project-10

December 13, 2023

[25]:
```
import pandas as pd
import numpy as np
```

[26]:
```
df=pd.read_csv('country_vaccinations.csv')
```

[27]:
```
df.head()
```

[27]:

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

	people_fully_vaccinated	daily_vaccinations_raw	daily_vaccinations
0	NaN	NaN	NaN
1	NaN	NaN	1367.0
2	NaN	NaN	1367.0
3	NaN	NaN	1367.0
4	NaN	NaN	1367.0

	total_vaccinations_per_hundred	people_vaccinated_per_hundred
0	0.0	0.0
1	NaN	NaN
2	NaN	NaN
3	NaN	NaN
4	NaN	NaN

	people_fully_vaccinated_per_hundred	daily_vaccinations_per_million
0	NaN	NaN
1	NaN	34.0
2	NaN	34.0
3	NaN	34.0
4	NaN	34.0

	vaccines
0	Johnson&Johnson, Oxford/AstraZeneca, Pfizer/Bi…

```
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…
```

	source_name	source_website
0	World Health Organization	https://covid19.who.int/
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/

0.0.1 1.find the number of total vaccinations in india in year 2020,2021 and 2022

[28]:
```
df['date'] = pd.to_datetime(df['date'])
df['year'] = df['date'].dt.year

# Filter data for India
india_data = df[df['country'] == 'India']

# Filter data for each year (2020, 2021, 2022)
vaccinations_2020 = india_data[india_data['year'] ==␣
 ↪2020]['total_vaccinations'].sum()
vaccinations_2021 = india_data[india_data['year'] ==␣
 ↪2021]['total_vaccinations'].sum()
vaccinations_2022 = india_data[india_data['year'] ==␣
 ↪2022]['total_vaccinations'].sum()

# Display the results
print(f'Total vaccinations in India in 2020: {vaccinations_2020}')
print(f'Total vaccinations in India in 2021: {vaccinations_2021}')
print(f'Total vaccinations in India in 2022: {vaccinations_2022}')
```

```
Total vaccinations in India in 2020: 0.0
Total vaccinations in India in 2021: 174118546779.0
Total vaccinations in India in 2022: 149321759019.0
```

0.0.2 2.Comapre number of total vaccinations in year 2020 of india and USA¶

[32]:
```
# Convert the 'date' column to datetime format
df['date'] = pd.to_datetime(df['date'])

# Filter data for the year 2020
df_2020 = df[df['date'].dt.year == 2020]

# Filter data for India and the USA
df_india = df_2020[df_2020['country'] == 'India']
```

```
df_usa = df_2020[df_2020['country'] == 'United States']

# Calculate the total vaccinations for India and the USA in 2020
total_vaccinations_india = df_india['total_vaccinations'].max()
total_vaccinations_usa = df_usa['total_vaccinations'].max()

# Display the results
print(f"Total Vaccinations in India in 2020: {total_vaccinations_india}")
print(f"Total Vaccinations in the USA in 2020: {total_vaccinations_usa}")
```

```
Total Vaccinations in India in 2020: nan
Total Vaccinations in the USA in 2020: 5650180.0
```

0.0.3 3.Comapre Number of total vaccinations in year 2021 of india and china

[34]:
```
# Filter data for the year 2021
df_2021 = df[df['date'].dt.year == 2021]

# Filter data for India and China
india_data = df_2021[df_2021['country'] == 'India']
china_data = df_2021[df_2021['country'] == 'China']

# Calculate the total vaccinations for India and China
total_vaccinations_india = india_data['total_vaccinations'].sum()
total_vaccinations_china = china_data['total_vaccinations'].sum()

# Print the results
print(f"Total vaccinations in India in 2021: {total_vaccinations_india}")
print(f"Total vaccinations in China in 2021: {total_vaccinations_china}")

# Compare the numbers
if total_vaccinations_india > total_vaccinations_china:
    print("India had more total vaccinations in 2021.")
elif total_vaccinations_india < total_vaccinations_china:
    print("China had more total vaccinations in 2021.")
else:
    print("India and China had the same total vaccinations in 2021.")
```

```
Total vaccinations in India in 2021: 174118546779.0
Total vaccinations in China in 2021: 440076944700.0
China had more total vaccinations in 2021.
```

0.0.4 4.find the number of vaccinations in each month in india in the year 2021

[35]:
```
# Convert the 'date' column to datetime format
df['date'] = pd.to_datetime(df['date'])
```

```
# Filter data for India and the year 2021
india_data_2021 = df[(df['country'] == 'India') & (df['date'].dt.year == 2021)]

# Extract month from the date and create a new column 'month'
india_data_2021['month'] = india_data_2021['date'].dt.month

# Group by month and sum the number of vaccinations
monthly_vaccinations = india_data_2021.groupby('month')['total_vaccinations'].
  ↪sum()

# Display the result
monthly_vaccinations
```

```
C:\Users\shivp\AppData\Local\Temp\ipykernel_11856\3348733675.py:8:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  india_data_2021['month'] = india_data_2021['date'].dt.month
```

[35]:
```
month
1     2.832321e+07
2     2.377621e+08
3     1.166007e+09
4     3.038130e+09
5     5.432545e+09
6     7.930055e+09
7     1.112970e+10
8     1.559347e+10
9     2.316289e+10
10    3.029975e+10
11    3.416499e+10
12    4.193492e+10
Name: total_vaccinations, dtype: float64
```

0.0.5 5.Which month has the most number of total vaccinations in india in 2021

[36]:
```
max_vaccinations_month = vaccinations_by_month.idxmax()
max_vaccinations_count = vaccinations_by_month.max()

print("\nMonth with the Most Number of Total Vaccinations in India in 2021:")
print(f"Month: {max_vaccinations_month} | Total Vaccinations:␣
  ↪{max_vaccinations_count}")
```

```
Month with the Most Number of Total Vaccinations in India in 2021:
Month: 12 | Total Vaccinations: 1448865422.0
```

[ ]: