In [1]:

```
#Task:Task to compare the rate of people vaccinated in United States and India
#Program By:Ayush Pandey
#Email Id:1805290@kiit.ac.in
#DATE:4-Oct-2021
#Python Version:3.7
#CAVEATS:None
#LICENSE:None
```

In [2]:

```
import pandas as pd
import numpy as np
from statistics import mode

# data extraction
df = pd.read_csv('country_vaccinations.csv')
df.head()
# vaccine combinations being used
```

Out[2]:

	Unnamed: 0	country	iso_code	date	total_vaccinations	people_vaccinated	people_fully_vaccinated	daily_vaccinations_raw	daily_vaccinations	total_vaccinations_per_hundred	pe
0	0	Albania	ALB	1/10/2021	0.0	0.0	0.0	0.0	0.0	0.00	
1	1	Albania	ALB	1/11/2021	0.0	0.0	0.0	0.0	64.0	0.00	
2	2	Albania	ALB	1/12/2021	128.0	128.0	0.0	0.0	64.0	0.00	
3	3	Albania	ALB	1/13/2021	188.0	188.0	0.0	60.0	63.0	0.01	
4	4	Albania	ALB	1/14/2021	266.0	266.0	0.0	78.0	66.0	0.01	

In [3]:

```
from collections import Counter

country_list = df['country'].unique()

vaccine_ava = df['vaccines'].unique()
print(country_list)
print("")
print(vaccine_ava)
```

```
['Albania' 'Algeria' 'Andorra' 'Anguilla' 'Argentina' 'Australia'
 'Austria' 'Azerbaijan' 'Bahrain' 'Bangladesh' 'Barbados' 'Belarus'
 'Belgium' 'Bermuda' 'Bolivia' 'Brazil' 'Bulgaria' 'Cambodia' 'Canada'
 'Cayman Islands' 'Chile' 'China' 'Colombia' 'Costa Rica' 'Croatia'
 'Cyprus' 'Czechia' 'Denmark' 'Dominican Republic' 'Ecuador' 'Egypt'
 'England' 'Estonia' 'Faeroe Islands' 'Falkland Islands' 'Finland'
 'France' 'Germany' 'Gibraltar' 'Greece' 'Greenland' 'Guernsey' 'Guyana'
 'Hungary' 'Iceland' 'India' 'Indonesia' 'Iran' 'Ireland' 'Isle of Man'
 'Israel' 'Italy' 'Japan' 'Jersey' 'Kuwait' 'Latvia' 'Lebanon'
 'Liechtenstein' 'Lithuania' 'Luxembourg' 'Macao' 'Maldives' 'Malta'
 'Mauritius' 'Mexico' 'Monaco' 'Morocco' 'Myanmar' 'Nepal' 'Netherlands'
 'New Zealand' 'Northern Cyprus' 'Northern Ireland' 'Norway' 'Oman'
 'Pakistan' 'Panama' 'Peru' 'Poland' 'Portugal' 'Qatar' 'Romania' 'Russia'
 'Saint Helena' 'Saudi Arabia' 'Scotland' 'Serbia' 'Seychelles'
 'Singapore' 'Slovakia' 'Slovenia' 'South Africa' 'Spain' 'Sri Lanka'
 'Sweden' 'Switzerland' 'Turkey' 'Turks and Caicos Islands'
 'United Arab Emirates' 'United Kingdom' 'United States' 'Wales']

['Pfizer/BioNTech' 'Sputnik V' 'Oxford/AstraZeneca'
 'Moderna, Oxford/AstraZeneca, Pfizer/BioNTech'
 'Oxford/AstraZeneca, Sputnik V' 'Pfizer/BioNTech, Sinopharm/Beijing'
 'Oxford/AstraZeneca, Sinovac' 'Sinopharm/Beijing'
 'Moderna, Pfizer/BioNTech' 'Pfizer/BioNTech, Sinovac'
 'Sinopharm/Beijing, Sinopharm/Wuhan, Sinovac'
 'Oxford/AstraZeneca, Pfizer/BioNTech' 'Covaxin, Oxford/AstraZeneca'
 'Sinovac' 'Oxford/AstraZeneca, Sinopharm/Beijing'
 'Oxford/AstraZeneca, Sinopharm/Beijing, Sputnik V'
 'Pfizer/BioNTech, Sinopharm/Beijing, Sputnik V' 'Johnson&Johnson'
 'Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sinopharm/Wuhan, Sputnik V']
```

In [4]:

```
vaccine_used = []
for i in country_list:
    vaccine_used.append(df[df['country']==i]['vaccines'].mode(0))
vaccine_used = pd.Series(vaccine_used)
print(vaccine_used)
```

```
0                    0    Pfizer/BioNTech
dtype: object
1                          0    Sputnik V
dtype: object
2                    0    Pfizer/BioNTech
dtype: object
3                 0    Oxford/AstraZeneca
dtype: object
4                          0    Sputnik V
dtype: object
                          ...
97                   0    Pfizer/BioNTech
dtype: object
98     0    Oxford/AstraZeneca, Pfizer/BioNTech, Sino...
99     0    Oxford/AstraZeneca, Pfizer/BioNTech
dtype...
100          0    Moderna, Pfizer/BioNTech
dtype: object
101    0    Oxford/AstraZeneca, Pfizer/BioNTech
dtype...
Length: 102, dtype: object
```

In [5]:

```
vaccine_mode = []
for i in range(0, 102):
    vaccine_mode.append(vaccine_used[i][0])

#1st method
"""words_to_count = (word for word in vaccine_mode if word[:1].isupper())
c = Counter(words_to_count)
print(c)"""
#Till here

c={}
#print(vaccine_mode)

#2nd method
for i in vaccine_mode:
    if(i not in c):
        c[i]=1
    else:
        c[i]+=1
print(c)
c=Counter(c)
print(len(c))
#Till here

count = []

names_vaccine = []
for i in range(0,19):
    count.append(c.most_common()[i][1])
    names_vaccine.append(c.most_common()[i][0])
#print("Most widely used vaccine: ", names_vaccine[0])
print("Most widely used vaccine: ", mode(vaccine_mode))

print("vaccines used in India:  ", df[df['country']=='India']['vaccines'].mode(0)[0])
```

```
{'Pfizer/BioNTech': 27, 'Sputnik V': 6, 'Oxford/AstraZeneca': 12, 'Moderna, Oxford/AstraZeneca, Pfizer/BioNTech': 20, 'Oxford/AstraZeneca, Sputnik
V': 1, 'Pfizer/BioNTech, Sinopharm/Beijing': 1, 'Oxford/AstraZeneca, Sinovac': 1, 'Sinopharm/Beijing': 4, 'Moderna, Pfizer/BioNTech': 7, 'Pfizer/B
ioNTech, Sinovac': 2, 'Sinopharm/Beijing, Sinopharm/Wuhan, Sinovac': 1, 'Oxford/AstraZeneca, Pfizer/BioNTech': 11, 'Covaxin, Oxford/AstraZeneca':
1, 'Sinovac': 2, 'Oxford/AstraZeneca, Sinopharm/Beijing': 2, 'Oxford/AstraZeneca, Sinopharm/Beijing, Sputnik V': 1, 'Pfizer/BioNTech, Sinopharm/Be
ijing, Sputnik V': 1, 'Johnson&Johnson': 1, 'Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sinopharm/Wuhan, Sputnik V': 1}
19
Most widely used vaccine:  Pfizer/BioNTech
vaccines used in India:   Covaxin, Oxford/AstraZeneca
```

In [6]:

```
#For each country checking the vaccine used frequently
def get(x):
    ans=df[df['country']==x]['vaccines'].mode(0)[0]
    if(x not in see):
        see[x]=ans
```

In [7]:

```
ans=0
see={}
q=df['country'].apply(get)
```

In [8]:

```
print(see)
print(len(see))
```

```
{'Albania': 'Pfizer/BioNTech', 'Algeria': 'Sputnik V', 'Andorra': 'Pfizer/BioNTech', 'Anguilla': 'Oxford/AstraZeneca', 'Argentina': 'Sputnik V',
'Australia': 'Pfizer/BioNTech', 'Austria': 'Moderna, Oxford/AstraZeneca, Pfizer/BioNTech', 'Azerbaijan': 'Oxford/AstraZeneca, Sputnik V', 'Bahrai
n': 'Pfizer/BioNTech, Sinopharm/Beijing', 'Bangladesh': 'Oxford/AstraZeneca', 'Barbados': 'Oxford/AstraZeneca', 'Belarus': 'Sputnik V', 'Belgium':
'Moderna, Oxford/AstraZeneca, Pfizer/BioNTech', 'Bermuda': 'Pfizer/BioNTech', 'Bolivia': 'Sputnik V', 'Brazil': 'Oxford/AstraZeneca, Sinovac', 'Bu
lgaria': 'Moderna, Oxford/AstraZeneca, Pfizer/BioNTech', 'Cambodia': 'Sinopharm/Beijing', 'Canada': 'Moderna, Pfizer/BioNTech', 'Cayman Islands':
'Pfizer/BioNTech', 'Chile': 'Pfizer/BioNTech, Sinovac', 'China': 'Sinopharm/Beijing, Sinopharm/Wuhan, Sinovac', 'Colombia': 'Pfizer/BioNTech', 'Co
sta Rica': 'Pfizer/BioNTech', 'Croatia': 'Pfizer/BioNTech', 'Cyprus': 'Pfizer/BioNTech', 'Czechia': 'Moderna, Oxford/AstraZeneca, Pfizer/BioNTec
h', 'Denmark': 'Moderna, Pfizer/BioNTech', 'Dominican Republic': 'Oxford/AstraZeneca', 'Ecuador': 'Pfizer/BioNTech', 'Egypt': 'Sinopharm/Beijing',
'England': 'Oxford/AstraZeneca, Pfizer/BioNTech', 'Estonia': 'Moderna, Oxford/AstraZeneca, Pfizer/BioNTech', 'Faeroe Islands': 'Pfizer/BioNTech',
'Falkland Islands': 'Oxford/AstraZeneca', 'Finland': 'Moderna, Oxford/AstraZeneca, Pfizer/BioNTech', 'France': 'Moderna, Oxford/AstraZeneca, Pfize
r/BioNTech', 'Germany': 'Moderna, Oxford/AstraZeneca, Pfizer/BioNTech', 'Gibraltar': 'Pfizer/BioNTech', 'Greece': 'Moderna, Oxford/AstraZeneca, Pf
izer/BioNTech', 'Greenland': 'Pfizer/BioNTech', 'Guernsey': 'Pfizer/BioNTech', 'Guyana': 'Oxford/AstraZeneca', 'Hungary': 'Pfizer/BioNTech', 'Icel
and': 'Moderna, Oxford/AstraZeneca, Pfizer/BioNTech', 'India': 'Covaxin, Oxford/AstraZeneca', 'Indonesia': 'Sinovac', 'Iran': 'Sputnik V', 'Irelan
d': 'Moderna, Oxford/AstraZeneca, Pfizer/BioNTech', 'Isle of Man': 'Oxford/AstraZeneca, Pfizer/BioNTech', 'Israel': 'Moderna, Pfizer/BioNTech', 'I
taly': 'Moderna, Oxford/AstraZeneca, Pfizer/BioNTech', 'Japan': 'Pfizer/BioNTech', 'Jersey': 'Oxford/AstraZeneca, Pfizer/BioNTech', 'Kuwait': 'Pfi
zer/BioNTech', 'Latvia': 'Moderna, Oxford/AstraZeneca, Pfizer/BioNTech', 'Lebanon': 'Pfizer/BioNTech', 'Liechtenstein': 'Moderna, Pfizer/BioNTec
h', 'Lithuania': 'Moderna, Oxford/AstraZeneca, Pfizer/BioNTech', 'Luxembourg': 'Moderna, Oxford/AstraZeneca, Pfizer/BioNTech', 'Macao': 'Sinophar
m/Beijing', 'Maldives': 'Oxford/AstraZeneca', 'Malta': 'Pfizer/BioNTech', 'Mauritius': 'Oxford/AstraZeneca', 'Mexico': 'Oxford/AstraZeneca, Pfize
r/BioNTech', 'Monaco': 'Pfizer/BioNTech', 'Morocco': 'Oxford/AstraZeneca, Sinopharm/Beijing', 'Myanmar': 'Oxford/AstraZeneca', 'Nepal': 'Oxford/As
traZeneca', 'Netherlands': 'Moderna, Oxford/AstraZeneca, Pfizer/BioNTech', 'New Zealand': 'Pfizer/BioNTech', 'Northern Cyprus': 'Pfizer/BioNTech,
Sinovac', 'Northern Ireland': 'Oxford/AstraZeneca, Pfizer/BioNTech', 'Norway': 'Moderna, Oxford/AstraZeneca, Pfizer/BioNTech', 'Oman': 'Oxford/Ast
raZeneca, Pfizer/BioNTech', 'Pakistan': 'Oxford/AstraZeneca, Sinopharm/Beijing, Sputnik V', 'Panama': 'Pfizer/BioNTech', 'Peru': 'Sinopharm/Beijin
g', 'Poland': 'Moderna, Oxford/AstraZeneca, Pfizer/BioNTech', 'Portugal': 'Moderna, Pfizer/BioNTech', 'Qatar': 'Pfizer/BioNTech', 'Romania': 'Mode
rna, Oxford/AstraZeneca, Pfizer/BioNTech', 'Russia': 'Sputnik V', 'Saint Helena': 'Oxford/AstraZeneca', 'Saudi Arabia': 'Pfizer/BioNTech', 'Scotla
nd': 'Oxford/AstraZeneca, Pfizer/BioNTech', 'Serbia': 'Pfizer/BioNTech, Sinopharm/Beijing, Sputnik V', 'Seychelles': 'Oxford/AstraZeneca, Sinophar
m/Beijing', 'Singapore': 'Pfizer/BioNTech', 'Slovakia': 'Pfizer/BioNTech', 'Slovenia': 'Oxford/AstraZeneca, Pfizer/BioNTech', 'South Africa': 'Joh
nson&Johnson', 'Spain': 'Moderna, Oxford/AstraZeneca, Pfizer/BioNTech', 'Sri Lanka': 'Oxford/AstraZeneca', 'Sweden': 'Oxford/AstraZeneca, Pfizer/B
ioNTech', 'Switzerland': 'Moderna, Pfizer/BioNTech', 'Turkey': 'Sinovac', 'Turks and Caicos Islands': 'Pfizer/BioNTech', 'United Arab Emirates':
'Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing, Sinopharm/Wuhan, Sputnik V', 'United Kingdom': 'Oxford/AstraZeneca, Pfizer/BioNTech', 'Un
ited States': 'Moderna, Pfizer/BioNTech', 'Wales': 'Oxford/AstraZeneca, Pfizer/BioNTech'}
102
```

In [9]:

```
#A logic to het sum of any column with respect to the country
a=df[df['country']=='Albania']['people_vaccinated'].sum()
print(a)
```

```
8428.0
```

In [10]:

```
#By above logic checking for countries which are unvaccinated
# unvaccinated countries

#Sir Logic
people_vaccinated = []

for i in country_list:
    people_vaccinated.append(df[df['country']==i]['people_vaccinated'].sum(0))

index_country = []

countries_unvaccinated = []

for i in range(1,102):
    if(people_vaccinated[i]==0):
        index_country.append(i)

for i in index_country:
    countries_unvaccinated.append(country_list[i])

print("countries unvaccinated # ", countries_unvaccinated)


#My attempt
see=[]
for i in country_list:
    see.append(df[df['country']==i]['people_vaccinated'].sum())
#print(see)
ans=[]
for i in range(len(see)):
    if(see[i]==0):
        ans.append(country_list[i])
print(ans)
```

```
countries unvaccinated #  ['Algeria', 'Bolivia', 'China', 'Dominican Republic', 'Egypt', 'Lebanon', 'Liechtenstein', 'Macao', 'Maldives', 'Monac
o', 'Myanmar', 'Nepal', 'Netherlands', 'Northern Cyprus', 'Panama', 'Saint Helena', 'Saudi Arabia', 'Sri Lanka']
['Algeria', 'Bolivia', 'China', 'Dominican Republic', 'Egypt', 'Lebanon', 'Liechtenstein', 'Macao', 'Maldives', 'Monaco', 'Myanmar', 'Nepal', 'Net
herlands', 'Northern Cyprus', 'Panama', 'Saint Helena', 'Saudi Arabia', 'Sri Lanka']
```

In [ ]:

In [ ]:

In [11]:

```
a=df[df['country']=='United States']['vaccines'].count()
a
```

Out[11]:

```
65
```

In [12]:

```
df.dtypes
```

Out[12]:

```
Unnamed: 0                               int64
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 [13]:

```
#Now converting type of date from object to datetime
import datetime
df['date']=pd.to_datetime(df['date'])
```

In [14]:

```
df.dtypes
```

Out[14]:

```
Unnamed: 0                                      int64
country                                        object
iso_code                                       object
date                                   datetime64[ns]
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 [15]:

```
df.head()
```

Out[15]:

	Unnamed: 0	country	iso_code	date	total_vaccinations	people_vaccinated	people_fully_vaccinated	daily_vaccinations_raw	daily_vaccinations	total_vaccinations_per_hundred	people
0	0	Albania	ALB	2021-01-10	0.0	0.0	0.0	0.0	0.0	0.00	
1	1	Albania	ALB	2021-01-11	0.0	0.0	0.0	0.0	64.0	0.00	
2	2	Albania	ALB	2021-01-12	128.0	128.0	0.0	0.0	64.0	0.00	
3	3	Albania	ALB	2021-01-13	188.0	188.0	0.0	60.0	63.0	0.01	
4	4	Albania	ALB	2021-01-14	266.0	266.0	0.0	78.0	66.0	0.01	

In [16]:

```
#Task

#For storing all data Where Country is 'India'
first_vaccines=df[df['country']=='India']
#For storing all data Where Country is 'United States'
second_vaccines=df[df['country']=='United States']
total_India=sum(first_vaccines['people_vaccinated'])
total_US=sum(second_vaccines['people_vaccinated'])
print("In India People Vaccinated are:",total_India)
print("In United States People Vaccinated are:",total_US)
print("Rate of United States with Respect to India is:")
ans=((total_US-total_India)/total_US)*100
print(ans)
```

```
In India People Vaccinated are: 181607289.0
In United States People Vaccinated are: 1064579940.0
Rate of United States with Respect to India is:
82.94094391821811
```

In [17]:

```
#After seeing the csv file I found that the date of India And US are different
#So to get proper Rate of vaccination of US and INDIA it is importtant to see for the dates which are similar
#I found that the date from "2021-01-15" to "2021-02-22" are present in both
#So I will be using the people vaccinated between these dates to get the more exact result
```

In [18]:

```
second_vaccines.head()
```

Out[18]:

	Unnamed: 0	country	iso_code	date	total_vaccinations	people_vaccinated	people_fully_vaccinated	daily_vaccinations_raw	daily_vaccinations	total_vaccinations_per_hundred	pe
3810	3810	United States	USA	2020-12-20	556208.0	556208.0	0.0	0.0	0.0	0.17	
3811	3811	United States	USA	2020-12-21	614117.0	614117.0	0.0	57909.0	57909.0	0.18	
3812	3812	United States	USA	2020-12-22	0.0	0.0	0.0	0.0	127432.0	0.00	
3813	3813	United States	USA	2020-12-23	1008025.0	1008025.0	0.0	0.0	150606.0	0.30	
3814	3814	United States	USA	2020-12-24	0.0	0.0	0.0	0.0	191001.0	0.00	

In [19]:

```
start="2021-01-15"
end="2021-02-22"
UnitedStates=second_vaccines[(second_vaccines['date']>=start) & (second_vaccines['date']<=end)]
```

In [20]:

```
UnitedStates.head()
```

Out[20]:

	Unnamed: 0	country	iso_code	date	total_vaccinations	people_vaccinated	people_fully_vaccinated	daily_vaccinations_raw	daily_vaccinations	total_vaccinations_per_hundred	pe
3836	3836	United States	USA	2021-01-15	12279180.0	10595866.0	1610524.0	1130189.0	798707.0	3.67	
3837	3837	United States	USA	2021-01-16	0.0	0.0	0.0	0.0	811670.0	0.00	
3838	3838	United States	USA	2021-01-17	0.0	0.0	0.0	0.0	824632.0	0.00	
3839	3839	United States	USA	2021-01-18	0.0	0.0	0.0	0.0	837595.0	0.00	
3840	3840	United States	USA	2021-01-19	15707588.0	13595803.0	2023124.0	0.0	911493.0	4.70	

In [21]:

```
total_vaccines_US=UnitedStates['people_vaccinated'].sum()
print(total_vaccines_US)
```

```
995990126.0
```

In [22]:

```
first_vaccines.head()
```

Out[22]:

	Unnamed: 0	country	iso_code	date	total_vaccinations	people_vaccinated	people_fully_vaccinated	daily_vaccinations_raw	daily_vaccinations	total_vaccinations_per_hundred	pe
1714	1714	India	IND	2021-01-15	0.0	0.0	0.0	0.0	0.0	0.00	
1715	1715	India	IND	2021-01-16	191181.0	191181.0	0.0	191181.0	191181.0	0.01	
1716	1716	India	IND	2021-01-17	224301.0	224301.0	0.0	33120.0	112150.0	0.02	
1717	1717	India	IND	2021-01-18	454049.0	454049.0	0.0	229748.0	151350.0	0.03	
1718	1718	India	IND	2021-01-19	674835.0	674835.0	0.0	220786.0	168709.0	0.05	

In [23]:

```
total_vaccines_India=first_vaccines['people_vaccinated'].sum()
total_vaccines_India
```

Out[23]:

```
181607289.0
```

In [24]:

```
print("Rate of United States with Respect to India is:")
ans=((total_vaccines_US-total_vaccines_India)/total_vaccines_US)*100
print(ans)
```

```
Rate of United States with Respect to India is:
81.7661556817482
```

In [25]:

```
#We can also calculate the Rate of vaccinations for Month of January and February
df['Month']=df['date'].dt.month
```

In [26]:

```
df1=df.groupby(['country','Month']).people_vaccinated.sum()
```

In [27]:

```
people_vaccinated_India=df1['India']
people_vaccinated_US=df1['United States']
print("For India:")
print(people_vaccinated_India)
print("For United States:")
print(people_vaccinated_US)
```

```
For India:
Month
1     28323206.0
2    153284083.0
Name: people_vaccinated, dtype: float64
For United States:
Month
1     318860802.0
2     736674472.0
12      9044666.0
Name: people_vaccinated, dtype: float64
```

In [28]:

```
#For the Month Of January Rate of Vaccines In US with respect to India Was
ans=((people_vaccinated_US[1]-people_vaccinated_India[1])/people_vaccinated_US[1])*100
print(ans)
```

```
91.11737603921601
```

In [29]:

```
#For the Month Of February Rate of Vaccines In US with respect to India Was
ans=((people_vaccinated_US[2]-people_vaccinated_India[2])/people_vaccinated_US[2])*100
print(ans)
```

```
79.19242639373012
```

In [ ]: