*# This Python 3 environment comes with many helpful analytics libraries installed*

*# It is defined by the kaggle/python Docker image: https://github.com/kaggle/docker-python*

*# For example, here's several helpful packages to load*

import numpy as np *# linear algebra*

import pandas as pd *# data processing, CSV file I/O (e.g. pd.read\_csv)*

*# Input data files are available in the read-only "../input/" directory*

*# For example, running this (by clicking run or pressing Shift+Enter) will list all files under the input directory*

import os

for dirname, \_, filenames in os.walk('/kaggle/input'): for filename in filenames:

print(os.path.join(dirname, filename))

*# You can write up to 20GB to the current directory (/kaggle/working/) that gets preserved as output when you create a version using "Save & Run All"*

*# You can also write temporary files to /kaggle/temp/, but they won't be saved outside of the current session*

/kaggle/input/covid-world-vaccination-progress/ country\_vaccinations\_by\_manufacturer.csv

/kaggle/input/covid-world-vaccination-progress/country\_vaccinations.cs v

import folium

*# Create a base map centered around Chennai, India*

m = folium.Map(location=[13.0827, 80.2707], zoom\_start=12)

*# Add markers for confirmed cases, vaccination rates, and preferences* folium.Marker([13.0827, 80.2707], popup='Confirmed Cases: 5000', tooltip='Click for Details').add\_to(m)

folium.Marker([13.0837, 80.2717], popup='Vaccination Rate: 65%', tooltip='Click for Details').add\_to(m)

folium.Marker([13.0847, 80.2727], popup='Vaccine Preferences: Covishield', tooltip='Click for Details').add\_to(m)

*# Display the map*

m

<folium.folium.Map at 0x78c1c70d7fd0>

import matplotlib.pyplot as plt

*# Example Data*

vaccines = ['Covishield', 'Covivaccine']

effectiveness = [85, 90] *# Example effectiveness rates in percentage # Create a bar chart*

plt.figure(figsize=(8, 6))

plt.bar(vaccines, effectiveness, color=['blue', 'green']) plt.xlabel('Vaccine Type')

plt.ylabel('Effectiveness (%)') plt.title("Effectiveness of Vaccines in Tamil Nadu") plt.ylim(0, 100)

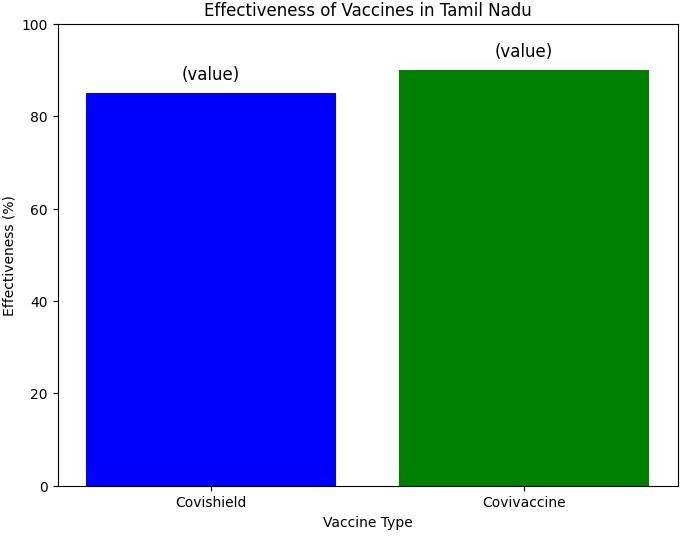
*# Add data labels*

for i, value in enumerate(effectiveness):

plt.text(i, value + 2, f'(value)', ha='center', va='bottom', fontsize=12, color="black")

*#Show the chart*

plt.show()



import pandas as pd

df =

pd.read\_csv("/kaggle/input/covid-world-vaccination-progress/country\_va ccinations.csv")

num\_rows\_before = df.shape[0]

df = df.dropna(subset=['total\_vaccinations', 'people\_vaccinated', 'people\_fully\_vaccinated'], how='all')

num\_rows\_after = df.shape[0]

rows\_removed = num\_rows\_before - num\_rows\_after print(f"Number of rows removed: {rows\_removed}")

Number of rows removed: 42483

import pandas as pd

df =

pd.read\_csv('../input/covid-world-vaccination-progress/country\_vaccina tions\_by\_manufacturer.csv',sep=',')

num\_rows\_before = df.shape[0]

df = df[(df['total\_vaccinations'].notna()) & (df['total\_vaccinations']

!= 0)]

num\_rows\_after = df.shape[0]

rows\_removed = num\_rows\_before - num\_rows\_after

\_

print(f"Number of rows removed: {rows\_removed}")

Number of rows removed: 1482 import pandas as pd

df\_vaccinations = pd.read\_csv("/kaggle/input/covid-world-vaccination- progress/country\_vaccinations.csv")

df\_manufacturer = pd.read\_csv('../input/covid-world-vaccination- progress/country\_vaccinations\_by\_manufacturer.csv',sep=',') merged\_df = pd.merge(df\_vaccinations, df\_manufacturer, on='total\_vaccinations')

print(merged\_df)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | country | iso\_code | date\_x | total\_vaccinations | \ |
| 0 | Afghanistan | AFG | 2021-02-22 | 0.0 |  |
| 1 | Afghanistan | AFG | 2021-02-22 | 0.0 |  |
| 2 | Afghanistan | AFG | 2021-02-22 | 0.0 |  |
| 3 | Afghanistan | AFG | 2021-02-22 | 0.0 |  |
| 4 | Afghanistan | AFG | 2021-02-22 | 0.0 |  |
| ... | ... | ... | ... | ... |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
| 187750 | Zimbabwe | ZWE | 2021-02-18 | 39.0 |
| 187751 | Zimbabwe | ZWE | 2021-02-18 | 39.0 |
| 187752 | Zimbabwe | ZWE | 2021-02-18 | 39.0 |
| 187753 | Zimbabwe | ZWE | 2021-03-09 | 36307.0 |
|  | 187754 | Zimbabwe | ZWE | 2021-03-23 | 45743.0 |

people\_vaccinated people\_fully\_vaccinated daily\_vaccinations\_raw \

0

NaN 1

NaN 2

NaN 3

NaN 4

NaN

...

... 187750

NaN 187751

NaN 187752

NaN 187753

243.0

187754

1623.0

0.0

NaN

0.0

NaN

0.0

NaN

0.0

NaN

0.0

NaN

...

...

39.0

NaN

39.0

NaN

39.0

NaN

36307.0

NaN

44681.0

1062.0

daily\_vaccinations

NaN NaN NaN NaN NaN

...

NaN NaN NaN 1568.0

807.0

total\_vaccinations\_per\_hundred

0.00

0.00

0.00

0.00

0.00

... 0.00

0.00

0.00

0.24

0.30

\

0

1

2

3

4

... 187750

187751

187752

187753

187754

people\_vaccinated\_per\_hundred people\_fully\_vaccinated\_per\_hundred \ 0 0.00

NaN

1 0.00

NaN

2 0.00

NaN

3 0.00

NaN

4 0.00

NaN

... ...

...

187750 0.00

NaN

187751 0.00

NaN

187752 0.00

NaN

187753 0.24

NaN

187754 0.30

0.01

daily\_vaccinations\_per\_million \

1. NaN
2. NaN
3. NaN
4. NaN
5. NaN

... ...

187750 NaN

187751 NaN

187752 NaN

187753 104.0

187754 53.0

vaccines \

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...
5. Johnson&Johnson, Oxford/AstraZeneca, Pfizer/Bi...

... ...

187750 Oxford/AstraZeneca, Sinopharm/Beijing, Sinovac... 187751 Oxford/AstraZeneca, Sinopharm/Beijing, Sinovac... 187752 Oxford/AstraZeneca, Sinopharm/Beijing, Sinovac... 187753 Oxford/AstraZeneca, Sinopharm/Beijing, Sinovac... 187754 Oxford/AstraZeneca, Sinopharm/Beijing, Sinovac...

source\_name \

1. World Health Organization
2. World Health Organization
3. World Health Organization
4. World Health Organization

4 World Health Organization

... ...

187750 Ministry of Health

187751 Ministry of Health

187752 Ministry of Health

187753 Ministry of Health

187754 Ministry of Health

source\_website

location \

2

Austria

https://covid19.who.int/

0

https://covid19.who.int/

1

Austria

https://covid19.who.int/

3

Austria

https://covid19.who.int/

...

Austria

...

.

..

187750 https://[www.arcgis.com/home/webmap/viewer.html...](http://www.arcgis.com/home/webmap/viewer.html) Germany

4

Austria

https://covid19.who.int/

187751 https://[www.arcgis.com/home/webmap/viewer.html...](http://www.arcgis.com/home/webmap/viewer.html) Italy

187752 https://[www.arcgis.com/home/webmap/viewer.html...](http://www.arcgis.com/home/webmap/viewer.html) Liechtenstein

187753 https://[www.arcgis.com/home/webmap/viewer.html...](http://www.arcgis.com/home/webmap/viewer.html) Latvia

187754 https://[www.arcgis.com/home/webmap/viewer.html...](http://www.arcgis.com/home/webmap/viewer.html)

|  |  |  |
| --- | --- | --- |
| Uruguay |  | |
|  |
|  | date\_y | vaccine |
| 0 | 2021-01-08 | Moderna |
| 1 | 2021-01-08 | Novavax |
| 2 | 2021-01-08 | Oxford/AstraZeneca |
| 3 | 2021-01-15 | Novavax |
| 4 | 2021-01-15 | Oxford/AstraZeneca |
| ... | ... | ... |
| 187750 | 2021-02-04 | Johnson&Johnson |
| 187751 | 2021-02-03 | Oxford/AstraZeneca |
| 187752 | 2021-01-15 | Pfizer/BioNTech |
| 187753 | 2021-03-18 | Pfizer/BioNTech |
| 187754 | 2021-06-02 | Oxford/AstraZeneca |

[187755 rows x 18 columns] df =

pd.read\_csv("/kaggle/input/covid-world-vaccination-progress/country\_va

ccinations.csv").head (5)

df['date'] = pd.to\_datetime(df['date']) df.fillna(0, inplace=True)

df

country iso\_code date total\_vaccinations

people\_vaccinated \

0 Afghanistan AFG 2021-02-22 0.0 1 Afghanistan AFG 2021-02-23 0.0 2 Afghanistan AFG 2021-02-24 0.0 3 Afghanistan AFG 2021-02-25 0.0 4 Afghanistan AFG 2021-02-26 0.0

0.0

0.0

0.0

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 |  |

|  |  |  |
| --- | --- | --- |
|  | people\_fully\_vaccinated\_per\_hundred | daily\_vaccinations\_per\_million |
| \ |  |  |
| 0 | 0.0 | 0.0 |
|  |  |  |
| 1 | 0.0 | 34.0 |
|  |  |  |
| 2 | 0.0 | 34.0 |
|  |  |  |
| 3 | 0.0 | 34.0 |
|  |  |  |
| 4 | 0.0 | 34.0 |

0

1

2

3

4

vaccines Johnson&Johnson, Oxford/AstraZeneca, Pfizer/Bi... Johnson&Johnson, Oxford/AstraZeneca, Pfizer/Bi... Johnson&Johnson, Oxford/AstraZeneca, Pfizer/Bi... Johnson&Johnson, Oxford/AstraZeneca, Pfizer/Bi... Johnson&Johnson, Oxford/AstraZeneca, Pfizer/Bi...

\

import pandas as pd

import matplotlib.pyplot as plt df =

pd.read\_csv('../input/covid-world-vaccination-progress/country\_vaccina tions\_by\_manufacturer.csv',sep=',')

additional\_vaccines = ['CanSino', 'Pfizer/BioNTech', 'Johnson&Johnson', 'Covaxin', 'Novavax'] selected\_vaccines = ['Moderna', 'Oxford/AstraZeneca', 'Sinopharm/Beijing', 'Sputnik V']+ additional\_vaccines

df\_selected\_vaccines = df[df['vaccine'].isin (selected\_vaccines)] vaccine\_totals=df\_selected\_vaccines.groupby('vaccine') ['total\_vaccinations'].sum()

plt.figure(figsize=(10,6))

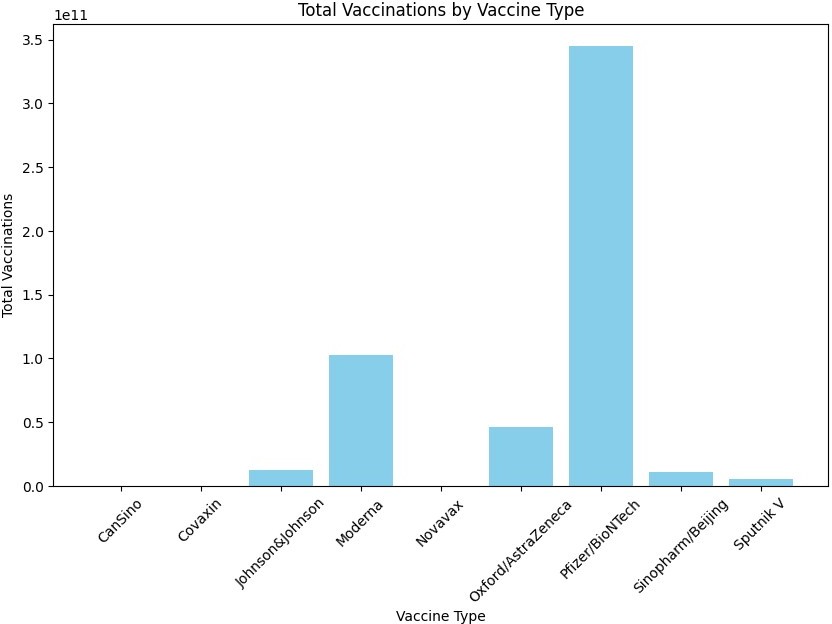
plt.bar(vaccine\_totals.index, vaccine\_totals.values, color='skyblue') plt.xlabel('Vaccine Type')

plt.ylabel('Total Vaccinations') plt.title('Total Vaccinations by Vaccine Type') plt.xticks(rotation=45)

plt.show()

source\_name source\_website

1. World Health Organization https://covid19.who.int/
2. World Health Organization https://covid19.who.int/
3. World Health Organization https://covid19.who.int/
4. World Health Organization https://covid19.who.int/
5. World Health Organization https://covid19.who.int/



import pandas as pd

import matplotlib.pyplot as plt df =

pd.read\_csv("/kaggle/input/covid-world-vaccination-progress/country\_va ccinations.csv")

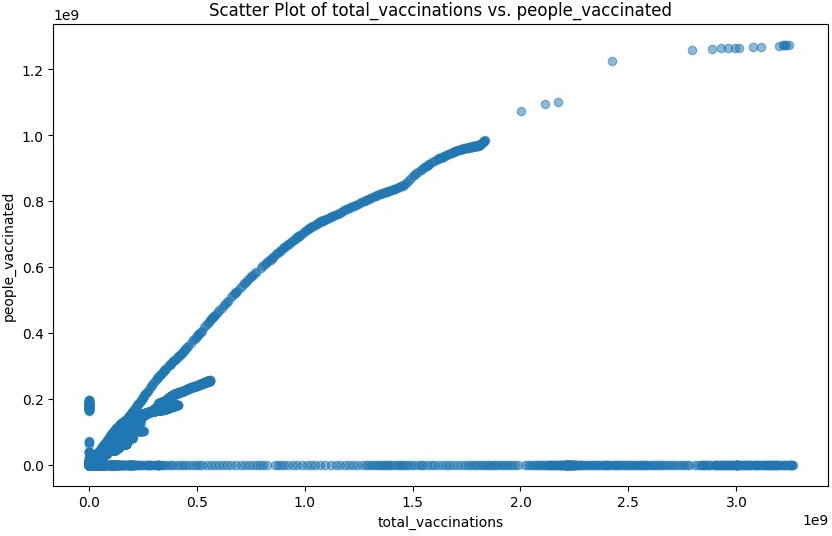
df['date'] = pd.to\_datetime (df['date']) df.fillna(0,inplace=True)

df = pd.get\_dummies (df, columns=['country','iso\_code'], drop\_first=True)

plt.figure(figsize=(10, 6)) x\_variable='total\_vaccinations' y\_variable="people\_vaccinated" plt.scatter(df[x\_variable], df[y\_variable], alpha=0.5) plt.xlabel(x\_variable)

plt.ylabel(y\_variable)

plt.title(f'Scatter Plot of {x\_variable} vs. {y\_variable}') plt.show()



import seaborn as sns

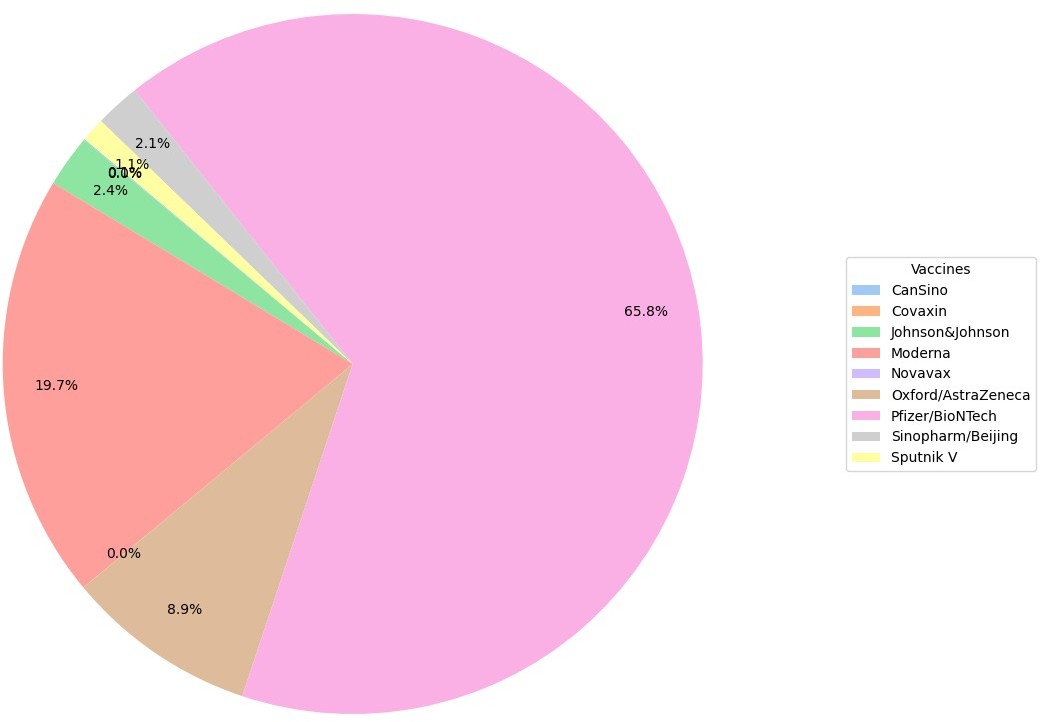
colors = sns.color\_palette('pastel', len(vaccine\_totals)) plt.figure(figsize=(12, 8))

wedges, texts, autotexts = plt.pie(vaccine\_totals, autopct='%1.1f%%', startangle=140, colors=colors, pctdistance=0.85)

plt.axis('equal')

plt.legend(wedges, vaccine\_totals.index, title="Vaccines", loc="center left", bbox\_to\_anchor=(1, 0, 0.5, 1))

plt.tight\_layout() plt.show()



import pandas as pd

import matplotlib.pyplot as plt df =

pd.read\_csv("/kaggle/input/covid-world-vaccination-progress/country\_va ccinations.csv").head(100)

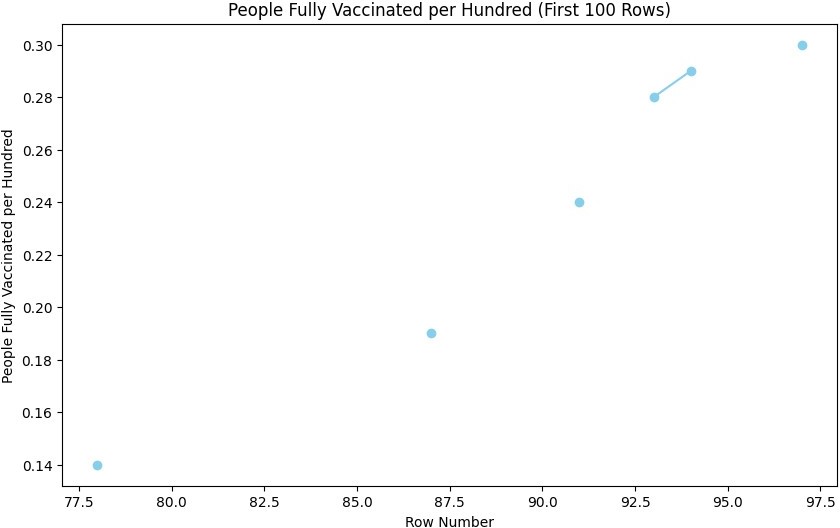
plt.figure(figsize=(10, 6))

plt.plot(df['people\_fully\_vaccinated\_per\_hundred'], marker='o', color='skyblue', linestyle='-')

plt.xlabel('Row Number')

plt.ylabel('People Fully Vaccinated per Hundred') plt.title('People Fully Vaccinated per Hundred (First 100 Rows)')

plt.show()



import pandas as pd

df =

pd.read\_csv("/kaggle/input/covid-world-vaccination-progress/country\_va ccinations.csv")

*# Convert 'date' column to datetime format*

df['date'] = pd.to\_datetime(df['date'])

*# Extract year and month*

df['year'] = df['date'].dt.year

df['month'] = df['date'].dt.month

*# Calculate the percentage*

df['percentage\_fully\_vaccinated'] = (df['people\_fully\_vaccinated'] / df['total\_vaccinations']) \* 100

*# Sort the DataFrame by country, year, and month*

df\_sorted = df.sort\_values(by=['country', 'year', 'month'])

*# Group by country, year, and month, then calculate the average percentage for each month*

result = df\_sorted.groupby(['country', 'year', 'month']) ['percentage\_fully\_vaccinated'].mean()

*# Reset the index to get the DataFrame format*

result\_df = result.reset\_index()

*# Print the result DataFrame*

print(result\_df)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | country | year | month | percentage\_fully\_vaccinated |
| 0 | Afghanistan | 2021 | 2 | NaN |
| 1 | Afghanistan | 2021 | 3 | NaN |
| 2 | Afghanistan | 2021 | 4 | NaN |
| 3 | Afghanistan | 2021 | 5 | 16.675279 |
| 4 | Afghanistan | 2021 | 6 | 23.651193 |
| ... | ... | ... | ... | ... |
| 3013 | Zimbabwe | 2021 | 11 | 43.366796 |
| 3014 | Zimbabwe | 2021 | 12 | 42.953565 |
| 3015 | Zimbabwe | 2022 | 1 | 43.225408 |
| 3016 | Zimbabwe | 2022 | 2 | 43.138566 |
| 3017 | Zimbabwe | 2022 | 3 | 41.875884 |

[3018 rows x 4 columns]

import pandas as pd import folium

*# Load the dataset with total vaccinations by manufacturer*

data =

pd.read\_csv('../input/covid-world-vaccination-progress/country\_vaccina tions\_by\_manufacturer.csv',sep=',')

*# Filter data for Argentina,we can give any country we want.* argentina\_data = data[data['location'] == 'Argentina'] total\_vaccinations\_by\_vaccine = argentina\_data.groupby('vaccine') ['total\_vaccinations'].sum().reset\_index()

*# Get the latitude and longitude for Argentina*

argentina\_lat = -38.4161

argentina\_lon = -63.6167

*# Create a base map centered on Argentina*

map = folium.Map(location=[argentina\_lat, argentina\_lon], zoom\_start=4)

*# Generate popup content with vaccine names and total vaccinations*

popup\_content = '<br>'.join(f"{row['vaccine']} -

{row['total\_vaccinations']}" for \_, row in total\_vaccinations\_by\_vaccine.iterrows())

*# Add a marker with the popup content*

folium.Marker(

location=[argentina\_lat, argentina\_lon], popup=folium.Popup(popup\_content, max\_width=300),

icon=folium.Icon(color='blue')

).add\_to(map)

*# Show the map*

map

<folium.folium.Map at 0x78c1c71144f0> import pandas as pd

*# Load the dataset*

data =

pd.read\_csv('../input/covid-world-vaccination-progress/country\_vaccina tions\_by\_manufacturer.csv',sep=',')

*# Calculate vaccination rate by vaccine type* vaccination\_rate\_by\_vaccine = data.groupby('vaccine') ['total\_vaccinations'].sum() / data.groupby('vaccine').size()

*# Create a new DataFrame to store the result*

result\_df = pd.DataFrame({'vaccination\_rate\_by\_vaccine': vaccination\_rate\_by\_vaccine})

*# Merge the result DataFrame with the original data on 'vaccine'*

data = data.merge(result\_df, on='vaccine', how='left')

*# Show the result*

print(data)

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 |  |  |  |
| ... | ... | ... | ... |
| ... |  |  |  |

35618 European Union 2022-03-29 Oxford/AstraZeneca 67403106

35619 European Union 2022-03-29 Pfizer/BioNTech 600519998

35620 European Union 2022-03-29 Sinopharm/Beijing 2301516

35621 European Union 2022-03-29 Sinovac 1809

35622 European Union 2022-03-29 Sputnik V

1845103

vaccination\_rate\_by\_vaccine 0 1.552058e+07

1 7.003092e+06

2 7.241682e+06

3 5.923586e+06

4 1.552058e+07

... ...

35618 7.003092e+06

35619 3.879792e+07

35620 7.241682e+06

35621 5.556222e+06

35622 5.923586e+06

[35623 rows x 5 columns] import pandas as pd

*# Load the dataset*

data =

pd.read\_csv('../input/covid-world-vaccination-progress/country\_vaccina tions\_by\_manufacturer.csv',sep=',')

*# Calculate total vaccinations by country* total\_vaccinations\_by\_country = data.groupby('location') ['total\_vaccinations'].sum()

*# Convert the result to a DataFrame*

result\_df = total\_vaccinations\_by\_country.reset\_index()

*# Rename the columns for better clarity*

result\_df.columns = ['Country', 'Total Vaccinations']

*# Show the result*

print(result\_df)

|  |  |  |  |
| --- | --- | --- | --- |
|  | Country | Total | Vaccinations |
| 0 | Argentina |  | 18678951288 |
| 1 | Austria |  | 596514759 |
| 2 | Belgium |  | 834395863 |
| 3 | Bulgaria |  | 134238313 |
| 4 | Chile |  | 9931757732 |
| 5 | Croatia |  | 180468380 |
| 6 | Cyprus |  | 55063408 |
| 7 | Czechia |  | 4104977325 |
| 8 | Denmark |  | 410370596 |
| 9 | Ecuador |  | 4599104870 |
| 10 | Estonia |  | 70424761 |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| 11 | European Union | 193020139251 |
| 12 | Finland | 373909900 |
| 13 | France | 32177039372 |
| 14 | Germany | 38982830332 |
| 15 | Hong Kong | 2684669290 |
| 16 | Hungary | 591312341 |
| 17 | Iceland | 4560663 |
| 18 | Ireland | 346279647 |
| 19 | Italy | 28306588441 |
| 20 | Japan | 3910422920 |
| 21 | Latvia | 596129865 |
| 22 | Liechtenstein | 2299059 |
| 23 | Lithuania | 154484516 |
| 24 | Luxembourg | 37541839 |
| 25 | Malta | 42587205 |
| 26 | Nepal | 1343230706 |
| 27 | Netherlands | 1116537275 |
| 28 | Norway | 356592335 |
| 29 | Peru | 10520125014 |
| 30 | Poland | 1824070504 |
| 31 | Portugal | 701877350 |
| 32 | Romania | 4151784964 |
| 33 | Slovakia | 223998563 |
| 34 | Slovenia | 99272786 |
| 35 | South Africa | 820872071 |
| 36 | South Korea | 21786515369 |
| 37 | Spain | 3262817401 |
| 38 | Sweden | 629157444 |
| 39 | Switzerland | 3691948362 |
| 40 | Ukraine | 4357666978 |
| 41 | United States | 139944068653 |
|  | 42 | Uruguay | 1664572676 |

import pandas as pd

from sklearn.linear\_model import LinearRegression import matplotlib.pyplot as plt

*# Load the dataset*

data =

pd.read\_csv('../input/covid-world-vaccination-progress/country\_vaccina tions\_by\_manufacturer.csv',sep=',')

*# Convert 'date' to numerical format (number of days since the start date)*

data['date'] = (pd.to\_datetime(data['date']) -

pd.to\_datetime(data['date']).min()).dt.days

*# Prepare the feature matrix X and the target variable y*

X = data[['date']]

y = data['total\_vaccinations']

*# Initialize the Linear Regression model*

model = LinearRegression() *# Train the model* model.fit(X, y)

*# Print the coefficients* print('Coefficient:', model.coef\_[0]) print('Intercept:', model.intercept\_)

*# Predict total vaccinations*

predictions = model.predict(X)

*# Create a scatter plot of the actual data* plt.scatter(X, y, color='blue', label='Actual Data') *# Plot the regression line*

plt.plot(X, predictions, color='red', linewidth=2, label='Regression Line')

*# Add labels and title* plt.xlabel('Days') plt.ylabel('Total Vaccinations')

plt.title('Linear Regression: Total Vaccinations vs. Days')

*# Add legend*

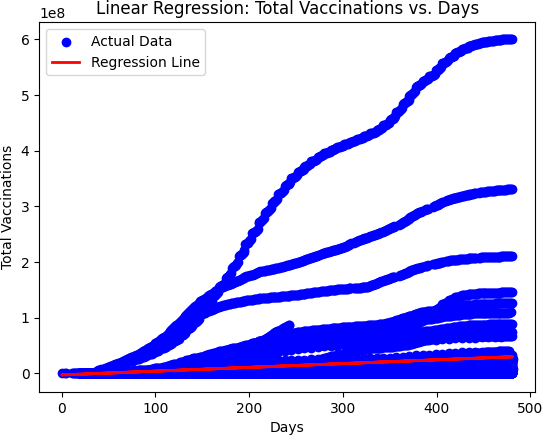
plt.legend()

*# Show the plot*

plt.show()

Coefficient: 67905.85580252815

Intercept: -2954051.2822800055



import pandas as pd

from sklearn.linear\_model import LinearRegression

from sklearn.metrics import mean\_squared\_error, mean\_absolute\_error, r2\_score

import numpy as np

*# Load the dataset*

data =

pd.read\_csv('../input/covid-world-vaccination-progress/country\_vaccina tions\_by\_manufacturer.csv',sep=',')

*# Convert 'date' to numerical format (number of days since the start date)*

data['date'] = (pd.to\_datetime(data['date']) -

pd.to\_datetime(data['date']).min()).dt.days

*# Prepare the feature matrix X and the target variable y*

X = data[['date']]

y = data['total\_vaccinations']

*# Initialize the Linear Regression model*

model = LinearRegression()

*# Training the model x,y*

model.fit(X, y)

*# Predicting total vaccinations*

predictions = model.predict(X)

*# Calculate evaluation metrics*

rmse = np.sqrt(mean\_squared\_error(y, predictions)) mae = mean\_absolute\_error(y, predictions) r\_squared = r2\_score(y, predictions)

*# Printing the evaluation metrics*

print(f'Root Mean Squared Error (RMSE): {rmse:.2f}') print(f'Mean Absolute Error (MAE): {mae:.2f}') print(f'R-squared (R2): {r\_squared:.2f}')

Root Mean Squared Error (RMSE): 51122830.05 Mean Absolute Error (MAE): 20286696.16

R-squared (R2): 0.03