PHASE-3

PREPROCESSING

3.1 DATASET AND IT'S DETAIL EXPLAINATION IMPLEMENTATION

The Dataset is taken from the kaggle , where Kaggle is a data science competition platform and online community of data scientists and machine learning practitioners under Google LLC. Kaggle enables users to find and publish datasets, explore and build models in a web-based data science environment, work with other data scientists and machine learning engineers, and enter competitions to solve data science challenges.

My Dataset Link: https://www.kaggle.com/datasets/gpreda/covid-world-vaccination-progress

My Dataset about Covid-19 Vaccines Analysis. The Dataset contains many columns they are,

- ***Country-** this is the country for which the vaccination information is provided;*
- ***Country ISO Code -** ISO code for the country;*
- ***Date -** date for the data entry; for some of the dates we have only the daily vaccinations, for others, only the (cumulative) total;*
- ***Total number of vaccinations -** this is the absolute number of total immunizations in the country;*
- ***Total number of people vaccinated -** a person, depending on the immunization scheme, will receive one or more (typically 2) vaccines; at a certain moment, the number of vaccination might be larger than the number of people;*
- ***Total number of people fully vaccinated -** this is the number of people that received the entire set of immunization according to the immunization scheme (typically 2); at a certain moment in time, there might be a certain number of people that received one vaccine and another number (smaller) of people that received all vaccines in the scheme;*
- ***Daily vaccinations (raw) -** for a certain data entry, the number of vaccination for that date/country;*
- ***Daily vaccinations -** for a certain data entry, the number of vaccination for that date/country;*
- ***Total vaccinations per hundred -** ratio (in percent) between vaccination number and total population up to the date in the country;*
- ***Total number of people vaccinated per hundred -** ratio (in percent) between population immunized and total population up to the date in the country;*

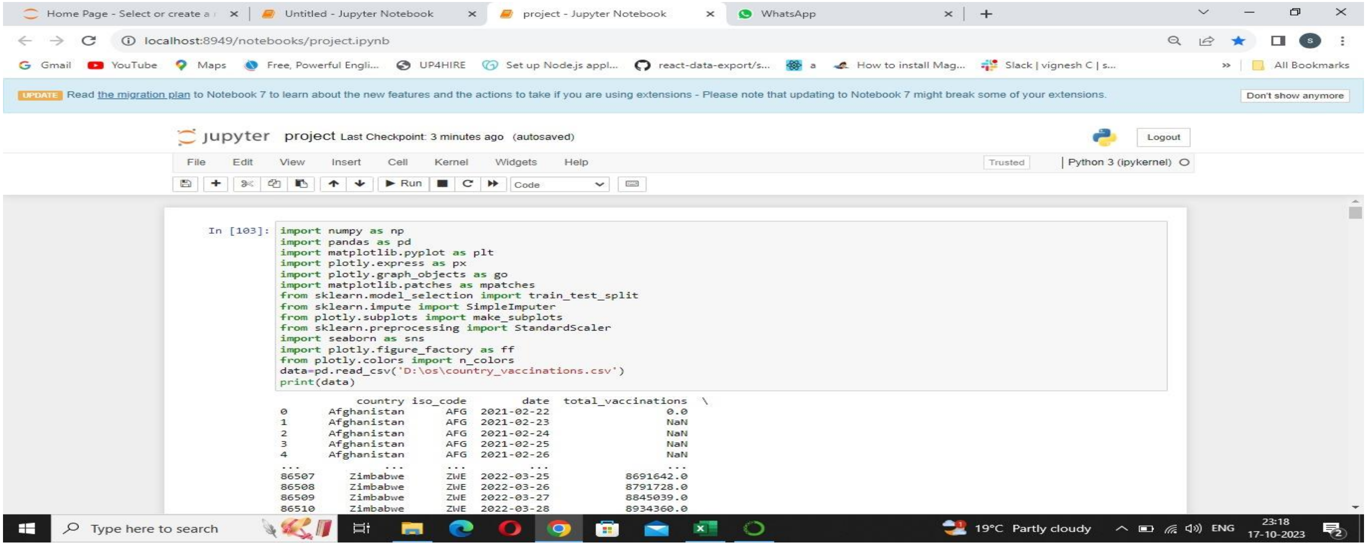
- ***Total number of people fully vaccinated per hundred** - ratio (in percent) between population fully immunized and total population up to the date in the country;*
- ***Number of vaccinations per day** - number of daily vaccination for that day and country;*
- ***Daily vaccinations per million** - ratio (in ppm) between vaccination number and total population for the current date in the country;*
- ***Vaccines used in the country** - total number of vaccines used in the country (up to date);*
- ***Source name** - source of the information (national authority, international organization, local organization etc.);*
- ***Source website** - website of the source of information;*

3.2 BEGIN BUILDING THE PROJECT BY LOAD THE DATASET

To import the required libraries and read a CSV file,

data=pd.read_csv('D:\os\country_vaccinations.csv')

Print(data)



```
In [103]: import numpy as np
          import pandas as pd
          import matplotlib.pyplot as plt
          import plotly.express as px
          import plotly.graph_objects as go
          import matplotlib.patches as mpatches
          from sklearn.model_selection import train_test_split
          from sklearn.impute import SimpleImputer
          from plotly.subplots import make_subplots
          from sklearn.preprocessing import StandardScaler
          import seaborn as sns
          import plotly.figure_factory as ff
          from plotly.colors import n_colors
          data=pd.read_csv('D:\os\country_vaccinations.csv')
          print(data)

                 country iso_code        date  total_vaccinations  \
          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
```

Home Page - Select or create a
Untitled - Jupyter Notebook
project - Jupyter Notebook
WhatsApp
localhost:8949/notebooks/project.ipynb
Gmail
YouTube
Maps
Free, Powerful Engli...
UP4HIRE
Set up Node.js appl...
react-data-export/s...
How to install Mag...
Slack | vignesh C | s...
All Bookmarks
UPDATE Read the migration plan to Notebook 7 to learn about the new features and the actions to take if you are using extensions - Please note that updating to Notebook 7 might break some of your extensions.
Don't show anymore
jupyter project Last Checkpoint: 3 minutes ago (autosaved)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3 (ipykernel)
Run
Code
Type here to search
19°C Partly cloudy
ENG
23:18
17-10-2023

3.3 PREPROCESS DATASET

Import The Required Libraries:

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import plotly.express as px

import plotly.graph_objects as go

import matplotlib.patches as mpatches

from sklearn.model_selection import train_test_split

from sklearn.impute import SimpleImputer

from plotly.subplots import make_subplots

from sklearn.preprocessing import StandardScaler

import seaborn as sns

import plotly.figure_factory as ff

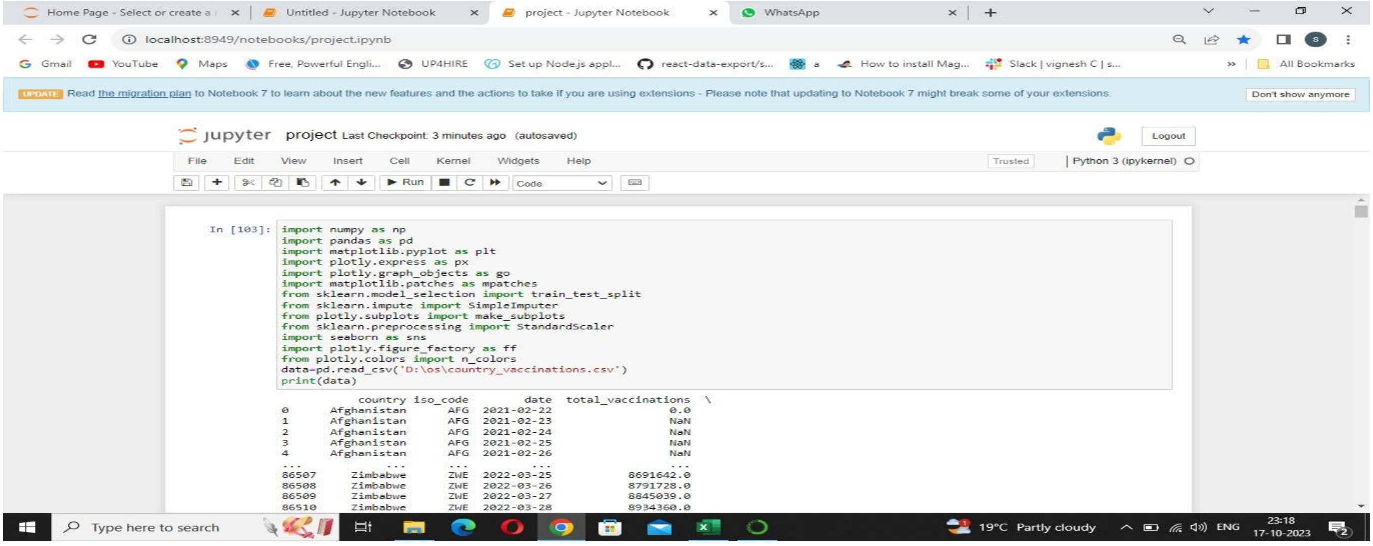
from plotly.colors import n_colors

Importing the Dataset:

- *Read Dataset,*

data=pd.read_csv('D:\os\country_vaccinations.csv')

print(data)



Home Page - Select or create a
Untitled - Jupyter Notebook
project - Jupyter Notebook
WhatsApp
localhost:8949/notebooks/project.ipynb
Gmail
YouTube
Maps
Free, Powerful Engli...
UP4HIRE
Set up Node.js appl...
react-data-export/s...
How to install Mag...
Slack | vignesh C | s...
All Bookmarks
UPDATE Read the migration plan to Notebook 7 to learn about the new features and the actions to take if you are using extensions - Please note that updating to Notebook 7 might break some of your extensions.
Don't show anymore
jupyter project Last Checkpoint: 3 minutes ago (autosaved)
Logout
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Trusted
Python 3 (ipykernel)
Run
Code
Type here to search
19°C Partly cloudy
ENG
23:18
17-10-2023

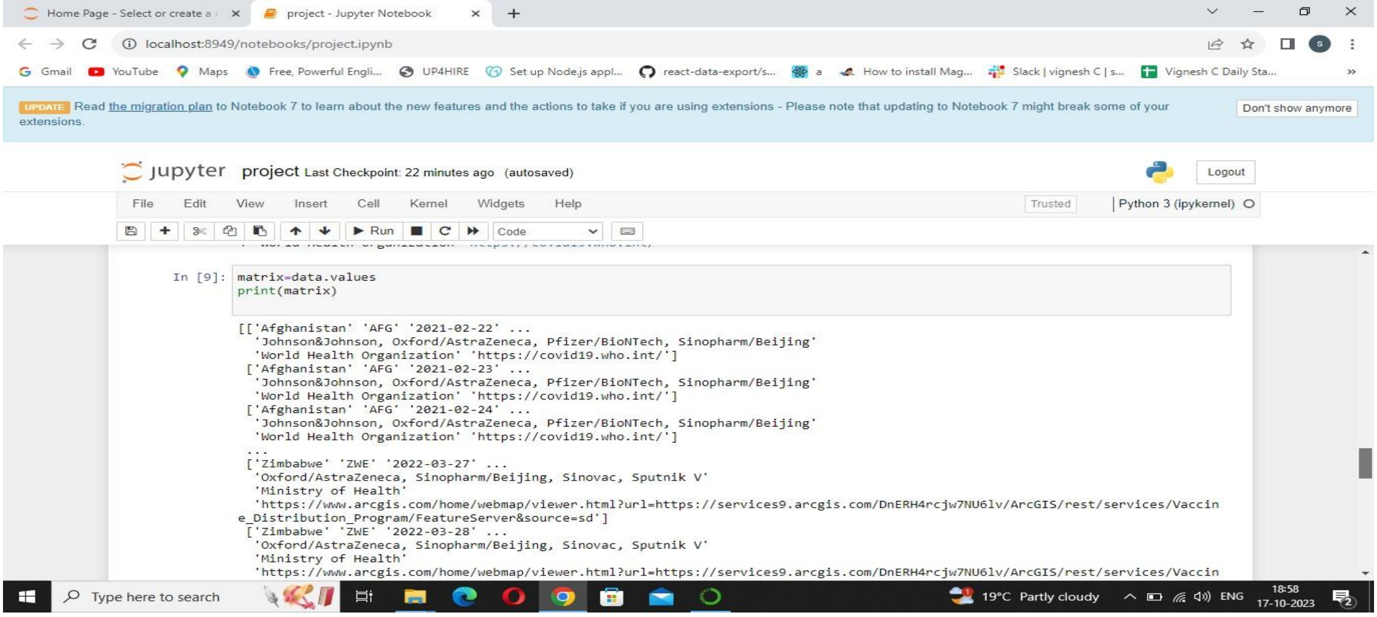
```
In [103]: import numpy as np
          import pandas as pd
          import matplotlib.pyplot as plt
          import plotly.express as px
          import plotly.graph_objects as go
          import matplotlib.patches as mpatches
          from sklearn.model_selection import train_test_split
          from sklearn.impute import SimpleImputer
          from plotly.subplots import make_subplots
          from sklearn.preprocessing import StandardScaler
          import seaborn as sns
          import plotly.figure_factory as ff
          from plotly.colors import n_colors
          data=pd.read_csv('D:\os\country_vaccinations.csv')
          print(data)
```

```
           country iso_code        date  total_vaccinations  \
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
```

- *Create Matrix,*

matrix=data.values

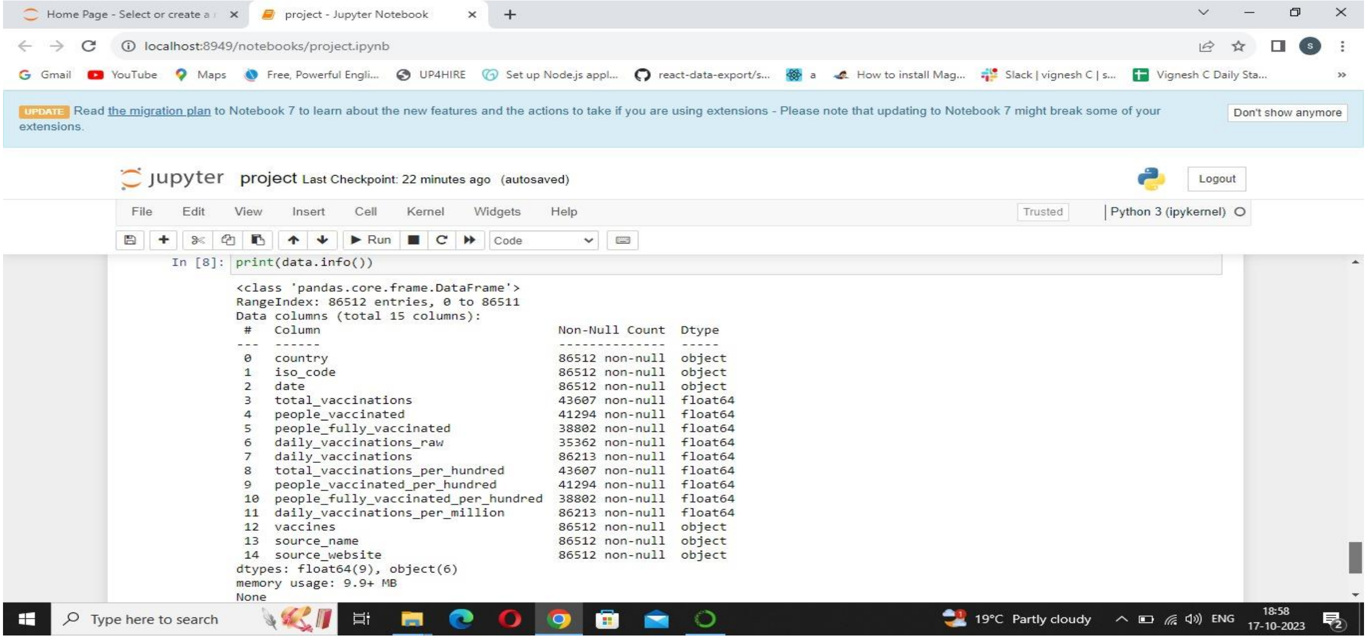
print(matrix)


Home Page - Select or create a
project - Jupyter Notebook
localhost:8949/notebooks/project.ipynb
UPDATE Read the migration plan to Notebook 7 to learn about the new features and the actions to take if you are using extensions - Please note that updating to Notebook 7 might break some of your extensions.
Don't show anymore
jupyter project Last Checkpoint: 22 minutes ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code
Type here to search
19°C Partly cloudy
ENG
18:58
17-10-2023

```
In [9]: matrix=data.values
        print(matrix)
```

```
[['Afghanistan' 'AFG' '2021-02-22' ...
  'Johnson&Johnson, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing'
  'World Health Organization' 'https://covid19.who.int/']
 ['Afghanistan' 'AFG' '2021-02-23' ...
  'Johnson&Johnson, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing'
  'World Health Organization' 'https://covid19.who.int/']
 ['Afghanistan' 'AFG' '2021-02-24' ...
  'Johnson&Johnson, Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm/Beijing'
  'World Health Organization' 'https://covid19.who.int/']
 ...
 ['Zimbabwe' 'ZWE' '2022-03-27' ...
  'Oxford/AstraZeneca, Sinopharm/Beijing, Sinovac, Sputnik V'
  'Ministry of Health'
  'https://www.arcgis.com/home/webmap/viewer.html?url=https://services9.arcgis.com/DnERH4rcjw7NU6lv/ArcGIS/rest/services/Vaccin
e_Distribution_Program/FeatureServer&source=sd']
 ['Zimbabwe' 'ZWE' '2022-03-28' ...
  'Oxford/AstraZeneca, Sinopharm/Beijing, Sinovac, Sputnik V'
  'Ministry of Health'
  'https://www.arcgis.com/home/webmap/viewer.html?url=https://services9.arcgis.com/DnERH4rcjw7NU6lv/ArcGIS/rest/services/Vaccin
```

- Other Imformation about dataset,

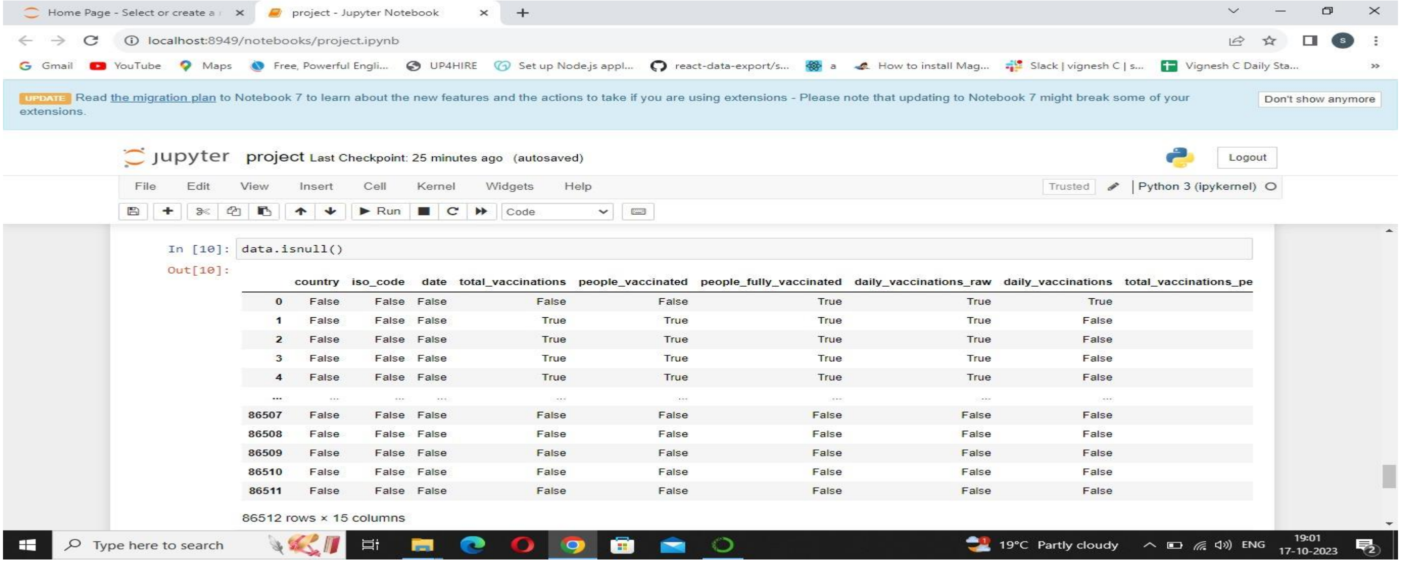

Home Page - Select or create a
project - Jupyter Notebook
localhost:8949/notebooks/project.ipynb
UPDATE Read the migration plan to Notebook 7 to learn about the new features and the actions to take if you are using extensions - Please note that updating to Notebook 7 might break some of your extensions.
Don't show anymore
jupyter project Last Checkpoint: 22 minutes ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code
Type here to search
19°C Partly cloudy
ENG
18:58
17-10-2023

```
In [8]: print(data.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 86512 entries, 0 to 86511
Data columns (total 15 columns):
 #   Column                               Non-Null Count  Dtype  
---  ------                               --------------  -----  
 0   country                              86512 non-null  object 
 1   iso_code                             86512 non-null  object 
 2   date                                 86512 non-null  object 
 3   total_vaccinations                   43607 non-null  float64
 4   people_vaccinated                    41294 non-null  float64
 5   people_fully_vaccinated              38802 non-null  float64
 6   daily_vaccinations_raw               35362 non-null  float64
 7   daily_vaccinations                   86213 non-null  float64
 8   total_vaccinations_per_hundred       43607 non-null  float64
 9   people_vaccinated_per_hundred        41294 non-null  float64
 10  people_fully_vaccinated_per_hundred  38802 non-null  float64
 11  daily_vaccinations_per_million       86213 non-null  float64
 12  vaccines                             86512 non-null  object 
 13  source_name                          86512 non-null  object 
 14  source_website                       86512 non-null  object 
dtypes: float64(9), object(6)
memory usage: 9.9+ MB
None
```

Handling The Missing Data:

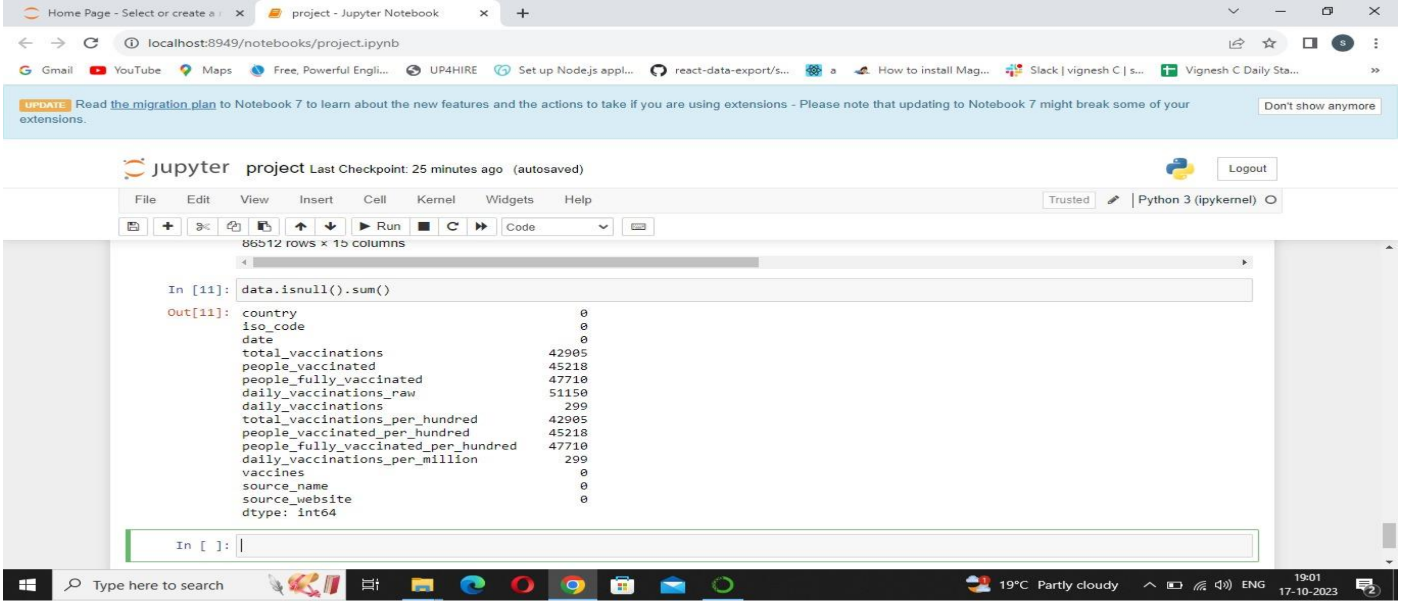
Before Handling the Missing data, we use isnull () to show the null values and using isnull().sum to get total number of null values.



```
In [10]: data.isnull()
Out[10]:
```

	country	iso_code	date	total_vaccinations	people_vaccinated	people_fully_vaccinated	daily_vaccinations_raw	daily_vaccinations	total_vaccinations_pe
0	False	False	False	False	False	True	True	True	
1	False	False	False	True	True	True	True	False	
2	False	False	False	True	True	True	True	False	
3	False	False	False	True	True	True	True	False	
4	False	False	False	True	True	True	True	False	
...	...	...	...	...	...	...	...	...	
86507	False	False	False	False	False	False	False	False	
86508	False	False	False	False	False	False	False	False	
86509	False	False	False	False	False	False	False	False	
86510	False	False	False	False	False	False	False	False	
86511	False	False	False	False	False	False	False	False	

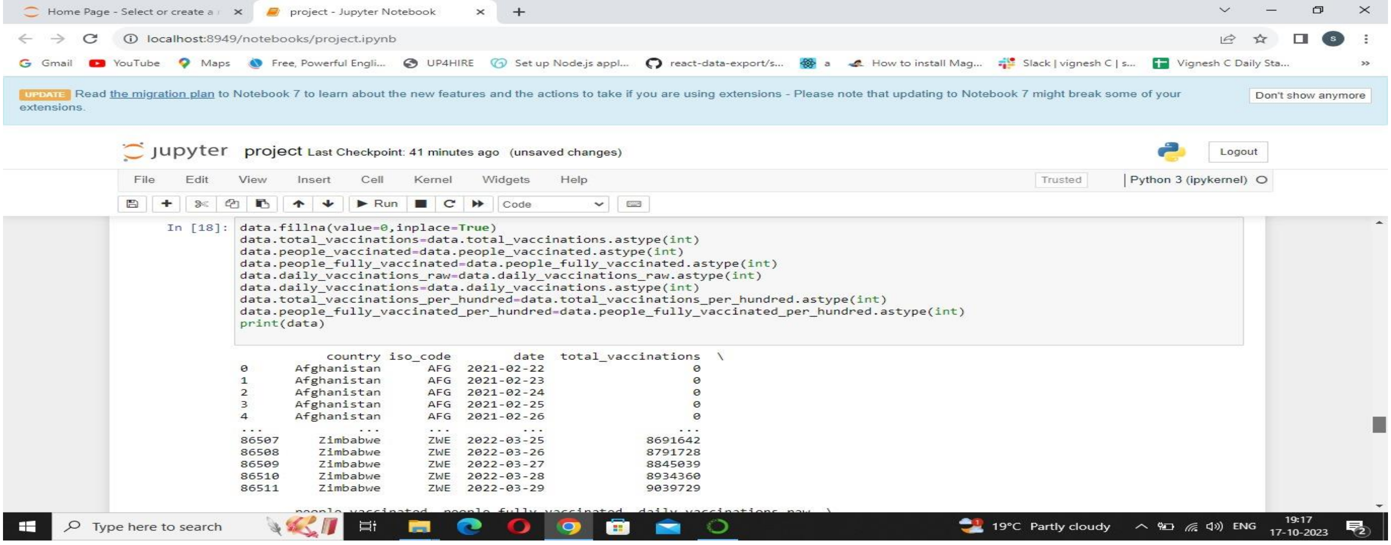
86512 rows × 15 columns



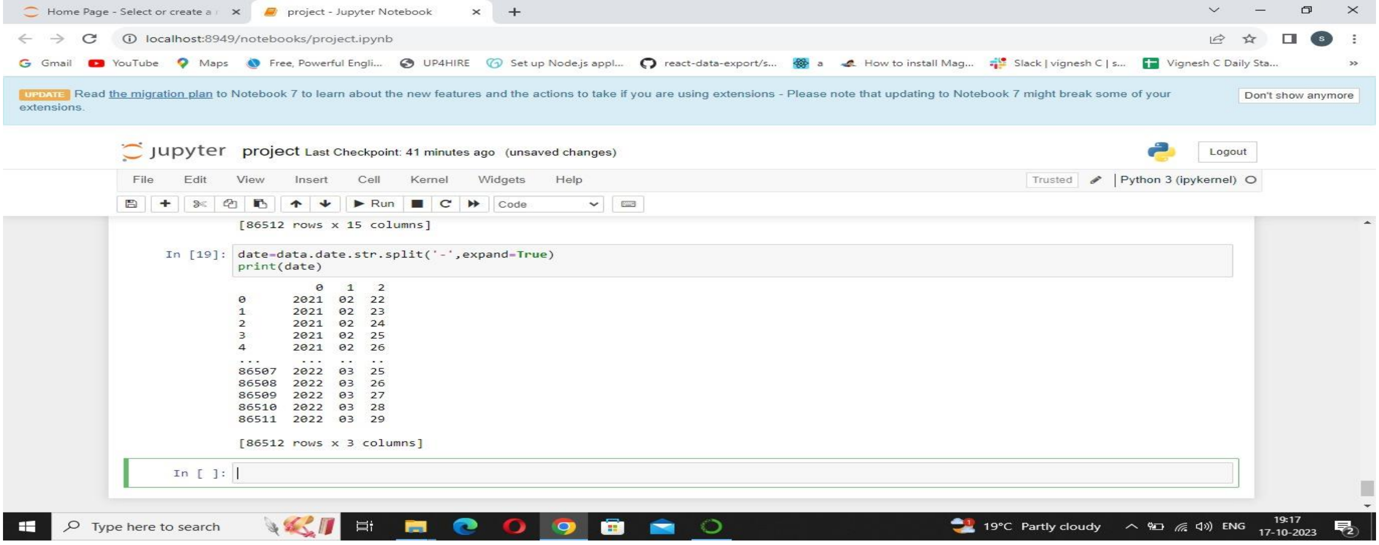
86512 rows × 15 columns

```
In [11]: data.isnull().sum()
Out[11]: 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 [ ]:
```

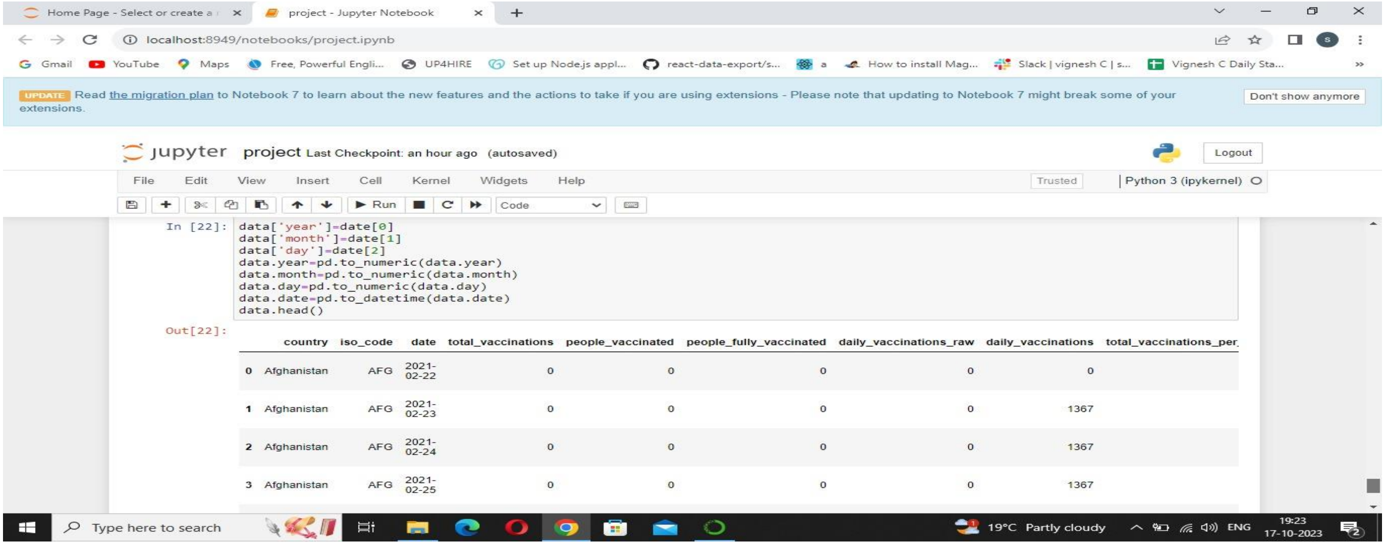
The below code for all the data cleaning,


project - Jupyter Notebook
localhost:8949/notebooks/project.ipynb
jupyter project Last Checkpoint: 41 minutes ago (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code

```
In [18]: data.fillna(value=0,inplace=True)
         data.total_vaccinations=data.total_vaccinations.astype(int)
         data.people_vaccinated=data.people_vaccinated.astype(int)
         data.people_fully_vaccinated=data.people_fully_vaccinated.astype(int)
         data.daily_vaccinations_raw=data.daily_vaccinations_raw.astype(int)
         data.daily_vaccinations=data.daily_vaccinations.astype(int)
         data.total_vaccinations_per_hundred=data.total_vaccinations_per_hundred.astype(int)
         data.people_fully_vaccinated_per_hundred=data.people_fully_vaccinated_per_hundred.astype(int)
         print(data)

              country iso_code        date  total_vaccinations  \
0      Afghanistan      AFG  2021-02-22                   0
1      Afghanistan      AFG  2021-02-23                   0
2      Afghanistan      AFG  2021-02-24                   0
3      Afghanistan      AFG  2021-02-25                   0
4      Afghanistan      AFG  2021-02-26                   0
...            ...      ...         ...                 ...
86507     Zimbabwe      ZWE  2022-03-25             8691642
86508     Zimbabwe      ZWE  2022-03-26             8791728
86509     Zimbabwe      ZWE  2022-03-27             8845039
86510     Zimbabwe      ZWE  2022-03-28             8934360
86511     Zimbabwe      ZWE  2022-03-29             9039729

       people_vaccinated  people_fully_vaccinated  daily_vaccinations_raw  \
```


project - Jupyter Notebook
localhost:8949/notebooks/project.ipynb
jupyter project Last Checkpoint: 41 minutes ago (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code

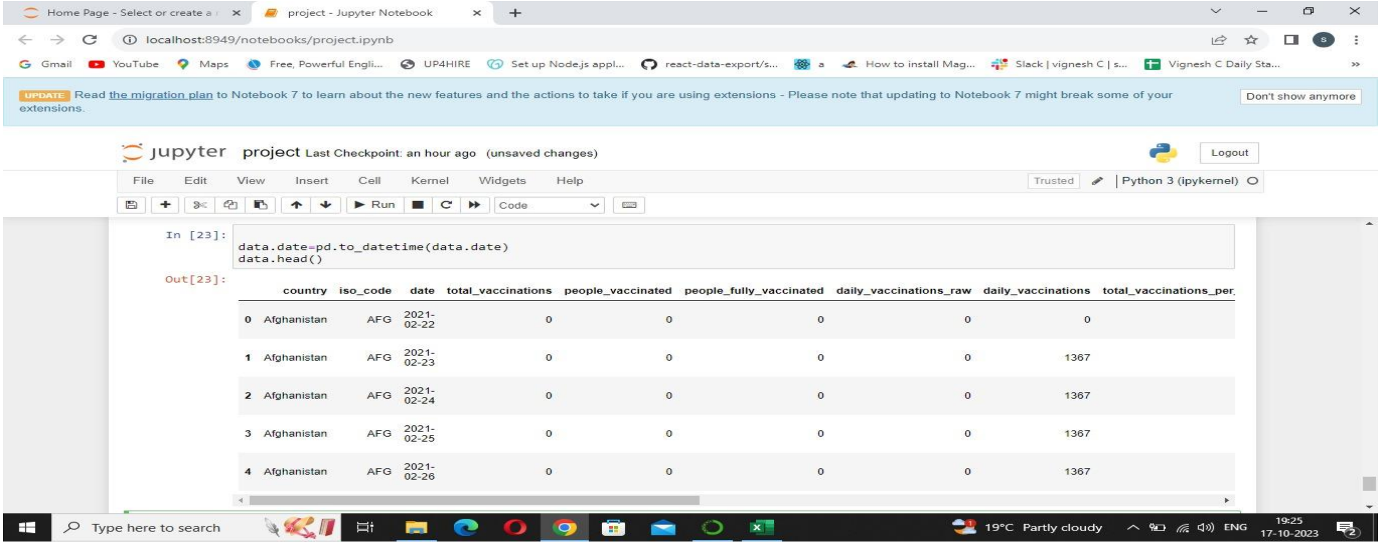
```
[86512 rows x 15 columns]

In [19]: date=data.date.str.split('-',expand=True)
         print(date)

          0   1   2
0      2021  02  22
1      2021  02  23
2      2021  02  24
3      2021  02  25
4      2021  02  26
...     ...  ..  ..
86507  2022  03  25
86508  2022  03  26
86509  2022  03  27
86510  2022  03  28
86511  2022  03  29

[86512 rows x 3 columns]

In [ ]: 
```


project - Jupyter Notebook
localhost:8949/notebooks/project.ipynb
jupyter project Last Checkpoint: an hour ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code

```
In [22]: data['year']=date[0]
         data['month']=date[1]
         data['day']=date[2]
         data.year=pd.to_numeric(data.year)
         data.month=pd.to_numeric(data.month)
         data.day=pd.to_numeric(data.day)
         data.date=pd.to_datetime(data.date)
         data.head()
```

Out[22]:

	country	iso_code	date	total_vaccinations	people_vaccinated	people_fully_vaccinated	daily_vaccinations_raw	daily_vaccinations	total_vaccinations_per_
0	Afghanistan	AFG	2021-02-22	0	0	0	0	0	
1	Afghanistan	AFG	2021-02-23	0	0	0	0	1367	
2	Afghanistan	AFG	2021-02-24	0	0	0	0	1367	
3	Afghanistan	AFG	2021-02-25	0	0	0	0	1367	

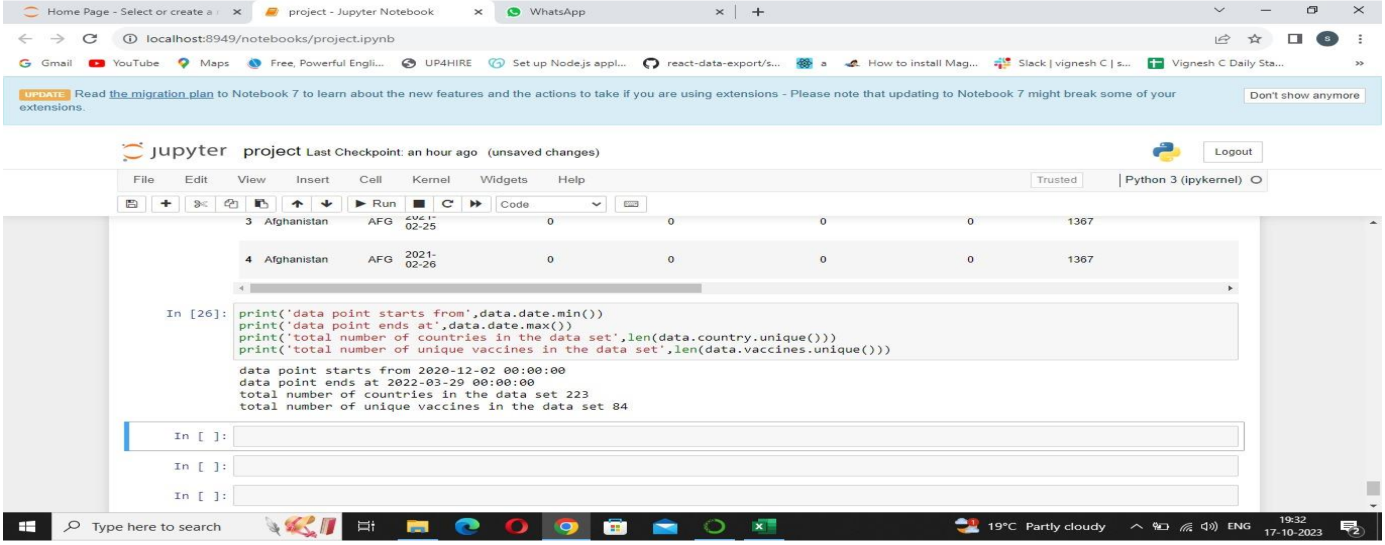

jupyter project Last Checkpoint: an hour ago (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code

In [23]:
```
data.date=pd.to_datetime(data.date)
data.head()
```

Out[23]:

	country	iso_code	date	total_vaccinations	people_vaccinated	people_fully_vaccinated	daily_vaccinations_raw	daily_vaccinations	total_vaccinations_per_
0	Afghanistan	AFG	2021-02-22	0	0	0	0	0	
1	Afghanistan	AFG	2021-02-23	0	0	0	0	1367	
2	Afghanistan	AFG	2021-02-24	0	0	0	0	1367	
3	Afghanistan	AFG	2021-02-25	0	0	0	0	1367	
4	Afghanistan	AFG	2021-02-26	0	0	0	0	1367	

Some detailed features to specify the details using the below code,


jupyter project Last Checkpoint: an hour ago (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code

3	Afghanistan	AFG	2021-02-25	0	0	0	0	1367
4	Afghanistan	AFG	2021-02-26	0	0	0	0	1367

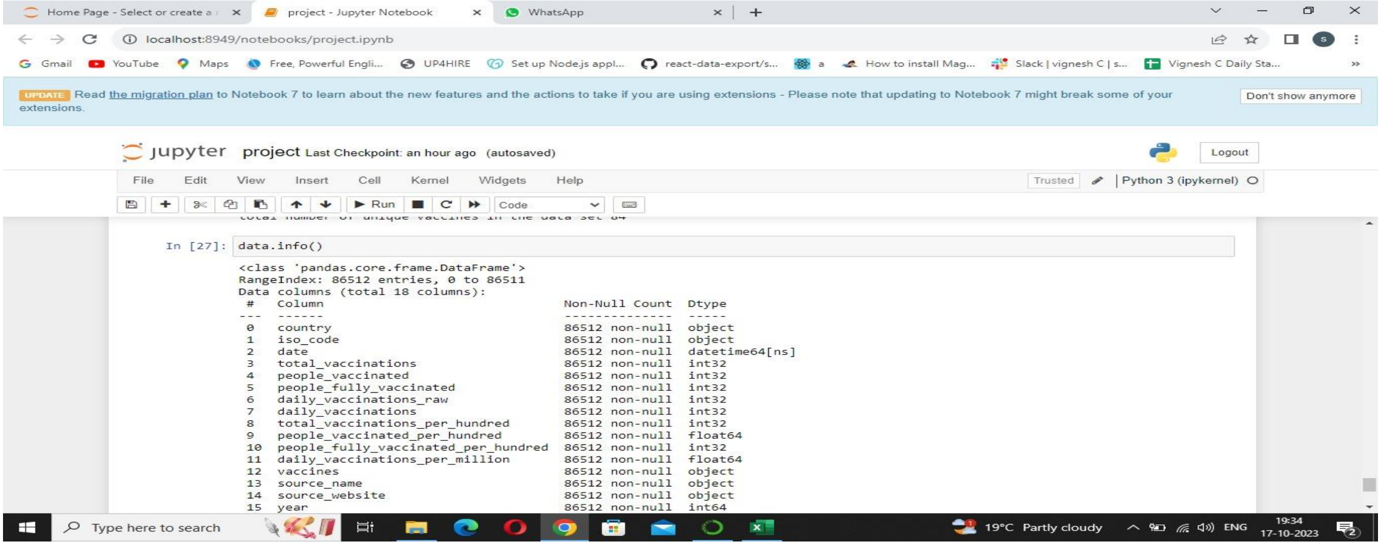
In [26]:
```
print('data point starts from',data.date.min())
print('data point ends at',data.date.max())
print('total number of countries in the data set',len(data.country.unique()))
print('total number of unique vaccines in the data set',len(data.vaccines.unique()))
```

```
data point starts from 2020-12-02 00:00:00
data point ends at 2022-03-29 00:00:00
total number of countries in the data set 223
total number of unique vaccines in the data set 84
```

In [ ]:

In [ ]:

In [ ]:

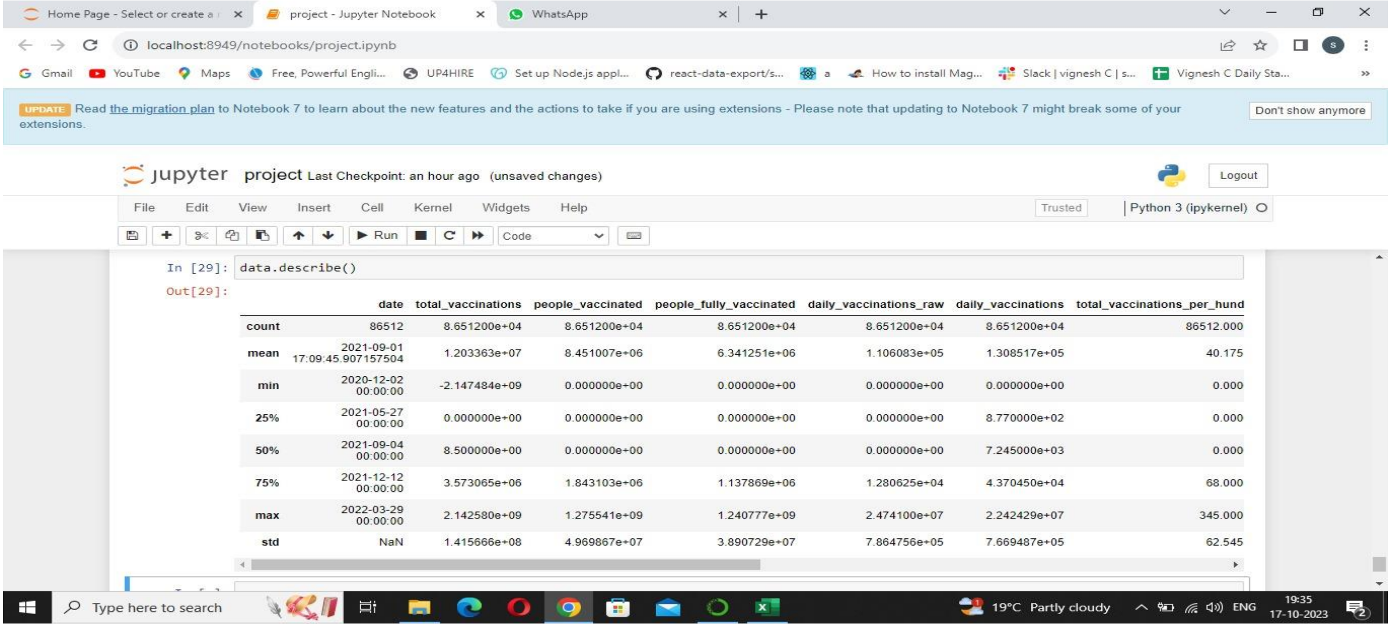

jupyter project Last Checkpoint: an hour ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code

In [27]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 86512 entries, 0 to 86511
Data columns (total 18 columns):
 #   Column                              Non-Null Count  Dtype
---  ------                              --------------  -----
 0   country                             86512 non-null  object
 1   iso_code                            86512 non-null  object
 2   date                                86512 non-null  datetime64[ns]
 3   total_vaccinations                  86512 non-null  int32
 4   people_vaccinated                   86512 non-null  int32
 5   people_fully_vaccinated             86512 non-null  int32
 6   daily_vaccinations_raw              86512 non-null  int32
 7   daily_vaccinations                  86512 non-null  int32
 8   total_vaccinations_per_hundred      86512 non-null  int32
 9   people_vaccinated_per_hundred       86512 non-null  float64
 10  people_fully_vaccinated_per_hundred 86512 non-null  int32
 11  daily_vaccinations_per_million      86512 non-null  float64
 12  vaccines                            86512 non-null  object
 13  source_name                         86512 non-null  object
 14  source_website                      86512 non-null  object
 15  year                                86512 non-null  int64
```

Using Data visualization we are going to draw some visuals to get insights from dataset,

Describe () function is used to get the statistics of each feature in dataset to get count, min, max, standard deviation, median, etc.,

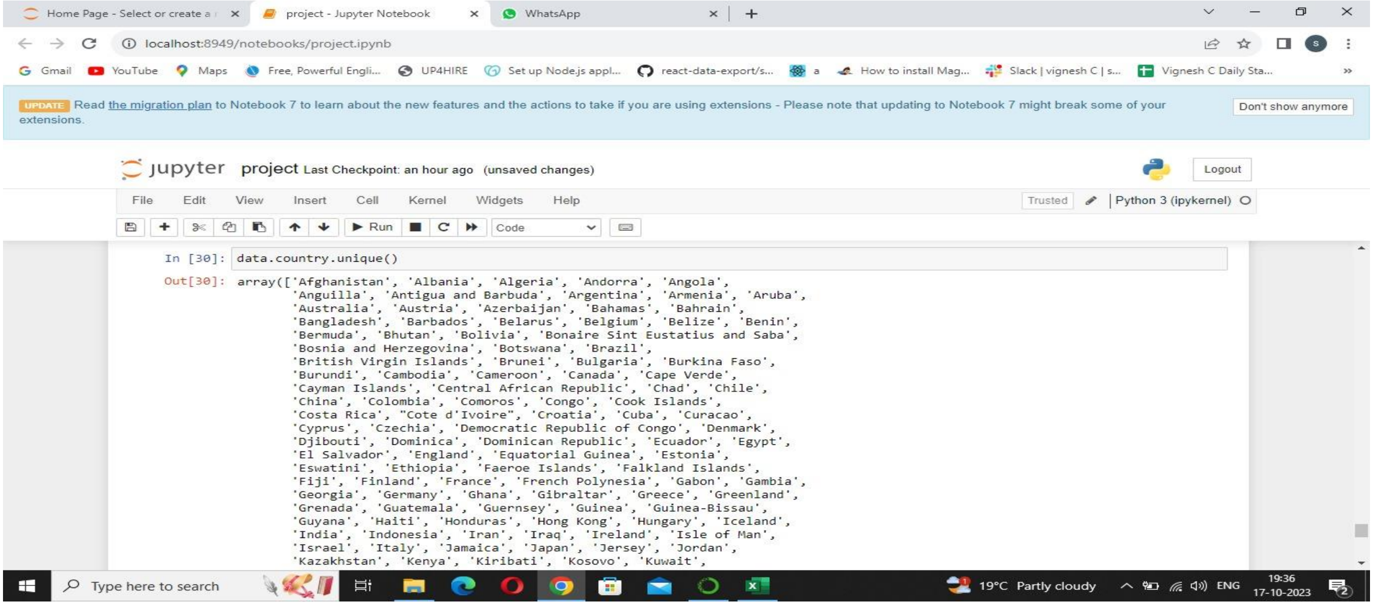


Home Page - Select or create a
project - Jupyter Notebook
WhatsApp
localhost:8949/notebooks/project.ipynb
UPDATE Read the migration plan to Notebook 7 to learn about the new features and the actions to take if you are using extensions - Please note that updating to Notebook 7 might break some of your extensions.
Don't show anymore
jupyter project Last Checkpoint: an hour ago (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code
Type here to search
19°C Partly cloudy
ENG
19:35
17-10-2023

```
In [29]: data.describe()
Out[29]:
```

	date	total_vaccinations	people_vaccinated	people_fully_vaccinated	daily_vaccinations_raw	daily_vaccinations	total_vaccinations_per_hund
count	86512	8.651200e+04	8.651200e+04	8.651200e+04	8.651200e+04	8.651200e+04	86512.000
mean	2021-09-01 17:09:45.907157504	1.203363e+07	8.451007e+06	6.341251e+06	1.106083e+05	1.308517e+05	40.175
min	2020-12-02 00:00:00	-2.147484e+09	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000
25%	2021-05-27 00:00:00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	8.770000e+02	0.000
50%	2021-09-04 00:00:00	8.500000e+00	0.000000e+00	0.000000e+00	0.000000e+00	7.245000e+03	0.000
75%	2021-12-12 00:00:00	3.573065e+06	1.843103e+06	1.137869e+06	1.280625e+04	4.370450e+04	68.000
max	2022-03-29 00:00:00	2.142580e+09	1.275541e+09	1.240777e+09	2.474100e+07	2.242429e+07	345.000
std	NaN	1.415666e+08	4.969867e+07	3.890729e+07	7.864756e+05	7.669487e+05	62.545

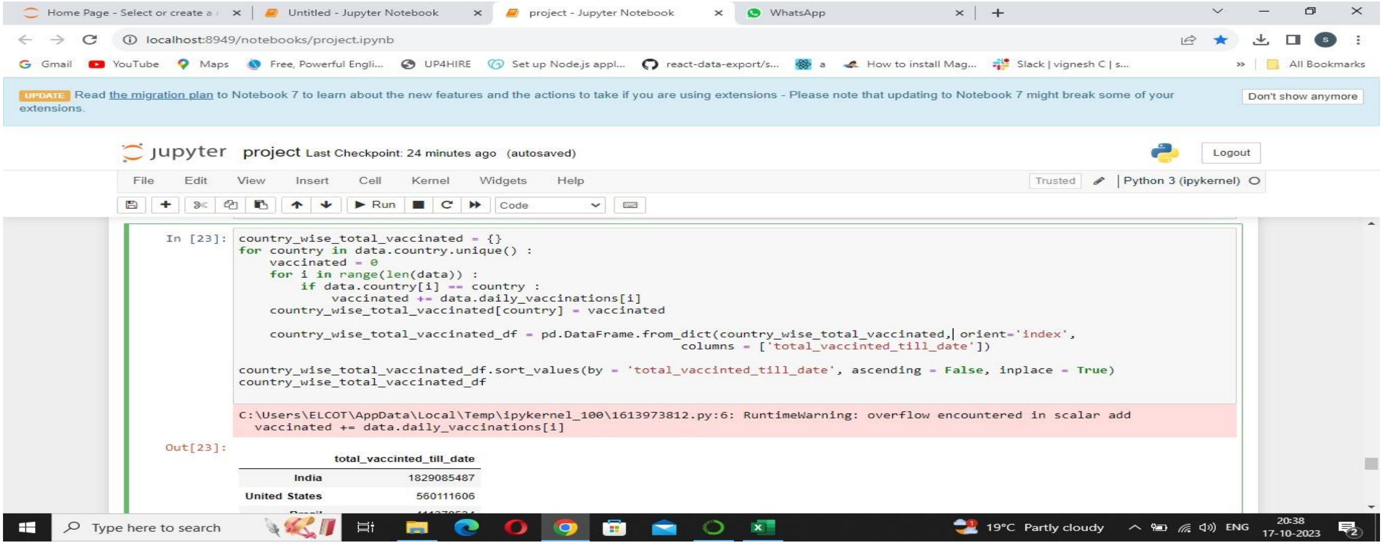
Unique () function helps to get unique values,



Home Page - Select or create a
project - Jupyter Notebook
WhatsApp
localhost:8949/notebooks/project.ipynb
UPDATE Read the migration plan to Notebook 7 to learn about the new features and the actions to take if you are using extensions - Please note that updating to Notebook 7 might break some of your extensions.
Don't show anymore
jupyter project Last Checkpoint: an hour ago (unsaved changes)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code
Type here to search
19°C Partly cloudy
ENG
19:36
17-10-2023

```
In [30]: data.country.unique()

Out[30]: array(['Afghanistan', 'Albania', 'Algeria', 'Andorra', 'Angola',
                'Anguilla', 'Antigua and Barbuda', 'Argentina', 'Armenia', 'Aruba',
                'Australia', 'Austria', 'Azerbaijan', 'Bahamas', 'Bahrain',
                'Bangladesh', 'Barbados', 'Belarus', 'Belgium', 'Belize', 'Benin',
                'Bermuda', 'Bhutan', 'Bolivia', 'Bonaire Sint Eustatius and Saba',
                'Bosnia and Herzegovina', 'Botswana', 'Brazil',
                'British Virgin Islands', 'Brunei', 'Bulgaria', 'Burkina Faso',
                'Burundi', 'Cambodia', 'Cameroon', 'Canada', 'Cape Verde',
                'Cayman Islands', 'Central African Republic', 'Chad', 'Chile',
                'China', 'Colombia', 'Comoros', 'Congo', 'Cook Islands',
                'Costa Rica', "Cote d'Ivoire", 'Croatia', 'Cuba', 'Curacao',
                'Cyprus', 'Czechia', 'Democratic Republic of Congo', 'Denmark',
                'Djibouti', 'Dominica', 'Dominican Republic', 'Ecuador', 'Egypt',
                'El Salvador', 'England', 'Equatorial Guinea', 'Estonia',
                'Eswatini', 'Ethiopia', 'Faeroe Islands', 'Falkland Islands',
                'Fiji', 'Finland', 'France', 'French Polynesia', 'Gabon', 'Gambia',
                'Georgia', 'Germany', 'Ghana', 'Gibraltar', 'Greece', 'Greenland',
                'Grenada', 'Guatemala', 'Guernsey', 'Guinea', 'Guinea-Bissau',
                'Guyana', 'Haiti', 'Honduras', 'Hong Kong', 'Hungary', 'Iceland',
                'India', 'Indonesia', 'Iran', 'Iraq', 'Ireland', 'Isle of Man',
                'Israel', 'Italy', 'Jamaica', 'Japan', 'Jersey', 'Jordan',
                'Kazakhstan', 'Kenya', 'Kiribati', 'Kosovo', 'Kuwait',
```

To see how many total vaccines have been used in each country using the code below,


localhost:8949/notebooks/project.ipynb
jupyter project Last Checkpoint: 24 minutes ago (autosaved)
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code
Type here to search
19°C Partly cloudy
ENG
20:38
17-10-2023

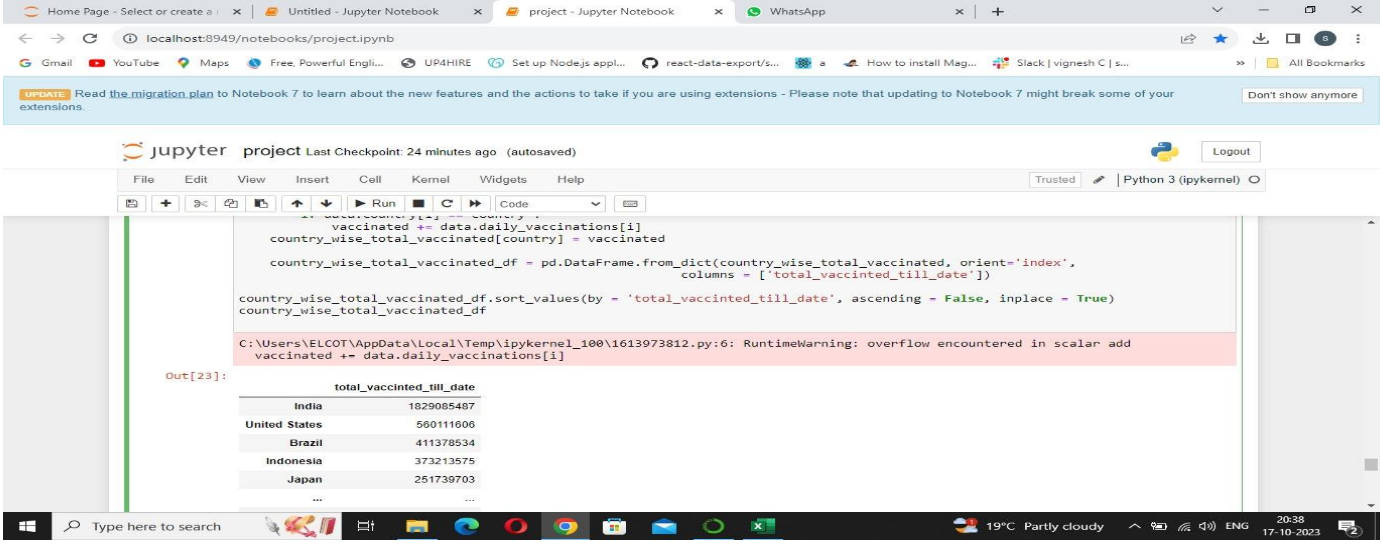
In [23]:

```
country_wise_total_vaccinated = {}
for country in data.country.unique() :
    vaccinated = 0
    for i in range(len(data)) :
        if data.country[i] == country :
            vaccinated += data.daily_vaccinations[i]
    country_wise_total_vaccinated[country] = vaccinated

    country_wise_total_vaccinated_df = pd.DataFrame.from_dict(country_wise_total_vaccinated, orient='index',
                                                              columns = ['total_vaccinted_till_date'])

country_wise_total_vaccinated_df.sort_values(by = 'total_vaccinted_till_date', ascending = False, inplace = True)
country_wise_total_vaccinated_df
```

```
C:\Users\ELCOT\AppData\Local\Temp\ipykernel_100\1613973812.py:6: RuntimeWarning: overflow encountered in scalar add
  vaccinated += data.daily_vaccinations[i]
```

Out[23]:

	total_vaccinted_till_date
India	1829085487
United States	560111606

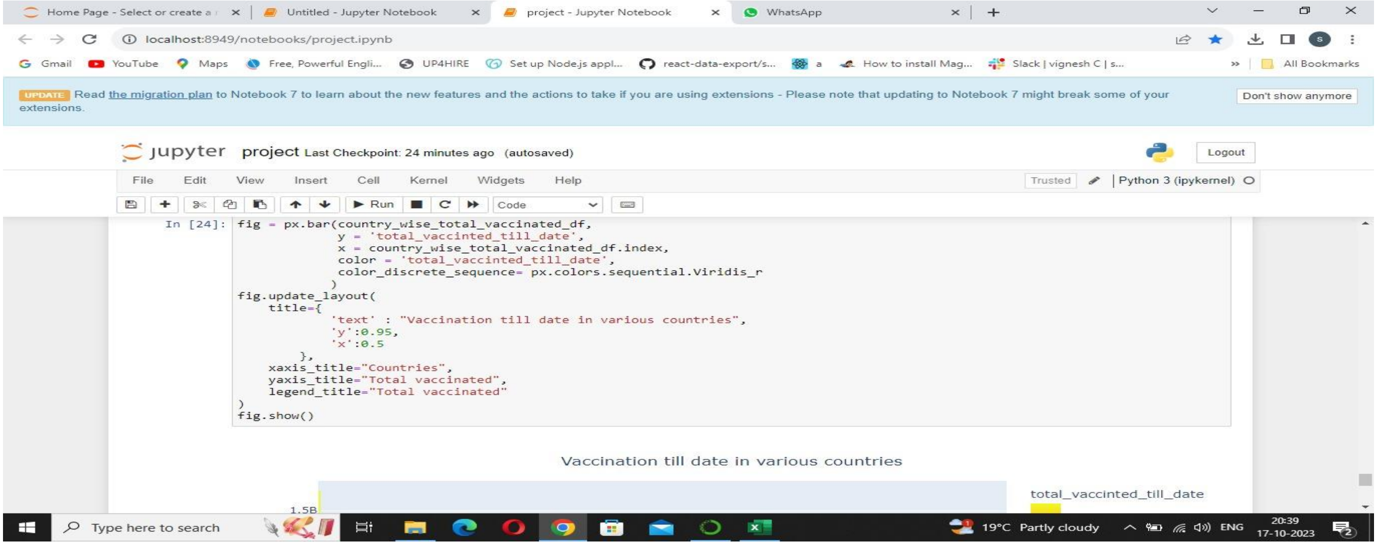

localhost:8949/notebooks/project.ipynb
jupyter project Last Checkpoint: 24 minutes ago (autosaved)
Type here to search
20:38
17-10-2023

```
            vaccinated += data.daily_vaccinations[i]
    country_wise_total_vaccinated[country] = vaccinated

    country_wise_total_vaccinated_df = pd.DataFrame.from_dict(country_wise_total_vaccinated, orient='index',
                                                              columns = ['total_vaccinted_till_date'])

country_wise_total_vaccinated_df.sort_values(by = 'total_vaccinted_till_date', ascending = False, inplace = True)
country_wise_total_vaccinated_df
```

```
C:\Users\ELCOT\AppData\Local\Temp\ipykernel_100\1613973812.py:6: RuntimeWarning: overflow encountered in scalar add
  vaccinated += data.daily_vaccinations[i]
```

Out[23]:

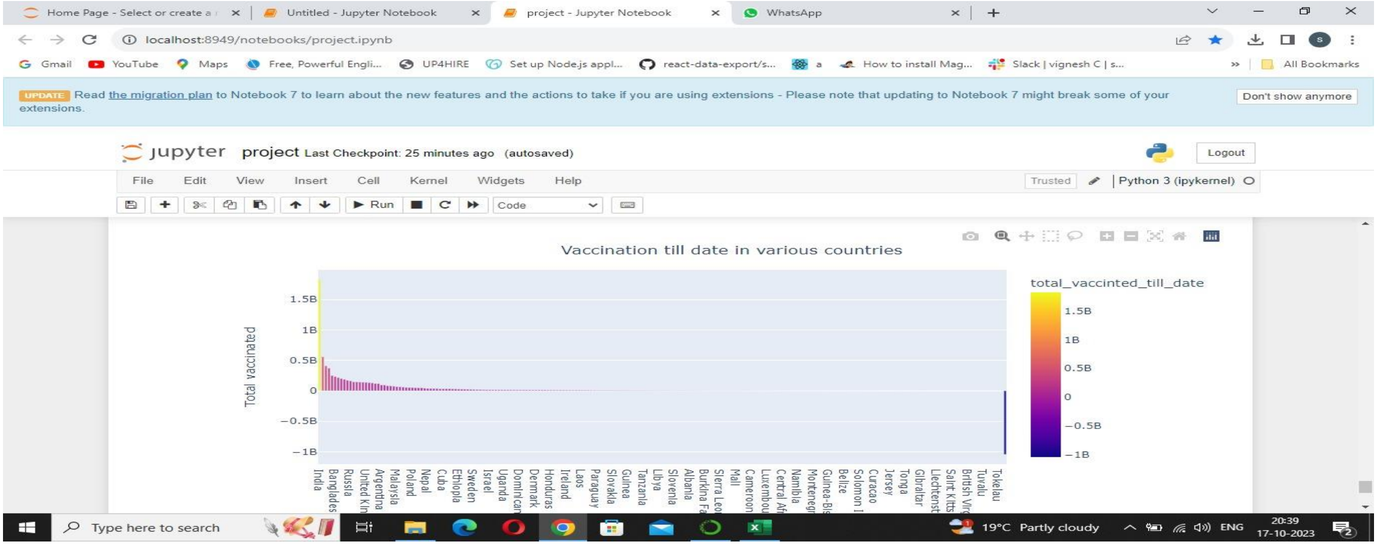
	total_vaccinted_till_date
India	1829085487
United States	560111606
Brazil	411378534
Indonesia	373213575
Japan	251739703
...	...


localhost:8949/notebooks/project.ipynb
jupyter project Last Checkpoint: 24 minutes ago (autosaved)
Type here to search
20:39
17-10-2023

In [24]:

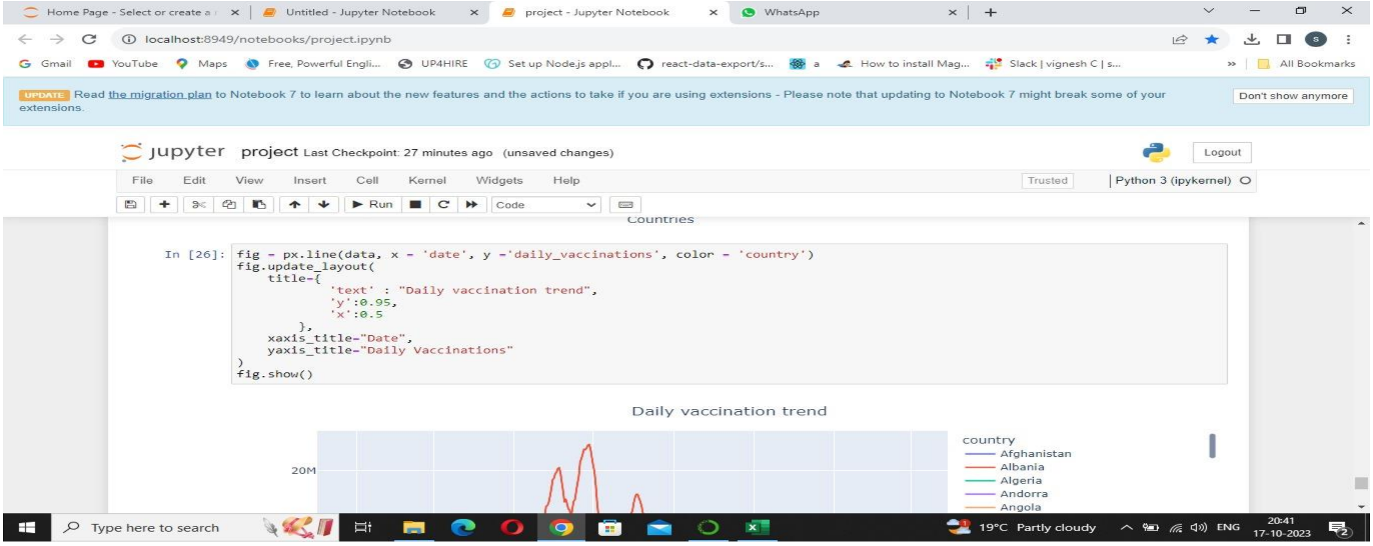
```
fig = px.bar(country_wise_total_vaccinated_df,
             y = 'total_vaccinted_till_date',
             x = country_wise_total_vaccinated_df.index,
             color = 'total_vaccinted_till_date',
             color_discrete_sequence= px.colors.sequential.Viridis_r
            )
fig.update_layout(
    title={
            'text' : "Vaccination till date in various countries",
            'y':0.95,
            'x':0.5
        },
    xaxis_title="Countries",
    yaxis_title="Total vaccinated",
    legend_title="Total vaccinated"
)
fig.show()
```

Vaccination till date in various countries



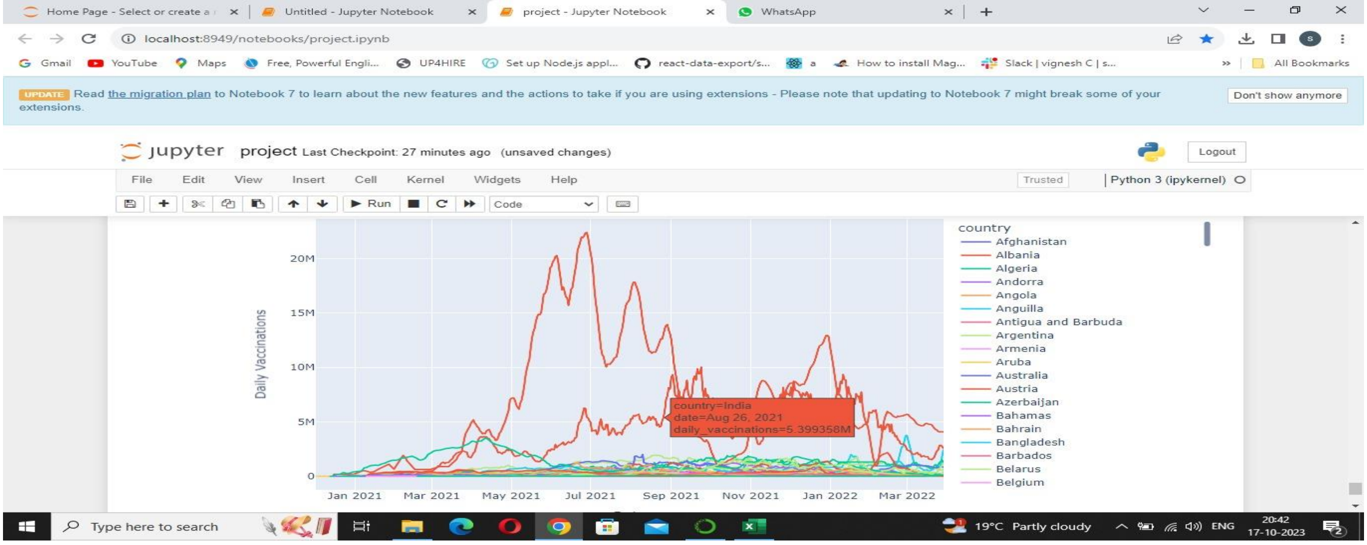
UPDATE Read the migration plan to Notebook 7 to learn about the new features and the actions to take if you are using extensions - Please note that updating to Notebook 7 might break some of your extensions.

To draw a line plot where x-axis is Date and the y-axis is daily_vaccination using the in the image,



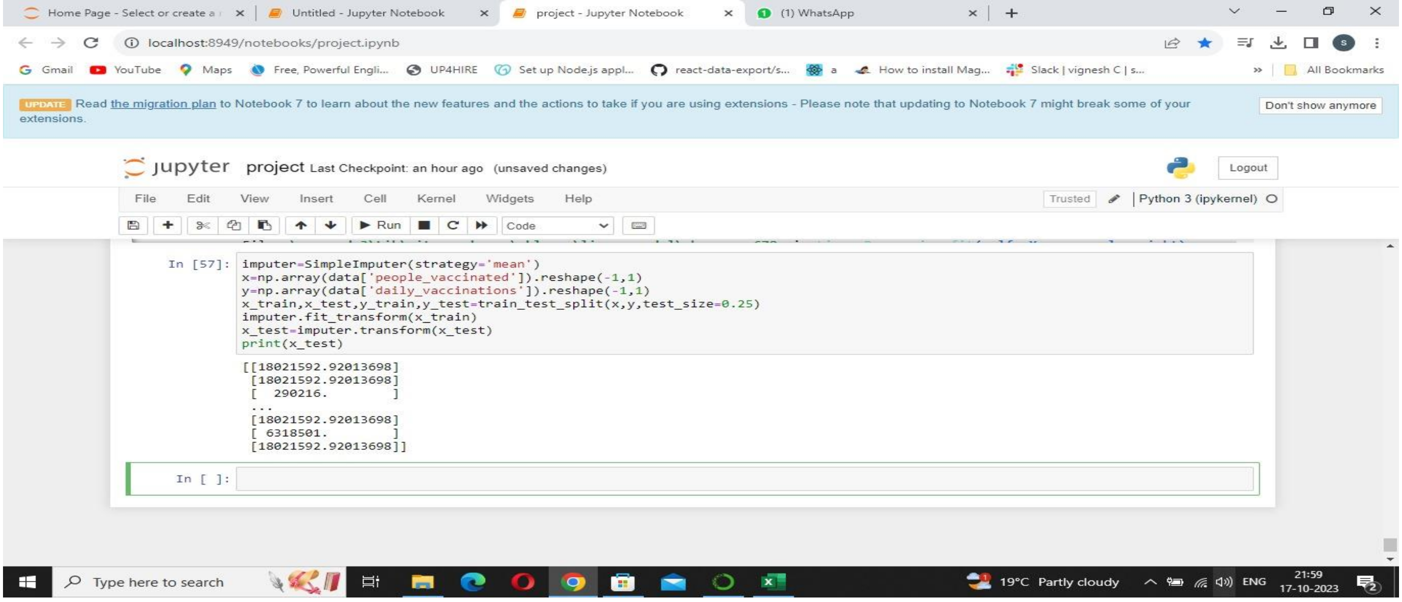
UPDATE Read the migration plan to Notebook 7 to learn about the new features and the actions to take if you are using extensions - Please note that updating to Notebook 7 might break some of your extensions.

```
In [26]: fig = px.line(data, x = 'date', y ='daily_vaccinations', color = 'country')
         fig.update_layout(
             title={
                  'text' : "Daily vaccination trend",
                  'y':0.95,
                  'x':0.5
             },
             xaxis_title="Date",
             yaxis_title="Daily Vaccinations"
         )
         fig.show()
```



UPDATE Read the migration plan to Notebook 7 to learn about the new features and the actions to take if you are using extensions - Please note that updating to Notebook 7 might break some of your extensions.

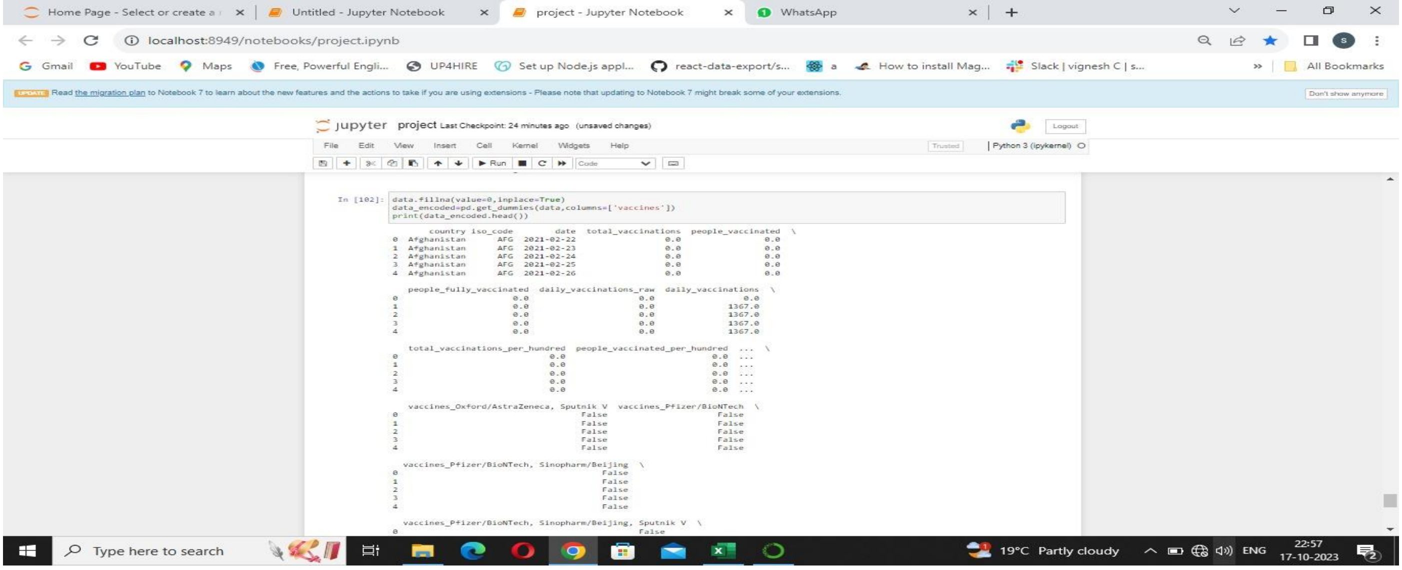
Now, using the sklearn.preprocessing library contains class called imputer, helps in missing data by using the below:


jupyter project Last Checkpoint: an hour ago (unsaved changes)
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code
Logout

```
In [57]: imputer=SimpleImputer(strategy='mean')
         x=np.array(data['people_vaccinated']).reshape(-1,1)
         y=np.array(data['daily_vaccinations']).reshape(-1,1)
         x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.25)
         imputer.fit_transform(x_train)
         x_test=imputer.transform(x_test)
         print(x_test)

         [[18021592.92013698]
          [18021592.92013698]
          [  290216.        ]
          ...
          [18021592.92013698]
          [ 6318501.        ]
          [18021592.92013698]]

In [ ]:
```

Encoding categorical data(one-hot encoding)

One-hot encoding is a technique used to convert categorical data into a numerical format that machine learning algorithms can work with. Here's how you can perform one-hot encoding in Python, assuming you have a dataset with categorical variables:

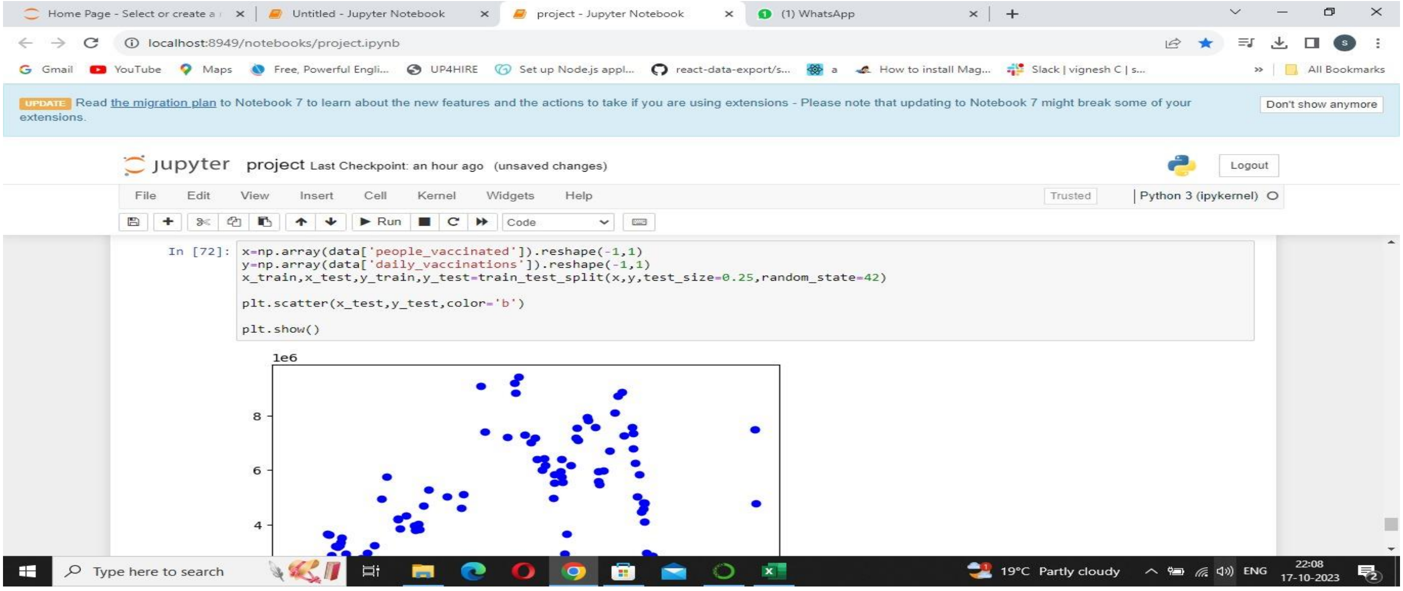

jupyter project Last Checkpoint: 24 minutes ago (unsaved changes)
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code
Logout

```
In [102]: data.fillna(value=0,inplace=True)
          data_encoded=pd.get_dummies(data,columns=['vaccines'])
          print(data_encoded.head())

              country iso_code        date  total_vaccinations  people_vaccinated  \
          0  Afghanistan      AFG  2021-02-22                 0.0                0.0
          1  Afghanistan      AFG  2021-02-23                 0.0                0.0
          2  Afghanistan      AFG  2021-02-24                 0.0                0.0
          3  Afghanistan      AFG  2021-02-25                 0.0                0.0
          4  Afghanistan      AFG  2021-02-26                 0.0                0.0

             people_fully_vaccinated  daily_vaccinations_raw  daily_vaccinations  \
          0                      0.0                     0.0                 0.0
          1                      0.0                     0.0              1367.0
          2                      0.0                     0.0              1367.0
          3                      0.0                     0.0              1367.0
          4                      0.0                     0.0              1367.0

             total_vaccinations_per_hundred  people_vaccinated_per_hundred  ...  \
          0                             0.0                            0.0  ...
          1                             0.0                            0.0  ...
          2                             0.0                            0.0  ...
          3                             0.0                            0.0  ...
          4                             0.0                            0.0  ...

             vaccines_Oxford/AstraZeneca, Sputnik V  vaccines_Pfizer/BioNTech  \
          0                                   False                     False
          1                                   False                     False
          2                                   False                     False
          3                                   False                     False
          4                                   False                     False

            vaccines_Pfizer/BioNTech, Sinopharm/Beijing  \
          0                                        False
          1                                        False
          2                                        False
          3                                        False
          4                                        False

            vaccines_Pfizer/BioNTech, Sinopharm/Beijing, Sputnik V  \
          0                                              False
```

Splitting the data set into test set and training set

By using,

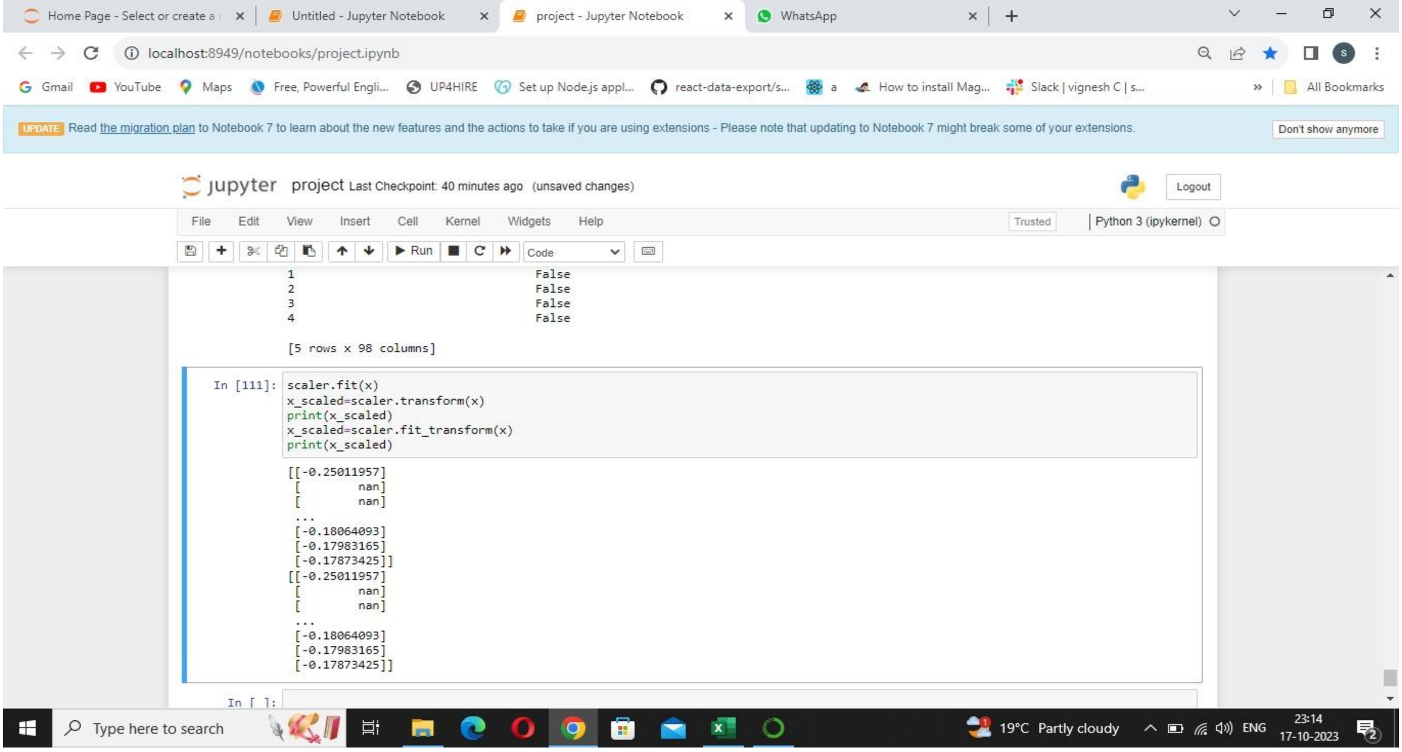
Import train_test_split



```
In [72]: x=np.array(data['people_vaccinated']).reshape(-1,1)
         y=np.array(data['daily_vaccinations']).reshape(-1,1)
         x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.25,random_state=42)

         plt.scatter(x_test,y_test,color='b')

         plt.show()
```

Feature Scaling

By using, import StandardScaler



```
1                                    False
2                                    False
3                                    False
4                                    False

[5 rows x 98 columns]

In [111]: scaler.fit(x)
          x_scaled=scaler.transform(x)
          print(x_scaled)
          x_scaled=scaler.fit_transform(x)
          print(x_scaled)

          [[-0.25011957]
           [         nan]
           [         nan]
           ...
           [-0.18064093]
           [-0.17983165]
           [-0.17873425]]
          [[-0.25011957]
           [         nan]
           [         nan]
           ...
           [-0.18064093]
           [-0.17983165]
           [-0.17873425]]

In [ ]:
```

3.4 PERFORMING DIFFERENT ANALYSIS

1. Descriptive Analysis:

```
import pandas as pd
data = pd.read_csv('your_dataset.csv')
print(data.describe())
```

2. Time Series Analysis:

```
import pandas as pd
data = pd.read_csv('time_series_data.csv')
# Perform time series analysis, e.g., using the statsmodels library
```

3. Correlation Analysis:

```
import pandas as pd
data = pd.read_csv('correlation_data.csv')
correlation_matrix = data.corr()
print(correlation_matrix)
```

4. Regression Analysis:

```
import pandas as pd
from sklearn.linear_model import LinearRegression
data = pd.read_csv('regression_data.csv')
X = data[['independent_variable']]
y = data['dependent_variable']
model = LinearRegression()
model.fit(X, y)
# You can make predictions using the model
```

5. Classification Analysis:

```
import pandas as pd
```

```
from sklearn.model_selection import train_test_split

from sklearn.ensemble import RandomForestClassifier

data = pd.read_csv('classification_data.csv')

X = data.drop('target_class', axis=1)

y = data['target_class']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)

model = RandomForestClassifier()

model.fit(X_train, y_train)

# Evaluate the model's performance
```

6. Text Analysis:

```
from textblob import TextBlob

text = "Your text data here."

blob = TextBlob(text)

# Perform sentiment analysis, extract keywords, etc.
```

7. Cluster Analysis:

```
from sklearn.cluster import KMeans

data = pd.read_csv('cluster_data.csv')

model = KMeans(n_clusters=3)

model.fit(data)

# Explore cluster assignments and centroids
```

8. Financial Ratio Analysis:

```
import pandas as pd

data = pd.read_csv('financial_data.csv')

# Calculate financial ratios, e.g., profit margin, debt-to-equity ratio, etc.
```

9. Statistical Hypothesis Testing:

```
import scipy.stats as stats

group1 = [1, 2, 3, 4, 5]

group2 = [6, 7, 8, 9, 10]

t_stat, p_value = stats.ttest_ind(group1, group2)

# Check if the p-value is significant
```