# COVID-19 VACCINATIONS ANALYSIS



# TEAM DETAILS:

### SS Ameesha-961721104012

**Gowri Suresh-961721104009**

**Devichandana AS-961721104006**

**Nima P-961721104010**

# PROBLEM STATEMENT

### Forecasting of time taken for completing 100% total vaccinations of particular region over the time period.By this, vaccine manufacturing companies get to know the prior requirements of vaccine which helps to produce the vaccines in large scale and complete the vaccination drive with in calculated time.

# 1.Data Understanding

### importing data and libraries

import pandas as pd  
import seaborn as sns  
import numpy as np  
import matplotlib.pyplot as plt  
import plotly.express as px  
import io  
import requests  
import warnings  
warnings.filterwarnings('ignore')  
url = "https://raw.githubusercontent.com/owid/covid-19-data/master/public/data/owid-covid-data.csv"  
read\_data = requests.get(url).content

address = pd.read\_csv(io.StringIO(read\_data.decode('utf-8')))  
address.head()

iso\_code continent location date total\_cases new\_cases \  
0 AFG Asia Afghanistan 2020-02-24 5.0 5.0   
1 AFG Asia Afghanistan 2020-02-25 5.0 0.0   
2 AFG Asia Afghanistan 2020-02-26 5.0 0.0   
3 AFG Asia Afghanistan 2020-02-27 5.0 0.0   
4 AFG Asia Afghanistan 2020-02-28 5.0 0.0   
  
 new\_cases\_smoothed total\_deaths new\_deaths new\_deaths\_smoothed ... \  
0 NaN NaN NaN NaN ...   
1 NaN NaN NaN NaN ...   
2 NaN NaN NaN NaN ...   
3 NaN NaN NaN NaN ...   
4 NaN NaN NaN NaN ...   
  
 female\_smokers male\_smokers handwashing\_facilities \  
0 NaN NaN 37.746   
1 NaN NaN 37.746   
2 NaN NaN 37.746   
3 NaN NaN 37.746   
4 NaN NaN 37.746   
  
 hospital\_beds\_per\_thousand life\_expectancy human\_development\_index \  
0 0.5 64.83 0.511   
1 0.5 64.83 0.511   
2 0.5 64.83 0.511   
3 0.5 64.83 0.511   
4 0.5 64.83 0.511   
  
 excess\_mortality\_cumulative\_absolute excess\_mortality\_cumulative \  
0 NaN NaN   
1 NaN NaN   
2 NaN NaN   
3 NaN NaN   
4 NaN NaN   
  
 excess\_mortality excess\_mortality\_cumulative\_per\_million   
0 NaN NaN   
1 NaN NaN   
2 NaN NaN   
3 NaN NaN   
4 NaN NaN   
  
[5 rows x 67 columns]

url2= "https://raw.githubusercontent.com/owid/covid-19-data/master/public/data/vaccinations/vaccinations-by-manufacturer.csv"  
read\_data = requests.get(url2).content

vaccine=pd.read\_csv(io.StringIO(read\_data.decode('utf-8')))

data=address

data.columns

Index(['iso\_code', 'continent', 'location', 'date', 'total\_cases', 'new\_cases',  
 'new\_cases\_smoothed', 'total\_deaths', 'new\_deaths',  
 'new\_deaths\_smoothed', 'total\_cases\_per\_million',  
 'new\_cases\_per\_million', 'new\_cases\_smoothed\_per\_million',  
 'total\_deaths\_per\_million', 'new\_deaths\_per\_million',  
 'new\_deaths\_smoothed\_per\_million', 'reproduction\_rate', 'icu\_patients',  
 'icu\_patients\_per\_million', 'hosp\_patients',  
 'hosp\_patients\_per\_million', 'weekly\_icu\_admissions',  
 'weekly\_icu\_admissions\_per\_million', 'weekly\_hosp\_admissions',  
 'weekly\_hosp\_admissions\_per\_million', 'total\_tests', 'new\_tests',  
 'total\_tests\_per\_thousand', 'new\_tests\_per\_thousand',  
 'new\_tests\_smoothed', 'new\_tests\_smoothed\_per\_thousand',  
 'positive\_rate', 'tests\_per\_case', 'tests\_units', 'total\_vaccinations',  
 'people\_vaccinated', 'people\_fully\_vaccinated', 'total\_boosters',  
 'new\_vaccinations', 'new\_vaccinations\_smoothed',  
 'total\_vaccinations\_per\_hundred', 'people\_vaccinated\_per\_hundred',  
 'people\_fully\_vaccinated\_per\_hundred', 'total\_boosters\_per\_hundred',  
 'new\_vaccinations\_smoothed\_per\_million',  
 'new\_people\_vaccinated\_smoothed',  
 'new\_people\_vaccinated\_smoothed\_per\_hundred', 'stringency\_index',  
 'population', 'population\_density', 'median\_age', 'aged\_65\_older',  
 'aged\_70\_older', 'gdp\_per\_capita', 'extreme\_poverty',  
 'cardiovasc\_death\_rate', 'diabetes\_prevalence', 'female\_smokers',  
 'male\_smokers', 'handwashing\_facilities', 'hospital\_beds\_per\_thousand',  
 'life\_expectancy', 'human\_development\_index',  
 'excess\_mortality\_cumulative\_absolute', 'excess\_mortality\_cumulative',  
 'excess\_mortality', 'excess\_mortality\_cumulative\_per\_million'],  
 dtype='object')

data.info()

<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 191376 entries, 0 to 191375  
Data columns (total 67 columns):  
 # Column Non-Null Count Dtype   
--- ------ -------------- -----   
 0 iso\_code 191376 non-null object   
 1 continent 180250 non-null object   
 2 location 191376 non-null object   
 3 date 191376 non-null object   
 4 total\_cases 183834 non-null float64  
 5 new\_cases 183621 non-null float64  
 6 new\_cases\_smoothed 182447 non-null float64  
 7 total\_deaths 165368 non-null float64  
 8 new\_deaths 165361 non-null float64  
 9 new\_deaths\_smoothed 164198 non-null float64  
 10 total\_cases\_per\_million 182986 non-null float64  
 11 new\_cases\_per\_million 182773 non-null float64  
 12 new\_cases\_smoothed\_per\_million 181604 non-null float64  
 13 total\_deaths\_per\_million 164533 non-null float64  
 14 new\_deaths\_per\_million 164526 non-null float64  
 15 new\_deaths\_smoothed\_per\_million 163368 non-null float64  
 16 reproduction\_rate 140710 non-null float64  
 17 icu\_patients 25496 non-null float64  
 18 icu\_patients\_per\_million 25496 non-null float64  
 19 hosp\_patients 26747 non-null float64  
 20 hosp\_patients\_per\_million 26747 non-null float64  
 21 weekly\_icu\_admissions 6222 non-null float64  
 22 weekly\_icu\_admissions\_per\_million 6222 non-null float64  
 23 weekly\_hosp\_admissions 12397 non-null float64  
 24 weekly\_hosp\_admissions\_per\_million 12397 non-null float64  
 25 total\_tests 77683 non-null float64  
 26 new\_tests 74008 non-null float64  
 27 total\_tests\_per\_thousand 77683 non-null float64  
 28 new\_tests\_per\_thousand 74008 non-null float64  
 29 new\_tests\_smoothed 101315 non-null float64  
 30 new\_tests\_smoothed\_per\_thousand 101315 non-null float64  
 31 positive\_rate 93441 non-null float64  
 32 tests\_per\_case 91681 non-null float64  
 33 tests\_units 104079 non-null object   
 34 total\_vaccinations 52388 non-null float64  
 35 people\_vaccinated 49909 non-null float64  
 36 people\_fully\_vaccinated 47375 non-null float64  
 37 total\_boosters 24452 non-null float64  
 38 new\_vaccinations 42912 non-null float64  
 39 new\_vaccinations\_smoothed 103578 non-null float64  
 40 total\_vaccinations\_per\_hundred 52388 non-null float64  
 41 people\_vaccinated\_per\_hundred 49909 non-null float64  
 42 people\_fully\_vaccinated\_per\_hundred 47375 non-null float64  
 43 total\_boosters\_per\_hundred 24452 non-null float64  
 44 new\_vaccinations\_smoothed\_per\_million 103578 non-null float64  
 45 new\_people\_vaccinated\_smoothed 102491 non-null float64  
 46 new\_people\_vaccinated\_smoothed\_per\_hundred 102491 non-null float64  
 47 stringency\_index 148621 non-null float64  
 48 population 190211 non-null float64  
 49 population\_density 170524 non-null float64  
 50 median\_age 158052 non-null float64  
 51 aged\_65\_older 156377 non-null float64  
 52 aged\_70\_older 157223 non-null float64  
 53 gdp\_per\_capita 157205 non-null float64  
 54 extreme\_poverty 102625 non-null float64  
 55 cardiovasc\_death\_rate 157692 non-null float64  
 56 diabetes\_prevalence 165401 non-null float64  
 57 female\_smokers 119268 non-null float64  
 58 male\_smokers 117633 non-null float64  
 59 handwashing\_facilities 77477 non-null float64  
 60 hospital\_beds\_per\_thousand 139914 non-null float64  
 61 life\_expectancy 178964 non-null float64  
 62 human\_development\_index 153621 non-null float64  
 63 excess\_mortality\_cumulative\_absolute 6553 non-null float64  
 64 excess\_mortality\_cumulative 6553 non-null float64  
 65 excess\_mortality 6553 non-null float64  
 66 excess\_mortality\_cumulative\_per\_million 6553 non-null float64  
dtypes: float64(62), object(5)  
memory usage: 97.8+ MB

data.describe(include='all')

iso\_code continent location date total\_cases new\_cases \  
count 191376 180250 191376 191376 1.838340e+05 1.836210e+05   
unique 244 6 244 885 NaN NaN   
top MEX Africa Mexico 2021-08-26 NaN NaN   
freq 885 44368 885 243 NaN NaN   
mean NaN NaN NaN NaN 3.296788e+06 1.233562e+04   
std NaN NaN NaN NaN 2.070422e+07 8.723816e+04   
min NaN NaN NaN NaN 1.000000e+00 0.000000e+00   
25% NaN NaN NaN NaN 2.696000e+03 0.000000e+00   
50% NaN NaN NaN NaN 3.569300e+04 7.100000e+01   
75% NaN NaN NaN NaN 3.787148e+05 1.027000e+03   
max NaN NaN NaN NaN 5.314561e+08 4.079835e+06   
  
 new\_cases\_smoothed total\_deaths new\_deaths new\_deaths\_smoothed \  
count 1.824470e+05 1.653680e+05 165361.000000 164198.000000   
unique NaN NaN NaN NaN   
top NaN NaN NaN NaN   
freq NaN NaN NaN NaN   
mean 1.237184e+04 6.477486e+04 157.962422 158.909722   
std 8.568677e+04 3.375227e+05 790.438648 773.912384   
min 0.000000e+00 1.000000e+00 0.000000 0.000000   
25% 6.714000e+00 9.400000e+01 0.000000 0.143000   
50% 1.008570e+02 9.170000e+02 1.000000 2.000000   
75% 1.136571e+03 8.457500e+03 17.000000 19.143000   
max 3.437236e+06 6.297755e+06 18151.000000 14795.286000   
  
 ... female\_smokers male\_smokers handwashing\_facilities \  
count ... 119268.000000 117633.000000 77477.000000   
unique ... NaN NaN NaN   
top ... NaN NaN NaN   
freq ... NaN NaN NaN   
mean ... 10.643791 32.792077 50.923861   
std ... 10.580629 13.529850 31.856571   
min ... 0.100000 7.700000 1.188000   
25% ... 1.900000 21.600000 20.859000   
50% ... 6.300000 31.400000 49.839000   
75% ... 19.300000 41.300000 83.241000   
max ... 44.000000 78.100000 100.000000   
  
 hospital\_beds\_per\_thousand life\_expectancy human\_development\_index \  
count 139914.000000 178964.000000 153621.000000   
unique NaN NaN NaN   
top NaN NaN NaN   
freq NaN NaN NaN   
mean 3.068646 73.649521 0.725215   
std 2.520030 7.449589 0.149731   
min 0.100000 53.280000 0.394000   
25% 1.300000 69.500000 0.602000   
50% 2.400000 75.050000 0.743000   
75% 4.000000 79.070000 0.845000   
max 13.800000 86.750000 0.957000   
  
 excess\_mortality\_cumulative\_absolute excess\_mortality\_cumulative \  
count 6.553000e+03 6553.000000   
unique NaN NaN   
top NaN NaN   
freq NaN NaN   
mean 4.093095e+04 9.773072   
std 1.138387e+05 16.007568   
min -3.772610e+04 -28.450000   
25% -1.100000e+01 -0.120000   
50% 4.269400e+03 6.570000   
75% 2.786380e+04 14.700000   
max 1.213593e+06 111.010000   
  
 excess\_mortality excess\_mortality\_cumulative\_per\_million   
count 6553.000000 6553.000000   
unique NaN NaN   
top NaN NaN   
freq NaN NaN   
mean 15.594497 1117.569401   
std 28.965876 1563.316733   
min -95.920000 -1826.595723   
25% -0.280000 -6.412607   
50% 7.460000 574.011450   
75% 21.800000 1794.278748   
max 375.000000 9725.192865   
  
[11 rows x 67 columns]

vaccine.info()

<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 42395 entries, 0 to 42394  
Data columns (total 4 columns):  
 # Column Non-Null Count Dtype   
--- ------ -------------- -----   
 0 location 42395 non-null object  
 1 date 42395 non-null object  
 2 vaccine 42395 non-null object  
 3 total\_vaccinations 42395 non-null int64   
dtypes: int64(1), object(3)  
memory usage: 1.3+ MB

vaccine.describe()

total\_vaccinations  
count 4.239500e+04  
mean 1.782378e+07  
std 5.733925e+07  
min 0.000000e+00  
25% 1.075450e+05  
50% 1.528400e+06  
75% 9.792642e+06  
max 6.141617e+08

# 2.data preprocessing

data.isnull().sum()

iso\_code 0  
continent 11126  
location 0  
date 0  
total\_cases 7542  
 ...   
human\_development\_index 37755  
excess\_mortality\_cumulative\_absolute 184823  
excess\_mortality\_cumulative 184823  
excess\_mortality 184823  
excess\_mortality\_cumulative\_per\_million 184823  
Length: 67, dtype: int64

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

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

data.drop([ 'new\_cases\_smoothed','new\_deaths\_smoothed', 'new\_cases\_smoothed\_per\_million',  
 'new\_deaths\_smoothed\_per\_million', 'reproduction\_rate', 'icu\_patients',  
 'new\_tests\_smoothed', 'new\_tests\_smoothed\_per\_thousand',  
 'new\_vaccinations\_smoothed',  
 'new\_vaccinations\_smoothed\_per\_million',  
 'new\_people\_vaccinated\_smoothed',  
 'new\_people\_vaccinated\_smoothed\_per\_hundred'], axis=1, inplace=True)

data.drop(['icu\_patients\_per\_million','hosp\_patients','hosp\_patients\_per\_million','weekly\_icu\_admissions',  
 'weekly\_icu\_admissions\_per\_million','weekly\_hosp\_admissions','weekly\_hosp\_admissions\_per\_million',  
 'new\_tests\_per\_thousand','excess\_mortality\_cumulative\_absolute','excess\_mortality\_cumulative',  
 'excess\_mortality','excess\_mortality\_cumulative\_per\_million','stringency\_index','life\_expectancy','human\_development\_index','extreme\_poverty',   
'cardiovasc\_death\_rate',   
'diabetes\_prevalence',   
'female\_smokers',   
'male\_smokers',   
'handwashing\_facilities',   
'hospital\_beds\_per\_thousand'],axis= 1,inplace=True)

### checking for the null values

x=data.isnull().sum()\*100/len(data)  
x

iso\_code 0.000000  
continent 5.813686  
location 0.000000  
date 0.000000  
total\_cases 3.940933  
new\_cases 4.052232  
total\_deaths 13.590001  
new\_deaths 13.593659  
total\_cases\_per\_million 4.384040  
new\_cases\_per\_million 4.495339  
total\_deaths\_per\_million 14.026315  
new\_deaths\_per\_million 14.029972  
total\_tests 59.408181  
new\_tests 61.328484  
total\_tests\_per\_thousand 59.408181  
positive\_rate 51.174128  
tests\_per\_case 52.093784  
tests\_units 45.615438  
total\_vaccinations 72.625617  
people\_vaccinated 73.920972  
people\_fully\_vaccinated 75.245067  
total\_boosters 87.223058  
new\_vaccinations 77.577126  
total\_vaccinations\_per\_hundred 72.625617  
people\_vaccinated\_per\_hundred 73.920972  
people\_fully\_vaccinated\_per\_hundred 75.245067  
total\_boosters\_per\_hundred 87.223058  
population 0.608749  
population\_density 10.895828  
median\_age 17.412842  
aged\_65\_older 18.288082  
aged\_70\_older 17.846020  
gdp\_per\_capita 17.855426  
dtype: float64

### checking for duplicate values

duplicate = data[data.duplicated()]   
duplicate

Empty DataFrame  
Columns: [iso\_code, continent, location, date, total\_cases, new\_cases, total\_deaths, new\_deaths, total\_cases\_per\_million, new\_cases\_per\_million, total\_deaths\_per\_million, new\_deaths\_per\_million, total\_tests, new\_tests, total\_tests\_per\_thousand, positive\_rate, tests\_per\_case, tests\_units, total\_vaccinations, people\_vaccinated, people\_fully\_vaccinated, total\_boosters, new\_vaccinations, total\_vaccinations\_per\_hundred, people\_vaccinated\_per\_hundred, people\_fully\_vaccinated\_per\_hundred, total\_boosters\_per\_hundred, population, population\_density, median\_age, aged\_65\_older, aged\_70\_older, gdp\_per\_capita]  
Index: []  
  
[0 rows x 33 columns]

print(data.isnull().values.any())

True

data['total\_deaths'].mean()

64774.858037830774

data['total\_deaths'].median()

917.0

data['total\_deaths'].replace(np.nan,data['total\_deaths'].median()).head(10)

0 917.0  
1 917.0  
2 917.0  
3 917.0  
4 917.0  
5 917.0  
6 917.0  
7 917.0  
8 917.0  
9 917.0  
Name: total\_deaths, dtype: float64

### using bfill method to fill nan cells

data.fillna(method="bfill")

iso\_code continent location date total\_cases new\_cases \  
0 AFG Asia Afghanistan 2020-02-24 5.0 5.0   
1 AFG Asia Afghanistan 2020-02-25 5.0 0.0   
2 AFG Asia Afghanistan 2020-02-26 5.0 0.0   
3 AFG Asia Afghanistan 2020-02-27 5.0 0.0   
4 AFG Asia Afghanistan 2020-02-28 5.0 0.0   
... ... ... ... ... ... ...   
191371 ZWE Africa Zimbabwe 2022-05-30 252092.0 0.0   
191372 ZWE Africa Zimbabwe 2022-05-31 252398.0 306.0   
191373 ZWE Africa Zimbabwe 2022-06-01 252874.0 476.0   
191374 ZWE Africa Zimbabwe 2022-06-02 253051.0 177.0   
191375 ZWE Africa Zimbabwe 2022-06-03 253236.0 185.0   
  
 total\_deaths new\_deaths total\_cases\_per\_million \  
0 1.0 1.0 0.126   
1 1.0 1.0 0.126   
2 1.0 1.0 0.126   
3 1.0 1.0 0.126   
4 1.0 1.0 0.126   
... ... ... ...   
191371 5500.0 0.0 16703.495   
191372 5503.0 3.0 16723.770   
191373 5507.0 4.0 16755.310   
191374 5508.0 1.0 16767.038   
191375 5509.0 1.0 16779.296   
  
 new\_cases\_per\_million ... total\_vaccinations\_per\_hundred \  
0 0.126 ... 0.00   
1 0.000 ... 0.00   
2 0.000 ... 0.00   
3 0.000 ... 0.00   
4 0.000 ... 0.00   
... ... ... ...   
191371 0.000 ... 77.95   
191372 20.275 ... 78.00   
191373 31.540 ... 78.02   
191374 11.728 ... NaN   
191375 12.258 ... NaN   
  
 people\_vaccinated\_per\_hundred people\_fully\_vaccinated\_per\_hundred \  
0 0.00 0.14   
1 0.00 0.14   
2 0.00 0.14   
3 0.00 0.14   
4 0.00 0.14   
... ... ...   
191371 41.39 29.94   
191372 41.41 29.96   
191373 41.41 29.98   
191374 NaN NaN   
191375 NaN NaN   
  
 total\_boosters\_per\_hundred population population\_density \  
0 0.00 39835428.0 54.422   
1 0.00 39835428.0 54.422   
2 0.00 39835428.0 54.422   
3 0.00 39835428.0 54.422   
4 0.00 39835428.0 54.422   
... ... ... ...   
191371 6.61 15092171.0 42.729   
191372 6.63 15092171.0 42.729   
191373 6.63 15092171.0 42.729   
191374 NaN 15092171.0 42.729   
191375 NaN 15092171.0 42.729   
  
 median\_age aged\_65\_older aged\_70\_older gdp\_per\_capita   
0 18.6 2.581 1.337 1803.987   
1 18.6 2.581 1.337 1803.987   
2 18.6 2.581 1.337 1803.987   
3 18.6 2.581 1.337 1803.987   
4 18.6 2.581 1.337 1803.987   
... ... ... ... ...   
191371 19.6 2.822 1.882 1899.775   
191372 19.6 2.822 1.882 1899.775   
191373 19.6 2.822 1.882 1899.775   
191374 19.6 2.822 1.882 1899.775   
191375 19.6 2.822 1.882 1899.775   
  
[191376 rows x 33 columns]

data.isnull().values.any() #Checking fo nan values in whole dataframe

True

data.head()

iso\_code continent location date total\_cases new\_cases \  
0 AFG Asia Afghanistan 2020-02-24 5.0 5.0   
1 AFG Asia Afghanistan 2020-02-25 5.0 0.0   
2 AFG Asia Afghanistan 2020-02-26 5.0 0.0   
3 AFG Asia Afghanistan 2020-02-27 5.0 0.0   
4 AFG Asia Afghanistan 2020-02-28 5.0 0.0   
  
 total\_deaths new\_deaths total\_cases\_per\_million new\_cases\_per\_million \  
0 NaN NaN 0.126 0.126   
1 NaN NaN 0.126 0.000   
2 NaN NaN 0.126 0.000   
3 NaN NaN 0.126 0.000   
4 NaN NaN 0.126 0.000   
  
 ... total\_vaccinations\_per\_hundred people\_vaccinated\_per\_hundred \  
0 ... NaN NaN   
1 ... NaN NaN   
2 ... NaN NaN   
3 ... NaN NaN   
4 ... NaN NaN   
  
 people\_fully\_vaccinated\_per\_hundred total\_boosters\_per\_hundred \  
0 NaN NaN   
1 NaN NaN   
2 NaN NaN   
3 NaN NaN   
4 NaN NaN   
  
 population population\_density median\_age aged\_65\_older aged\_70\_older \  
0 39835428.0 54.422 18.6 2.581 1.337   
1 39835428.0 54.422 18.6 2.581 1.337   
2 39835428.0 54.422 18.6 2.581 1.337   
3 39835428.0 54.422 18.6 2.581 1.337   
4 39835428.0 54.422 18.6 2.581 1.337   
  
 gdp\_per\_capita   
0 1803.987   
1 1803.987   
2 1803.987   
3 1803.987   
4 1803.987   
  
[5 rows x 33 columns]

data.info(  
)

<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 191376 entries, 0 to 191375  
Data columns (total 33 columns):  
 # Column Non-Null Count Dtype   
--- ------ -------------- -----   
 0 iso\_code 191376 non-null object   
 1 continent 180250 non-null object   
 2 location 191376 non-null object   
 3 date 191376 non-null datetime64[ns]  
 4 total\_cases 183834 non-null float64   
 5 new\_cases 183621 non-null float64   
 6 total\_deaths 165368 non-null float64   
 7 new\_deaths 165361 non-null float64   
 8 total\_cases\_per\_million 182986 non-null float64   
 9 new\_cases\_per\_million 182773 non-null float64   
 10 total\_deaths\_per\_million 164533 non-null float64   
 11 new\_deaths\_per\_million 164526 non-null float64   
 12 total\_tests 77683 non-null float64   
 13 new\_tests 74008 non-null float64   
 14 total\_tests\_per\_thousand 77683 non-null float64   
 15 positive\_rate 93441 non-null float64   
 16 tests\_per\_case 91681 non-null float64   
 17 tests\_units 104079 non-null object   
 18 total\_vaccinations 52388 non-null float64   
 19 people\_vaccinated 49909 non-null float64   
 20 people\_fully\_vaccinated 47375 non-null float64   
 21 total\_boosters 24452 non-null float64   
 22 new\_vaccinations 42912 non-null float64   
 23 total\_vaccinations\_per\_hundred 52388 non-null float64   
 24 people\_vaccinated\_per\_hundred 49909 non-null float64   
 25 people\_fully\_vaccinated\_per\_hundred 47375 non-null float64   
 26 total\_boosters\_per\_hundred 24452 non-null float64   
 27 population 190211 non-null float64   
 28 population\_density 170524 non-null float64   
 29 median\_age 158052 non-null float64   
 30 aged\_65\_older 156377 non-null float64   
 31 aged\_70\_older 157223 non-null float64   
 32 gdp\_per\_capita 157205 non-null float64   
dtypes: datetime64[ns](1), float64(28), object(4)  
memory usage: 48.2+ MB

data.drop(['tests\_units'],axis=1,inplace=True)

null\_percentage=data.isna().sum()\*100/len(data)  
null\_percentage.head(38)

iso\_code 0.000000  
continent 5.813686  
location 0.000000  
date 0.000000  
total\_cases 3.940933  
new\_cases 4.052232  
total\_deaths 13.590001  
new\_deaths 13.593659  
total\_cases\_per\_million 4.384040  
new\_cases\_per\_million 4.495339  
total\_deaths\_per\_million 14.026315  
new\_deaths\_per\_million 14.029972  
total\_tests 59.408181  
new\_tests 61.328484  
total\_tests\_per\_thousand 59.408181  
positive\_rate 51.174128  
tests\_per\_case 52.093784  
total\_vaccinations 72.625617  
people\_vaccinated 73.920972  
people\_fully\_vaccinated 75.245067  
total\_boosters 87.223058  
new\_vaccinations 77.577126  
total\_vaccinations\_per\_hundred 72.625617  
people\_vaccinated\_per\_hundred 73.920972  
people\_fully\_vaccinated\_per\_hundred 75.245067  
total\_boosters\_per\_hundred 87.223058  
population 0.608749  
population\_density 10.895828  
median\_age 17.412842  
aged\_65\_older 18.288082  
aged\_70\_older 17.846020  
gdp\_per\_capita 17.855426  
dtype: float64

data=data.fillna(method="bfill")

null\_percentage=data.isna().sum()\*100/len(data)  
  
null\_percentage.head(38)

iso\_code 0.000000  
continent 0.000000  
location 0.000000  
date 0.000000  
total\_cases 0.000000  
new\_cases 0.000000  
total\_deaths 0.000000  
new\_deaths 0.000000  
total\_cases\_per\_million 0.000000  
new\_cases\_per\_million 0.000000  
total\_deaths\_per\_million 0.000000  
new\_deaths\_per\_million 0.000000  
total\_tests 0.000523  
new\_tests 0.007838  
total\_tests\_per\_thousand 0.000523  
positive\_rate 0.000523  
tests\_per\_case 0.000523  
total\_vaccinations 0.001045  
people\_vaccinated 0.001045  
people\_fully\_vaccinated 0.001045  
total\_boosters 0.001045  
new\_vaccinations 0.001045  
total\_vaccinations\_per\_hundred 0.001045  
people\_vaccinated\_per\_hundred 0.001045  
people\_fully\_vaccinated\_per\_hundred 0.001045  
total\_boosters\_per\_hundred 0.001045  
population 0.000000  
population\_density 0.000000  
median\_age 0.000000  
aged\_65\_older 0.000000  
aged\_70\_older 0.000000  
gdp\_per\_capita 0.000000  
dtype: float64

data.isnull().sum()

iso\_code 0  
continent 0  
location 0  
date 0  
total\_cases 0  
new\_cases 0  
total\_deaths 0  
new\_deaths 0  
total\_cases\_per\_million 0  
new\_cases\_per\_million 0  
total\_deaths\_per\_million 0  
new\_deaths\_per\_million 0  
total\_tests 1  
new\_tests 15  
total\_tests\_per\_thousand 1  
positive\_rate 1  
tests\_per\_case 1  
total\_vaccinations 2  
people\_vaccinated 2  
people\_fully\_vaccinated 2  
total\_boosters 2  
new\_vaccinations 2  
total\_vaccinations\_per\_hundred 2  
people\_vaccinated\_per\_hundred 2  
people\_fully\_vaccinated\_per\_hundred 2  
total\_boosters\_per\_hundred 2  
population 0  
population\_density 0  
median\_age 0  
aged\_65\_older 0  
aged\_70\_older 0  
gdp\_per\_capita 0  
dtype: int64

data['new\_tests'].replace(np.nan,data['new\_tests'].median(),inplace=True)  
data['positive\_rate'].replace(np.nan,data['positive\_rate'].median(),inplace=True)  
data['tests\_per\_case'].replace(np.nan,data['tests\_per\_case'].median(),inplace=True)  
data['new\_vaccinations'].replace(np.nan,data['new\_vaccinations'].median(),inplace=True)

data.isnull().sum()

iso\_code 0  
continent 0  
location 0  
date 0  
total\_cases 0  
new\_cases 0  
total\_deaths 0  
new\_deaths 0  
total\_cases\_per\_million 0  
new\_cases\_per\_million 0  
total\_deaths\_per\_million 0  
new\_deaths\_per\_million 0  
total\_tests 1  
new\_tests 0  
total\_tests\_per\_thousand 1  
positive\_rate 0  
tests\_per\_case 0  
total\_vaccinations 2  
people\_vaccinated 2  
people\_fully\_vaccinated 2  
total\_boosters 2  
new\_vaccinations 0  
total\_vaccinations\_per\_hundred 2  
people\_vaccinated\_per\_hundred 2  
people\_fully\_vaccinated\_per\_hundred 2  
total\_boosters\_per\_hundred 2  
population 0  
population\_density 0  
median\_age 0  
aged\_65\_older 0  
aged\_70\_older 0  
gdp\_per\_capita 0  
dtype: int64

v=vaccine.drop(['total\_vaccinations'], axis = 1)  
v

location date vaccine  
0 Argentina 2020-02-24 Moderna  
1 Argentina 2020-02-25 Oxford/AstraZeneca  
2 Argentina 2020-02-26 Pfizer/BioNTech  
3 Argentina 2020-02-27 Sinopharm/Beijing  
4 Argentina 2020-02-28 Sputnik V  
... ... ... ...  
42390 European Union 2020-05-27 Oxford/AstraZeneca  
42391 European Union 2020-05-28 Pfizer/BioNTech  
42392 European Union 2020-05-29 Sinopharm/Beijing  
42393 European Union 2020-05-30 Sinovac  
42394 European Union 2020-05-31 Sputnik V  
  
[42395 rows x 3 columns]

### Integrating two datasets,vaccines and vaccination names

final=pd.merge(data,v,on=['date','location'],how='left')

final.isnull().sum()

iso\_code 0  
continent 0  
location 0  
date 0  
total\_cases 0  
new\_cases 0  
total\_deaths 0  
new\_deaths 0  
total\_cases\_per\_million 0  
new\_cases\_per\_million 0  
total\_deaths\_per\_million 0  
new\_deaths\_per\_million 0  
total\_tests 1  
new\_tests 0  
total\_tests\_per\_thousand 1  
positive\_rate 0  
tests\_per\_case 0  
total\_vaccinations 2  
people\_vaccinated 2  
people\_fully\_vaccinated 2  
total\_boosters 2  
new\_vaccinations 0  
total\_vaccinations\_per\_hundred 2  
people\_vaccinated\_per\_hundred 2  
people\_fully\_vaccinated\_per\_hundred 2  
total\_boosters\_per\_hundred 2  
population 0  
population\_density 0  
median\_age 0  
aged\_65\_older 0  
aged\_70\_older 0  
gdp\_per\_capita 0  
vaccine 168644  
dtype: int64

n2=final.isna().sum()\*100/len(final)

final['vaccine'].isna().sum()/len(final)\*100

79.99165192314078

fin=pd.merge(data,v,on=['date','location'],how='left')  
fin

iso\_code continent location date total\_cases new\_cases \  
0 AFG Asia Afghanistan 2020-02-24 5.0 5.0   
1 AFG Asia Afghanistan 2020-02-25 5.0 0.0   
2 AFG Asia Afghanistan 2020-02-26 5.0 0.0   
3 AFG Asia Afghanistan 2020-02-27 5.0 0.0   
4 AFG Asia Afghanistan 2020-02-28 5.0 0.0   
... ... ... ... ... ... ...   
210822 ZWE Africa Zimbabwe 2022-05-30 252092.0 0.0   
210823 ZWE Africa Zimbabwe 2022-05-31 252398.0 306.0   
210824 ZWE Africa Zimbabwe 2022-06-01 252874.0 476.0   
210825 ZWE Africa Zimbabwe 2022-06-02 253051.0 177.0   
210826 ZWE Africa Zimbabwe 2022-06-03 253236.0 185.0   
  
 total\_deaths new\_deaths total\_cases\_per\_million \  
0 1.0 1.0 0.126   
1 1.0 1.0 0.126   
2 1.0 1.0 0.126   
3 1.0 1.0 0.126   
4 1.0 1.0 0.126   
... ... ... ...   
210822 5500.0 0.0 16703.495   
210823 5503.0 3.0 16723.770   
210824 5507.0 4.0 16755.310   
210825 5508.0 1.0 16767.038   
210826 5509.0 1.0 16779.296   
  
 new\_cases\_per\_million ... people\_vaccinated\_per\_hundred \  
0 0.126 ... 0.00   
1 0.000 ... 0.00   
2 0.000 ... 0.00   
3 0.000 ... 0.00   
4 0.000 ... 0.00   
... ... ... ...   
210822 0.000 ... 41.39   
210823 20.275 ... 41.41   
210824 31.540 ... 41.41   
210825 11.728 ... NaN   
210826 12.258 ... NaN   
  
 people\_fully\_vaccinated\_per\_hundred total\_boosters\_per\_hundred \  
0 0.14 0.00   
1 0.14 0.00   
2 0.14 0.00   
3 0.14 0.00   
4 0.14 0.00   
... ... ...   
210822 29.94 6.61   
210823 29.96 6.63   
210824 29.98 6.63   
210825 NaN NaN   
210826 NaN NaN   
  
 population population\_density median\_age aged\_65\_older \  
0 39835428.0 54.422 18.6 2.581   
1 39835428.0 54.422 18.6 2.581   
2 39835428.0 54.422 18.6 2.581   
3 39835428.0 54.422 18.6 2.581   
4 39835428.0 54.422 18.6 2.581   
... ... ... ... ...   
210822 15092171.0 42.729 19.6 2.822   
210823 15092171.0 42.729 19.6 2.822   
210824 15092171.0 42.729 19.6 2.822   
210825 15092171.0 42.729 19.6 2.822   
210826 15092171.0 42.729 19.6 2.822   
  
 aged\_70\_older gdp\_per\_capita vaccine   
0 1.337 1803.987 NaN   
1 1.337 1803.987 NaN   
2 1.337 1803.987 NaN   
3 1.337 1803.987 NaN   
4 1.337 1803.987 NaN   
... ... ... ...   
210822 1.882 1899.775 NaN   
210823 1.882 1899.775 NaN   
210824 1.882 1899.775 NaN   
210825 1.882 1899.775 NaN   
210826 1.882 1899.775 NaN   
  
[210827 rows x 33 columns]

### one hot encoding

def convert\_to\_binary(df, column\_to\_convert):  
 categories = list(df[column\_to\_convert].drop\_duplicates())  
  
 for category in categories:  
 cat\_name = str(category).replace(" ", "\_").replace("(", "").replace(")", "").replace("/", "\_").replace("-", "").lower()  
 col\_name = column\_to\_convert[:5] + '\_' + cat\_name[:10]  
 df[col\_name] = 0  
 df.loc[(df[column\_to\_convert] == category), col\_name] = 1  
  
 return df  
  
  
print("One Hot Encoding categorical data...")  
columns\_to\_convert = ['vaccine']  
  
for column in columns\_to\_convert:  
 df\_all = convert\_to\_binary(df=final, column\_to\_convert=column)  
 df\_all.drop(column, axis=1, inplace=True)  
print("One Hot Encoding categorical data...completed")

One Hot Encoding categorical data...  
One Hot Encoding categorical data...completed

fin=fin.dropna()  
fin

iso\_code continent location date total\_cases new\_cases \  
6617 ARG South America Argentina 2020-02-13 1.0 1.0   
6618 ARG South America Argentina 2020-02-14 1.0 1.0   
6619 ARG South America Argentina 2020-02-15 1.0 1.0   
6620 ARG South America Argentina 2020-02-16 1.0 1.0   
6621 ARG South America Argentina 2020-02-17 1.0 1.0   
... ... ... ... ... ... ...   
203088 URY South America Uruguay 2022-05-30 916388.0 0.0   
203089 URY South America Uruguay 2022-05-31 925777.0 9389.0   
203090 URY South America Uruguay 2022-06-01 925777.0 1.0   
203091 URY South America Uruguay 2022-06-02 925777.0 1.0   
203092 URY South America Uruguay 2022-06-03 925777.0 1.0   
  
 total\_deaths new\_deaths total\_cases\_per\_million \  
6617 1.0 1.0 0.022   
6618 1.0 1.0 0.022   
6619 1.0 1.0 0.022   
6620 1.0 1.0 0.022   
6621 1.0 1.0 0.022   
... ... ... ...   
203088 7227.0 0.0 262940.612   
203089 7238.0 11.0 265634.612   
203090 7238.0 1.0 265634.612   
203091 7238.0 1.0 265634.612   
203092 7238.0 1.0 265634.612   
  
 new\_cases\_per\_million ... people\_vaccinated\_per\_hundred \  
6617 0.022 ... 0.04   
6618 0.022 ... 0.04   
6619 0.022 ... 0.04   
6620 0.022 ... 0.04   
6621 0.022 ... 0.04   
... ... ... ...   
203088 0.000 ... 85.83   
203089 2694.000 ... 85.84   
203090 0.029 ... 85.86   
203091 0.029 ... 85.88   
203092 0.029 ... 85.88   
  
 people\_fully\_vaccinated\_per\_hundred total\_boosters\_per\_hundred \  
6617 0.00 0.00   
6618 0.00 0.00   
6619 0.00 0.00   
6620 0.00 0.00   
6621 0.00 0.00   
... ... ...   
203088 82.50 72.56   
203089 82.52 72.91   
203090 82.55 73.26   
203091 82.58 73.60   
203092 82.60 74.07   
  
 population population\_density median\_age aged\_65\_older \  
6617 45605823.0 16.177 31.9 11.198   
6618 45605823.0 16.177 31.9 11.198   
6619 45605823.0 16.177 31.9 11.198   
6620 45605823.0 16.177 31.9 11.198   
6621 45605823.0 16.177 31.9 11.198   
... ... ... ... ...   
203088 3485152.0 19.751 35.6 14.655   
203089 3485152.0 19.751 35.6 14.655   
203090 3485152.0 19.751 35.6 14.655   
203091 3485152.0 19.751 35.6 14.655   
203092 3485152.0 19.751 35.6 14.655   
  
 aged\_70\_older gdp\_per\_capita vaccine   
6617 7.441 18933.907 Sputnik V   
6618 7.441 18933.907 CanSino   
6619 7.441 18933.907 Moderna   
6620 7.441 18933.907 Oxford/AstraZeneca   
6621 7.441 18933.907 Pfizer/BioNTech   
... ... ... ...   
203088 10.361 20551.409 Pfizer/BioNTech   
203089 10.361 20551.409 Sinovac   
203090 10.361 20551.409 Oxford/AstraZeneca   
203091 10.361 20551.409 Pfizer/BioNTech   
203092 10.361 20551.409 Sinovac   
  
[42183 rows x 33 columns]

idf2=vaccine.groupby('vaccine',as\_index=False).sum()

idf2=idf2[['vaccine','total\_vaccinations']]

idf2.total\_vaccinations[0]

165653252

idf2

vaccine total\_vaccinations  
0 CanSino 165653252  
1 Covaxin 16898  
2 Johnson&Johnson 16124009967  
3 Moderna 146511657511  
4 Novavax 56784475  
5 Oxford/AstraZeneca 56807596055  
6 Pfizer/BioNTech 497342012032  
7 Sinopharm/Beijing 15479222902  
8 Sinovac 15887323481  
9 Sputnik V 7264694689

idf=vaccine['vaccine']

idf=idf.to\_frame()

idf=idf.dropna()

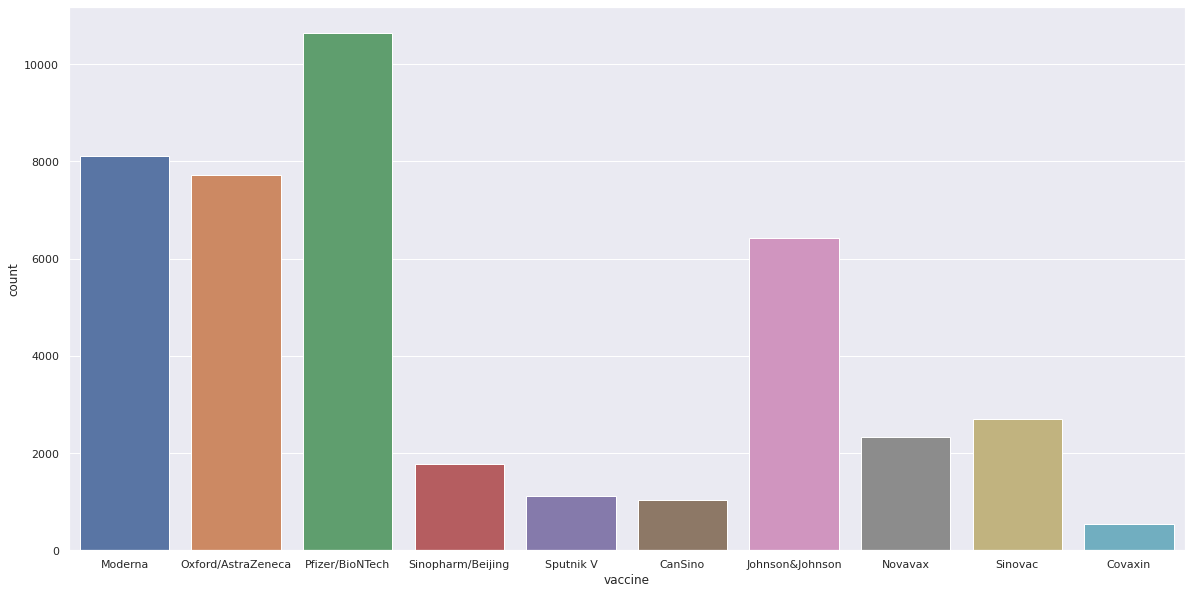
idf

vaccine  
0 Moderna  
1 Oxford/AstraZeneca  
2 Pfizer/BioNTech  
3 Sinopharm/Beijing  
4 Sputnik V  
... ...  
42390 Oxford/AstraZeneca  
42391 Pfizer/BioNTech  
42392 Sinopharm/Beijing  
42393 Sinovac  
42394 Sputnik V  
  
[42395 rows x 1 columns]

# 3. Questions

### 1.which is the most used vaccines?

sns.set\_theme(style="darkgrid")  
sns.set(rc = {'figure.figsize':(20,10)})  
ax = sns.countplot(x="vaccine", data=vaccine)



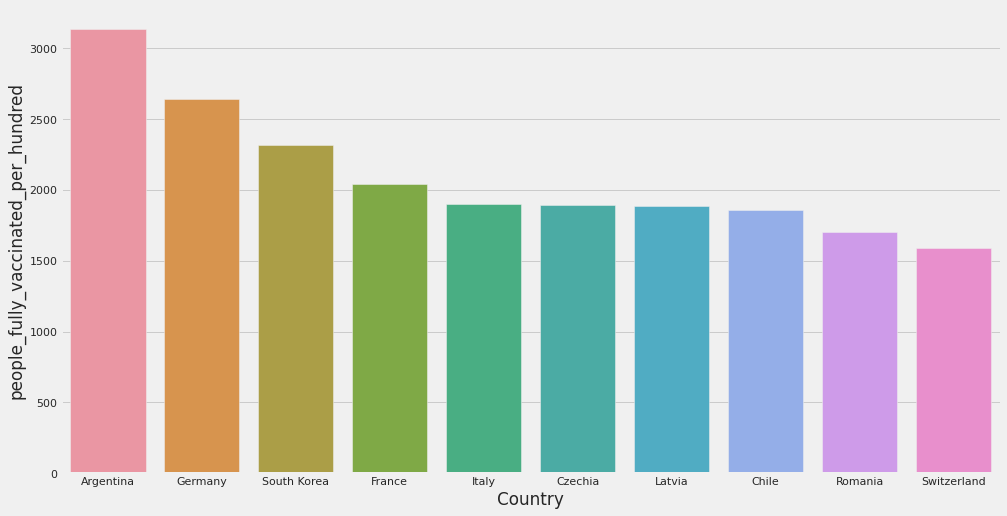
#### pfizer is the most used vaccine because it got approved in many countries very quickly.

## 2. Which countries has highest people fully vaccinated per hundred

c=final['location'].value\_counts().loc[lambda x:x>1500]  
c=pd.DataFrame(c)  
c.rename(columns={'location':"people\_fully\_vaccinated\_per\_hundred"},inplace=True)  
c[1:]

people\_fully\_vaccinated\_per\_hundred  
Argentina 3137  
Germany 2643  
South Korea 2316  
France 2045  
Italy 1900  
Czechia 1896  
Latvia 1888  
Chile 1859  
Romania 1701  
Switzerland 1592

plt.style.use("fivethirtyeight")  
plt.figure(figsize=(15,8))  
plt.xlabel("Country")  
plt.ylabel("people\_fully\_vaccinated\_per\_hundred")  
sns.barplot(y=c['people\_fully\_vaccinated\_per\_hundred'][1:],x=c.index[1:])  
plt.show()



## 3. what is the share of total vaccinationsof covid-19 in each country

df\_loc=final.groupby('location',as\_index=False)

fig = px.treemap(final, path=[px.Constant('total\_vaccinations'),'location'], values='total\_vaccinations',  
 hover\_data=['location'])  
fig.show()

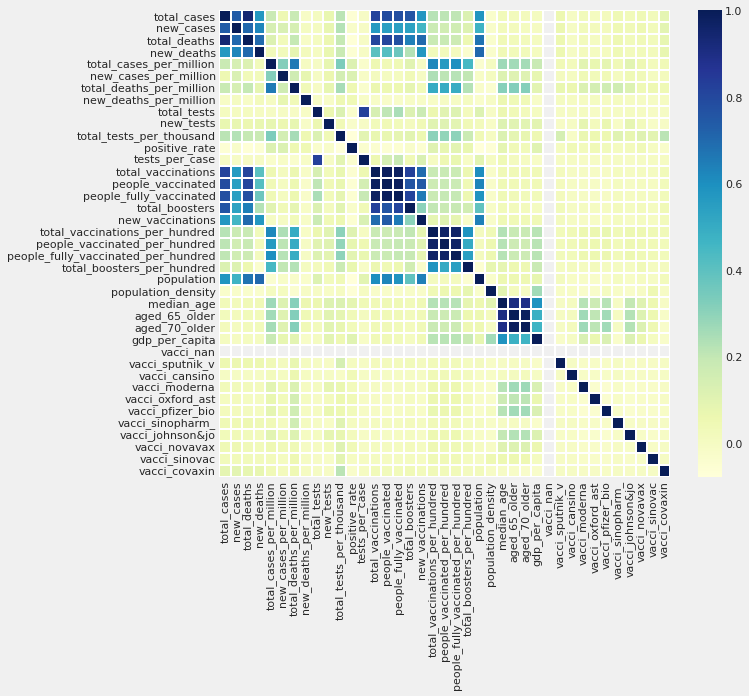
{"config":{"plotlyServerURL":"https://plot.ly"},"data":[{"branchvalues":"total","customdata":[["Afghanistan"],["Africa"],["Albania"],["Algeria"],["Andorra"],["Angola"],["Anguilla"],["Antigua and Barbuda"],["Argentina"],["Armenia"],["Aruba"],["Asia"],["Australia"],["Austria"],["Azerbaijan"],["Bahamas"],["Bahrain"],["Bangladesh"],["Barbados"],["Belarus"],["Belgium"],["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"],["Equatorial Guinea"],["Eritrea"],["Estonia"],["Eswatini"],["Ethiopia"],["Europe"],["European Union"],["Faeroe Islands"],["Falkland Islands"],["Fiji"],["Finland"],["France"],["French Polynesia"],["Gabon"],["Gambia"],["Georgia"],["Germany"],["Ghana"],["Gibraltar"],["Greece"],["Greenland"],["Grenada"],["Guam"],["Guatemala"],["Guernsey"],["Guinea"],["Guinea-Bissau"],["Guyana"],["Haiti"],["High income"],["Honduras"],["Hong Kong"],["Hungary"],["Iceland"],["India"],["Indonesia"],["International"],["Iran"],["Iraq"],["Ireland"],["Isle of Man"],["Israel"],["Italy"],["Jamaica"],["Japan"],["Jersey"],["Jordan"],["Kazakhstan"],["Kenya"],["Kiribati"],["Kosovo"],["Kuwait"],["Kyrgyzstan"],["Laos"],["Latvia"],["Lebanon"],["Lesotho"],["Liberia"],["Libya"],["Liechtenstein"],["Lithuania"],["Low income"],["Lower middle income"],["Luxembourg"],["Macao"],["Madagascar"],["Malawi"],["Malaysia"],["Maldives"],["Mali"],["Malta"],["Marshall Islands"],["Mauritania"],["Mauritius"],["Mexico"],["Micronesia (country)"],["Moldova"],["Monaco"],["Mongolia"],["Montenegro"],["Montserrat"],["Morocco"],["Mozambique"],["Myanmar"],["Namibia"],["Nauru"],["Nepal"],["Netherlands"],["New Caledonia"],["New Zealand"],["Nicaragua"],["Niger"],["Nigeria"],["Niue"],["North America"],["North Korea"],["North Macedonia"],["Northern Cyprus"],["Northern Mariana Islands"],["Norway"],["Oceania"],["Oman"],["Pakistan"],["Palau"],["Palestine"],["Panama"],["Papua New Guinea"],["Paraguay"],["Peru"],["Philippines"],["Pitcairn"],["Poland"],["Portugal"],["Puerto Rico"],["Qatar"],["Romania"],["Russia"],["Rwanda"],["Saint Helena"],["Saint Kitts and Nevis"],["Saint Lucia"],["Saint Pierre and Miquelon"],["Saint Vincent and the Grenadines"],["Samoa"],["San Marino"],["Sao Tome and Principe"],["Saudi Arabia"],["Senegal"],["Serbia"],["Seychelles"],["Sierra Leone"],["Singapore"],["Sint Maarten (Dutch part)"],["Slovakia"],["Slovenia"],["Solomon Islands"],["Somalia"],["South Africa"],["South America"],["South Korea"],["South Sudan"],["Spain"],["Sri Lanka"],["Sudan"],["Suriname"],["Sweden"],["Switzerland"],["Syria"],["Taiwan"],["Tajikistan"],["Tanzania"],["Thailand"],["Timor"],["Togo"],["Tokelau"],["Tonga"],["Trinidad and Tobago"],["Tunisia"],["Turkey"],["Turkmenistan"],["Turks and Caicos Islands"],["Tuvalu"],["Uganda"],["Ukraine"],["United Arab Emirates"],["United Kingdom"],["United States"],["United States Virgin Islands"],["Upper middle income"],["Uruguay"],["Uzbekistan"],["Vanuatu"],["Vatican"],["Venezuela"],["Vietnam"],["Wallis and Futuna"],["Western Sahara"],["World"],["Yemen"],["Zambia"],["Zimbabwe"],["(?)"]],"domain":{"x":[0,1],"y":[0,1]},"hovertemplate":"labels=%{label}<br>total\_vaccinations=%{value}<br>parent=%{parent}<br>id=%{id}<br>location=%{customdata[0]}<extra></extra>","ids":["total\_vaccinations/Afghanistan","total\_vaccinations/Africa","total\_vaccinations/Albania","total\_vaccinations/Algeria","total\_vaccinations/Andorra","total\_vaccinations/Angola","total\_vaccinations/Anguilla","total\_vaccinations/Antigua and Barbuda","total\_vaccinations/Argentina","total\_vaccinations/Armenia","total\_vaccinations/Aruba","total\_vaccinations/Asia","total\_vaccinations/Australia","total\_vaccinations/Austria","total\_vaccinations/Azerbaijan","total\_vaccinations/Bahamas","total\_vaccinations/Bahrain","total\_vaccinations/Bangladesh","total\_vaccinations/Barbados","total\_vaccinations/Belarus","total\_vaccinations/Belgium","total\_vaccinations/Belize","total\_vaccinations/Benin","total\_vaccinations/Bermuda","total\_vaccinations/Bhutan","total\_vaccinations/Bolivia","total\_vaccinations/Bonaire Sint Eustatius and Saba","total\_vaccinations/Bosnia and Herzegovina","total\_vaccinations/Botswana","total\_vaccinations/Brazil","total\_vaccinations/British Virgin Islands","total\_vaccinations/Brunei","total\_vaccinations/Bulgaria","total\_vaccinations/Burkina Faso","total\_vaccinations/Burundi","total\_vaccinations/Cambodia","total\_vaccinations/Cameroon","total\_vaccinations/Canada","total\_vaccinations/Cape Verde","total\_vaccinations/Cayman Islands","total\_vaccinations/Central African Republic","total\_vaccinations/Chad","total\_vaccinations/Chile","total\_vaccinations/China","total\_vaccinations/Colombia","total\_vaccinations/Comoros","total\_vaccinations/Congo","total\_vaccinations/Cook Islands","total\_vaccinations/Costa Rica","total\_vaccinations/Cote d'Ivoire","total\_vaccinations/Croatia","total\_vaccinations/Cuba","total\_vaccinations/Curacao","total\_vaccinations/Cyprus","total\_vaccinations/Czechia","total\_vaccinations/Democratic Republic of Congo","total\_vaccinations/Denmark","total\_vaccinations/Djibouti","total\_vaccinations/Dominica","total\_vaccinations/Dominican Republic","total\_vaccinations/Ecuador","total\_vaccinations/Egypt","total\_vaccinations/El Salvador","total\_vaccinations/Equatorial Guinea","total\_vaccinations/Eritrea","total\_vaccinations/Estonia","total\_vaccinations/Eswatini","total\_vaccinations/Ethiopia","total\_vaccinations/Europe","total\_vaccinations/European Union","total\_vaccinations/Faeroe Islands","total\_vaccinations/Falkland Islands","total\_vaccinations/Fiji","total\_vaccinations/Finland","total\_vaccinations/France","total\_vaccinations/French Polynesia","total\_vaccinations/Gabon","total\_vaccinations/Gambia","total\_vaccinations/Georgia","total\_vaccinations/Germany","total\_vaccinations/Ghana","total\_vaccinations/Gibraltar","total\_vaccinations/Greece","total\_vaccinations/Greenland","total\_vaccinations/Grenada","total\_vaccinations/Guam","total\_vaccinations/Guatemala","total\_vaccinations/Guernsey","total\_vaccinations/Guinea","total\_vaccinations/Guinea-Bissau","total\_vaccinations/Guyana","total\_vaccinations/Haiti","total\_vaccinations/High income","total\_vaccinations/Honduras","total\_vaccinations/Hong Kong","total\_vaccinations/Hungary","total\_vaccinations/Iceland","total\_vaccinations/India","total\_vaccinations/Indonesia","total\_vaccinations/International","total\_vaccinations/Iran","total\_vaccinations/Iraq","total\_vaccinations/Ireland","total\_vaccinations/Isle of Man","total\_vaccinations/Israel","total\_vaccinations/Italy","total\_vaccinations/Jamaica","total\_vaccinations/Japan","total\_vaccinations/Jersey","total\_vaccinations/Jordan","total\_vaccinations/Kazakhstan","total\_vaccinations/Kenya","total\_vaccinations/Kiribati","total\_vaccinations/Kosovo","total\_vaccinations/Kuwait","total\_vaccinations/Kyrgyzstan","total\_vaccinations/Laos","total\_vaccinations/Latvia","total\_vaccinations/Lebanon","total\_vaccinations/Lesotho","total\_vaccinations/Liberia","total\_vaccinations/Libya","total\_vaccinations/Liechtenstein","total\_vaccinations/Lithuania","total\_vaccinations/Low income","total\_vaccinations/Lower middle income","total\_vaccinations/Luxembourg","total\_vaccinations/Macao","total\_vaccinations/Madagascar","total\_vaccinations/Malawi","total\_vaccinations/Malaysia","total\_vaccinations/Maldives","total\_vaccinations/Mali","total\_vaccinations/Malta","total\_vaccinations/Marshall Islands","total\_vaccinations/Mauritania","total\_vaccinations/Mauritius","total\_vaccinations/Mexico","total\_vaccinations/Micronesia (country)","total\_vaccinations/Moldova","total\_vaccinations/Monaco","total\_vaccinations/Mongolia","total\_vaccinations/Montenegro","total\_vaccinations/Montserrat","total\_vaccinations/Morocco","total\_vaccinations/Mozambique","total\_vaccinations/Myanmar","total\_vaccinations/Namibia","total\_vaccinations/Nauru","total\_vaccinations/Nepal","total\_vaccinations/Netherlands","total\_vaccinations/New Caledonia","total\_vaccinations/New Zealand","total\_vaccinations/Nicaragua","total\_vaccinations/Niger","total\_vaccinations/Nigeria","total\_vaccinations/Niue","total\_vaccinations/North America","total\_vaccinations/North Korea","total\_vaccinations/North Macedonia","total\_vaccinations/Northern Cyprus","total\_vaccinations/Northern Mariana Islands","total\_vaccinations/Norway","total\_vaccinations/Oceania","total\_vaccinations/Oman","total\_vaccinations/Pakistan","total\_vaccinations/Palau","total\_vaccinations/Palestine","total\_vaccinations/Panama","total\_vaccinations/Papua New Guinea","total\_vaccinations/Paraguay","total\_vaccinations/Peru","total\_vaccinations/Philippines","total\_vaccinations/Pitcairn","total\_vaccinations/Poland","total\_vaccinations/Portugal","total\_vaccinations/Puerto Rico","total\_vaccinations/Qatar","total\_vaccinations/Romania","total\_vaccinations/Russia","total\_vaccinations/Rwanda","total\_vaccinations/Saint Helena","total\_vaccinations/Saint Kitts and Nevis","total\_vaccinations/Saint Lucia","total\_vaccinations/Saint Pierre and Miquelon","total\_vaccinations/Saint Vincent and the Grenadines","total\_vaccinations/Samoa","total\_vaccinations/San Marino","total\_vaccinations/Sao Tome and Principe","total\_vaccinations/Saudi Arabia","total\_vaccinations/Senegal","total\_vaccinations/Serbia","total\_vaccinations/Seychelles","total\_vaccinations/Sierra Leone","total\_vaccinations/Singapore","total\_vaccinations/Sint Maarten (Dutch part)","total\_vaccinations/Slovakia","total\_vaccinations/Slovenia","total\_vaccinations/Solomon Islands","total\_vaccinations/Somalia","total\_vaccinations/South Africa","total\_vaccinations/South America","total\_vaccinations/South Korea","total\_vaccinations/South Sudan","total\_vaccinations/Spain","total\_vaccinations/Sri Lanka","total\_vaccinations/Sudan","total\_vaccinations/Suriname","total\_vaccinations/Sweden","total\_vaccinations/Switzerland","total\_vaccinations/Syria","total\_vaccinations/Taiwan","total\_vaccinations/Tajikistan","total\_vaccinations/Tanzania","total\_vaccinations/Thailand","total\_vaccinations/Timor","total\_vaccinations/Togo","total\_vaccinations/Tokelau","total\_vaccinations/Tonga","total\_vaccinations/Trinidad and Tobago","total\_vaccinations/Tunisia","total\_vaccinations/Turkey","total\_vaccinations/Turkmenistan","total\_vaccinations/Turks and Caicos Islands","total\_vaccinations/Tuvalu","total\_vaccinations/Uganda","total\_vaccinations/Ukraine","total\_vaccinations/United Arab Emirates","total\_vaccinations/United Kingdom","total\_vaccinations/United States","total\_vaccinations/United States Virgin Islands","total\_vaccinations/Upper middle income","total\_vaccinations/Uruguay","total\_vaccinations/Uzbekistan","total\_vaccinations/Vanuatu","total\_vaccinations/Vatican","total\_vaccinations/Venezuela","total\_vaccinations/Vietnam","total\_vaccinations/Wallis and Futuna","total\_vaccinations/Western Sahara","total\_vaccinations/World","total\_vaccinations/Yemen","total\_vaccinations/Zambia","total\_vaccinations/Zimbabwe","total\_vaccinations"],"labels":["Afghanistan","Africa","Albania","Algeria","Andorra","Angola","Anguilla","Antigua and Barbuda","Argentina","Armenia","Aruba","Asia","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","Equatorial Guinea","Eritrea","Estonia","Eswatini","Ethiopia","Europe","European Union","Faeroe Islands","Falkland Islands","Fiji","Finland","France","French Polynesia","Gabon","Gambia","Georgia","Germany","Ghana","Gibraltar","Greece","Greenland","Grenada","Guam","Guatemala","Guernsey","Guinea","Guinea-Bissau","Guyana","Haiti","High income","Honduras","Hong Kong","Hungary","Iceland","India","Indonesia","International","Iran","Iraq","Ireland","Isle of Man","Israel","Italy","Jamaica","Japan","Jersey","Jordan","Kazakhstan","Kenya","Kiribati","Kosovo","Kuwait","Kyrgyzstan","Laos","Latvia","Lebanon","Lesotho","Liberia","Libya","Liechtenstein","Lithuania","Low income","Lower middle income","Luxembourg","Macao","Madagascar","Malawi","Malaysia","Maldives","Mali","Malta","Marshall Islands","Mauritania","Mauritius","Mexico","Micronesia (country)","Moldova","Monaco","Mongolia","Montenegro","Montserrat","Morocco","Mozambique","Myanmar","Namibia","Nauru","Nepal","Netherlands","New Caledonia","New Zealand","Nicaragua","Niger","Nigeria","Niue","North America","North Korea","North Macedonia","Northern Cyprus","Northern Mariana Islands","Norway","Oceania","Oman","Pakistan","Palau","Palestine","Panama","Papua New Guinea","Paraguay","Peru","Philippines","Pitcairn","Poland","Portugal","Puerto Rico","Qatar","Romania","Russia","Rwanda","Saint Helena","Saint Kitts and Nevis","Saint Lucia","Saint Pierre and Miquelon","Saint Vincent and the Grenadines","Samoa","San Marino","Sao Tome and Principe","Saudi Arabia","Senegal","Serbia","Seychelles","Sierra Leone","Singapore","Sint Maarten (Dutch part)","Slovakia","Slovenia","Solomon Islands","Somalia","South Africa","South America","South Korea","South Sudan","Spain","Sri Lanka","Sudan","Suriname","Sweden","Switzerland","Syria","Taiwan","Tajikistan","Tanzania","Thailand","Timor","Togo","Tokelau","Tonga","Trinidad and Tobago","Tunisia","Turkey","Turkmenistan","Turks and Caicos Islands","Tuvalu","Uganda","Ukraine","United Arab Emirates","United Kingdom","United States","United States Virgin Islands","Upper middle income","Uruguay","Uzbekistan","Vanuatu","Vatican","Venezuela","Vietnam","Wallis and Futuna","Western Sahara","World","Yemen","Zambia","Zimbabwe","total\_vaccinations"],"name":"","parents":["total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations","total\_vaccinations",""],"type":"treemap","values":[1458879336,96308723410,750627381,3887617098,43534820,3362221973,8160197,40883468,80928476518,399654024,71963982,1992140290909,13471890825,5414416056,3664909134,82530472,1197435656,43142928504,104923707,2401694156,7435550996,133849937,508150497,42144076,483679703,3133866193,7954368,324955048,399120663,109817394172,18369686,238782495,1258437728,421444996,1909840,9317192237,320986832,24430898843,176754649,48807830,184090427,199355738,32006812619,933204782700,18739747345,166398648,204897094,7444097,2518696548,2002857096,1591753726,9969621452,90220862,498650140,10688739199,205893165,3894032629,50622531,22016432,4881878136,12920037165,15548246815,3030986678,155220454,153755,612393732,120616000,3604862640,371656556936,1666923348868,23394883,176941,393312263,3374929681,86636423711,113618633,119702179,128390733,712119695,159278965188,2427028304,37771112,5863671688,21456109,26198608,0,3568634960,49006608,1134074301,93552459,251306142,51115108,716019965677,2691754304,4301164340,5269683162,196263667,451596393387,85529138518,0,32725917344,3940789598,3207998797,51632111,6926153235,91799747573,363916188,67652284689,73739997,2839121200,3838794345,3129636348,28763924,489461246,2396713223,693646459,2511307057,2115157648,1366889183,187376015,289504939,785156282,20431039,1410397323,24718034182,871217789895,363797321,329386663,320582048,507173051,17928256793,308876780,356395722,394967436,0,516683689,642187221,46608739298,1920651,598618150,13699167,1764577473,197486690,1307427,16572000126,4459298793,10281477533,199038555,5059191,8470069820,9831096753,113796791,2551639404,1933949661,498258868,5776074591,830741,299930352842,0,532110713,89182017,0,3307890419,17060562675,1866607254,49383649100,23435412,986814488,2130098761,130777632,2064845098,30771220495,30703137284,6983,16824689477,6871374531,0,2013092380,10726556854,44506548846,3754514115,467195,14397378,36472667,7072080,21918946,83308845,21604760,47410774,17834813344,682265407,2980481525,74140157,409404322,4025229381,19454343,2237448793,936064001,76112806,516808900,8226298375,212131448464,85740636145,105000898,28936935597,10077148546,1275621089,148719360,6222309391,8888453801,689197244,10698860857,2020355250,1014452116,29743619604,352558821,669681031,163930,36737610,425656739,3345748228,43471320844,2648991746,25102561,3512869,3196074140,8864417627,9035817769,46157734063,382948432045,1219743000,1377001839853,3743838039,10421451921,39095633,0,6367772233,41213161046,4505461,0,2988797081981,221039344,651866598,2322213776,14182146723295]}],"layout":{"legend":{"tracegroupgap":0},"margin":{"t":60},"template":{"data":{"bar":[{"error\_x":{"color":"#2a3f5f"},"error\_y":{"color":"#2a3f5f"},"marker":{"line":{"color":"#E5ECF6","width":0.5},"pattern":{"fillmode":"overlay","size":10,"solidity":0.2}},"type":"bar"}],"barpolar":[{"marker":{"line":{"color":"#E5ECF6","width":0.5},"pattern":{"fillmode":"overlay","size":10,"solidity":0.2}},"type":"barpolar"}],"carpet":[{"aaxis":{"endlinecolor":"#2a3f5f","gridcolor":"white","linecolor":"white","minorgridcolor":"white","startlinecolor":"#2a3f5f"},"baxis":{"endlinecolor":"#2a3f5f","gridcolor":"white","linecolor":"white","minorgridcolor":"white","startlinecolor":"#2a3f5f"},"type":"carpet"}],"choropleth":[{"colorbar":{"outlinewidth":0,"ticks":""},"type":"choropleth"}],"contour":[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":[[0,"#0d0887"],[0.1111111111111111,"#46039f"],[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],[0.8888888888888888,"#fdca26"],[1,"#f0f921"]],"type":"contour"}],"contourcarpet":[{"colorbar":{"outlinewidth":0,"ticks":""},"type":"contourcarpet"}],"heatmap":[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":[[0,"#0d0887"],[0.1111111111111111,"#46039f"],[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],[0.8888888888888888,"#fdca26"],[1,"#f0f921"]],"type":"heatmap"}],"heatmapgl":[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":[[0,"#0d0887"],[0.1111111111111111,"#46039f"],[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],[0.8888888888888888,"#fdca26"],[1,"#f0f921"]],"type":"heatmapgl"}],"histogram":[{"marker":{"pattern":{"fillmode":"overlay","size":10,"solidity":0.2}},"type":"histogram"}],"histogram2d":[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":[[0,"#0d0887"],[0.1111111111111111,"#46039f"],[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],[0.8888888888888888,"#fdca26"],[1,"#f0f921"]],"type":"histogram2d"}],"histogram2dcontour":[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":[[0,"#0d0887"],[0.1111111111111111,"#46039f"],[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],[0.8888888888888888,"#fdca26"],[1,"#f0f921"]],"type":"histogram2dcontour"}],"mesh3d":[{"colorbar":{"outlinewidth":0,"ticks":""},"type":"mesh3d"}],"parcoords":[{"line":{"colorbar":{"outlinewidth":0,"ticks":""}},"type":"parcoords"}],"pie":[{"automargin":true,"type":"pie"}],"scatter":[{"fillpattern":{"fillmode":"overlay","size":10,"solidity":0.2},"type":"scatter"}],"scatter3d":[{"line":{"colorbar":{"outlinewidth":0,"ticks":""}},"marker":{"colorbar":{"outlinewidth":0,"ticks":""}},"type":"scatter3d"}],"scattercarpet":[{"marker":{"colorbar":{"outlinewidth":0,"ticks":""}},"type":"scattercarpet"}],"scattergeo":[{"marker":{"colorbar":{"outlinewidth":0,"ticks":""}},"type":"scattergeo"}],"scattergl":[{"marker":{"colorbar":{"outlinewidth":0,"ticks":""}},"type":"scattergl"}],"scattermapbox":[{"marker":{"colorbar":{"outlinewidth":0,"ticks":""}},"type":"scattermapbox"}],"scatterpolar":[{"marker":{"colorbar":{"outlinewidth":0,"ticks":""}},"type":"scatterpolar"}],"scatterpolargl":[{"marker":{"colorbar":{"outlinewidth":0,"ticks":""}},"type":"scatterpolargl"}],"scatterternary":[{"marker":{"colorbar":{"outlinewidth":0,"ticks":""}},"type":"scatterternary"}],"surface":[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":[[0,"#0d0887"],[0.1111111111111111,"#46039f"],[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],[0.8888888888888888,"#fdca26"],[1,"#f0f921"]],"type":"surface"}],"table":[{"cells":{"fill":{"color":"#EBF0F8"},"line":{"color":"white"}},"header":{"fill":{"color":"#C8D4E3"},"line":{"color":"white"}},"type":"table"}]},"layout":{"annotationdefaults":{"arrowcolor":"#2a3f5f","arrowhead":0,"arrowwidth":1},"autotypenumbers":"strict","coloraxis":{"colorbar":{"outlinewidth":0,"ticks":""}},"colorscale":{"diverging":[[0,"#8e0152"],[0.1,"#c51b7d"],[0.2,"#de77ae"],[0.3,"#f1b6da"],[0.4,"#fde0ef"],[0.5,"#f7f7f7"],[0.6,"#e6f5d0"],[0.7,"#b8e186"],[0.8,"#7fbc41"],[0.9,"#4d9221"],[1,"#276419"]],"sequential":[[0,"#0d0887"],[0.1111111111111111,"#46039f"],[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],[0.8888888888888888,"#fdca26"],[1,"#f0f921"]],"sequentialminus":[[0,"#0d0887"],[0.1111111111111111,"#46039f"],[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],[0.8888888888888888,"#fdca26"],[1,"#f0f921"]]},"colorway":["#636efa","#EF553B","#00cc96","#ab63fa","#FFA15A","#19d3f3","#FF6692","#B6E880","#FF97FF","#FECB52"],"font":{"color":"#2a3f5f"},"geo":{"bgcolor":"white","lakecolor":"white","landcolor":"#E5ECF6","showlakes":true,"showland":true,"subunitcolor":"white"},"hoverlabel":{"align":"left"},"hovermode":"closest","mapbox":{"style":"light"},"paper\_bgcolor":"white","plot\_bgcolor":"#E5ECF6","polar":{"angularaxis":{"gridcolor":"white","linecolor":"white","ticks":""},"bgcolor":"#E5ECF6","radialaxis":{"gridcolor":"white","linecolor":"white","ticks":""}},"scene":{"xaxis":{"backgroundcolor":"#E5ECF6","gridcolor":"white","gridwidth":2,"linecolor":"white","showbackground":true,"ticks":"","zerolinecolor":"white"},"yaxis":{"backgroundcolor":"#E5ECF6","gridcolor":"white","gridwidth":2,"linecolor":"white","showbackground":true,"ticks":"","zerolinecolor":"white"},"zaxis":{"backgroundcolor":"#E5ECF6","gridcolor":"white","gridwidth":2,"linecolor":"white","showbackground":true,"ticks":"","zerolinecolor":"white"}},"shapedefaults":{"line":{"color":"#2a3f5f"}},"ternary":{"aaxis":{"gridcolor":"white","linecolor":"white","ticks":""},"baxis":{"gridcolor":"white","linecolor":"white","ticks":""},"bgcolor":"#E5ECF6","caxis":{"gridcolor":"white","linecolor":"white","ticks":""}},"title":{"x":5.0e-2},"xaxis":{"automargin":true,"gridcolor":"white","linecolor":"white","ticks":"","title":{"standoff":15},"zerolinecolor":"white","zerolinewidth":2},"yaxis":{"automargin":true,"gridcolor":"white","linecolor":"white","ticks":"","title":{"standoff":15},"zerolinecolor":"white","zerolinewidth":2}}}}}

### 8 countries in top 10 countries with people\_fully\_vaccinatd\_per\_hundred belong to europe

# 4. Data Analysis and Visualisation

corrmat = final.corr()  
   
f, ax = plt.subplots(figsize =(9, 8))  
sns.heatmap(corrmat, ax = ax, cmap ="YlGnBu", linewidths = 0.1)

<AxesSubplot:>

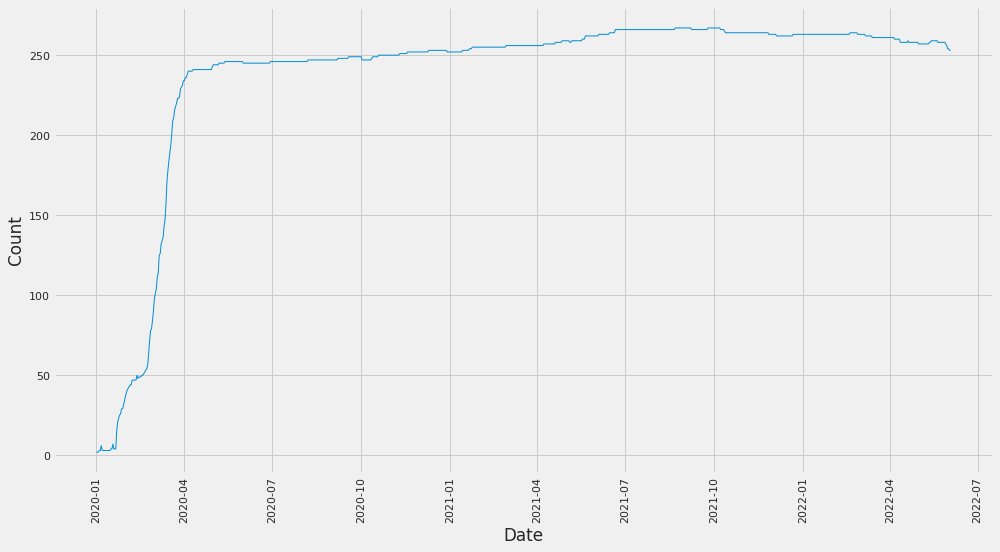


### Correlation describes how two attributes are related and indicates that as one variable changes in value, the other variable tends to change in a specific direction

time\_series = pd.DataFrame(final['date'].value\_counts().reset\_index())  
time\_series.columns = ['date', 'count']

time\_series= time\_series.sort\_values('date', ascending=True)  
plt.style.use("fivethirtyeight")  
plt.figure(figsize=(15,8))  
plt.plot(time\_series['date'], time\_series['count'],linewidth=1)  
plt.xticks(rotation='vertical')  
plt.xlabel("Date")  
plt.ylabel("Count")

Text(0, 0.5, 'Count')



a=final['median\_age'].values

d=final['total\_boosters'].values

X=final[['date','total\_vaccinations\_per\_hundred']]

X

date total\_vaccinations\_per\_hundred  
0 2020-02-24 0.00  
1 2020-02-25 0.00  
2 2020-02-26 0.00  
3 2020-02-27 0.00  
4 2020-02-28 0.00  
... ... ...  
210822 2022-05-30 77.95  
210823 2022-05-31 78.00  
210824 2022-06-01 78.02  
210825 2022-06-02 NaN  
210826 2022-06-03 NaN  
  
[210827 rows x 2 columns]

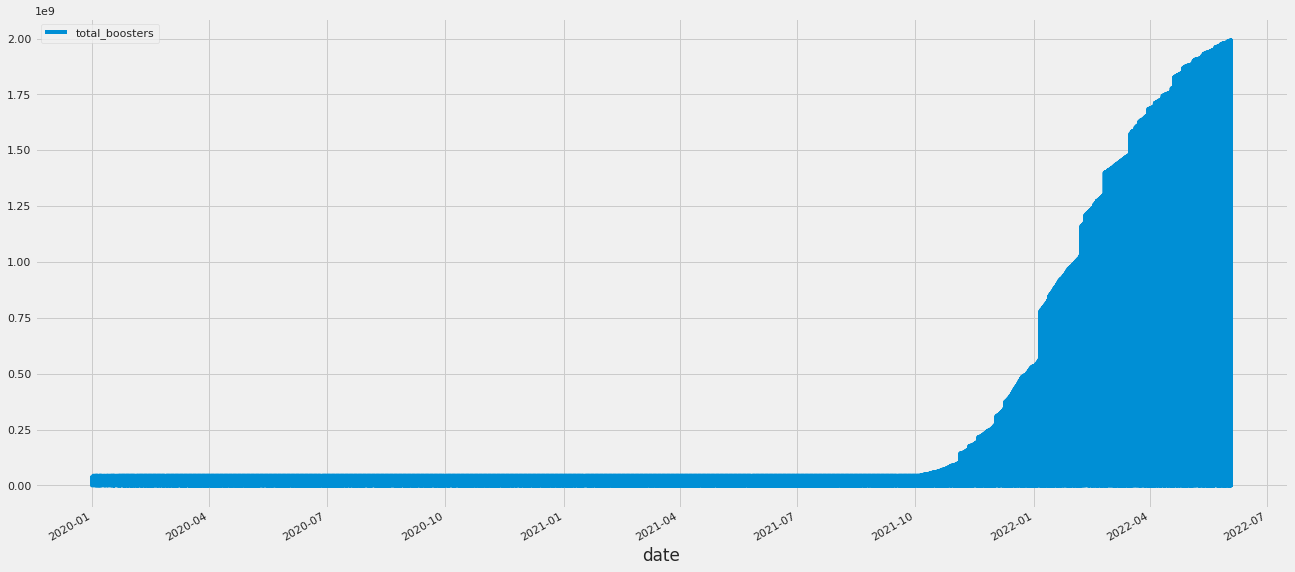
df=X

df

date total\_vaccinations\_per\_hundred  
0 2020-02-24 0.00  
1 2020-02-25 0.00  
2 2020-02-26 0.00  
3 2020-02-27 0.00  
4 2020-02-28 0.00  
... ... ...  
210822 2022-05-30 77.95  
210823 2022-05-31 78.00  
210824 2022-06-01 78.02  
210825 2022-06-02 NaN  
210826 2022-06-03 NaN  
  
[210827 rows x 2 columns]

final.plot(x='date',y='total\_boosters')

<AxesSubplot:xlabel='date'>



### Booster doses drive started around november 2021.

final[['date','total\_vaccinations\_per\_hundred']]

date total\_vaccinations\_per\_hundred  
0 2020-02-24 0.00  
1 2020-02-25 0.00  
2 2020-02-26 0.00  
3 2020-02-27 0.00  
4 2020-02-28 0.00  
... ... ...  
210822 2022-05-30 77.95  
210823 2022-05-31 78.00  
210824 2022-06-01 78.02  
210825 2022-06-02 NaN  
210826 2022-06-03 NaN  
  
[210827 rows x 2 columns]

df['total\_vaccinations']=final['total\_vaccinations']

df[['total\_vaccinations','total\_vaccinations\_per\_hundred']]=final[['total\_vaccinations','total\_vaccinations\_per\_hundred']]

df

date total\_vaccinations\_per\_hundred total\_vaccinations  
0 2020-02-24 0.00 0.0  
1 2020-02-25 0.00 0.0  
2 2020-02-26 0.00 0.0  
3 2020-02-27 0.00 0.0  
4 2020-02-28 0.00 0.0  
... ... ... ...  
210822 2022-05-30 77.95 11763851.0  
210823 2022-05-31 78.00 11771952.0  
210824 2022-06-01 78.02 11775542.0  
210825 2022-06-02 NaN NaN  
210826 2022-06-03 NaN NaN  
  
[210827 rows x 3 columns]

temp=pd.DataFrame()

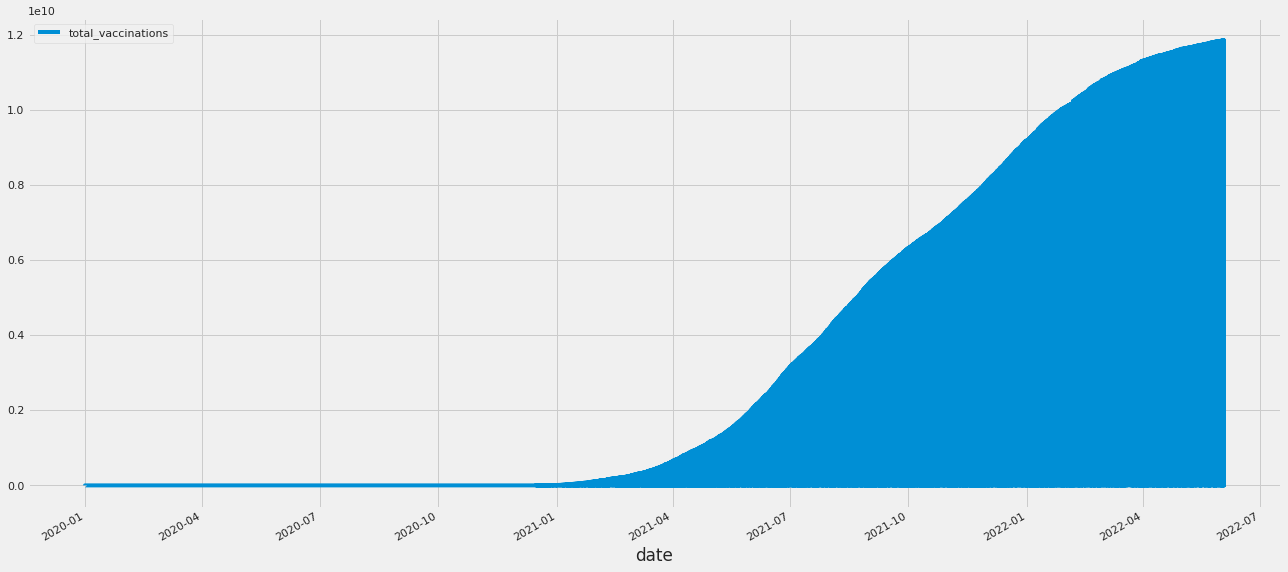
temp[['date','total\_vaccinations','total\_vaccinations\_per\_hundred']]=final[['date','total\_vaccinations','total\_vaccinations\_per\_hundred']]

temp

date total\_vaccinations total\_vaccinations\_per\_hundred  
0 2020-02-24 0.0 0.00  
1 2020-02-25 0.0 0.00  
2 2020-02-26 0.0 0.00  
3 2020-02-27 0.0 0.00  
4 2020-02-28 0.0 0.00  
... ... ... ...  
210822 2022-05-30 11763851.0 77.95  
210823 2022-05-31 11771952.0 78.00  
210824 2022-06-01 11775542.0 78.02  
210825 2022-06-02 NaN NaN  
210826 2022-06-03 NaN NaN  
  
[210827 rows x 3 columns]

final.plot(x='date',y='total\_vaccinations')

<AxesSubplot:xlabel='date'>



### there is a linear increase of total vaccinations

df

date total\_vaccinations\_per\_hundred total\_vaccinations  
0 2020-02-24 0.00 0.0  
1 2020-02-25 0.00 0.0  
2 2020-02-26 0.00 0.0  
3 2020-02-27 0.00 0.0  
4 2020-02-28 0.00 0.0  
... ... ... ...  
210822 2022-05-30 77.95 11763851.0  
210823 2022-05-31 78.00 11771952.0  
210824 2022-06-01 78.02 11775542.0  
210825 2022-06-02 NaN NaN  
210826 2022-06-03 NaN NaN  
  
[210827 rows x 3 columns]

temp=temp.set\_index('date')

temp

total\_vaccinations total\_vaccinations\_per\_hundred  
date   
2020-02-24 0.0 0.00  
2020-02-25 0.0 0.00  
2020-02-26 0.0 0.00  
2020-02-27 0.0 0.00  
2020-02-28 0.0 0.00  
... ... ...  
2022-05-30 11763851.0 77.95  
2022-05-31 11771952.0 78.00  
2022-06-01 11775542.0 78.02  
2022-06-02 NaN NaN  
2022-06-03 NaN NaN  
  
[210827 rows x 2 columns]

%matplotlib inline  
from matplotlib.pylab import rcParams  
rcParams['figure.figsize'] = 10, 6

df2=temp

df2

total\_vaccinations total\_vaccinations\_per\_hundred  
date   
2020-02-24 0.0 0.00  
2020-02-25 0.0 0.00  
2020-02-26 0.0 0.00  
2020-02-27 0.0 0.00  
2020-02-28 0.0 0.00  
... ... ...  
2022-05-30 11763851.0 77.95  
2022-05-31 11771952.0 78.00  
2022-06-01 11775542.0 78.02  
2022-06-02 NaN NaN  
2022-06-03 NaN NaN  
  
[210827 rows x 2 columns]

df\_loc=final[['date' ]]

df\_loc=final.groupby('location',as\_index=False).sum()

df\_loc

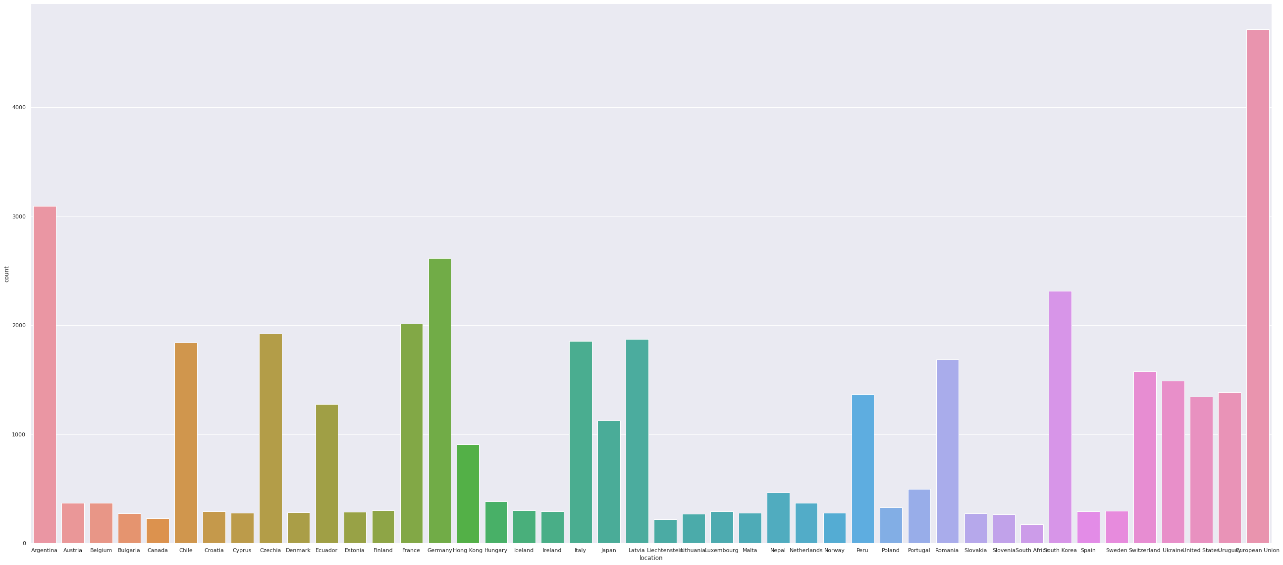
location total\_cases new\_cases total\_deaths new\_deaths \  
0 Afghanistan 7.463695e+07 180614.0 3.243220e+06 7736.0   
1 Africa 4.254702e+09 11883114.0 1.021116e+08 253979.0   
2 Albania 9.560575e+07 276336.0 1.488508e+06 3512.0   
3 Algeria 1.074541e+08 265889.0 2.982005e+06 6891.0   
4 Andorra 1.161060e+07 43067.0 8.325300e+04 175.0   
.. ... ... ... ... ...   
239 Western Sahara 5.570000e+02 0.0 1.700000e+01 0.0   
240 World 1.438550e+11 530350318.0 2.563715e+09 6255973.0   
241 Yemen 4.555049e+06 11833.0 9.140290e+05 2189.0   
242 Zambia 1.029198e+08 322207.0 1.509855e+06 4003.0   
243 Zimbabwe 6.831400e+07 253242.0 1.896900e+06 5512.0   
  
 total\_cases\_per\_million new\_cases\_per\_million total\_deaths\_per\_million \  
0 1.873632e+06 4534.008 81415.455   
1 3.097739e+06 8651.798 74344.809   
2 3.327809e+07 96185.992 518114.217   
3 2.408386e+06 5959.407 66836.197   
4 1.500970e+08 556752.077 1076259.852   
.. ... ... ...   
239 7.100000e-02 0.000 0.002   
240 1.826738e+07 67346.374 325552.501   
241 1.493917e+05 388.117 29977.389   
242 5.439545e+06 17029.407 79799.290   
243 4.526452e+06 16779.698 125687.682   
  
 new\_deaths\_per\_million total\_tests ... vacci\_sputnik\_v \  
0 194.161 7.114556e+08 ... 0   
1 184.896 6.736000e+03 ... 0   
2 1222.373 5.639359e+08 ... 0   
3 154.421 1.751062e+08 ... 0   
4 2262.367 2.376456e+08 ... 0   
.. ... ... ... ...   
239 0.000 3.943000e+03 ... 0   
240 794.399 2.847675e+08 ... 0   
241 71.838 2.560731e+08 ... 0   
242 211.613 1.220336e+09 ... 0   
243 365.233 6.508981e+08 ... 0   
  
 vacci\_cansino vacci\_moderna vacci\_oxford\_ast vacci\_pfizer\_bio \  
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   
.. ... ... ... ...   
239 0 0 0 0   
240 0 0 0 0   
241 0 0 0 0   
242 0 0 0 0   
243 0 0 0 0   
  
 vacci\_sinopharm\_ vacci\_johnson&jo vacci\_novavax vacci\_sinovac \  
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   
.. ... ... ... ...   
239 0 0 0 0   
240 0 0 0 0   
241 0 0 0 0   
242 0 0 0 0   
243 0 0 0 0   
  
 vacci\_covaxin   
0 0   
1 0   
2 0   
3 0   
4 0   
.. ...   
239 0   
240 0   
241 0   
242 0   
243 0   
  
[244 rows x 40 columns]

fig = px.treemap(df\_loc, path=[px.Constant('gdp\_per\_capita'),'location'], values='gdp\_per\_capita',  
 hover\_data=['location'])  
fig.show()

{"config":{"plotlyServerURL":"https://plot.ly"},"data":[{"branchvalues":"total","customdata":[["Afghanistan"],["Africa"],["Albania"],["Algeria"],["Andorra"],["Angola"],["Anguilla"],["Antigua and Barbuda"],["Argentina"],["Armenia"],["Aruba"],["Asia"],["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"],["Equatorial Guinea"],["Eritrea"],["Estonia"],["Eswatini"],["Ethiopia"],["Europe"],["European Union"],["Faeroe Islands"],["Falkland Islands"],["Fiji"],["Finland"],["France"],["French Polynesia"],["Gabon"],["Gambia"],["Georgia"],["Germany"],["Ghana"],["Gibraltar"],["Greece"],["Greenland"],["Grenada"],["Guam"],["Guatemala"],["Guernsey"],["Guinea"],["Guinea-Bissau"],["Guyana"],["Haiti"],["High income"],["Honduras"],["Hong Kong"],["Hungary"],["Iceland"],["India"],["Indonesia"],["International"],["Iran"],["Iraq"],["Ireland"],["Isle of Man"],["Israel"],["Italy"],["Jamaica"],["Japan"],["Jersey"],["Jordan"],["Kazakhstan"],["Kenya"],["Kiribati"],["Kosovo"],["Kuwait"],["Kyrgyzstan"],["Laos"],["Latvia"],["Lebanon"],["Lesotho"],["Liberia"],["Libya"],["Liechtenstein"],["Lithuania"],["Low income"],["Lower middle income"],["Luxembourg"],["Macao"],["Madagascar"],["Malawi"],["Malaysia"],["Maldives"],["Mali"],["Malta"],["Marshall Islands"],["Mauritania"],["Mauritius"],["Mexico"],["Micronesia (country)"],["Moldova"],["Monaco"],["Mongolia"],["Montenegro"],["Montserrat"],["Morocco"],["Mozambique"],["Myanmar"],["Namibia"],["Nauru"],["Nepal"],["Netherlands"],["New Caledonia"],["New Zealand"],["Nicaragua"],["Niger"],["Nigeria"],["Niue"],["North America"],["North Korea"],["North Macedonia"],["Northern Cyprus"],["Northern Mariana Islands"],["Norway"],["Oceania"],["Oman"],["Pakistan"],["Palau"],["Palestine"],["Panama"],["Papua New Guinea"],["Paraguay"],["Peru"],["Philippines"],["Pitcairn"],["Poland"],["Portugal"],["Puerto Rico"],["Qatar"],["Romania"],["Russia"],["Rwanda"],["Saint Helena"],["Saint Kitts and Nevis"],["Saint Lucia"],["Saint Pierre and Miquelon"],["Saint Vincent and the Grenadines"],["Samoa"],["San Marino"],["Sao Tome and Principe"],["Saudi Arabia"],["Senegal"],["Serbia"],["Seychelles"],["Sierra Leone"],["Singapore"],["Sint Maarten (Dutch part)"],["Slovakia"],["Slovenia"],["Solomon Islands"],["Somalia"],["South Africa"],["South America"],["South Korea"],["South Sudan"],["Spain"],["Sri Lanka"],["Sudan"],["Suriname"],["Sweden"],["Switzerland"],["Syria"],["Taiwan"],["Tajikistan"],["Tanzania"],["Thailand"],["Timor"],["Togo"],["Tokelau"],["Tonga"],["Trinidad and Tobago"],["Tunisia"],["Turkey"],["Turkmenistan"],["Turks and Caicos Islands"],["Tuvalu"],["Uganda"],["Ukraine"],["United Arab Emirates"],["United Kingdom"],["United States"],["United States Virgin Islands"],["Upper middle income"],["Uruguay"],["Uzbekistan"],["Vanuatu"],["Vatican"],["Venezuela"],["Vietnam"],["Wallis and Futuna"],["Western Sahara"],["World"],["Yemen"],["Zambia"],["Zimbabwe"],["(?)"]],"domain":{"x":[0,1],"y":[0,1]},"hovertemplate":"labels=%{label}<br>gdp\_per\_capita=%{value}<br>parent=%{parent}<br>id=%{id}<br>location=%{customdata[0]}<extra></extra>","ids":["gdp\_per\_capita/Afghanistan","gdp\_per\_capita/Africa","gdp\_per\_capita/Albania","gdp\_per\_capita/Algeria","gdp\_per\_capita/Andorra","gdp\_per\_capita/Angola","gdp\_per\_capita/Anguilla","gdp\_per\_capita/Antigua and Barbuda","gdp\_per\_capita/Argentina","gdp\_per\_capita/Armenia","gdp\_per\_capita/Aruba","gdp\_per\_capita/Asia","gdp\_per\_capita/Australia","gdp\_per\_capita/Austria","gdp\_per\_capita/Azerbaijan","gdp\_per\_capita/Bahamas","gdp\_per\_capita/Bahrain","gdp\_per\_capita/Bangladesh","gdp\_per\_capita/Barbados","gdp\_per\_capita/Belarus","gdp\_per\_capita/Belgium","gdp\_per\_capita/Belize","gdp\_per\_capita/Benin","gdp\_per\_capita/Bermuda","gdp\_per\_capita/Bhutan","gdp\_per\_capita/Bolivia","gdp\_per\_capita/Bonaire Sint Eustatius and Saba","gdp\_per\_capita/Bosnia and Herzegovina","gdp\_per\_capita/Botswana","gdp\_per\_capita/Brazil","gdp\_per\_capita/British Virgin Islands","gdp\_per\_capita/Brunei","gdp\_per\_capita/Bulgaria","gdp\_per\_capita/Burkina Faso","gdp\_per\_capita/Burundi","gdp\_per\_capita/Cambodia","gdp\_per\_capita/Cameroon","gdp\_per\_capita/Canada","gdp\_per\_capita/Cape Verde","gdp\_per\_capita/Cayman Islands","gdp\_per\_capita/Central African Republic","gdp\_per\_capita/Chad","gdp\_per\_capita/Chile","gdp\_per\_capita/China","gdp\_per\_capita/Colombia","gdp\_per\_capita/Comoros","gdp\_per\_capita/Congo","gdp\_per\_capita/Cook Islands","gdp\_per\_capita/Costa Rica","gdp\_per\_capita/Cote d'Ivoire","gdp\_per\_capita/Croatia","gdp\_per\_capita/Cuba","gdp\_per\_capita/Curacao","gdp\_per\_capita/Cyprus","gdp\_per\_capita/Czechia","gdp\_per\_capita/Democratic Republic of Congo","gdp\_per\_capita/Denmark","gdp\_per\_capita/Djibouti","gdp\_per\_capita/Dominica","gdp\_per\_capita/Dominican Republic","gdp\_per\_capita/Ecuador","gdp\_per\_capita/Egypt","gdp\_per\_capita/El Salvador","gdp\_per\_capita/Equatorial Guinea","gdp\_per\_capita/Eritrea","gdp\_per\_capita/Estonia","gdp\_per\_capita/Eswatini","gdp\_per\_capita/Ethiopia","gdp\_per\_capita/Europe","gdp\_per\_capita/European Union","gdp\_per\_capita/Faeroe Islands","gdp\_per\_capita/Falkland Islands","gdp\_per\_capita/Fiji","gdp\_per\_capita/Finland","gdp\_per\_capita/France","gdp\_per\_capita/French Polynesia","gdp\_per\_capita/Gabon","gdp\_per\_capita/Gambia","gdp\_per\_capita/Georgia","gdp\_per\_capita/Germany","gdp\_per\_capita/Ghana","gdp\_per\_capita/Gibraltar","gdp\_per\_capita/Greece","gdp\_per\_capita/Greenland","gdp\_per\_capita/Grenada","gdp\_per\_capita/Guam","gdp\_per\_capita/Guatemala","gdp\_per\_capita/Guernsey","gdp\_per\_capita/Guinea","gdp\_per\_capita/Guinea-Bissau","gdp\_per\_capita/Guyana","gdp\_per\_capita/Haiti","gdp\_per\_capita/High income","gdp\_per\_capita/Honduras","gdp\_per\_capita/Hong Kong","gdp\_per\_capita/Hungary","gdp\_per\_capita/Iceland","gdp\_per\_capita/India","gdp\_per\_capita/Indonesia","gdp\_per\_capita/International","gdp\_per\_capita/Iran","gdp\_per\_capita/Iraq","gdp\_per\_capita/Ireland","gdp\_per\_capita/Isle of Man","gdp\_per\_capita/Israel","gdp\_per\_capita/Italy","gdp\_per\_capita/Jamaica","gdp\_per\_capita/Japan","gdp\_per\_capita/Jersey","gdp\_per\_capita/Jordan","gdp\_per\_capita/Kazakhstan","gdp\_per\_capita/Kenya","gdp\_per\_capita/Kiribati","gdp\_per\_capita/Kosovo","gdp\_per\_capita/Kuwait","gdp\_per\_capita/Kyrgyzstan","gdp\_per\_capita/Laos","gdp\_per\_capita/Latvia","gdp\_per\_capita/Lebanon","gdp\_per\_capita/Lesotho","gdp\_per\_capita/Liberia","gdp\_per\_capita/Libya","gdp\_per\_capita/Liechtenstein","gdp\_per\_capita/Lithuania","gdp\_per\_capita/Low income","gdp\_per\_capita/Lower middle income","gdp\_per\_capita/Luxembourg","gdp\_per\_capita/Macao","gdp\_per\_capita/Madagascar","gdp\_per\_capita/Malawi","gdp\_per\_capita/Malaysia","gdp\_per\_capita/Maldives","gdp\_per\_capita/Mali","gdp\_per\_capita/Malta","gdp\_per\_capita/Marshall Islands","gdp\_per\_capita/Mauritania","gdp\_per\_capita/Mauritius","gdp\_per\_capita/Mexico","gdp\_per\_capita/Micronesia (country)","gdp\_per\_capita/Moldova","gdp\_per\_capita/Monaco","gdp\_per\_capita/Mongolia","gdp\_per\_capita/Montenegro","gdp\_per\_capita/Montserrat","gdp\_per\_capita/Morocco","gdp\_per\_capita/Mozambique","gdp\_per\_capita/Myanmar","gdp\_per\_capita/Namibia","gdp\_per\_capita/Nauru","gdp\_per\_capita/Nepal","gdp\_per\_capita/Netherlands","gdp\_per\_capita/New Caledonia","gdp\_per\_capita/New Zealand","gdp\_per\_capita/Nicaragua","gdp\_per\_capita/Niger","gdp\_per\_capita/Nigeria","gdp\_per\_capita/Niue","gdp\_per\_capita/North America","gdp\_per\_capita/North Korea","gdp\_per\_capita/North Macedonia","gdp\_per\_capita/Northern Cyprus","gdp\_per\_capita/Northern Mariana Islands","gdp\_per\_capita/Norway","gdp\_per\_capita/Oceania","gdp\_per\_capita/Oman","gdp\_per\_capita/Pakistan","gdp\_per\_capita/Palau","gdp\_per\_capita/Palestine","gdp\_per\_capita/Panama","gdp\_per\_capita/Papua New Guinea","gdp\_per\_capita/Paraguay","gdp\_per\_capita/Peru","gdp\_per\_capita/Philippines","gdp\_per\_capita/Pitcairn","gdp\_per\_capita/Poland","gdp\_per\_capita/Portugal","gdp\_per\_capita/Puerto Rico","gdp\_per\_capita/Qatar","gdp\_per\_capita/Romania","gdp\_per\_capita/Russia","gdp\_per\_capita/Rwanda","gdp\_per\_capita/Saint Helena","gdp\_per\_capita/Saint Kitts and Nevis","gdp\_per\_capita/Saint Lucia","gdp\_per\_capita/Saint Pierre and Miquelon","gdp\_per\_capita/Saint Vincent and the Grenadines","gdp\_per\_capita/Samoa","gdp\_per\_capita/San Marino","gdp\_per\_capita/Sao Tome and Principe","gdp\_per\_capita/Saudi Arabia","gdp\_per\_capita/Senegal","gdp\_per\_capita/Serbia","gdp\_per\_capita/Seychelles","gdp\_per\_capita/Sierra Leone","gdp\_per\_capita/Singapore","gdp\_per\_capita/Sint Maarten (Dutch part)","gdp\_per\_capita/Slovakia","gdp\_per\_capita/Slovenia","gdp\_per\_capita/Solomon Islands","gdp\_per\_capita/Somalia","gdp\_per\_capita/South Africa","gdp\_per\_capita/South America","gdp\_per\_capita/South Korea","gdp\_per\_capita/South Sudan","gdp\_per\_capita/Spain","gdp\_per\_capita/Sri Lanka","gdp\_per\_capita/Sudan","gdp\_per\_capita/Suriname","gdp\_per\_capita/Sweden","gdp\_per\_capita/Switzerland","gdp\_per\_capita/Syria","gdp\_per\_capita/Taiwan","gdp\_per\_capita/Tajikistan","gdp\_per\_capita/Tanzania","gdp\_per\_capita/Thailand","gdp\_per\_capita/Timor","gdp\_per\_capita/Togo","gdp\_per\_capita/Tokelau","gdp\_per\_capita/Tonga","gdp\_per\_capita/Trinidad and Tobago","gdp\_per\_capita/Tunisia","gdp\_per\_capita/Turkey","gdp\_per\_capita/Turkmenistan","gdp\_per\_capita/Turks and Caicos Islands","gdp\_per\_capita/Tuvalu","gdp\_per\_capita/Uganda","gdp\_per\_capita/Ukraine","gdp\_per\_capita/United Arab Emirates","gdp\_per\_capita/United Kingdom","gdp\_per\_capita/United States","gdp\_per\_capita/United States Virgin Islands","gdp\_per\_capita/Upper middle income","gdp\_per\_capita/Uruguay","gdp\_per\_capita/Uzbekistan","gdp\_per\_capita/Vanuatu","gdp\_per\_capita/Vatican","gdp\_per\_capita/Venezuela","gdp\_per\_capita/Vietnam","gdp\_per\_capita/Wallis and Futuna","gdp\_per\_capita/Western Sahara","gdp\_per\_capita/World","gdp\_per\_capita/Yemen","gdp\_per\_capita/Zambia","gdp\_per\_capita/Zimbabwe","gdp\_per\_capita"],"labels":["Afghanistan","Africa","Albania","Algeria","Andorra","Angola","Anguilla","Antigua and Barbuda","Argentina","Armenia","Aruba","Asia","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","Equatorial Guinea","Eritrea","Estonia","Eswatini","Ethiopia","Europe","European Union","Faeroe Islands","Falkland Islands","Fiji","Finland","France","French Polynesia","Gabon","Gambia","Georgia","Germany","Ghana","Gibraltar","Greece","Greenland","Grenada","Guam","Guatemala","Guernsey","Guinea","Guinea-Bissau","Guyana","Haiti","High income","Honduras","Hong Kong","Hungary","Iceland","India","Indonesia","International","Iran","Iraq","Ireland","Isle of Man","Israel","Italy","Jamaica","Japan","Jersey","Jordan","Kazakhstan","Kenya","Kiribati","Kosovo","Kuwait","Kyrgyzstan","Laos","Latvia","Lebanon","Lesotho","Liberia","Libya","Liechtenstein","Lithuania","Low income","Lower middle income","Luxembourg","Macao","Madagascar","Malawi","Malaysia","Maldives","Mali","Malta","Marshall Islands","Mauritania","Mauritius","Mexico","Micronesia (country)","Moldova","Monaco","Mongolia","Montenegro","Montserrat","Morocco","Mozambique","Myanmar","Namibia","Nauru","Nepal","Netherlands","New Caledonia","New Zealand","Nicaragua","Niger","Nigeria","Niue","North America","North Korea","North Macedonia","Northern Cyprus","Northern Mariana Islands","Norway","Oceania","Oman","Pakistan","Palau","Palestine","Panama","Papua New Guinea","Paraguay","Peru","Philippines","Pitcairn","Poland","Portugal","Puerto Rico","Qatar","Romania","Russia","Rwanda","Saint Helena","Saint Kitts and Nevis","Saint Lucia","Saint Pierre and Miquelon","Saint Vincent and the Grenadines","Samoa","San Marino","Sao Tome and Principe","Saudi Arabia","Senegal","Serbia","Seychelles","Sierra Leone","Singapore","Sint Maarten (Dutch part)","Slovakia","Slovenia","Solomon Islands","Somalia","South Africa","South America","South Korea","South Sudan","Spain","Sri Lanka","Sudan","Suriname","Sweden","Switzerland","Syria","Taiwan","Tajikistan","Tanzania","Thailand","Timor","Togo","Tokelau","Tonga","Trinidad and Tobago","Tunisia","Turkey","Turkmenistan","Turks and Caicos Islands","Tuvalu","Uganda","Ukraine","United Arab Emirates","United Kingdom","United States","United States Virgin Islands","Upper middle income","Uruguay","Uzbekistan","Vanuatu","Vatican","Venezuela","Vietnam","Wallis and Futuna","Western Sahara","World","Yemen","Zambia","Zimbabwe","gdp\_per\_capita"],"name":"","parents":["gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita","gdp\_per\_capita",""],"type":"treemap","values":[1499113.1970000002,9938488.902,9796847.73,1.154848637e7,4795263.88,4690512.97,1.7149772514e7,1.7472136658999998e7,5.9395666258999996e7,7487018.16,2.9246683953e7,3.857648544e7,3.83978906e7,3.771244938e7,1.3074120675e7,2.245145607e7,3.5974575855000004e7,2882618.912,1.3735257012e7,1.4197908709e7,3.6302448176e7,6282962.686,1672031.16,4.0890137205e7,7141049.54,5611950.635,9289118.735000001,9617107.795,1.2582669704e7,1.1691761707999999e7,5.7303782298e7,5.8668158067e7,1.5184785126e7,1389731.232,558268.875,3131115.1300000004,2759239.32,3.7987181033e7,5015378.524,4.0571162577e7,536265.64,1426899.471,4.2323921783e7,1.3226727168e7,1.086905818e7,1081625.85,3958820.266,5883973.105,1.27304959e7,2934819.8899999997,1.8815931509999998e7,2.6385917448000003e7,2.6321087184e7,2.6515577976e7,6.1820797776e7,658628.395,3.9820185295e7,2185968.048,7777387.068,1.2045710325000001e7,1.2740650944e7,8872723.246,5885013.606,1.8332552003e7,1215919.495,2.5118026704e7,6284047.7,1406430.6509999998,7510667.425000001,4.1330428275e7,7153845.45,6884053.225000001,7449746.600000001,3.4781962897e7,7.894859719500001e7,1.3465241769000001e7,1.3448679356e7,1263469.503,8078670.490999999,1.1954089453500001e8,3432835.56,2.0200142004e7,2.0445885824e7,1.1011040370000001e7,1.0929477108000001e7,5998436.864,6221151.104,987469.444,1625126.838,1240488.675,6052128.257999999,1332457.438,3924110.88,3701562.9250000003,5.302795432e7,2.2037932703e7,3.8441406266e7,5501232.944,9219525.056,1.618206176e7,1.5953070319999998e7,1.3016772366e7,5.561895201800001e7,2.6704649919999998e7,2.76654872e7,6.69181596e7,6677760.365,4.5671603133e7,3685170.58,6861754.27,1.9557193044e7,2454282.96,756792.424,7954217.208000001,5.4455876246999994e7,2741926.992,5130682.72,4.7320541248e7,1.114854921e7,2144067.056,609005.492,1.4698600397999998e7,2.426894583e7,2.438704289e7,7.843926688e7,8.145616175999999e7,7.834498891499999e7,9.0600639264e7,1141650.6400000001,872748.4739999999,2.3108637368e7,1.2420197888e7,1613459.1060000001,2.9977438183e7,2894955.1160000004,2921277.996,1.6396537959999999e7,1.5342775065000001e7,1646432.536,4245397.096,9780538.796,9662130.336,1.3275113992e7,6047890.504,6347291.024,913426.812,4467686.003,7747948.096000001,4758489.315,2103254.244,4.013526726e7,2.9121275301e7,2.9842992161000002e7,4294405.308,746356,4414901.458,3487582.924,1.1328088896e7,6673607.926,1.0869196406e7,2.0541618069000002e7,3.952803477e7,5.378404731e7,3.2684170449e7,3.1545349179e7,4178807.6399999997,3786755.83,3653366.258,1.8192169229e7,3081494.364,7229321.19,1.6800997338e7,6504904.928,2313397.825,2.237191779e7,2.30479392e7,2.877167407e7,9.658880560000001e7,3.9655751499e7,2.117489067e7,1505619.332,1.5655534475e7,1.9748162384999998e7,1.0516893268e7,8474445.34,8710442.552000001,3390136.591,4.696757422e7,2408591.346,4.0413418664e7,2043169.66,1.1646522349e7,2.1396034757e7,1105288.5,7.3817035529e7,1.3768020928000001e7,2.4727224639999997e7,2.678491652e7,1323553.7999999998,9958849.56,1.0426054848e7,2.9936665542000003e7,8.323327418400002e7,1240211.52,2.926859544e7,1.0023737143e7,3631270.1909999996,1.1178900628e7,4.0094687682000004e7,9.1396984272e7,2329118.052,2520314.31,2213241.532,2173476.24,1.4356905822e7,5282362.007999999,1175306.286,618520.794,2224504.61,2.3355613652e7,8918122.134,2.0480412915e7,6096716.556000001,2852932.9919999996,1229835.7759999998,1366654.135,1.1699490426e7,5.7670514930999994e7,3.398902362e7,7.287899942400001e7,1.6400024382e7,1.7756417376e7,2.897748669e7,5071267.3440000005,1668410.039,1.3730918040000001e7,1.3596957864e7,5326335.892,9173239.751,15469.207,1.3365394848000001e7,1161130.395,2980914.808,1531218.6500000001,4.400472229537e9]}],"layout":{"legend":{"tracegroupgap":0},"margin":{"t":60},"template":{"data":{"bar":[{"error\_x":{"color":"#2a3f5f"},"error\_y":{"color":"#2a3f5f"},"marker":{"line":{"color":"#E5ECF6","width":0.5},"pattern":{"fillmode":"overlay","size":10,"solidity":0.2}},"type":"bar"}],"barpolar":[{"marker":{"line":{"color":"#E5ECF6","width":0.5},"pattern":{"fillmode":"overlay","size":10,"solidity":0.2}},"type":"barpolar"}],"carpet":[{"aaxis":{"endlinecolor":"#2a3f5f","gridcolor":"white","linecolor":"white","minorgridcolor":"white","startlinecolor":"#2a3f5f"},"baxis":{"endlinecolor":"#2a3f5f","gridcolor":"white","linecolor":"white","minorgridcolor":"white","startlinecolor":"#2a3f5f"},"type":"carpet"}],"choropleth":[{"colorbar":{"outlinewidth":0,"ticks":""},"type":"choropleth"}],"contour":[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":[[0,"#0d0887"],[0.1111111111111111,"#46039f"],[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],[0.8888888888888888,"#fdca26"],[1,"#f0f921"]],"type":"contour"}],"contourcarpet":[{"colorbar":{"outlinewidth":0,"ticks":""},"type":"contourcarpet"}],"heatmap":[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":[[0,"#0d0887"],[0.1111111111111111,"#46039f"],[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],[0.8888888888888888,"#fdca26"],[1,"#f0f921"]],"type":"heatmap"}],"heatmapgl":[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":[[0,"#0d0887"],[0.1111111111111111,"#46039f"],[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],[0.8888888888888888,"#fdca26"],[1,"#f0f921"]],"type":"heatmapgl"}],"histogram":[{"marker":{"pattern":{"fillmode":"overlay","size":10,"solidity":0.2}},"type":"histogram"}],"histogram2d":[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":[[0,"#0d0887"],[0.1111111111111111,"#46039f"],[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],[0.8888888888888888,"#fdca26"],[1,"#f0f921"]],"type":"histogram2d"}],"histogram2dcontour":[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":[[0,"#0d0887"],[0.1111111111111111,"#46039f"],[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],[0.8888888888888888,"#fdca26"],[1,"#f0f921"]],"type":"histogram2dcontour"}],"mesh3d":[{"colorbar":{"outlinewidth":0,"ticks":""},"type":"mesh3d"}],"parcoords":[{"line":{"colorbar":{"outlinewidth":0,"ticks":""}},"type":"parcoords"}],"pie":[{"automargin":true,"type":"pie"}],"scatter":[{"fillpattern":{"fillmode":"overlay","size":10,"solidity":0.2},"type":"scatter"}],"scatter3d":[{"line":{"colorbar":{"outlinewidth":0,"ticks":""}},"marker":{"colorbar":{"outlinewidth":0,"ticks":""}},"type":"scatter3d"}],"scattercarpet":[{"marker":{"colorbar":{"outlinewidth":0,"ticks":""}},"type":"scattercarpet"}],"scattergeo":[{"marker":{"colorbar":{"outlinewidth":0,"ticks":""}},"type":"scattergeo"}],"scattergl":[{"marker":{"colorbar":{"outlinewidth":0,"ticks":""}},"type":"scattergl"}],"scattermapbox":[{"marker":{"colorbar":{"outlinewidth":0,"ticks":""}},"type":"scattermapbox"}],"scatterpolar":[{"marker":{"colorbar":{"outlinewidth":0,"ticks":""}},"type":"scatterpolar"}],"scatterpolargl":[{"marker":{"colorbar":{"outlinewidth":0,"ticks":""}},"type":"scatterpolargl"}],"scatterternary":[{"marker":{"colorbar":{"outlinewidth":0,"ticks":""}},"type":"scatterternary"}],"surface":[{"colorbar":{"outlinewidth":0,"ticks":""},"colorscale":[[0,"#0d0887"],[0.1111111111111111,"#46039f"],[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],[0.8888888888888888,"#fdca26"],[1,"#f0f921"]],"type":"surface"}],"table":[{"cells":{"fill":{"color":"#EBF0F8"},"line":{"color":"white"}},"header":{"fill":{"color":"#C8D4E3"},"line":{"color":"white"}},"type":"table"}]},"layout":{"annotationdefaults":{"arrowcolor":"#2a3f5f","arrowhead":0,"arrowwidth":1},"autotypenumbers":"strict","coloraxis":{"colorbar":{"outlinewidth":0,"ticks":""}},"colorscale":{"diverging":[[0,"#8e0152"],[0.1,"#c51b7d"],[0.2,"#de77ae"],[0.3,"#f1b6da"],[0.4,"#fde0ef"],[0.5,"#f7f7f7"],[0.6,"#e6f5d0"],[0.7,"#b8e186"],[0.8,"#7fbc41"],[0.9,"#4d9221"],[1,"#276419"]],"sequential":[[0,"#0d0887"],[0.1111111111111111,"#46039f"],[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],[0.8888888888888888,"#fdca26"],[1,"#f0f921"]],"sequentialminus":[[0,"#0d0887"],[0.1111111111111111,"#46039f"],[0.2222222222222222,"#7201a8"],[0.3333333333333333,"#9c179e"],[0.4444444444444444,"#bd3786"],[0.5555555555555556,"#d8576b"],[0.6666666666666666,"#ed7953"],[0.7777777777777778,"#fb9f3a"],[0.8888888888888888,"#fdca26"],[1,"#f0f921"]]},"colorway":["#636efa","#EF553B","#00cc96","#ab63fa","#FFA15A","#19d3f3","#FF6692","#B6E880","#FF97FF","#FECB52"],"font":{"color":"#2a3f5f"},"geo":{"bgcolor":"white","lakecolor":"white","landcolor":"#E5ECF6","showlakes":true,"showland":true,"subunitcolor":"white"},"hoverlabel":{"align":"left"},"hovermode":"closest","mapbox":{"style":"light"},"paper\_bgcolor":"white","plot\_bgcolor":"#E5ECF6","polar":{"angularaxis":{"gridcolor":"white","linecolor":"white","ticks":""},"bgcolor":"#E5ECF6","radialaxis":{"gridcolor":"white","linecolor":"white","ticks":""}},"scene":{"xaxis":{"backgroundcolor":"#E5ECF6","gridcolor":"white","gridwidth":2,"linecolor":"white","showbackground":true,"ticks":"","zerolinecolor":"white"},"yaxis":{"backgroundcolor":"#E5ECF6","gridcolor":"white","gridwidth":2,"linecolor":"white","showbackground":true,"ticks":"","zerolinecolor":"white"},"zaxis":{"backgroundcolor":"#E5ECF6","gridcolor":"white","gridwidth":2,"linecolor":"white","showbackground":true,"ticks":"","zerolinecolor":"white"}},"shapedefaults":{"line":{"color":"#2a3f5f"}},"ternary":{"aaxis":{"gridcolor":"white","linecolor":"white","ticks":""},"baxis":{"gridcolor":"white","linecolor":"white","ticks":""},"bgcolor":"#E5ECF6","caxis":{"gridcolor":"white","linecolor":"white","ticks":""}},"title":{"x":5.0e-2},"xaxis":{"automargin":true,"gridcolor":"white","linecolor":"white","ticks":"","title":{"standoff":15},"zerolinecolor":"white","zerolinewidth":2},"yaxis":{"automargin":true,"gridcolor":"white","linecolor":"white","ticks":"","title":{"standoff":15},"zerolinecolor":"white","zerolinewidth":2}}}}}

## North America,European Union,China and countries with higher gdp have higher total vaccination per hundred

sns.set\_theme(style="darkgrid")  
sns.set(rc = {'figure.figsize':(40,20)})  
ax = sns.countplot(x="location", data=vaccine)



# 5. Model Building

!pip install pycaret

import numpy as np  
import pandas as pd  
  
import plotly.graph\_objects as go  
import plotly.express as px  
import plotly.offline as pyo  
from plotly.subplots import make\_subplots  
pyo.init\_notebook\_mode()  
  
  
from datetime import date , datetime , timedelta  
  
import pycaret.regression as caret  
  
  
import warnings  
warnings.filterwarnings('ignore')

import io  
import requests  
url = "https://raw.githubusercontent.com/owid/covid-19-data/master/public/data/vaccinations/vaccinations.csv"  
read\_data = requests.get(url).content  
data\_detailed = pd.read\_csv(io.StringIO(read\_data.decode('utf-8')))  
  
url2 = "https://raw.githubusercontent.com/owid/covid-19-data/master/public/data/vaccinations/vaccinations-by-manufacturer.csv"  
read\_data2 = requests.get(url2).content  
data\_total = pd.read\_csv(io.StringIO(read\_data2.decode('utf-8')))

print("\* "\*10+" data\_detailed "+" \*"\*10)  
print("\nShape: rows = {} , columns = {}".format(data\_detailed.shape[0] , data\_detailed.shape[1]))  
print(data\_detailed.info())  
print("\* "\*10+" data\_total "+" \*"\*10)  
print("\nShape: rows = {} , columns = {}".format(data\_total.shape[0] , data\_total.shape[1]))  
print(data\_total.info())

data\_detailed.date.max()

# find the last date  
last\_date = data\_detailed.sort\_values(by = 'date' , ascending=False)['date'].iloc[0]  
# its ''2022-05-21'

countries = data\_total.location.unique()

data\_detailed[(data\_detailed.date == last\_date)&(data\_detailed.people\_fully\_vaccinated\_per\_hundred.isnull())]

data\_detailed[(data\_detailed.date == last\_date)&(data\_detailed.location == 'Germany')]

euro\_vaccines = data\_total[(data\_total.location == 'European Union') &  
 (data\_total.date == last\_date)][['vaccine','total\_vaccinations']]  
euro\_vaccines.sort\_values(by = 'total\_vaccinations' , ascending = False , inplace = True)

euro\_vaccines

pie\_euro\_vac = go.Figure(data = go.Pie(values = euro\_vaccines.total\_vaccinations,   
 labels = euro\_vaccines.vaccine, hole = 0.55))  
pie\_euro\_vac.update\_traces(textposition='outside', textinfo='percent+label')  
pie\_euro\_vac.update\_layout(annotations=[dict(text='Vaccines used by', x=0.5, y=0.55, font\_size=16, showarrow=False),  
 dict(text='European Union', x=0.5, y=0.45, font\_size=16, showarrow=False)])  
pie\_euro\_vac.show()

data\_detailed[data\_detailed.location == 'Germany']['date'].max() , data\_total[data\_total.location == 'Germany']['date'].max()

germany\_vaccines=data\_total[(data\_total.location=='Germany')&(data\_total.date=='2022-05-18')][['vaccine','total\_vaccinations']]  
germany\_vaccines.sort\_values(by = 'total\_vaccinations' , ascending = False , inplace = True)  
df\_germany\_info = data\_detailed[data\_detailed.location == 'Germany']

fig\_germany = make\_subplots(rows = 4 , cols = 2  
 , specs=[[{"type": "pie","rowspan": 2}, {"type": "scatter","rowspan": 2}]  
 ,[None , None]  
 ,[{"type": "scatter","colspan": 2,"rowspan": 2}, None]  
 ,[None ,  
 None]]  
   
 , subplot\_titles=[  
 '',   
 'temp',  
 'temp' # i will change the titles a few lines later ...  
 ])  
  
fig\_germany.add\_trace(go.Pie(labels = germany\_vaccines.vaccine , values = germany\_vaccines.total\_vaccinations  
 , hole = 0.5 , pull = [0,0.1,0.1,0.1] , title = "Vaccines" , titleposition='middle center'  
 , titlefont = {'family':'serif' , 'size':18}  
 , textinfo = 'percent+label' , textposition = 'inside')  
 , row = 1 , col = 1)  
  
fig\_germany.add\_trace(go.Scatter(x = df\_germany\_info['date']  
 , y = df\_germany\_info['daily\_vaccinations']  
 , name = "Daily vaccinations")  
 , row = 1 , col = 2)  
  
fig\_germany.add\_trace(go.Scatter(x = df\_germany\_info['date']  
 , y = df\_germany\_info['people\_fully\_vaccinated\_per\_hundred']  
 , name = "Fully vaccinated people percentage"  
 # <br> for the next line in hover  
 , hovertemplate = "<b>%{x}</b><br>" +"Fully vaccinated people = %{y:.2f} %" +"<extra></extra>")  
 , row = 3 , col = 1)  
  
  
fig\_germany.layout.annotations[0].update(text="Number of daily vaccinations" , x=0.75  
 , font = {'family':'serif','size':20})  
  
fig\_germany.layout.annotations[1].update(text="Fully vaccinated people percentage" , x=0.25   
 , font = {'family':'serif','size':20})  
  
fig\_germany.update\_yaxes(range=[0, 100], row=3, col=1)  
fig\_germany.update\_layout(width = 950,height=600, showlegend=True)  
fig\_germany.update\_layout(title\_text='Germany abstract informations'  
 ,title\_font={'family':'serif','size':26} , title = {'x':0.25 , 'y':0.95})  
  
fig\_germany.show()

data = pd.DataFrame()  
data['Date'] = pd.to\_datetime(df\_germany\_info['date'])  
data['Target'] = df\_germany\_info['people\_fully\_vaccinated\_per\_hundred']  
data.reset\_index(drop = True , inplace = True)

data.Date.min() , data.Date.max() , len(data)

from datetime import date, datetime  
  
d0 = date(2020 , 12 , 27)  
d1 = date(2022 , 5 , 18)  
delta = d1 - d0  
  
days = delta.days + 1  
print(days)

data=data.dropna()

data.isnull().sum()

data['Series'] = np.arange(1 , len(data)+1)  
  
# Shift1 is the previous value(Target) for each row :  
data['Shift1'] = data.Target.shift(1)  
  
# mean of the Target during 10 previous days :  
window\_len = 10  
window = data['Shift1'].rolling(window = window\_len)  
means = window.mean()  
data['Window\_mean'] = means  
  
  
# This approach will make some Missing values (for example we dont have the previous value for the first row)  
data.dropna(inplace = True)  
data.reset\_index(drop = True , inplace=True)  
  
dates = data['Date'] # we will need this  
  
data = data[['Series' , 'Window\_mean' , 'Shift1' , 'Target']]  
  
data

# 50% for train & 50% for test  
train = data.iloc[:230,:]   
test = data.iloc[230:,:]  
  
train.shape , test.shape

setup = caret.setup(data = train , test\_data = test , target = 'Target' , fold\_strategy = 'timeseries'  
 , remove\_perfect\_collinearity = False , numeric\_features = ['Series' , 'Window\_mean' , 'Shift1']   
 , fold = 5 , session\_id = 51)

best = caret.compare\_models(sort = 'MAE' , turbo = False)

best = caret.tune\_model(best)

\_ = caret.predict\_model(best)

# generate predictions on the original dataset  
predictions = caret.predict\_model(best , data=data)  
  
# add a date column in the dataset  
predictions['Date'] = dates  
  
# line plot  
fig = px.line(predictions.rename(columns = {'Target':'Actual' }), x='Date', y=["Actual"])  
fig.update\_layout(annotations=[dict(text='Test set', x='2022-4-15', y=30, font\_size=20, showarrow=False)])  
# add a vertical rectangle for test-set separation  
  
fig.add\_vrect(x0 = dates.iloc[230], x1 = dates.iloc[-1], fillcolor="grey", opacity=0.25, line\_width=1)  
fig.show()

## Future forecasting

* As we used lag and window features, forecasting the future is a little harder.
* For example we dont have the previous value for 2022-5-29 since we dont know the target value at 2022-5-28
* So we will start from the first future time step and both we make predictions and also fill the lag features for next time steps. (maybe something like recursive functions)

future = pd.DataFrame(columns = ['Series' , 'Window\_mean' , 'Shift1'])  
future['Series'] = np.arange(300,450) # for the next 150 time steps  
future['Window\_mean'] = np.nan  
future['Shift1'] = np.nan  
  
# initialize the first row :  
#------------------------------  
future.iloc[0,2] = data['Target'].max()  
sum = 0  
for i in range(window\_len):  
 sum += data.iloc[len(data)-1-i,3]  
   
future.iloc[0,1] = sum/window\_len  
future

for j in range(len(future)):  
 current\_row = j  
 next\_row = j+1  
   
 # for the next\_row we are going to fill :  
 # 1. Shift1 --> use currnet\_row predicted value  
 # 2. Window\_mean  
   
 if current\_row != len(future)-1 :  
 # fill Shift1 for the next\_row  
 future.iloc[next\_row,2] = caret.predict\_model(best , future.iloc[[current\_row]])['Label']  
# print(future.iloc[next\_row,2]-future.iloc[current\_row,2])  
   
   
 # fill the Window\_mean for the next\_row  
 if next\_row < 9 :  
 sum = 0  
 num\_rows\_from\_data = window\_len - (next\_row + 1)  
 num\_rows\_from\_future = window\_len - num\_rows\_from\_data  
  
 for i in range(num\_rows\_from\_data):  
 sum += data.iloc[len(data)-1-i , 2]  
  
  
 for i in range(num\_rows\_from\_future):  
 sum += future.iloc[next\_row - i , 2]  
  
 future.iloc[next\_row , 1] = sum/window\_len  
  
  
 elif next\_row >= 9:  
 sum = 0  
 for i in range(window\_len):  
 sum += future.iloc[next\_row-i,2]  
 future.iloc[next\_row,1] = sum/window\_len

* As you see in the above cell, in each row of the future data frame we have the previous value in 'Shift1' column.
* So with a reverse shift of this column, we have the current value for each row.

future['Predicted'] = future['Shift1'].shift(-1)  
  
start = datetime.strptime("2022-05-19", "%Y-%m-%d")  
date\_generated = [start + timedelta(days=x) for x in range(0, 150)]  
date\_list = []  
for date in date\_generated:  
 date\_list.append(date.strftime("%Y-%m-%d"))  
   
future['Date'] = date\_list  
  
future = future[['Date' , 'Predicted']]  
future.dropna(inplace = True)  
future

fig = go.Figure(data=go.Scatter(x=df\_germany\_info['date'], y = df\_germany\_info['people\_fully\_vaccinated\_per\_hundred']  
 ,mode='lines', line\_color='red' , name = 'Until now'))  
fig.add\_trace(go.Scatter(x=future['Date'], y=future['Predicted'],mode='lines', line=dict(color="#0000ff"), name = 'Future'))  
  
  
  
fig.show()

## About 13th october 2022

## 100% of people in Germany will get fully vaccinated ?! Maybe

# 6. conclusion

European union is one of the best example for the region which has many deaths and also which produced high number of total vaccinations. Also the gdp of countries of this region has not much differed after pandemic.

Comparing the Root-mean-square error (rmse) and coefficient of discrimination(R2) values of models, Huber regression is selected with least RMSE of 0.1678 and R2 value of 0.9974 as the best model to predict the total vaccinations of particular region over the time period.

According to our model,Total Vaccinations of Germany might be completed by 2nd week of October.

# 7. Future Scope

The country governments can improve the vaccination facilities and availibility of vaccines to different locations by demand of vaccines and by refering most used vaccines across different countries. The synthesis of current research will be helpful to researchers analyzing historical trends in the COVID-19 pandemic and individuals interested in better understanding and advocating for underserved communities across the globe.