Covid-19 Vaccines Analysis

**Phase 4: Development Part 2**

The steps of conducting exploratory data analysis (EDA), statistical analysis, and visualization of COVID-19 vaccine data using Python.

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

# Assuming you have a CSV file containing your data

data = pd.read\_csv('country\_vaccinations\_by\_manufacturer.csv')

# View the first few rows of the dataset

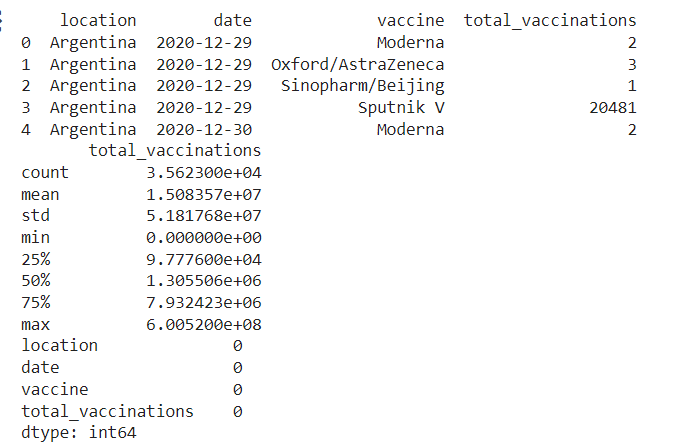
print(data.head())

# Get summary statistics of the data

print(data.describe())

# Check for missing values

print(data.isnull().sum())

Output:

#importing essential libraries

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

import warnings

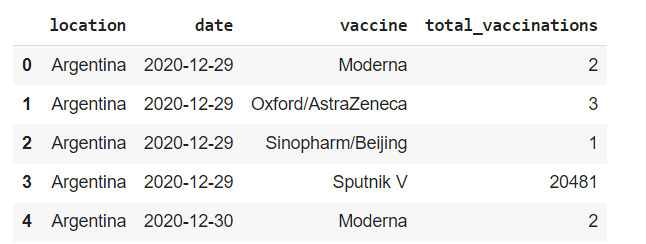
warnings.filterwarnings("ignore")

#reading dataset

df= pd.read\_csv("country\_vaccinations\_by\_manufacturer.csv")

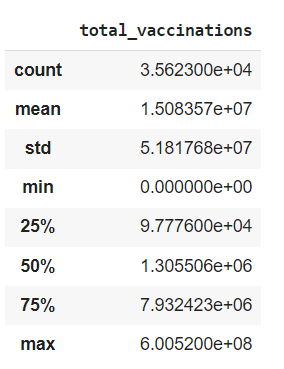
df.head()

output:



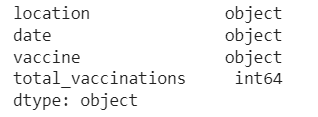
df.describe()

output:



df.dtypes

output:



#converting date column datatype to date

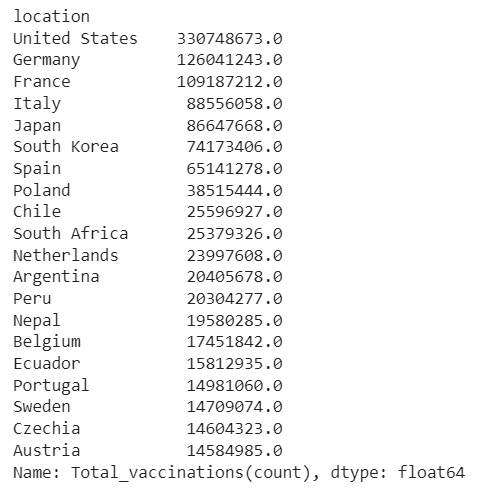
df["date"]= pd.to\_datetime(df.date)

df["Total\_vaccinations(count)"]= df.groupby("location").total\_vaccinations.tail(1)

#Top countries with most vaccinations

df.groupby("location")["Total\_vaccinations(count)"].mean().sort\_values(ascending= False).head(20)

output:



# Calculate mean, median, and standard deviation of a numerical column

mean\_value = data['total\_vaccinations'].mean()

median\_value = data['total\_vaccinations'].median()

std\_dev = data['total\_vaccinations'].std()

plt.hist(data['total\_vaccinations'], bins=20, color='blue', alpha=0.7)

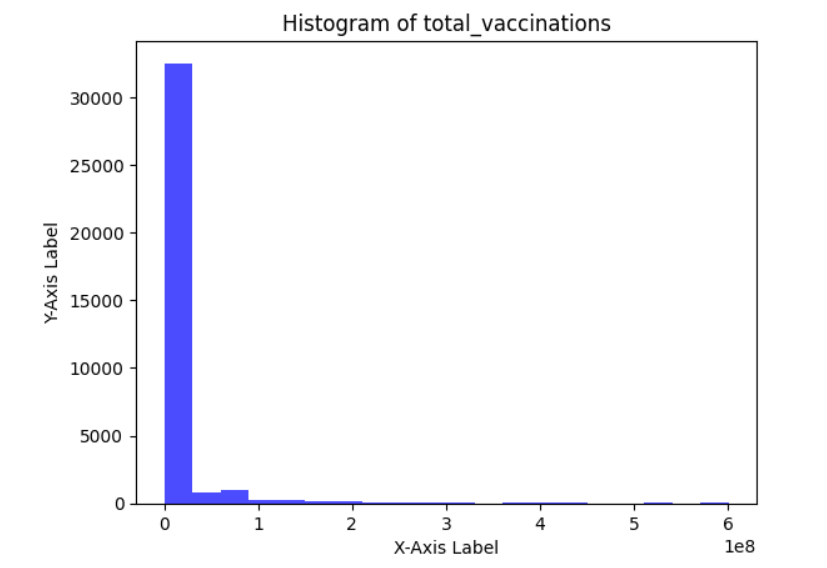
plt.xlabel('X-Axis Label')

plt.ylabel('Y-Axis Label')

plt.title('Histogram of total\_vaccinations')

plt.show()

output:



sns.boxplot(x='total\_vaccinations', y='date', data=data)

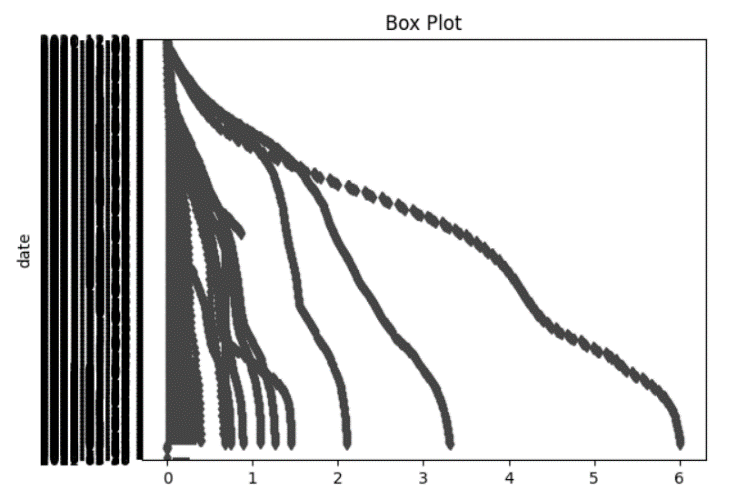
plt.xlabel('total\_vaccinations')

plt.ylabel('date')

plt.title('Box Plot')

plt.show()

output:



plt.scatter(data['total\_vaccinations'], data['date'] , alpha=0.5)

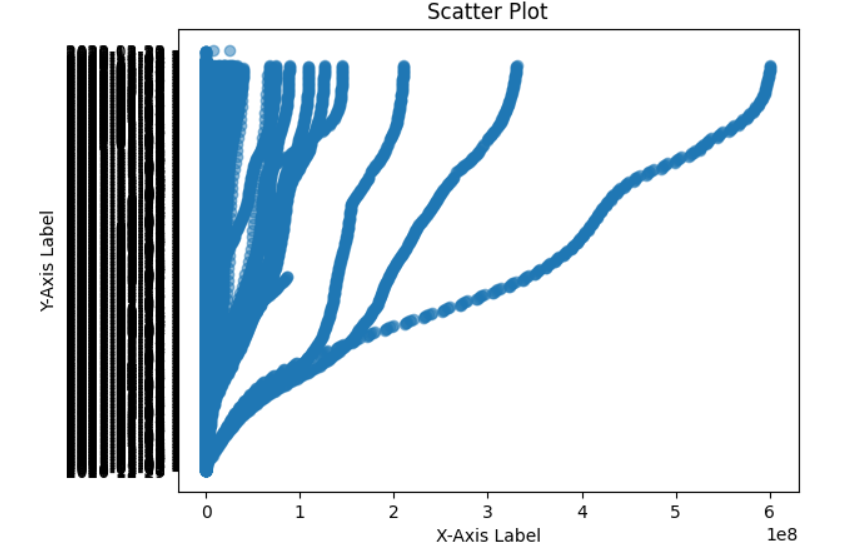
plt.xlabel('X-Axis Label')

plt.ylabel('Y-Axis Label')

plt.title('Scatter Plot')

plt.show()

output:



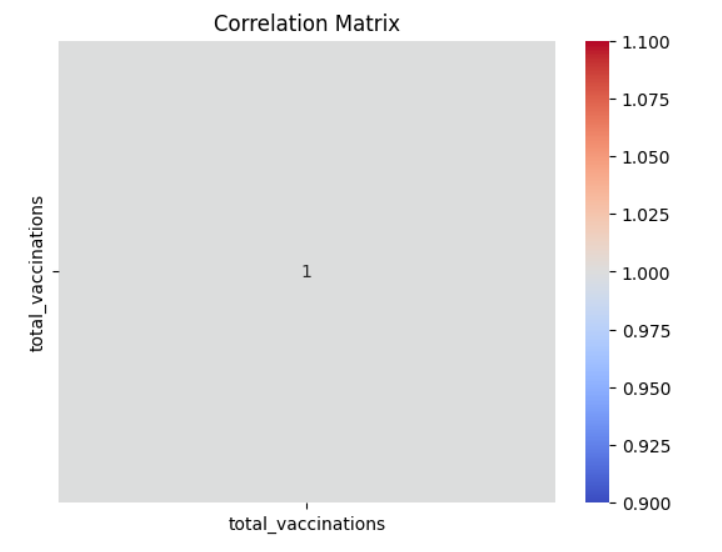
corr\_matrix = data.corr()

sns.heatmap(corr\_matrix, annot=True, cmap='coolwarm')

plt.title('Correlation Matrix')

plt.show()

output:



#barplot visualization of top countries with most vaccinations

x= df.groupby("location")["total\_vaccinations"].mean().sort\_values(ascending= False).head(20)

sns.set\_style("whitegrid")

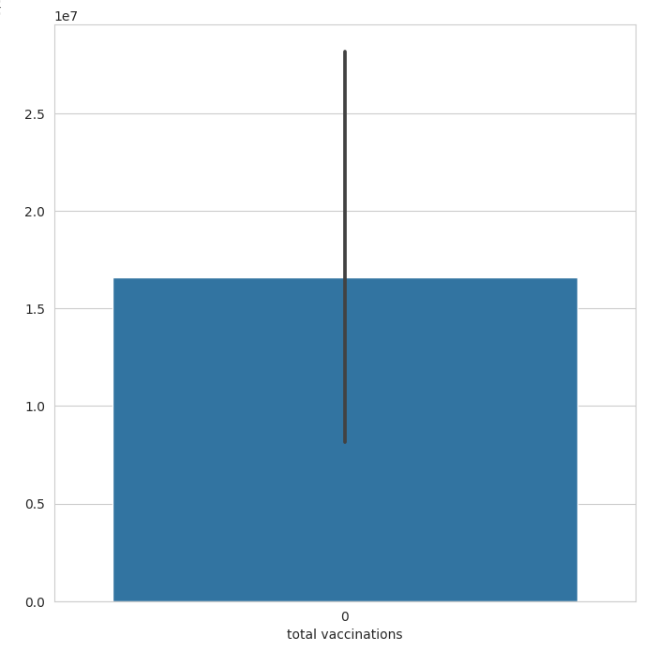
plt.figure(figsize= (8,8))

ax= sns.barplot(x.values)

ax.set\_xlabel("total vaccinations")

plt.show()

output:



#Top countries with fully  vaccinated peoples

df["total\_vaccinations"]= df.groupby("location").total\_vaccinations.tail(1)

df.groupby("location")["total\_vaccinations"].mean().sort\_values(ascending= False).head(20)

output:



sns.set\_style("whitegrid")

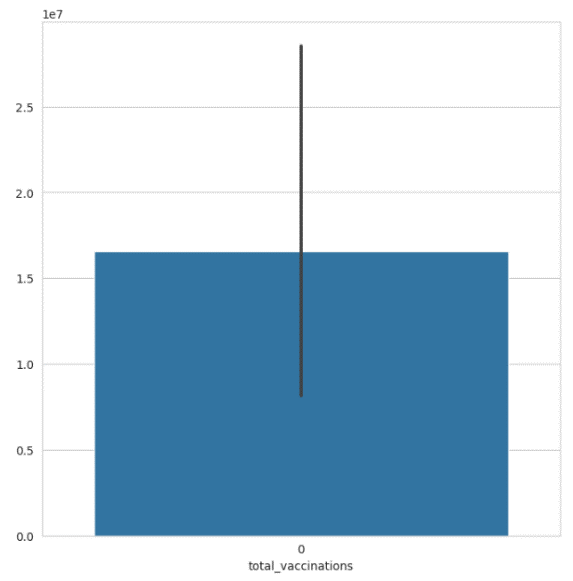
plt.figure(figsize= (8,8))

ax= sns.barplot(x.values)

ax.set\_xlabel("total\_vaccinations")

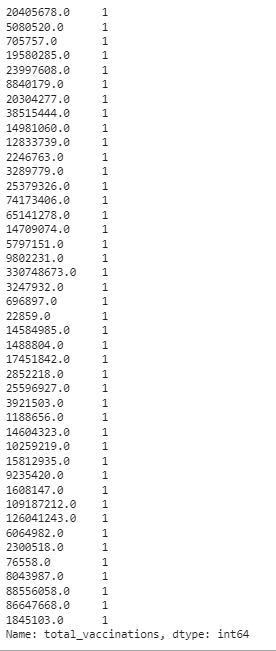
plt.show()

output:



df.total\_vaccinations.value\_counts()

output:



from wordcloud import WordCloud, STOPWORDS

plt.figure(figsize= (20,20))

words= "".join(df["vaccine"])

final = WordCloud(width = 2000, height = 800, background\_color ="black",min\_font\_size = 10).generate(words)

plt.imshow(final)

plt.axis("off")

plt.show()

output:



from wordcloud import WordCloud, STOPWORDS

plt.figure(figsize= (20,20))

words= "".join(df["location"])

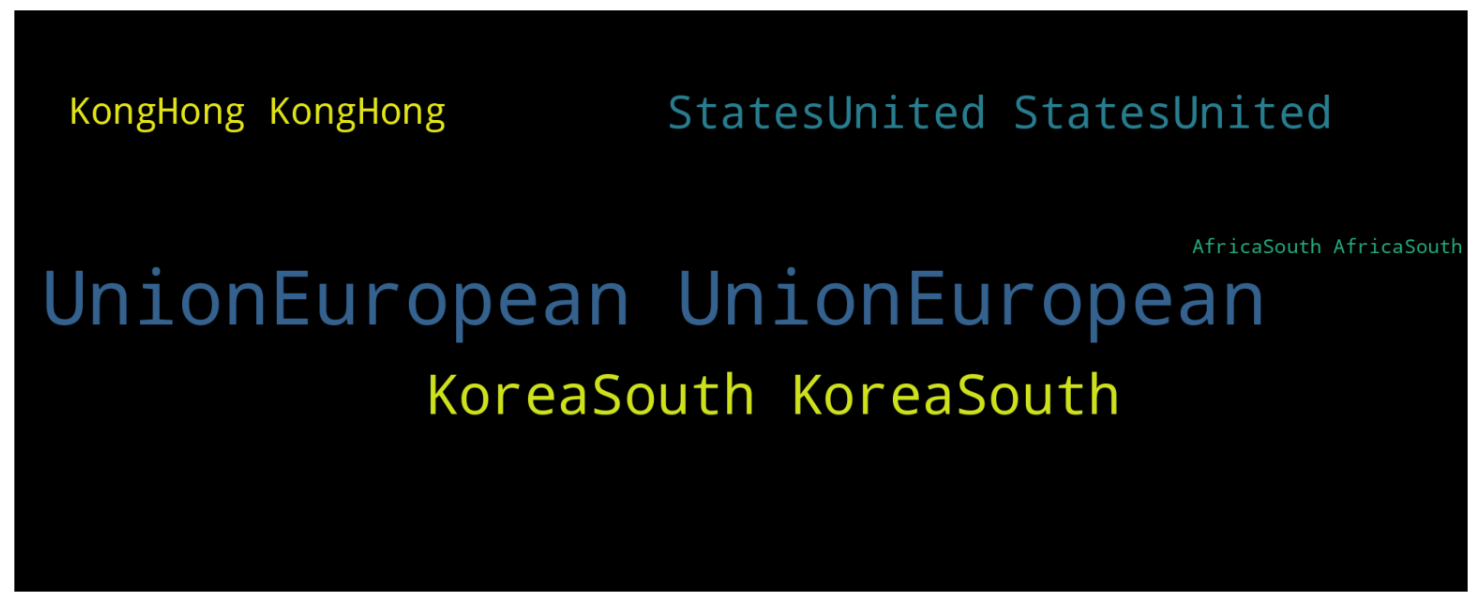
final = WordCloud(width = 2000, height = 800, background\_color ="black",min\_font\_size = 10).generate(words)

plt.imshow(final)

plt.axis("off")

plt.show()

output:

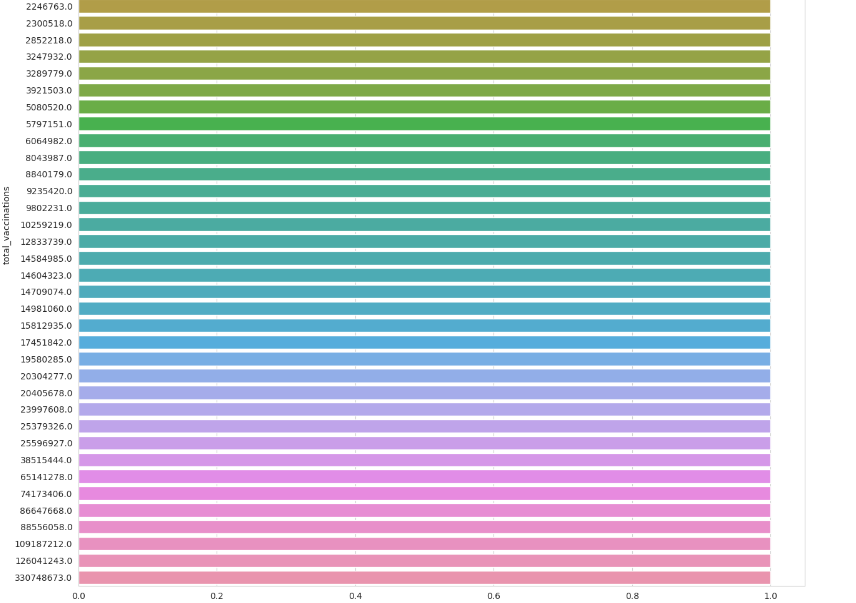


plt.figure(figsize=(15,15))

sns.countplot(y= "total\_vaccinations",data= df)

plt.show()

output:

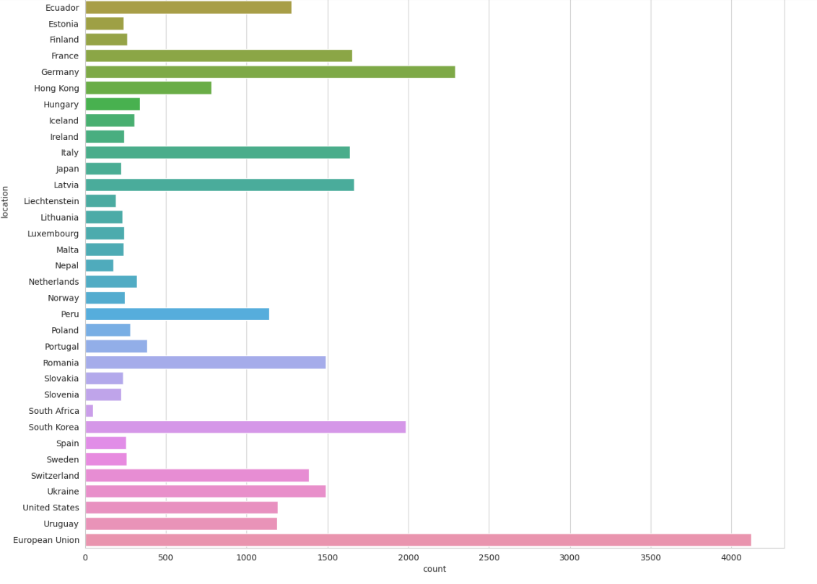


plt.figure(figsize=(15,15))

sns.countplot(y= "location",data= df)

plt.show()

output:



#daily vaccinations

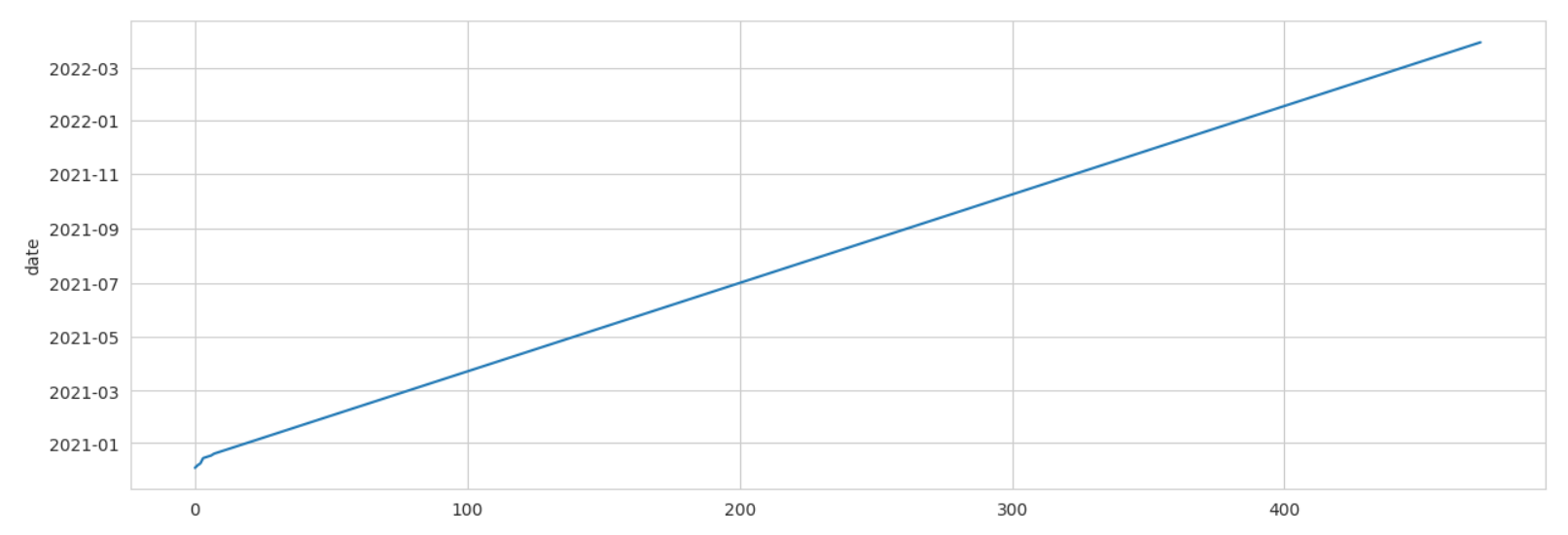
x= df.groupby("date").total\_vaccinations.sum()

plt.figure(figsize= (15,5))

sns.lineplot(x.index)

plt.show()

output:



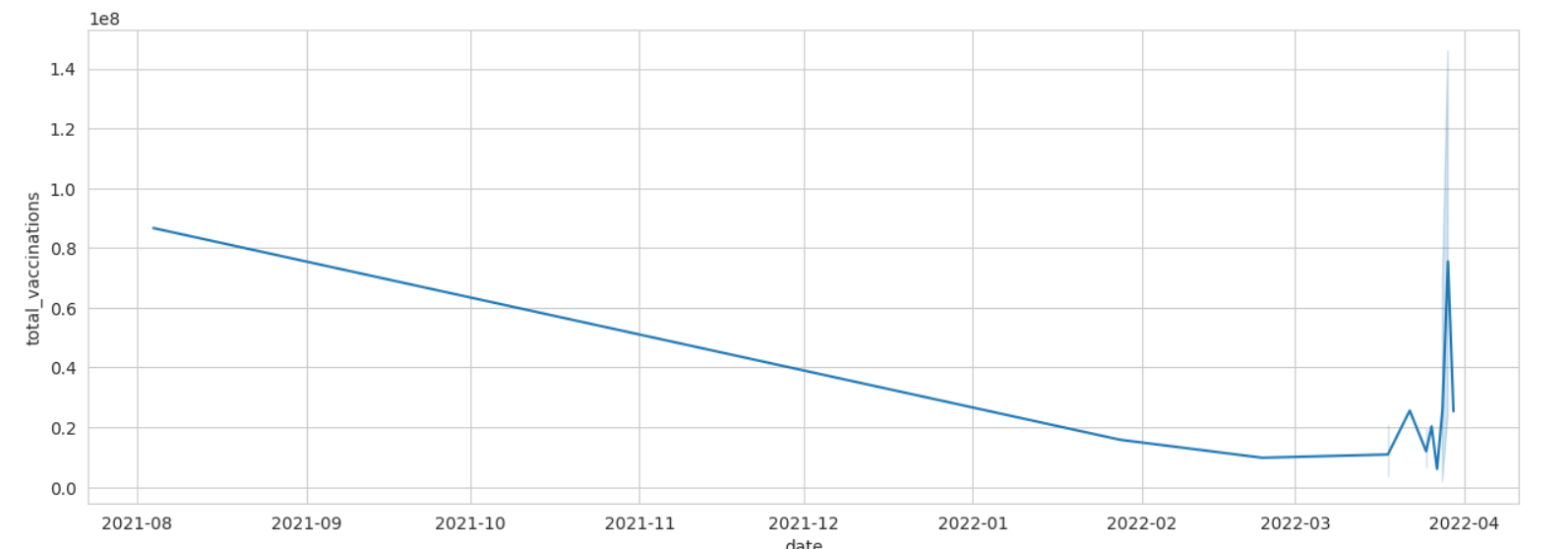
#total vaccinations

plt.figure(figsize= (15,5))

sns.lineplot(x= "date",y= "total\_vaccinations",data= df)

plt.show()

output:



#vaccination per hundred top countries

df["total\_vaccinations"]= df.groupby("location").total\_vaccinations.tail(1)

x= df.groupby("location")["total\_vaccinations"].mean().sort\_values(ascending= False).head(10)

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

ax= sns.barplot(x.values)

ax.set\_xlabel("vaccinations")

plt.show()

output:

