FIT1043 Introduction to Data Science

Assignment 1

Kang Hong Bo 32684673

19th August 2021

Introduction

For this assignment we are going to read data from files and manipulate the data in Python. Beside manipulate the data we are also going to produce non-graphical and graphical visualisation of the data.

Importing libraries

The first step I will do is importing libraries before starting the assignment. Which will be **pandas** and **matplotlib.pyplot**, it is an open source data analysis tool in python. In pandas function it consists data structures which will be used in the assignment. For matpltlib.pyplot, we will need to creat a graph with it.

```
In [1]:
import pandas as pd
import matplotlib.pyplot as plt
```

Reading Data

Getting data from files (data), which contain Country-Vaccinations.csv,2020-Population.csv,2020-GDP.csv.

```
In [2]:
df=pd.read_csv('Country-Vaccinations.csv')
df.head()
```

Out[2]:

	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	

In [3]:

```
df2=pd.read_csv('2020-Population.csv')
df2
```

Out[3]:

	Unnamed: 0	Unnamed: 1	Unnamed: 2	Unnamed: 3	Unnamed: 4	Unnamed: 5	Unnamed: 6	Unnamed: 7
0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	United Nations	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4	Population Division	NaN	NaN	NaN	NaN	NaN	NaN	NaN
...	...	...	...	...	...	...	...	...
300	285	Estimates	Bermuda	14	60	Country/Area	918	37
301	286	Estimates	Canada	NaN	124	Country/Area	918	13 733
302	287	Estimates	Greenland	26	304	Country/Area	918	23
303	288	Estimates	Saint Pierre and Miquelon	2	666	Country/Area	918	5
304	289	Estimates	United States of America	35	840	Country/Area	918	158 804

305 rows × 78 columns

In [4]:

```
df3=pd.read_csv('2020-GDP.csv')
df3
```

Out[4]:

	Unnamed: 0	Gross domestic product 2020	Unnamed: 2	Unnamed: 3	Unnamed: 4	Unnamed: 5
0	NaN	NaN	NaN	NaN	NaN	NaN
1	NaN	NaN	NaN	NaN	(millions of	NaN
2	NaN	Ranking	NaN	Economy	US dollars)	NaN
3	NaN	NaN	NaN	NaN	NaN	NaN
4	USA	1	NaN	United States	20,936,600	NaN
...	...	...	...	...	...	...
324	NaN	NaN	NaN	NaN	NaN	NaN
325	NaN	NaN	NaN	NaN	NaN	NaN
326	NaN	NaN	NaN	NaN	NaN	NaN
327	NaN	NaN	NaN	NaN	NaN	NaN
328	NaN	NaN	NaN	NaN	NaN	NaN

329 rows × 6 columns

Wrangling Data

Creating subsets from read data and wrangle into 3 different table by using groupby function and aggregation function. Then, in this assignment we only used 6 countries which are Indonesia, Malaysia, Singapore, Thailand, Philippines, and Australia to form the dataframe.

In [5]:

```
vac={'people_fully_vaccinated':'max','daily_vaccinations':'sum','vaccines':'max'}
dfgroupbycountry=df.groupby(['country']).agg(vac)
dfgroupbycountry
```

Out[5]:

	people_fully_vaccinated	daily_vaccinations	vaccines
country			
Afghanistan	219159.0	1649463.0	Johnson&Johnson, Oxford/AstraZeneca, Pfizer/Bi...
Albania	553482.0	1237306.0	Oxford/AstraZeneca, Pfizer/BioNTech, Sinovac, ...
Algeria	724812.0	4075025.0	Oxford/AstraZeneca, Sinopharm/Beijing, Sinovac...
Andorra	33904.0	76689.0	Oxford/AstraZeneca, Pfizer/BioNTech
Angola	722610.0	1690469.0	Oxford/AstraZeneca
...	...	...	...
Wales	2115477.0	4396339.0	Moderna, Oxford/AstraZeneca, Pfizer/BioNTech
Wallis and Futuna	4659.0	8937.0	Oxford/AstraZeneca
Yemen	13322.0	302943.0	Oxford/AstraZeneca
Zambia	193603.0	475566.0	Oxford/AstraZeneca, Sinopharm/Beijing
Zimbabwe	1022618.0	2693298.0	Sinopharm/Beijing, Sinovac, Sputnik V

222 rows × 3 columns

In [6]:

```
countrylst=['Indonesia', 'Malaysia', 'Singapore', 'Thailand', 'Philippines', 'Austra
findf=dfgroupbycountry.loc[countrylst].reset_index()
findf.rename(columns={'daily_vaccinations':'total_vaccinations'},inplace=True)
findf
```

Out[6]:

	country	people_fully_vaccinated	total_vaccinations	vaccines
0	Indonesia	24481296.0	72386296.0	Moderna, Oxford/AstraZeneca, Sinopharm/Beijing...
1	Malaysia	9048634.0	23687251.0	Oxford/AstraZeneca, Pfizer/BioNTech, Sinovac
2	Singapore	3862510.0	7911869.0	Moderna, Pfizer/BioNTech
3	Thailand	4277071.0	18349011.0	Oxford/AstraZeneca, Sinovac

	country	people_fully_vaccinated	total_vaccinations	vaccines
4	Philippines	11614590.0	23230492.0	Johnson&Johnson, Moderna, Oxford/AstraZeneca, ...
5	Australia	4614203.0	13222783.0	Oxford/AstraZeneca, Pfizer/BioNTech

In [7]:

```
val={'Unnamed: 4':'sum'}
df3new=df3.groupby(['Unnamed: 3']).agg(val)
findf3=df3new.loc[countrylst].reset_index()
findf3
```

Out[7]:

	Unnamed: 3	Unnamed: 4
0	Indonesia	1,058,424
1	Malaysia	336,664
2	Singapore	339,998
3	Thailand	501,795
4	Philippines	361,489
5	Australia	1,330,901

In [8]:

```
findf3.rename(columns={'Unnamed: 3':'country','Unnamed: 4':'GDP_million in US dollar
findf3
```

Out[8]:

	country	GDP_million in US dollars
0	Indonesia	1,058,424
1	Malaysia	336,664
2	Singapore	339,998
3	Thailand	501,795
4	Philippines	361,489
5	Australia	1,330,901

In [9]:

```
pop={'Unnamed: 77':'sum'}
df2new=df2.groupby(['Unnamed: 2']).agg(pop)
findf2=df2new.loc[countrylst].reset_index()
findf2
```

Out[9]:

	Unnamed: 2	Unnamed: 77
0	Indonesia	273 524
1	Malaysia	32 366
2	Singapore	5 850
3	Thailand	69 800
4	Philippines	109 581
5	Australia	25 500

In [10]:
```
findf2 = findf2.rename(columns={'Unnamed: 2':'country','Unnamed: 77':'Population_Tho
findf2
```

Out[10]:

	country	Population_Thousand
0	Indonesia	273 524
1	Malaysia	32 366
2	Singapore	5 850
3	Thailand	69 800
4	Philippines	109 581
5	Australia	25 500

Merging data

Merge the data from the wrangled data

In [11]:
```
mergedf=pd.merge(findf,findf2,on='country')
mergedf=pd.merge(mergedf,findf3,on='country')
mergedf
```

Out[11]:

	country	people_fully_vaccinated	total_vaccinations	vaccines	Population_Thousand
0	Indonesia	24481296.0	72386296.0	Moderna, Oxford/AstraZeneca, Sinopharm/Beijing...	273 524
1	Malaysia	9048634.0	23687251.0	Oxford/AstraZeneca, Pfizer/BioNTech, Sinovac	32 366
2	Singapore	3862510.0	7911869.0	Moderna, Pfizer/BioNTech	5 850
3	Thailand	4277071.0	18349011.0	Oxford/AstraZeneca, Sinovac	69 800
4	Philippines	11614590.0	23230492.0	Johnson&Johnson, Moderna, Oxford/AstraZeneca, ...	109 581
5	Australia	4614203.0	13222783.0	Oxford/AstraZeneca, Pfizer/BioNTech	25 500

In [12]:
```
mergedf['GDP_million in US dollars']=mergedf['GDP_million in US dollars'].str.replac
mergedf['GDP_million in US dollars']=mergedf['GDP_million in US dollars']
```

In [13]:
```
mergedf['Population_Thousand']=mergedf['Population_Thousand'].str.replace(' ','').as
mergedf['Population_Thousand']=mergedf['Population_Thousand']
```

In [14]:
```
mergedf['perCapitaGDP']=mergedf['GDP_million in US dollars']*1000/mergedf['Populatio
# since the unit of GDP is million per unit and the population is thousand per unit
# By dividing million by thousand which will be thousand.
mergedf['perVarVac']=mergedf['Population_Thousand']*1000//[4,3,2,2,6,2]
mergedf['Population']=mergedf['Population_Thousand']*1000
mergedf
```

Out[14]:

	country	people_fully_vaccinated	total_vaccinations	vaccines	Population_Thousand
0	Indonesia	24481296.0	72386296.0	Moderna, Oxford/AstraZeneca, Sinopharm/Beijing...	273524
1	Malaysia	9048634.0	23687251.0	Oxford/AstraZeneca, Pfizer/BioNTech, Sinovac	32366
2	Singapore	3862510.0	7911869.0	Moderna, Pfizer/BioNTech	5850
3	Thailand	4277071.0	18349011.0	Oxford/AstraZeneca, Sinovac	69800
4	Philippines	11614590.0	23230492.0	Johnson&Johnson, Moderna, Oxford/AstraZeneca, ...	109581
5	Australia	4614203.0	13222783.0	Oxford/AstraZeneca, Pfizer/BioNTech	25500

From the table above (mergedf) we can conclude that Indonesia has the most people which was fully vaccinated but in the ratio of people_fully_vaccinated:Population(Thousand) Singapore has the most ratio which will be 66% people has fully vaccinated.

In the table every country have different kinds of vaccine but AstraZeneca is the most used vaccine among this country.

Singapore and Australia has higher perCapitaGDP which is over $50000 among all other countries in this table.

In [15]:

```
various=['Johnson&Johnson','Moderna','Oxford/AstraZeneca','Pfizer/BioNTech','Sinopha


plot1=pd.DataFrame({
            'Country':['Indonesia','Indonesia','Indonesia','Indonesia','Indonesia','
            'Vaccines':various*6,
            'Amount':[0,68381000,68381000,0,68381000,68381000,0,0,0,10788666,1078866
})

plot1
```

Out[15]:

	Country	Vaccines	Amount
0	Indonesia	Johnson&Johnson	0
1	Indonesia	Moderna	68381000
2	Indonesia	Oxford/AstraZeneca	68381000
3	Indonesia	Pfizer/BioNTech	0

	Country	Vaccines	Amount
4	Indonesia	Sinopharm/Beijing	68381000
5	Indonesia	Sinovac	68381000
6	Indonesia	Sputnik V	0
7	Malaysia	Johnson&Johnson	0
8	Malaysia	Moderna	0
9	Malaysia	Oxford/AstraZeneca	10788666
10	Malaysia	Pfizer/BioNTech	10788666
11	Malaysia	Sinopharm/Beijing	0
12	Malaysia	Sinovac	10788666
13	Malaysia	Sputnik V	0
14	Singapore	Johnson&Johnson	0
15	Singapore	Moderna	2925000
16	Singapore	Oxford/AstraZeneca	0
17	Singapore	Pfizer/BioNTech	2925000
18	Singapore	Sinopharm/Beijing	0
19	Singapore	Sinovac	0
20	Singapore	Sputnik V	0
21	Thailand	Johnson&Johnson	0
22	Thailand	Moderna	0
23	Thailand	Oxford/AstraZeneca	34900000
24	Thailand	Pfizer/BioNTech	0
25	Thailand	Sinopharm/Beijing	0
26	Thailand	Sinovac	34900000
27	Thailand	Sputnik V	0
28	Philippines	Johnson&Johnson	18263500
29	Philippines	Moderna	18263500
30	Philippines	Oxford/AstraZeneca	18263500
31	Philippines	Pfizer/BioNTech	18263500
32	Philippines	Sinopharm/Beijing	0
33	Philippines	Sinovac	18263500
34	Philippines	Sputnik V	18263500
35	Australia	Johnson&Johnson	0
36	Australia	Moderna	0
37	Australia	Oxford/AstraZeneca	12750000
38	Australia	Pfizer/BioNTech	12750000
39	Australia	Sinopharm/Beijing	0

	Country	Vaccines	Amount
40	Australia	Sinovac	0
41	Australia	Sputnik V	0

In [16]:

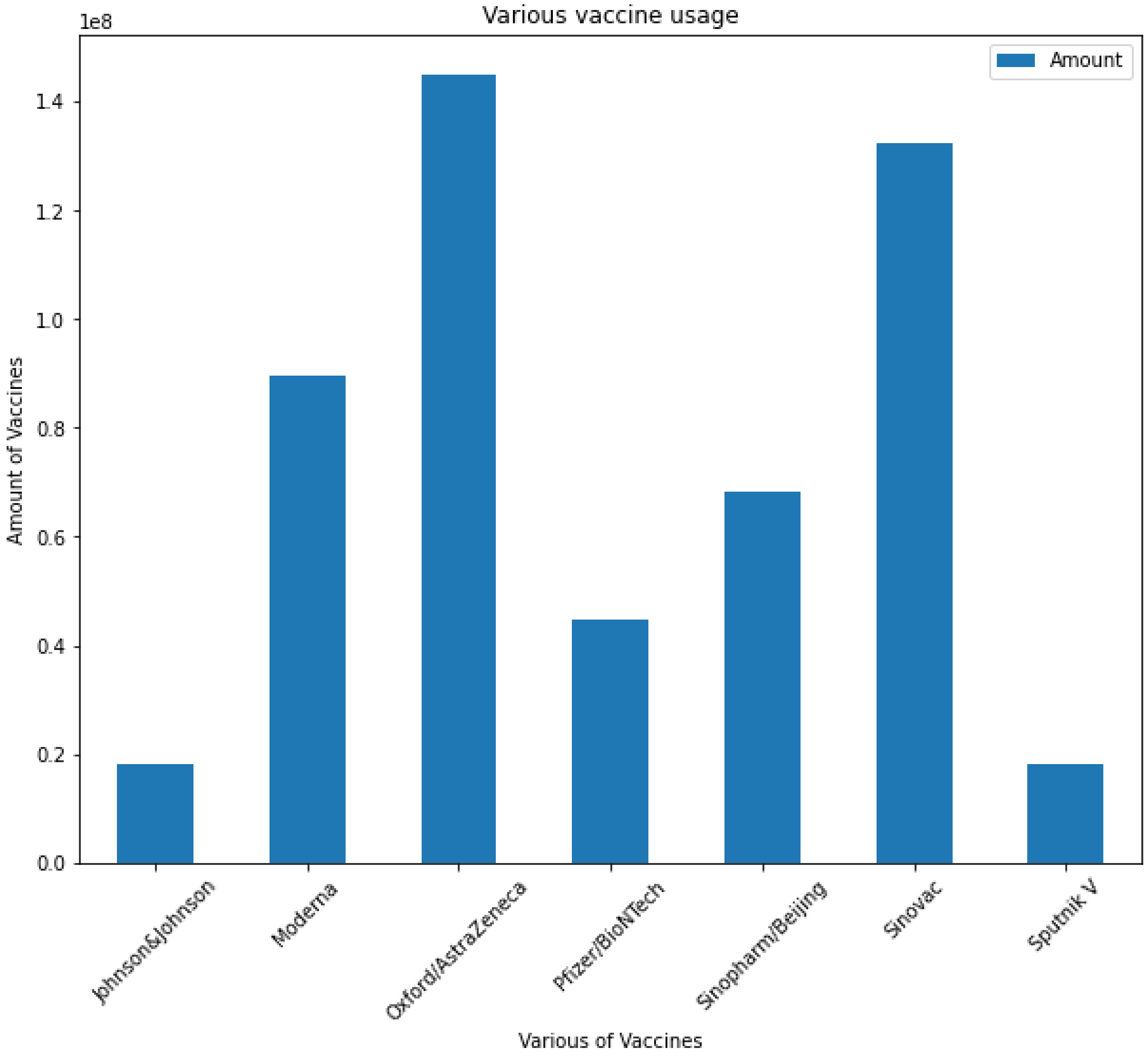
```
tot={'Amount':'sum'}
plot1_1=plot1.groupby('Vaccines').agg(tot).reset_index()
plot1_1
```

Out[16]:

	Vaccines	Amount
0	Johnson&Johnson	18263500
1	Moderna	89569500
2	Oxford/AstraZeneca	145083166
3	Pfizer/BioNTech	44727166
4	Sinopharm/Beijing	68381000
5	Sinovac	132333166
6	Sputnik V	18263500

In [17]:

```
ax=plot1_1.plot.bar(figsize=(10,8))
ax.set_xticklabels(plot1_1['Vaccines'],rotation=45)
plt.xlabel('Various of Vaccines')
plt.ylabel('Amount of Vaccines')
plt.title('Various vaccine usage')
plt.show()
```



By using bar graph we could easily know which data is the most and the least. From the graph above (Various vaccine usage) we can occur that AstraZeneca is the most frequently used vaccine among the others. Johnson&Johnson is the most least used vaccine among the others vacccines.

In [18]:

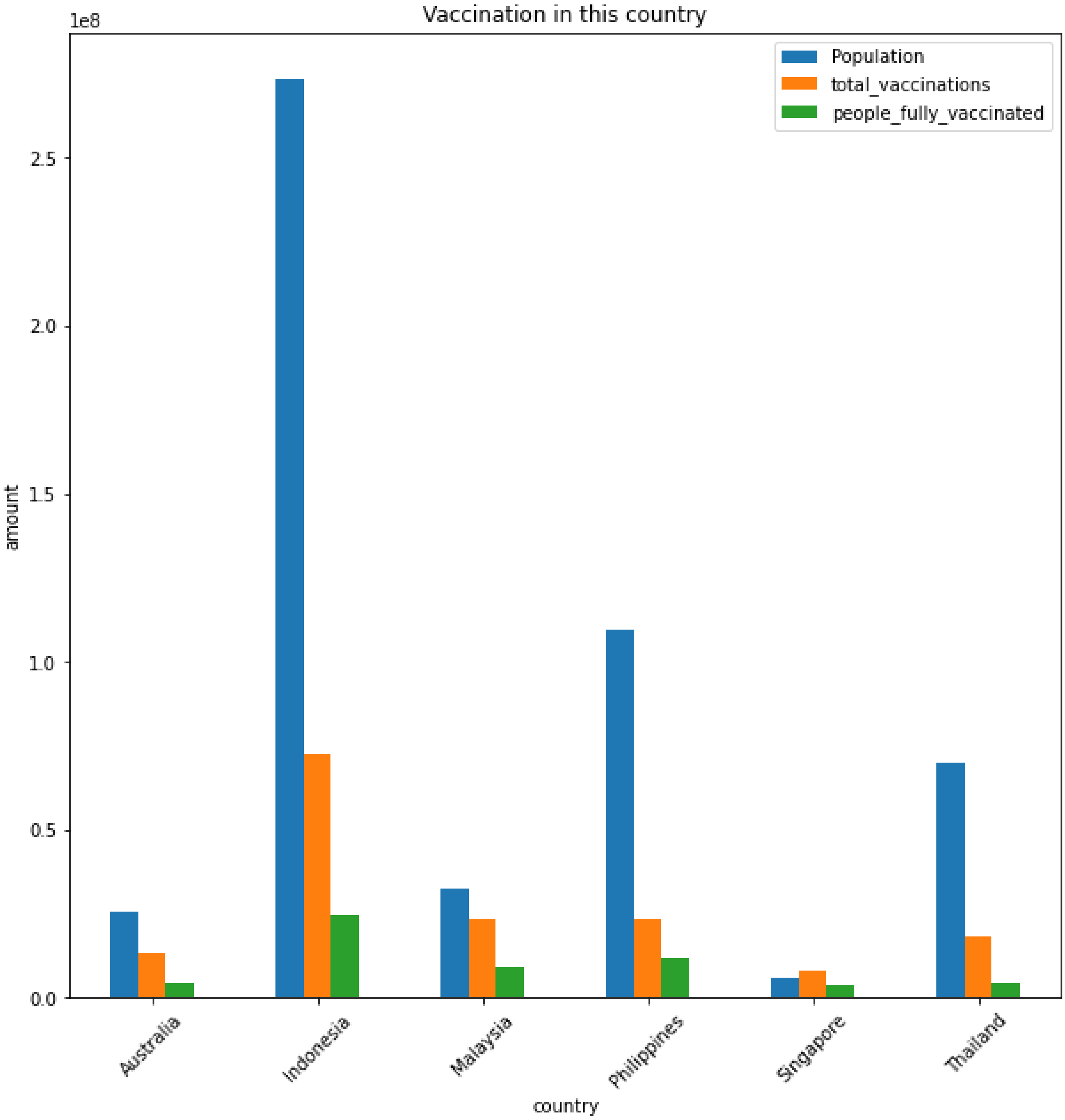
```
lst={'Population':'sum','total_vaccinations':'sum','people_fully_vaccinated':'sum'}
plot2=mergedf.groupby(['country']).agg(lst).reset_index()
plot2.head()
```

Out[18]:

	country	Population	total_vaccinations	people_fully_vaccinated
0	Australia	25500000	13222783.0	4614203.0
1	Indonesia	273524000	72386296.0	24481296.0
2	Malaysia	32366000	23687251.0	9048634.0
3	Philippines	109581000	23230492.0	11614590.0
4	Singapore	5850000	7911869.0	3862510.0

In [19]:

```
ax2=plot2.plot.bar(figsize=(10,10))
ax2.set_xticklabels(plot2['country'], rotation=45)
plt.xlabel('country')
plt.ylabel('amount')
plt.title('Vaccination in this country')
plt.show()
```



From the graph above we can find out that the bar is not obvious to observe data since it has a huge difference between the bar and for every country has different ratio of
Population : total_vaccinations : people_fully_vaccinated

In [20]:

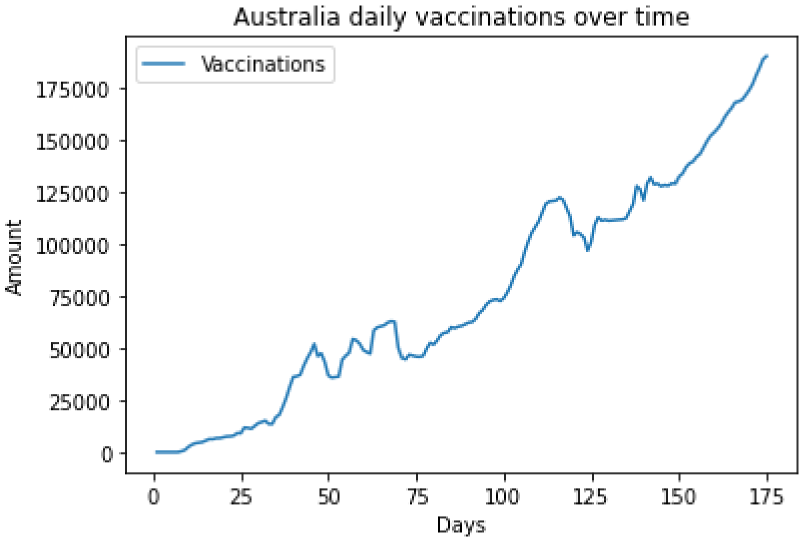
```
plot3=df[df['country']=='Australia'].reset_index()
plot3_3=plot3['daily_vaccinations']
plot3_3
```

Out[20]:

```
0           NaN
1           0.0
2           0.0
3           0.0
4           0.0
          ...   
171    176432.0
172    180617.0
173    184239.0
174    188408.0
175    189893.0
Name: daily_vaccinations, Length: 176, dtype: float64
```

In [21]:

```
line=plot3_3.plot.line()
plt.title('Australia daily vaccinations over time')
plt.xlabel('Days')
plt.ylabel('Amount')
plt.legend(['Vaccinations'])
plt.show()
```



In [22]:

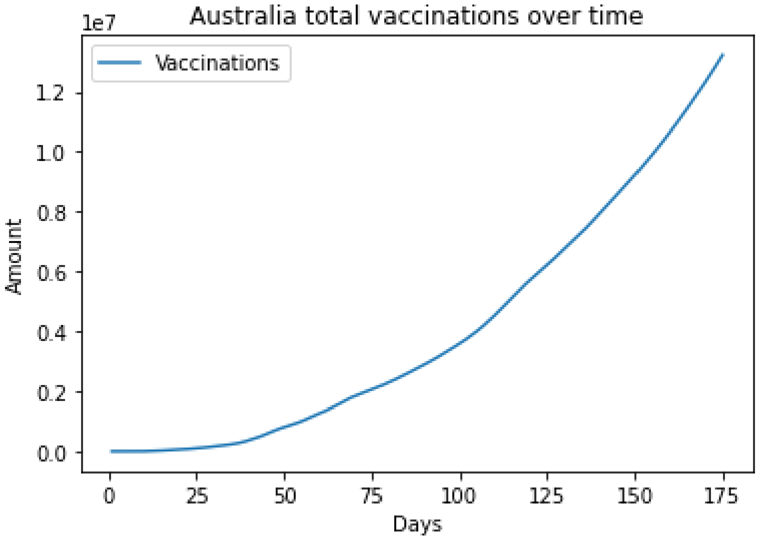
```
plot3_4=plot3['daily_vaccinations'].cumsum()
plot3_4
```

Out[22]:

```
0             NaN
1             0.0
2             0.0
3             0.0
4             0.0
          ...
171    12479626.0
172    12660243.0
173    12844482.0
174    13032890.0
175    13222783.0
Name: daily_vaccinations, Length: 176, dtype: float64
```

In [23]:

```
line=plot3_4.plot.line()
plt.title('Australia total vaccinations over time')
plt.xlabel('Days')
plt.ylabel('Amount')
plt.legend(['Vaccinations'])
plt.show()
```



As we can see the graph we use total vaccinations it will be a curve line because the data was sum together so that it will be more smooth compare to the graph with daily vaccinations. For the graph of daily vaccinations we can observe for every day and for the total vaccinations graph we can see the amount of vaccinated people over time which can track the total of vaccinated people and the speed of vaccinatting.