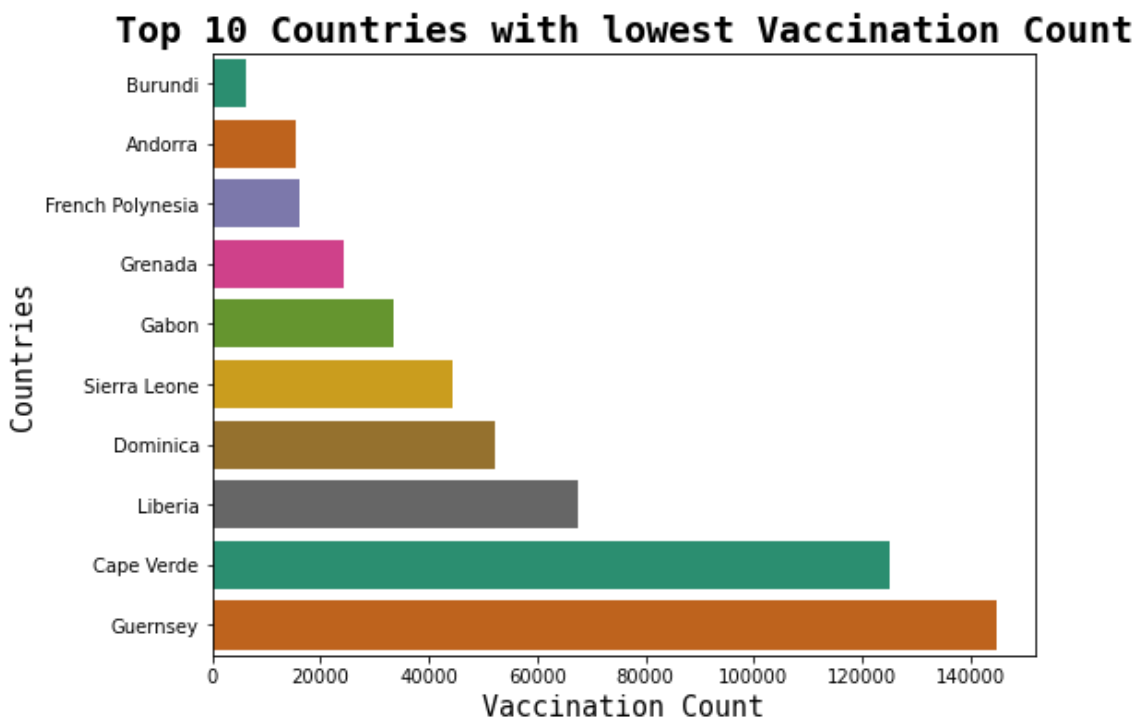
169 rows x 2 columns

```
In [19]:
```

```
plt.figure(figsize=[8,6])
sb.barplot(data = top_10_countries.head(10), x = 'total_vaccinations', y = "country", palette = 'Dark2')
plt.title('Top 10 Countries with lowest Vaccination Count', fontdict={'fontname': 'Monospace', 'fontsize': 20, 'fontweight': 'bold'})
plt.xlabel('Vaccination Count', fontdict={'fontname': 'Monospace', 'fontsize': 15,})
plt.ylabel('Countries', fontdict={'fontname': 'Monospace', 'fontsize': 15,})
```

```
Out[19]:
```

```
Text(0, 0.5, 'Countries')
```



```
In [58]:
```

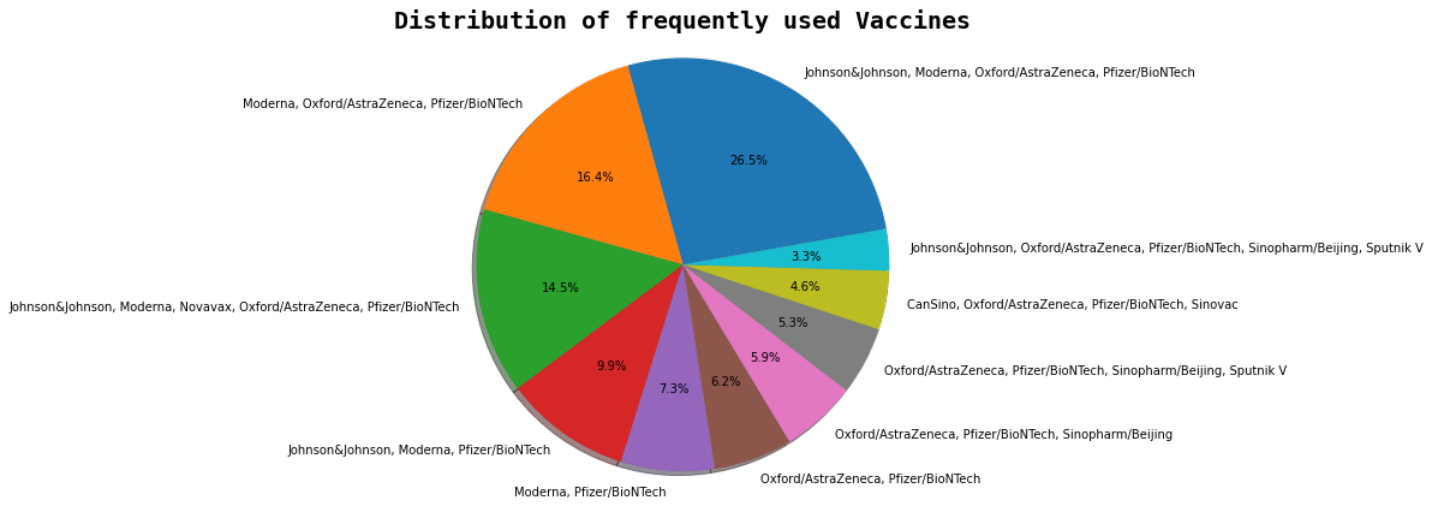
```
vac = pd.DataFrame(data['vaccines'].value_counts()).reset_index()
vac.rename(columns = {'index': 'Name', 'vaccines': 'Count'}, inplace = True)
vac = vac.head(10)
vac
```

Out [58]:

	Name	Count
0	Johnson&Johnson, Moderna, Oxford/AstraZeneca, ...	4832
1	Moderna, Oxford/AstraZeneca, Pfizer/BioNTech	2981
2	Johnson&Johnson, Moderna, Novavax, Oxford/Astr...	2647
3	Johnson&Johnson, Moderna, Pfizer/BioNTech	1808
4	Moderna, Pfizer/BioNTech	1334
5	Oxford/AstraZeneca, Pfizer/BioNTech	1121
6	Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm...	1083
7	Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm...	973
8	CanSino, Oxford/AstraZeneca, Pfizer/BioNTech, ...	838
9	Johnson&Johnson, Oxford/AstraZeneca, Pfizer/Bi...	592

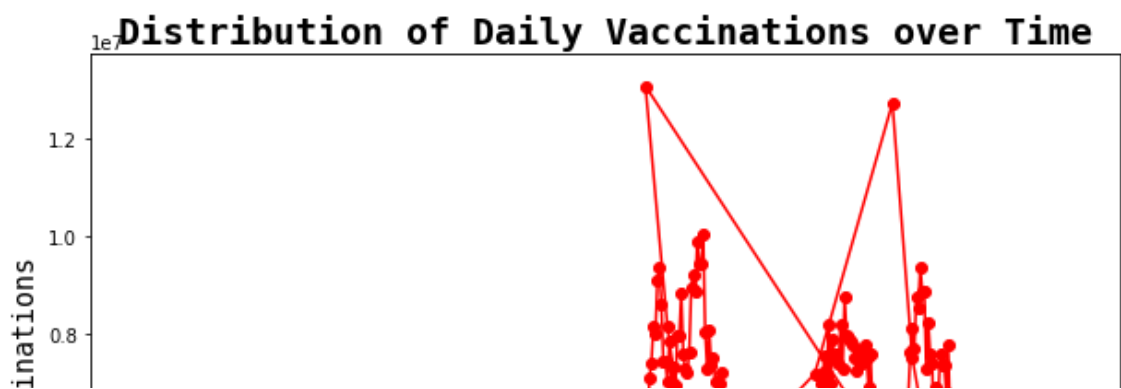
In [92]:

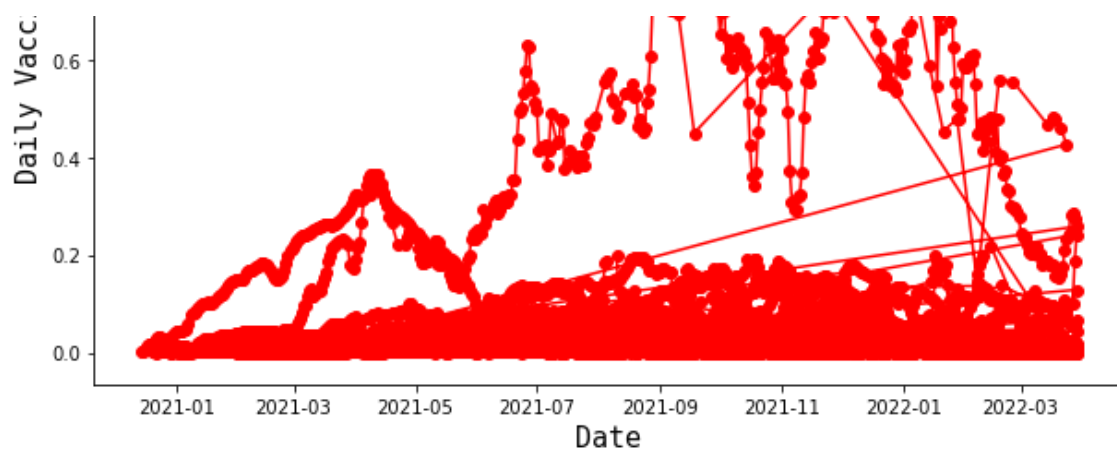
```
plt.figure(figsize=[10,7])
plt.pie(vac.Count, labels = vac.Name, startangle = 10, shadow = True, autopct = '%1.1f%%')
plt.title('Distribution of frequently used Vaccines', fontdict={'fontname': 'Monospace',
'fontsize': 20, 'fontweight': 'bold'})
plt.legend()
plt.axis('equal')
plt.show()
```



In [127]:

```
plt.figure(figsize=(10, 7))
plt.plot(data['date'], data['daily_vaccinations'], marker = 'o', color = 'r')
plt.title('Distribution of Daily Vaccinations over Time', fontdict={'fontname': 'Monospace',
'fontsize': 20, 'fontweight': 'bold'})
plt.xlabel('Date', fontdict={'fontname': 'Monospace', 'fontsize': 15,})
plt.ylabel('Daily Vaccinations', fontdict={'fontname': 'Monospace', 'fontsize': 15,})
plt.show()
```



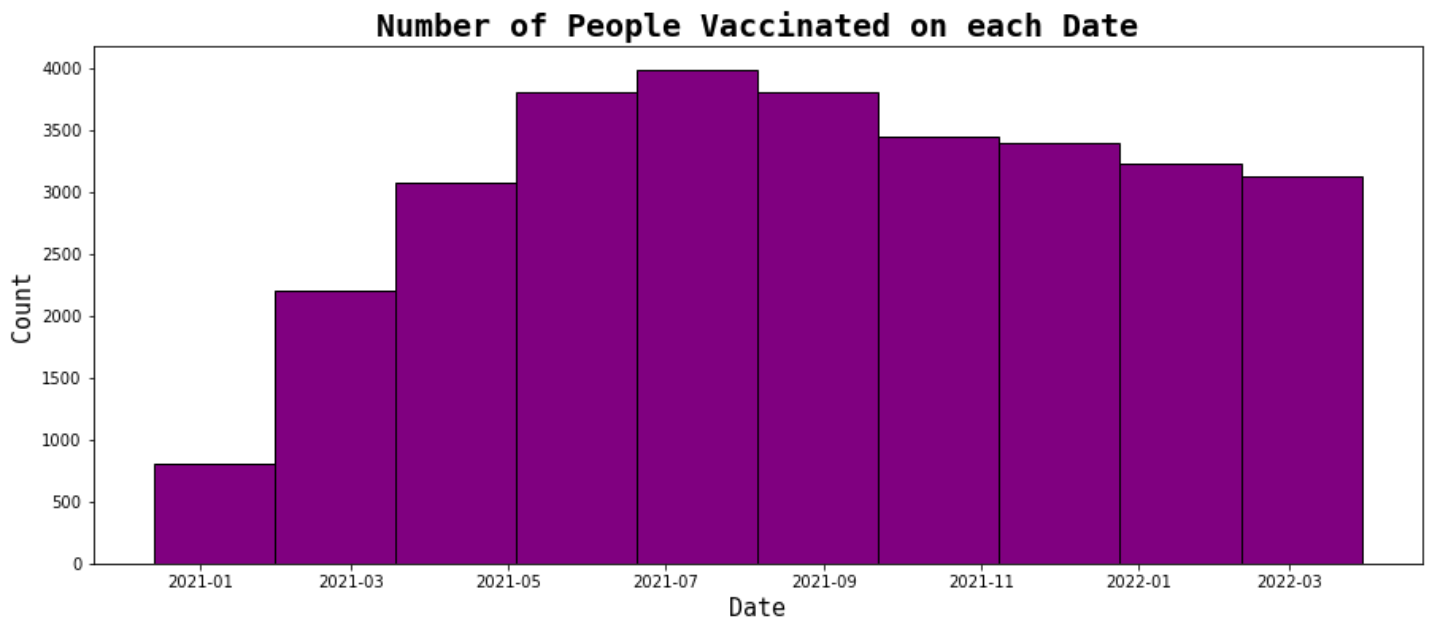


In [128]:

```
plt.figure(figsize=[15,6])
plt.hist(data = data, x = 'date', color = 'purple', ec = 'black')
plt.title('Number of People Vaccinated on each Date', fontdict={'fontname': 'Monospace',
'fontsize': 20, 'fontweight': 'bold'})
plt.xlabel('Date', fontdict={'fontname': 'Monospace', 'fontsize': 15,})
plt.ylabel('Count', fontdict={'fontname': 'Monospace', 'fontsize': 15,})
```

Out[128]:

Text(0, 0.5, 'Count')



In [21]:

```
count = data['source_name'].value_counts()
count = count.head(10)
count
```

Out[21]:

Ministry of Health	7488
Government of the United Kingdom	2169
World Health Organization	1184
Federal Office of Public Health	883
Centers for Disease Control and Prevention	470
Government of Israel	465
National Health Board	457
Robert Koch Institut	457
Statens Serum Institute	455
National Health Service	454

