```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
data=pd.read_csv("country_vaccinations.csv")
data_manu=pd.read_csv("country_vaccinations_by_manufacturer.csv")
```

```
print(data.shape)
print(data_manu.shape)
```

```
(52366, 15)
(35623, 4)
```

```
data.head()
```

	country	iso_code	date	total_vaccinations	people_vaccinated	people_fully_vaccinated	daily_vaccinations_raw	daily_vaccinations
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

```
data_manu.head()
```

	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

make statistical aalysis on the datas

```
data.describe()
```

	total_vaccinations	people_vaccinated	people_fully_vaccinated	daily_vaccinations_raw	daily_vaccinations	total_vaccinations_per
count	2.705700e+04	2.603900e+04	2.437700e+04	2.193500e+04	5.216000e+04	270
mean	5.691027e+07	1.923211e+07	1.487287e+07	3.420715e+05	1.613521e+05	
std	2.795499e+08	8.494562e+07	6.809048e+07	1.511664e+06	9.695998e+05	
min	0.000000e+00	0.000000e+00	1.000000e+00	0.000000e+00	0.000000e+00	
25%	3.963830e+05	2.793155e+05	1.731230e+05	3.230500e+03	7.380000e+02	
50%	2.589469e+06	1.580100e+06	1.190856e+06	2.008900e+04	5.568500e+03	
75%	1.521038e+07	7.464513e+06	6.170630e+06	1.160790e+05	3.537500e+04	1
max	3.263129e+09	1.275541e+09	1.240777e+09	2.474100e+07	2.242429e+07	3

on the manufactururas data we can only have total vaccination in the countries on day by day

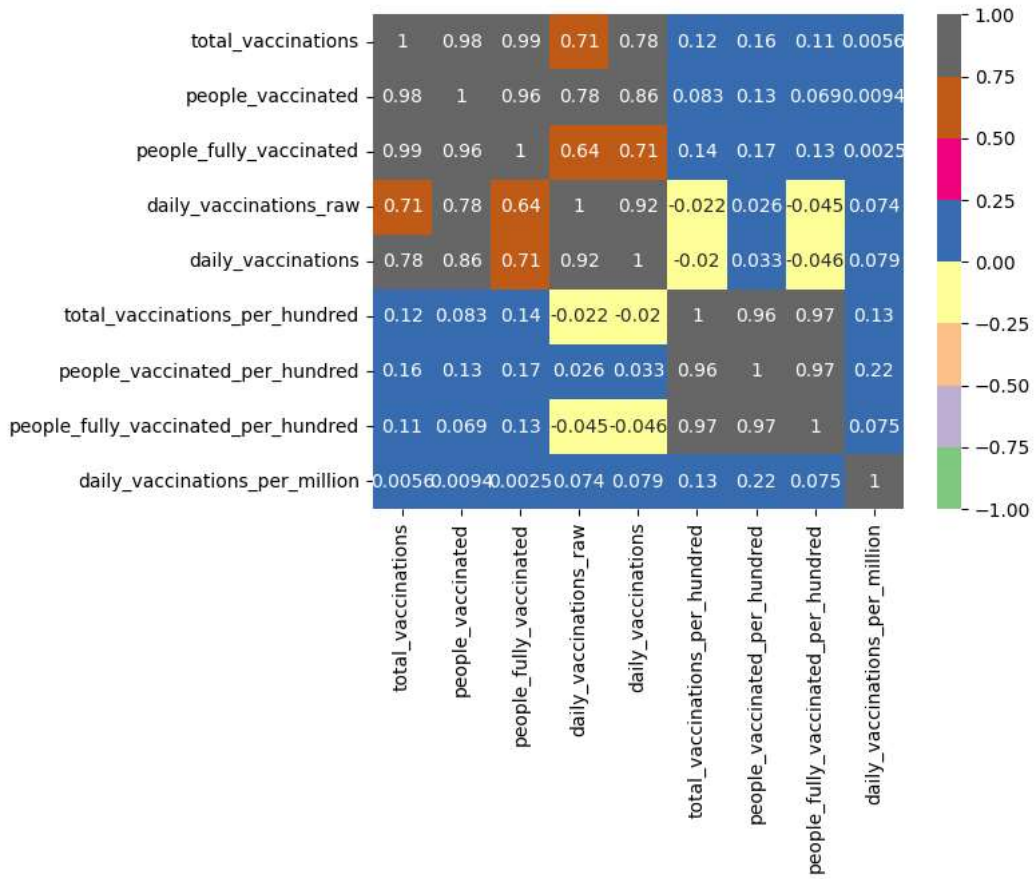
```
data_manu.describe()
```

	total_vaccinations
count	3.562300e+04
mean	1.508357e+07
std	5.181768e+07
min	0.000000e+00
25%	9.777600e+04
50%	1.305506e+06
75%	7.932423e+06
max	6.005200e+08

Find Correlation of the column

```
sns.heatmap(data.corr(),cmap="Accent",annot=True,vmin=-1,vmax=1,center=0)
```

```
<ipython-input-35-4236846674f2>:1: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a future version
  sns.heatmap(data.corr(),cmap="Accent",annot=True,vmin=-1,vmax=1,center=0)
<Axes: >
```



Number of Countries that are present in dataset

```
len(data["country"].unique())
```

```
134
```

checking the cont of missing values

```
print(data.isnull().sum())
print("----------------------------------------")
print(data_manu.isnull().sum())
```

```
country                                0
iso_code                               0
date                                   0
total_vaccinations                 25309
people_vaccinated                  26327
people_fully_vaccinated            27989
daily_vaccinations_raw             30431
daily_vaccinations                   206
total_vaccinations_per_hundred     25309
people_vaccinated_per_hundred      26327
people_fully_vaccinated_per_hundred    27989
daily_vaccinations_per_million       206
vaccines                               0
source_name                            0
source_website                         0
dtype: int64
----------------------------------------
location             0
date                 0
vaccine              0
total_vaccinations   0
dtype: int64
```

dataset given by manufacturer have no null value

But country vaccination datset have null values so remove null values or missing values make a good analysis

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

```
data.head()
```

	country	iso_code	date	total_vaccinations	people_vaccinated	people_fully_vaccinated	daily_vaccinations_raw	daily_vaccinatio
94	Afghanistan	AFG	2021-05-27	593313.0	479574.0	113739.0	2859.0	648
101	Afghanistan	AFG	2021-06-03	630305.0	481800.0	148505.0	4015.0	528
339	Afghanistan	AFG	2022-01-27	5081064.0	4517380.0	3868832.0	6868.0	980
433	Albania	ALB	2021-02-18	3049.0	2438.0	611.0	1348.0	25
515	Albania	ALB	2021-05-11	622507.0	440921.0	181586.0	9548.0	1216

Make some visualizations

Distributions of the data

```
sns.histplot(data=data_manu["total_vaccinations"],bins=20,color= 'red')
```

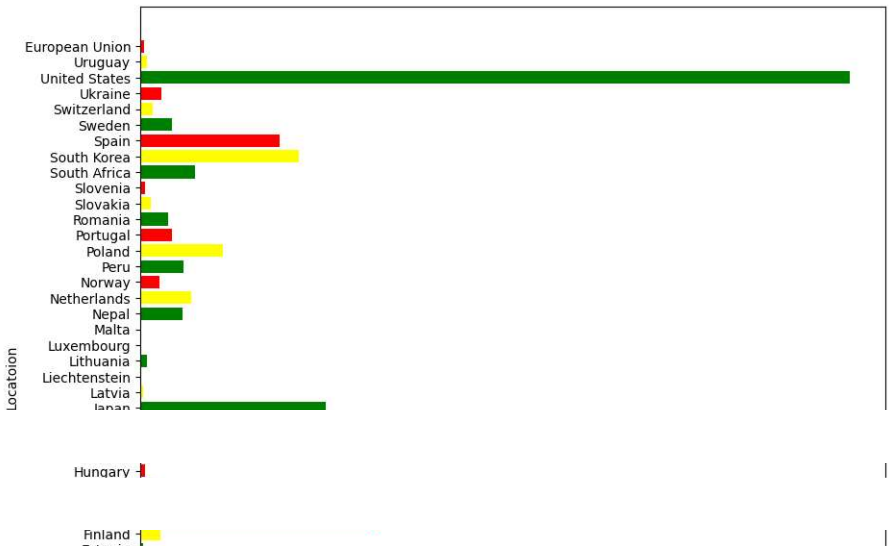
```
<Axes: xlabel='total_vaccinations', ylabel='Count'>
```

Calculate the Total vaccination on date wise

```
z=0
total_vac1=[]
for x in data_manu["location"].unique():
     for y in data_manu["location"]:
        if x == y:
            total= 0 + data_manu["total_vaccinations"].iloc[z]
            z=z+1
     total_vac1.append(total)


z=0
total_vac2=[]
for x in data["country"].unique():
     for y in data["country"]:
        if x == y:
            total= 0 + data["total_vaccinations"].iloc[z]
            z=z+1
     total_vac2.append(total)
```

Country wise vaccination

```
x=data_manu["location"].unique()
y=total_vac1
plt.figure(figsize=(10,10))
plt.ylabel("Locatoion")
plt.xlabel("Total_vaccination")
plt.barh(x,y,color=["red",'green','yellow'])
plt.show()
```



Calculate the peple_vaccinated and people_fully_vaccinated for countries

```
z=0
people_vac=[]
for x in data["country"].unique():
     for y in data["country"]:
        if x == y:
            total= 0 + data["people_vaccinated"].iloc[z]
            z=z+1
     people_vac.append(total)
```

```
z=0
people_fulvac=[]
for x in data["country"].unique():
     for y in data["country"]:
        if x == y:
            total= 0 + data["people_fully_vaccinated"].iloc[z]
            z=z+1
     people_fulvac.append(total)
```

Create dataframe for country wise vaccination distibution

```
datafre={"country":data['country'].unique(),"People_Vaccinated":people_vac,"people_Fully_vaccinated":people_fulvac}
new_data=pd.DataFrame(datafre)
new_data
```

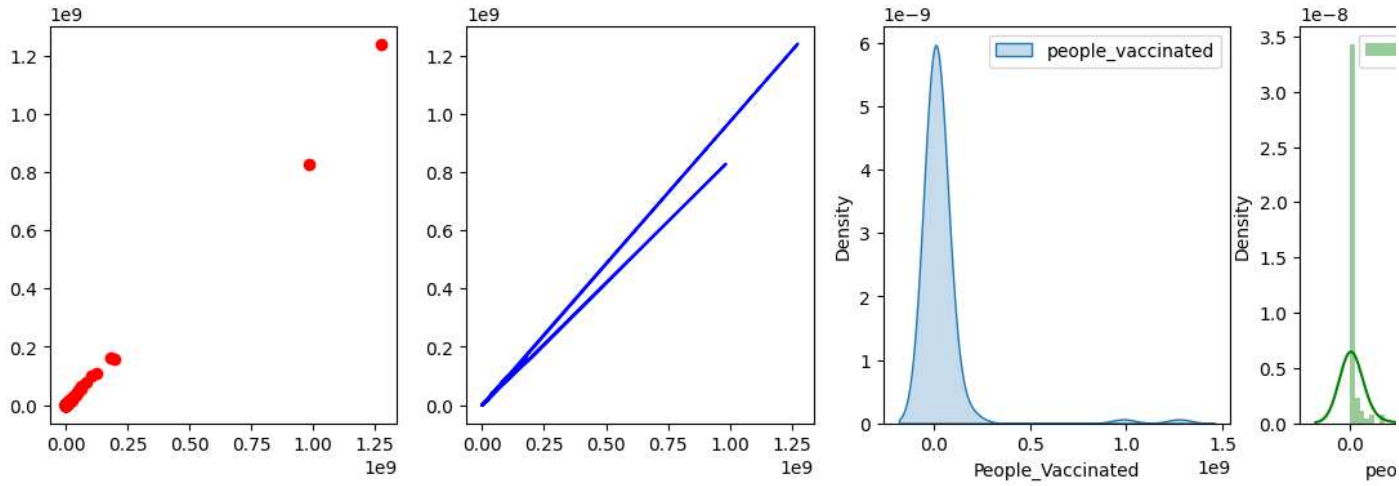
	country	People_Vaccinated	people_Fully_vaccinated
0	Afghanistan	4517380.0	3868832.0
1	Albania	1275907.0	1209791.0
2	Algeria	6875003.0	5391232.0
3	Andorra	9781.0	4484.0
4	Antigua and Barbuda	63704.0	61327.0
...	...	...	...
102	Mexico	85580293.0	79711762.0
103	Moldova	557291.0	513094.0
104	Mongolia	2272327.0	2174398.0
105	Montenegro	289687.0	281571.0
106	Morocco	4284060.0	2844916.0

107 rows × 3 columns

```
x1=new_data["People_Vaccinated"]
y1=new_data["people_Fully_vaccinated"]
```

```
plt.figure(figsize=(16,4))
plt.subplot(1,4,1)
plt.scatter(x1,y1,color="red")
plt.subplot(1,4,2)
plt.plot(x1,y1,color='blue')
plt.subplot(1,4,3)
sns.kdeplot(x1,shade=True,label="people_vaccinated")
plt.legend()
plt.subplot(1,4,4)
sns.distplot(y1,color='green',label='people_fully_vaccinated')
plt.legend()
plt.show()
```

```
<ipython-input-23-fffecbf8546b>:9: FutureWarning:

`shade` is now deprecated in favor of `fill`; setting `fill=True`.
This will become an error in seaborn v0.14.0; please update your code.

  sns.kdeplot(x1,shade=True,label="people_vaccinated")
<ipython-input-23-fffecbf8546b>:12: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(y1,color='green',label='people_fully_vaccinated')
```



From above graph we can say that these two fields are positively relatived

```
sns.heatmap(new_data.corr(),cmap='GnBu',annot=True)
```

```
<ipython-input-33-2bfd1593bd5d>:1: FutureWarning: The default value of numeric_only in DataFrame.corr is deprecated. In a future versior
  sns.heatmap(new_data.corr(),cmap='GnBu',annot=True)
<Axes: >
```