

TITLE: COVID-19 VACCINE ANALYSIS

Step-1: Problem Definition

- The objective of this project is to conduct a comprehensive analysis of Covid-19 vaccine data, with a primary focus on vaccine efficacy, distribution, and adverse effects.
- The ultimate goal is to provide valuable insights that can aid policymakers and health organizations in optimizing vaccine deployment strategies.
- This multifaceted project encompasses data collection, data preprocessing, exploratory data analysis (EDA), statistical analysis, visualization, and the formulation of actionable recommendations.

Step 2: Data Collection

- We will gather Covid-19 vaccine data from reliable sources, including health organizations (e.g., WHO, CDC), government databases, and peer-reviewed research publications.
- The dataset located at (<https://www.kaggle.com/datasets/gpreda/covid-worldvaccination-progress>) will serve as a primary source.
- Data is collected daily from Our World in Data GitHub repository for covid-19, merged and uploaded. Country level vaccination data is gathered and assembled in one single file.
- Then, this data file is merged with locations data file to include vaccination sources information. A second file, with manufacturers information, is included.

```
#import all relevant libraries
```

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

```
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, confusion_matrix, classification_report
```

```
#Loading the dataset
```

```
data=pd.read_csv("C:\\Users\\velpr\\Desktop\\nm\\country_vaccinations.csv")
data.head()
```

| | 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 | NaN | NaN | NaN | |
| 1 | Afghanistan | AFG | 2021-02-23 | NaN | NaN | NaN | NaN | 1367.0 | |
| 2 | Afghanistan | AFG | 2021-02-24 | NaN | NaN | NaN | NaN | 1367.0 | |
| 3 | Afghanistan | AFG | 2021-02-25 | NaN | NaN | NaN | NaN | 1367.0 | |
| 4 | Afghanistan | AFG | 2021-02-26 | NaN | NaN | NaN | NaN | 1367.0 | |

```
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
```

```
data.describe()
```

| | total_vaccinations | people_vaccinated | people_fully_vaccinated | \ |
|-------|--------------------|-------------------|-------------------------|---|
| count | 4.360700e+04 | 4.129400e+04 | 3.880200e+04 | |
| mean | 4.592964e+07 | 1.770508e+07 | 1.413830e+07 | |
| std | 2.246004e+08 | 7.078731e+07 | 5.713920e+07 | |
| min | 0.000000e+00 | 0.000000e+00 | 1.000000e+00 | |
| 25% | 5.264100e+05 | 3.494642e+05 | 2.439622e+05 | |
| 50% | 3.590096e+06 | 2.187310e+06 | 1.722140e+06 | |
| 75% | 1.701230e+07 | 9.152520e+06 | 7.559870e+06 | |
| max | 3.263129e+09 | 1.275541e+09 | 1.240777e+09 | |

| | daily_vaccinations_raw | daily_vaccinations \ |
|-------|------------------------|----------------------|
| count | 3.536200e+04 | 8.621300e+04 |
| mean | 2.705996e+05 | 1.313055e+05 |
| std | 1.212427e+06 | 7.682388e+05 |
| min | 0.000000e+00 | 0.000000e+00 |
| 25% | 4.668000e+03 | 9.000000e+02 |
| 50% | 2.530900e+04 | 7.343000e+03 |
| 75% | 1.234925e+05 | 4.409800e+04 |
| max | 2.474100e+07 | 2.242429e+07 |

| | total_vaccinations_per_hundred | people_vaccinated_per_hundred \ |
|-------|--------------------------------|---------------------------------|
| count | 43607.000000 | 41294.000000 |
| mean | 80.188543 | 40.927317 |
| std | 67.913577 | 29.290759 |
| min | 0.000000 | 0.000000 |
| 25% | 16.050000 | 11.370000 |
| 50% | 67.520000 | 41.435000 |
| 75% | 132.735000 | 67.910000 |
| max | 345.370000 | 124.760000 |

| | people_fully_vaccinated_per_hundred | daily_vaccinations_per_million |
|-------|-------------------------------------|--------------------------------|
| count | 38802.000000 | 86213.000000 |
| mean | 35.523243 | 3257.049157 |
| std | 28.376252 | 3934.312440 |
| min | 0.000000 | 0.000000 |
| 25% | 7.020000 | 636.000000 |
| 50% | 31.750000 | 2050.000000 |
| 75% | 62.080000 | 4682.000000 |
| max | 122.370000 | 117497.000000 |

Step 3: Data Preprocessing

- Cleaning and preprocessing the data are essential steps in preparing it for analysis.
- This involves addressing issues such as duplicate records, inconsistent formatting, handling missing values, and converting categorical features into numerical representations.

```
data.dtypes
```

```
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 |

```
data.isnull().sum()
```

| | |
|-------------------------------------|-------|
| country | 0 |
| iso_code | 0 |
| date | 0 |
| total_vaccinations | 0 |
| people_vaccinated | 0 |
| people_fully_vaccinated | 0 |
| daily_vaccinations_raw | 0 |
| daily_vaccinations | 0 |
| total_vaccinations_per_hundred | 0 |
| people_vaccinated_per_hundred | 0 |
| people_fully_vaccinated_per_hundred | 0 |
| daily_vaccinations_per_million | 0 |
| vaccines | 0 |
| dtype: | int64 |

Step 4: Data Exploration

- Perform exploratory data analysis (EDA) to understand the data's distribution, correlations, and trends.
- In this phase, we will dive into the dataset to gain a deeper understanding of its characteristics. EDA will involve generating statistical summaries, visualizing data distributions, and identifying trends and outliers.
- Key areas of exploration include vaccine distribution across regions, vaccination rates over time, and potential anomalies.
- Visualize the data to gain insights into vaccine distribution and adverse effects

```

#data cleaning data transformation data reduction
#drop irrelevant variables
data=data.drop(['source_name','source_website'],axis=1)
#identifying and treating missing values
data.isnull().sum()
data=data.fillna(0)

data.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 | |

| | 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 | |

| | 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... |

```
#convert the date to datetime
data['date'] = pd.to_datetime(data['date'])
data.dtypes
```

```
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
```

```
# Calculate mean and median total vaccinations
mean_total_vaccinations = data['total_vaccinations'].mean()
median_total_vaccinations = data['total_vaccinations'].median()

# Calculate the correlation between total vaccinations and people fully vaccinated
correlation = data['total_vaccinations'].corr(data['people_fully_vaccinated'])

# Display the results
print(f"Mean Total Vaccinations: {mean_total_vaccinations:.2f}")
print(f"Median Total Vaccinations: {median_total_vaccinations:.2f}")
print(f"Correlation (Total Vaccinations vs. People Fully Vaccinated): {correlation:.2f}")
```

```
Mean Total Vaccinations: 45929644.64
Median Total Vaccinations: 3590096.00
Correlation (Total Vaccinations vs. People Fully Vaccinated): 0.99
```

```
#eda
```

```
data.country.value_counts()
```

```
Norway                482
Latvia                480
Denmark               476
United States         471
Russia                470
...
```

```
Bonaire Sint Eustatius and Saba    146
Tokelau                            114
Saint Helena                      92
Pitcairn                          85
Falkland Islands                  67
Name: country, Length: 223, dtype: int64
```

```
data["Total_vaccinations(count)"] = data.groupby("country").total_vaccinations
.tail(1)
```

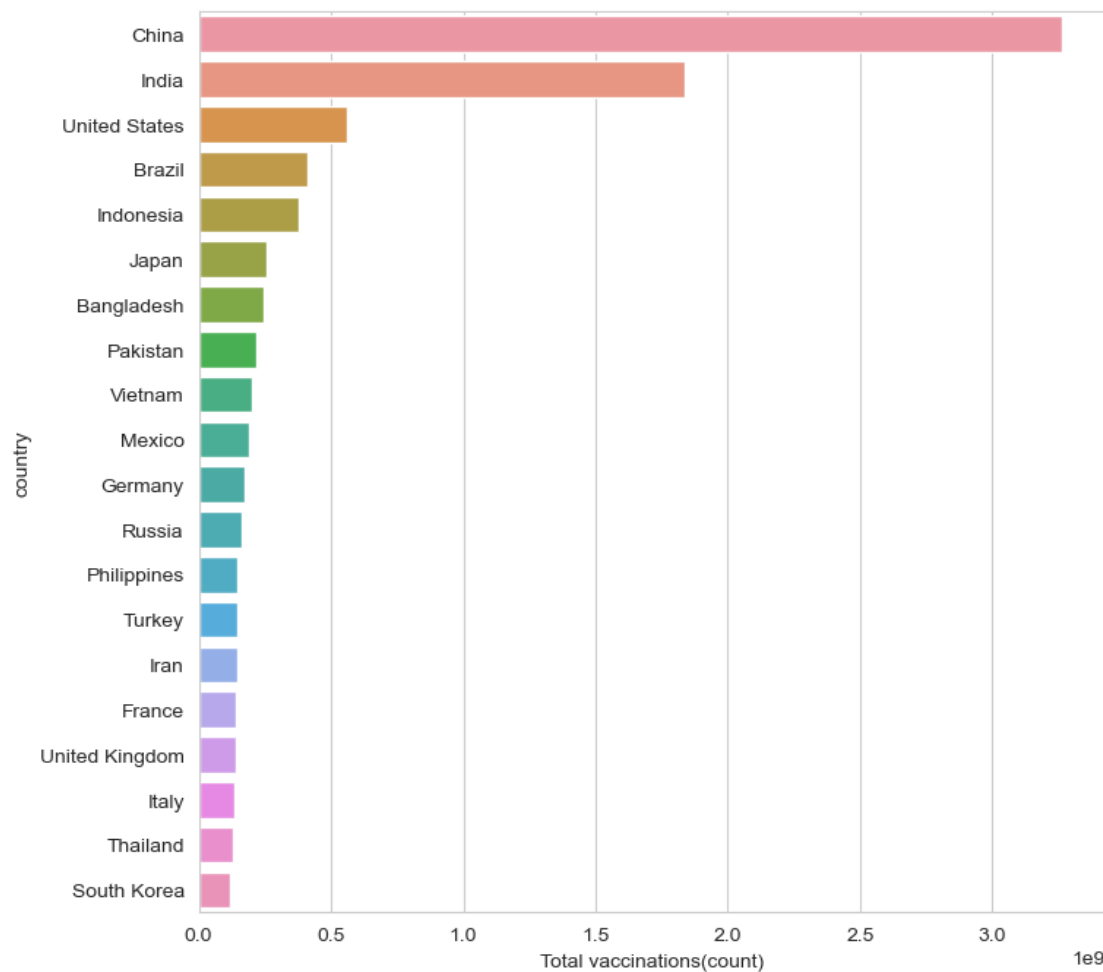
```
#Top countries with most vaccinations
```

```
data.groupby("country")["Total_vaccinations(count)"].mean().sort_values(ascending= False).head(20)
```

```
country
China          3.263129e+09
India          1.834501e+09
United States  5.601818e+08
Brazil         4.135596e+08
Indonesia      3.771089e+08
Japan          2.543456e+08
Bangladesh     2.436427e+08
Pakistan       2.193686e+08
Vietnam        2.031444e+08
Mexico         1.919079e+08
Germany        1.719400e+08
Russia         1.636012e+08
Philippines    1.487991e+08
Turkey         1.468819e+08
Iran           1.467926e+08
France         1.416662e+08
United Kingdom 1.409683e+08
Italy          1.358709e+08
Thailand       1.288824e+08
South Korea    1.206045e+08
Name: Total_vaccinations(count), dtype: float64
```

```
#barplot visualization of top countries with most vaccinations
```

```
x= data.groupby("country")["Total_vaccinations(count)"].mean().sort_values(ascending= False).head(20)
sns.set_style("whitegrid")
plt.figure(figsize= (8,8))
ax= sns.barplot(x.values,x.index)
ax.set_xlabel("Total vaccinations(count)")
plt.show()
```



```
#Top countries with fully vaccinated peoples
data["Full_vaccinations(count)"] = data.groupby("country").people_fully_vaccinated.tail(1)

data.groupby("country")["Full_vaccinations(count)"].mean().sort_values(ascending=False).head(20)
```

| | |
|---------------|-------------|
| country | |
| India | 828229455.0 |
| United States | 217498967.0 |
| Brazil | 160272858.0 |
| Indonesia | 158830466.0 |
| Bangladesh | 107712737.0 |
| Pakistan | 101881176.0 |
| Japan | 100633737.0 |
| Mexico | 79711762.0 |
| Vietnam | 77754108.0 |
| Russia | 72841232.0 |
| Philippines | 65804988.0 |
| Germany | 63142649.0 |


```

Iran          56810058.0
Turkey       52968985.0
France        52438706.0
Thailand       50159803.0
United Kingdom 49404026.0
Italy         47817555.0
South Korea   44482876.0
England       41501690.0
Name: Full_vaccinations(count), dtype: float64

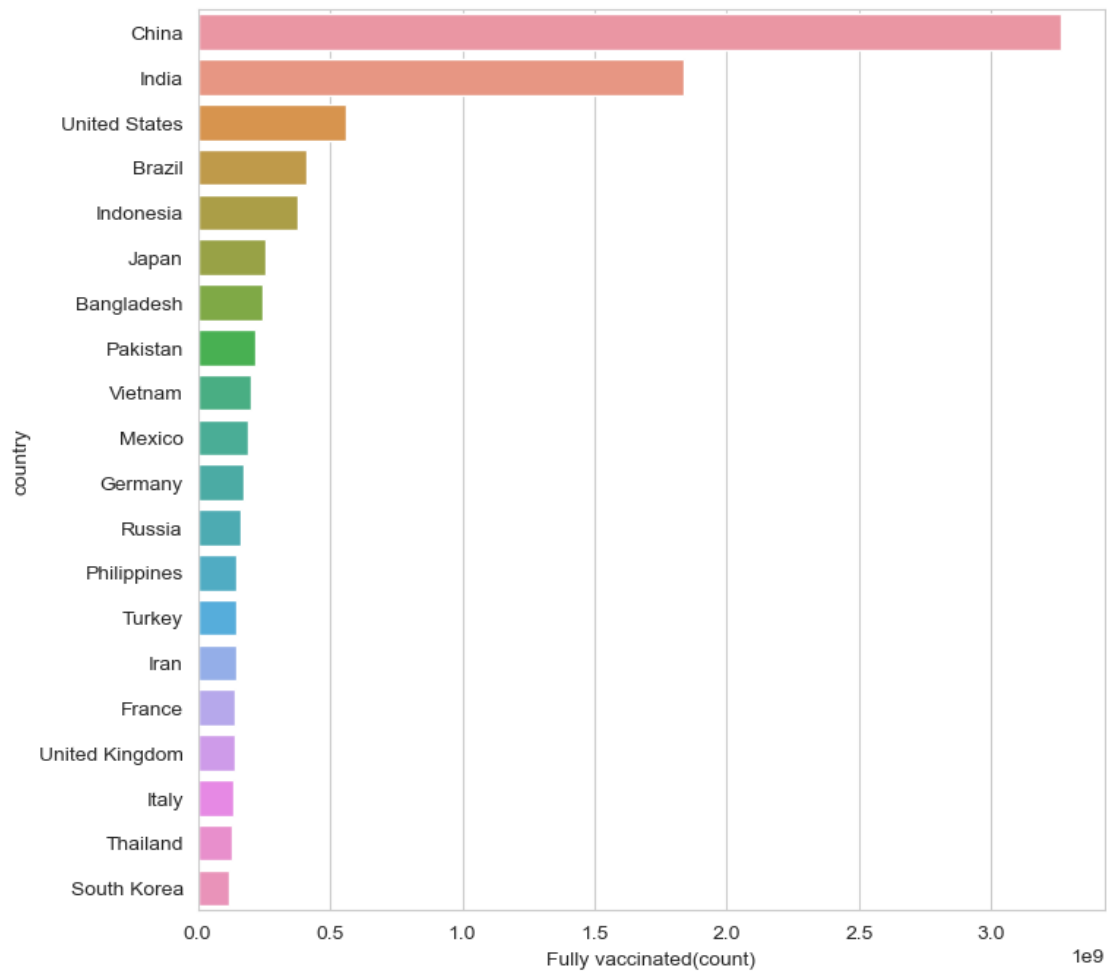
```

#barplot visualization of top countries with most full vaccinations

```

sns.set_style("whitegrid")
plt.figure(figsize= (8,8))
ax= sns.barplot(x.values,x.index)
ax.set_xlabel("Fully vaccinated(count)")
plt.show()

```



#most common vaccines
data.vaccines.value_counts()

```

Johnson&Johnson, Moderna, Oxford/AstraZeneca, Pfizer/BioNTech
7608
Moderna, Oxford/AstraZeneca, Pfizer/BioNTech
6263
Oxford/AstraZeneca
6022
Oxford/AstraZeneca, Pfizer/BioNTech
4629
Johnson&Johnson, Moderna, Novavax, Oxford/AstraZeneca, Pfizer/BioNTech
3564

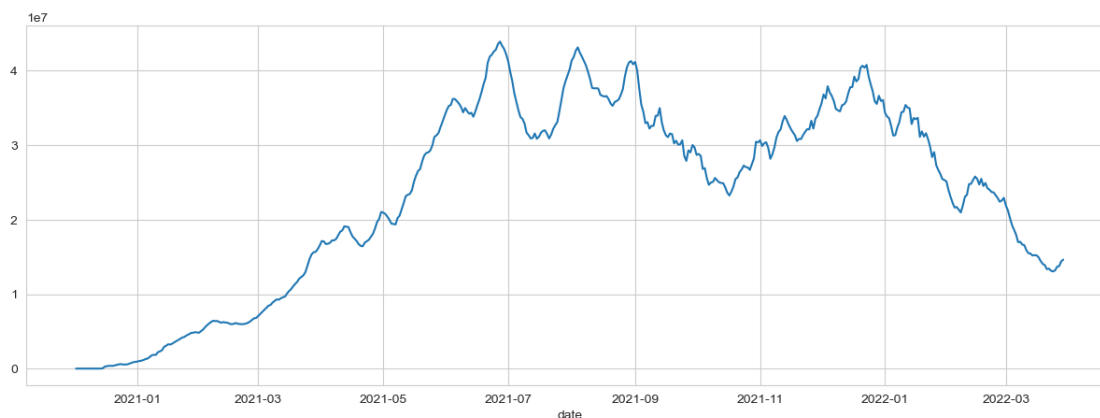
...
Johnson&Johnson, Oxford/AstraZeneca, Sinovac
312
Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, Sinovac, Sputnik V
311
Johnson&Johnson, Moderna
251
Johnson&Johnson, Pfizer/BioNTech, Sinopharm/Beijing
228
EpiVacCorona, Oxford/AstraZeneca, QazVac, Sinopharm/Beijing, Sputnik V, ZF200
1      190
Name: vaccines, Length: 84, dtype: int64

```

```

#daily vaccinations
x= data.groupby("date").daily_vaccinations.sum()
plt.figure(figsize= (15,5))
sns.lineplot(x.index,x.values)
plt.show()

```



```

#preferred vaccine in India
x= data[data["country"]=="India"]
z= x.vaccines.value_counts()
c= list(z.index)
c

```

```
['Covaxin, Oxford/AstraZeneca, Sputnik V']
```

```
#COMPARING TOP 5 COUNTRIES WITH MOST VACCINATIONS
```

```
data.groupby("country")["Total_vaccinations(count)"].mean().sort_values(ascending= False).head()
```

```
country
China          3.263129e+09
India          1.834501e+09
United States   5.601818e+08
Brazil         4.135596e+08
Indonesia      3.771089e+08
Name: Total_vaccinations(count), dtype: float64
```

```
#creating dataframe for top 5 vaccinated countries
```

```
x= data.loc[(data.country== "United States") | (data.country== "China") | (data.country== "India") | (data.country== "United Kingdom") | (data.country== "England")]
```

```
#total vaccination comparison
```

```
plt.figure(figsize= (15,5))
sns.lineplot(x= "date",y= "total_vaccinations" ,data= x,hue= "country")
plt.show()
```