Name: source\_name, dtype: int64

In [22]:

```
plt.figure(figsize=[10,8])
```



```

sb.countplot(count, palette = 'Dark2')
plt.title('Popular Source Names', fontdict={'fontname': 'Monospace', 'fontsize': 20, 'fontweight': 'bold'})
plt.xlabel('Source Name', fontdict={'fontname': 'Monospace', 'fontsize': 15,})
plt.ylabel('Count', fontdict={'fontname': 'Monospace', 'fontsize': 15,})

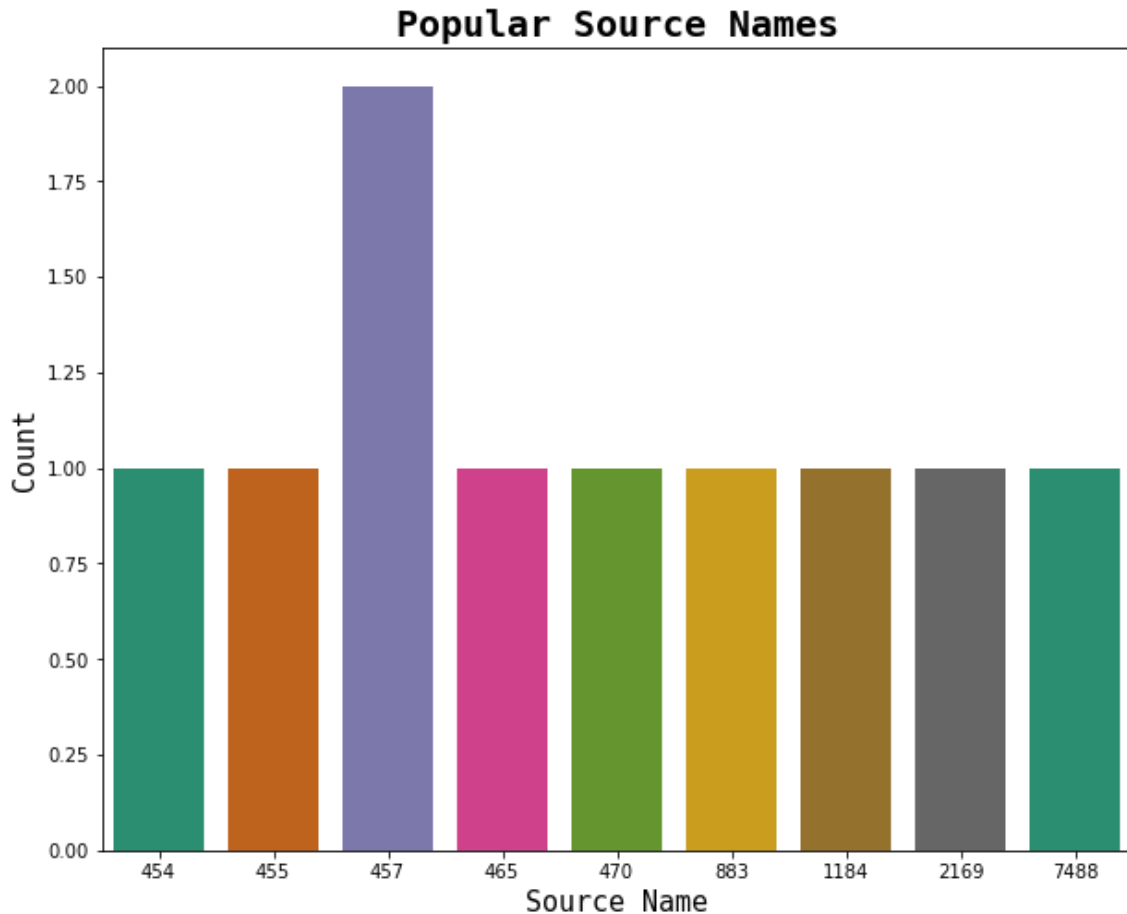
```

c:\Users\Mannahil Miftah\anaconda3\lib\site-packages\seaborn\\_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional argument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.

```
warnings.warn(
```

Out[22]:

```
Text(0, 0.5, 'Count')
```



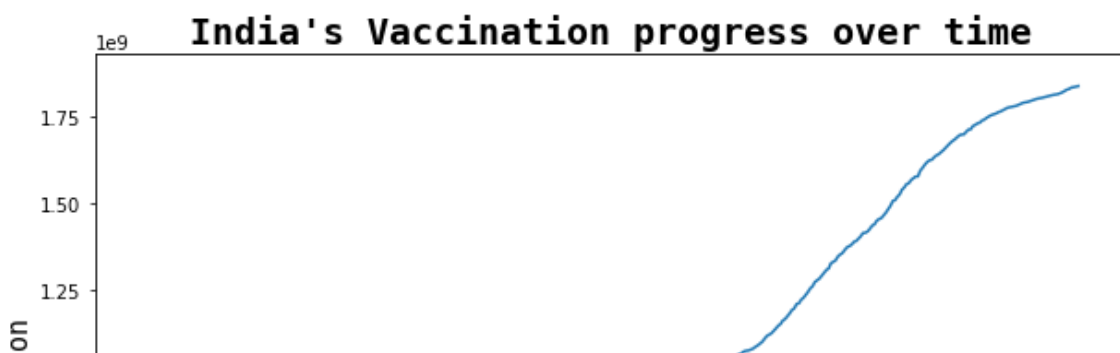
## Time series plots to visualize vaccination progress over time

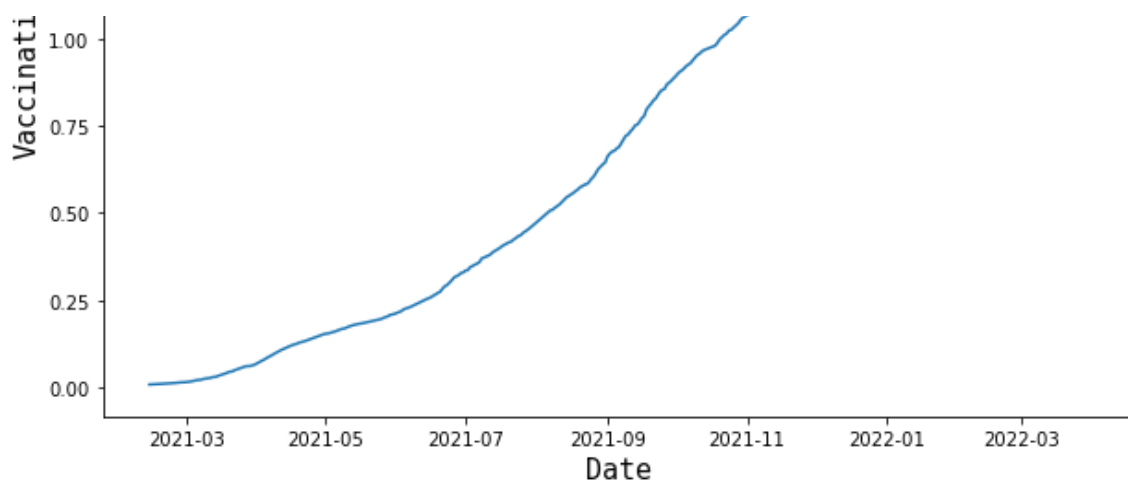
In [97]:

```

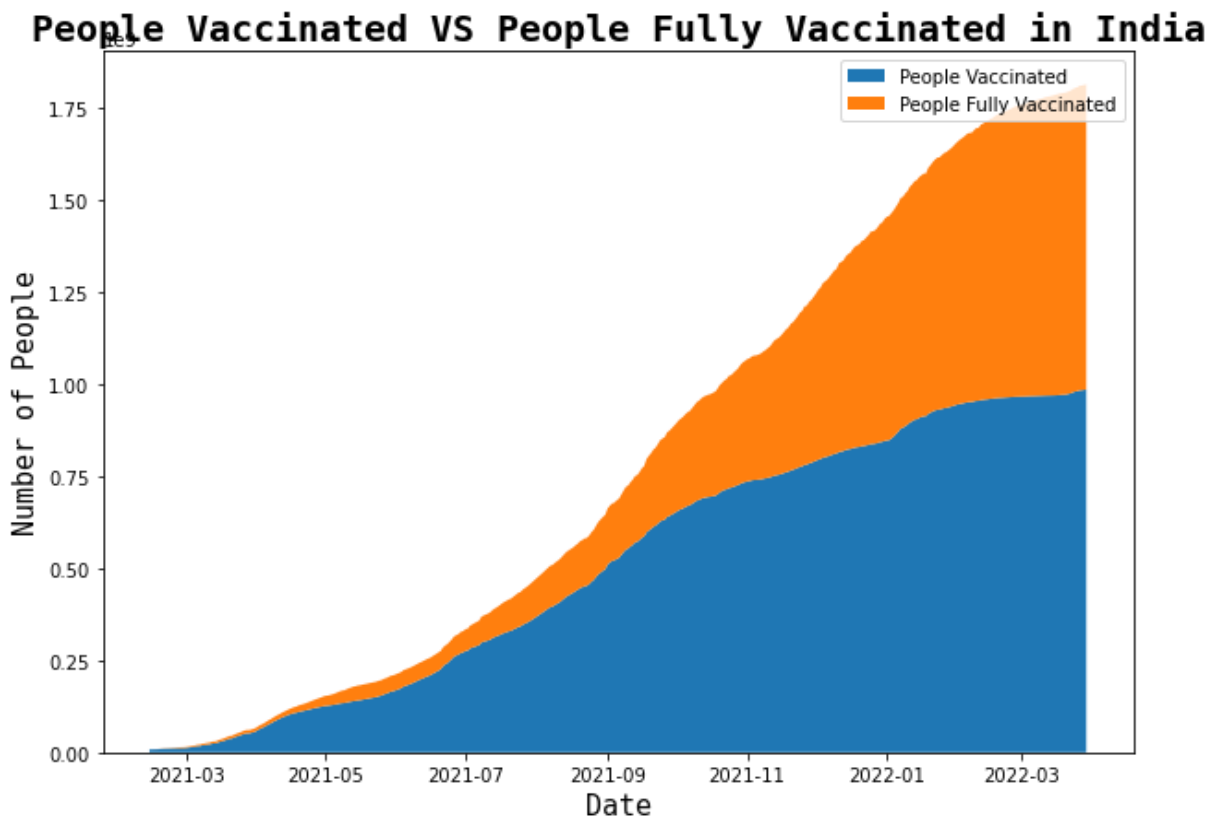
plt.figure(figsize=(10,7))
c = data[data['country'] == 'India']
plt.plot(c['date'], c['total_vaccinations'])
plt.title("India's Vaccination progress over time", fontdict={'fontname': 'Monospace', 'fontsize': 20, 'fontweight': 'bold'})
plt.xlabel('Date', fontdict={'fontname': 'Monospace', 'fontsize': 15,})
plt.ylabel('Vaccination', fontdict={'fontname': 'Monospace', 'fontsize': 15,})
plt.show()

```



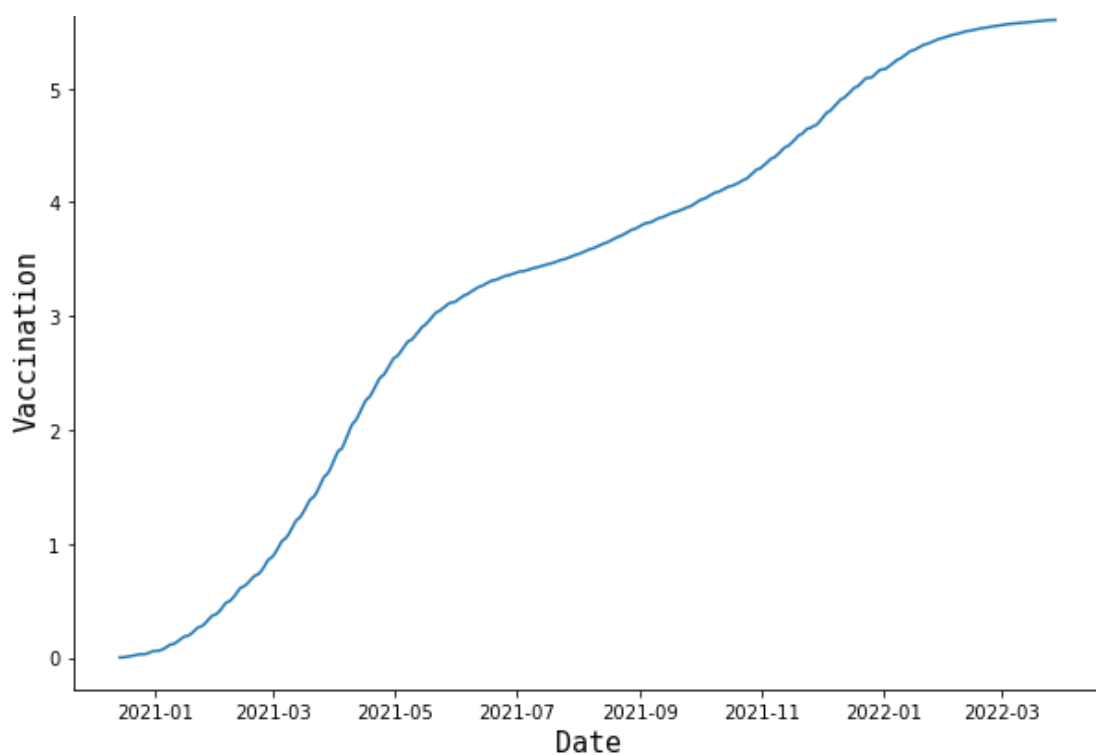


```
In [136]:
country= data[data['country'] == 'India']
plt.figure(figsize=(10,7))
plt.stackplot(country['date'], country['people_vaccinated'], country['people_fully_vaccinated'],
              labels=['People Vaccinated', 'People Fully Vaccinated'])
plt.title('People Vaccinated VS People Fully Vaccinated in India', fontdict={'fontname':
'Monospace', 'fontsize': 20, 'fontweight': 'bold'})
plt.xlabel('Date', fontdict={'fontname': 'Monospace', 'fontsize': 15,})
plt.ylabel('Number of People', fontdict={'fontname': 'Monospace', 'fontsize': 15,})
plt.legend()
plt.show()
```



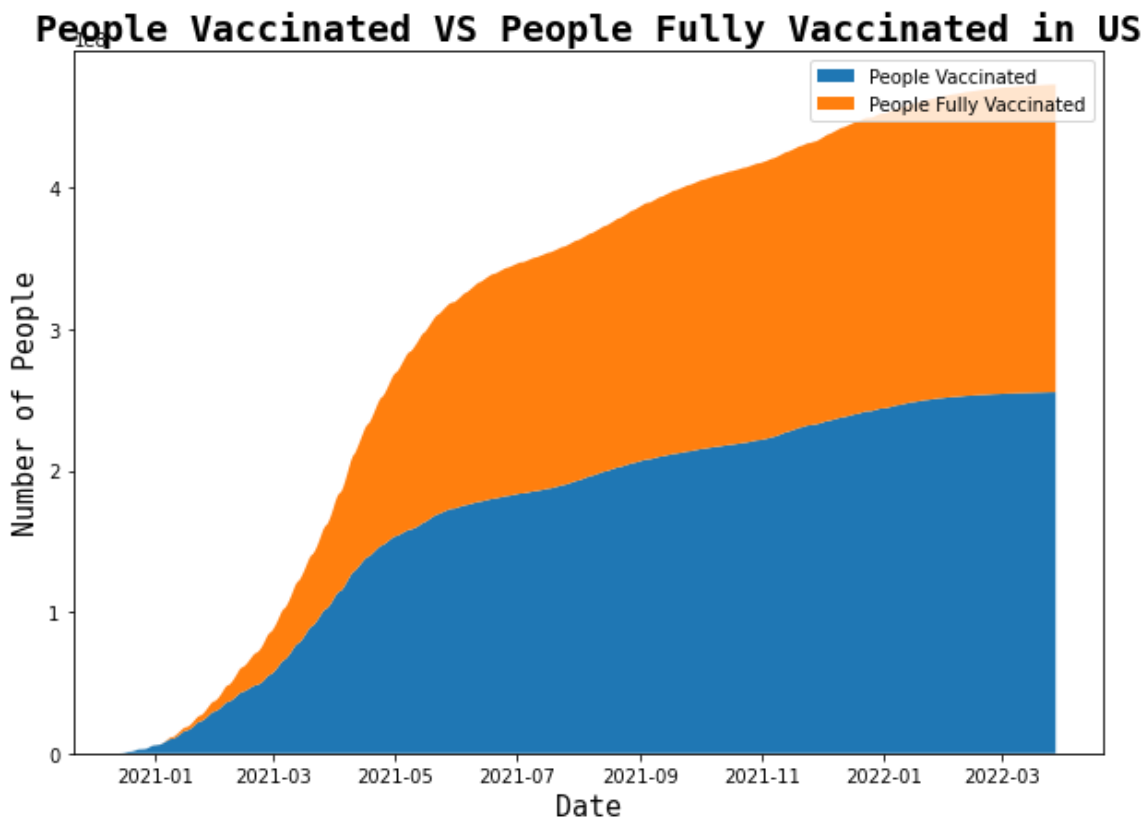
```
In [118]:
plt.figure(figsize=(10,7))
c = data[data['country'] == 'United States']
plt.plot(c['date'], c['total_vaccinations'])
plt.title("US's Vaccination progress over time", fontdict={'fontname': 'Monospace', 'font
size': 20, 'fontweight': 'bold'})
plt.xlabel('Date', fontdict={'fontname': 'Monospace', 'fontsize': 15,})
plt.ylabel('Vaccination', fontdict={'fontname': 'Monospace', 'fontsize': 15,})
plt.show()
```





In [135]:

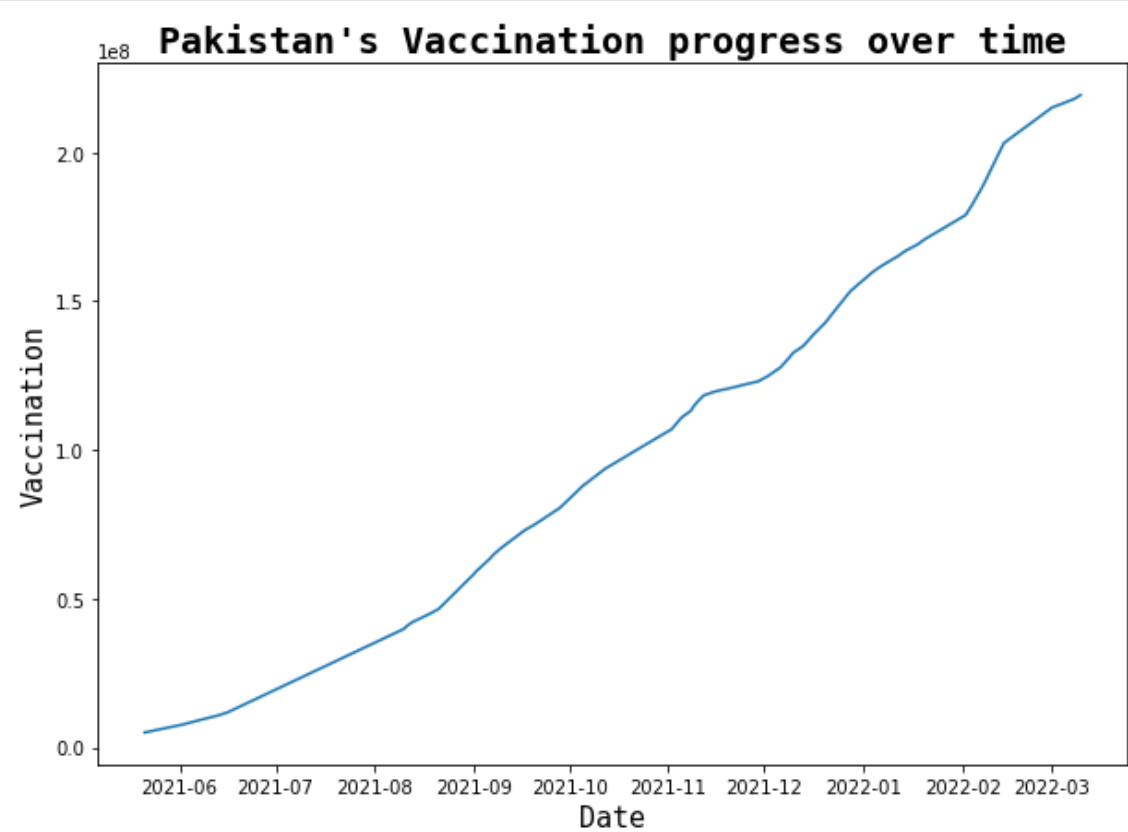
```
country= data[data['country'] == 'United States']
plt.figure(figsize=(10,7))
plt.stackplot(country['date'], country['people_vaccinated'], country['people_fully_vaccinated'],
              labels=['People Vaccinated', 'People Fully Vaccinated'])
plt.title('People Vaccinated VS People Fully Vaccinated in US', fontdict={'fontname': 'Monospace', 'fontsize': 20, 'fontweight': 'bold'})
plt.xlabel('Date', fontdict={'fontname': 'Monospace', 'fontsize': 15,})
plt.ylabel('Number of People', fontdict={'fontname': 'Monospace', 'fontsize': 15,})
plt.legend()
plt.show()
```



In [133]:

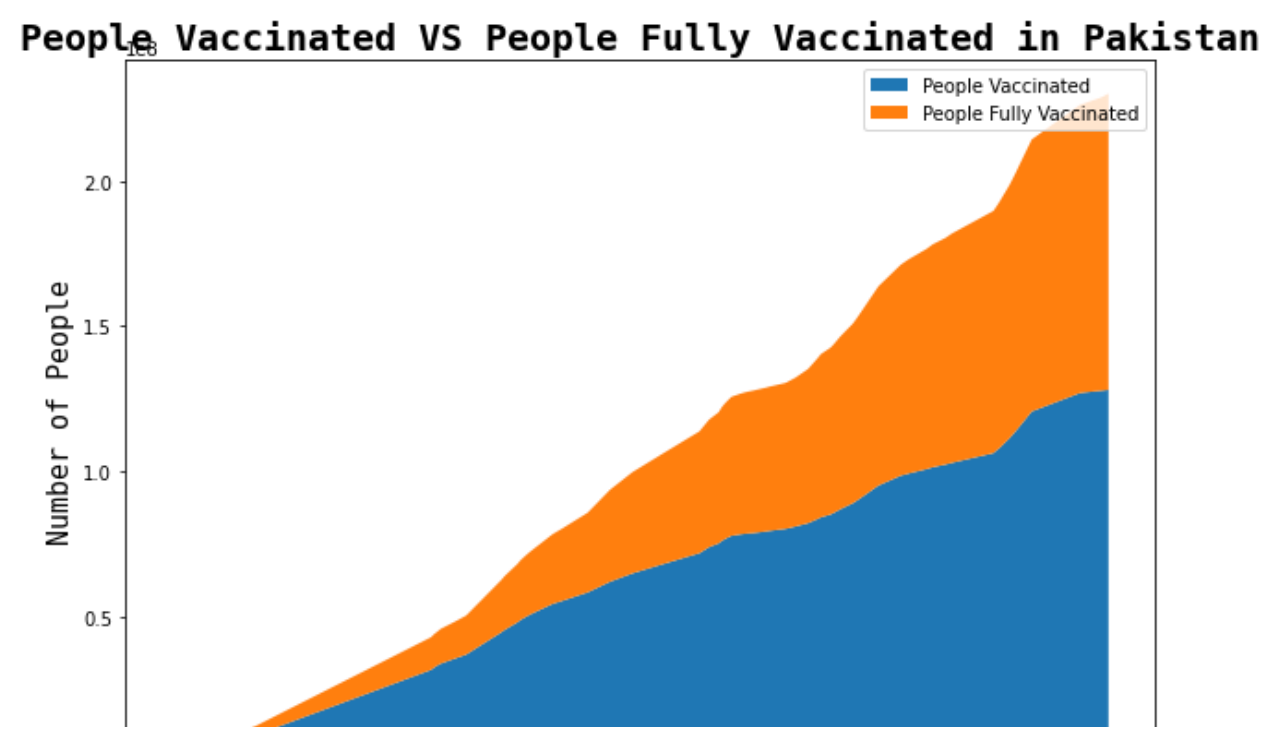
```
plt.figure(figsize=(10,7))
c = data[data['country'] == 'Pakistan']
```

```
plt.plot(c['date'], c['total_vaccinations'])
plt.title("Pakistan's Vaccination progress over time", fontdict={'fontname': 'Monospace',
'fontsize': 20, 'fontweight': 'bold'})
plt.xlabel('Date', fontdict={'fontname': 'Monospace', 'fontsize': 15,})
plt.ylabel('Vaccination', fontdict={'fontname': 'Monospace', 'fontsize': 15,})
plt.show()
```



In [134]:

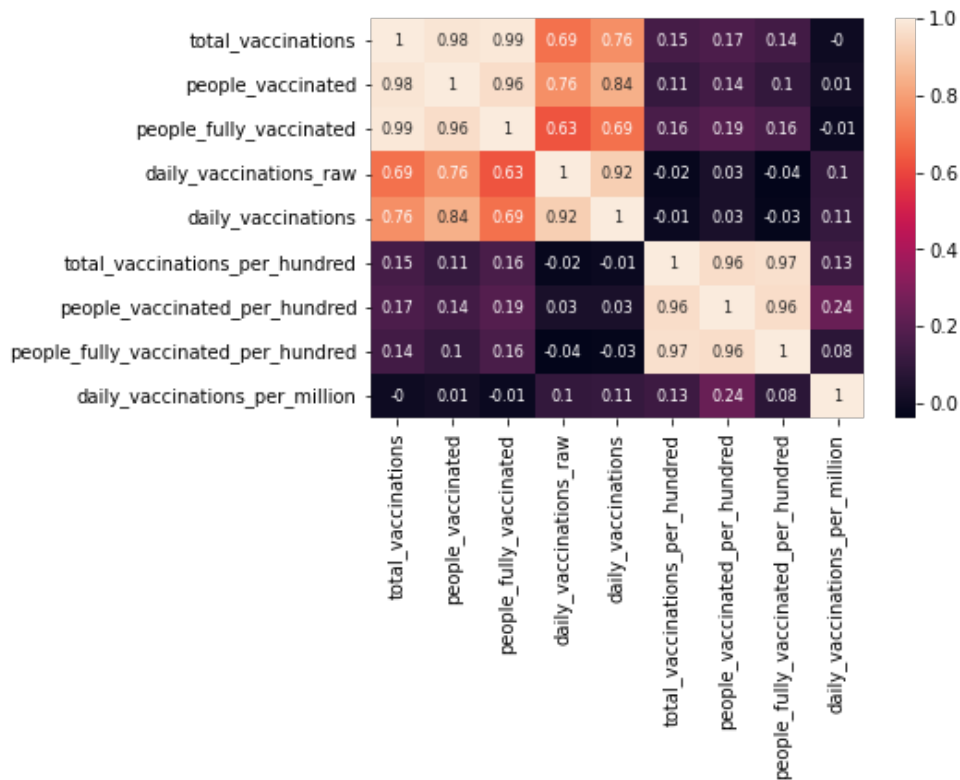
```
country= data[data['country'] == 'Pakistan']
plt.figure(figsize=(10,7))
plt.stackplot(country['date'], country['people_vaccinated'], country['people_fully_vacci
nated'],
              labels=['People Vaccinated', 'People Fully Vaccinated'])
plt.title('People Vaccinated VS People Fully Vaccinated in Pakistan', fontdict={'fontname
': 'Monospace', 'fontsize': 20, 'fontweight': 'bold'})
plt.xlabel('Date', fontdict={'fontname': 'Monospace', 'fontsize': 15,})
plt.ylabel('Number of People', fontdict={'fontname': 'Monospace', 'fontsize': 15,})
plt.legend()
plt.show()
```





In [113]:

```
sb.heatmap(data=data.corr().round(2), annot=True, annot_kws={"size":8})
plt.show()
```



In [ ]: