

Covid vaccines analysis

Introduction:

The global battle against COVID 19 pandemic can be won only if a large part of the world gets vaccinated against the SARS-CoV-2 virus. In this blog, we study the COVID 19 vaccination trends across the world using python, and we aim to derive key insights from the data which can help policymakers modify their policies. Give recommendation manufacturers for their vaccinations At country.

On the project we by manufactures. These two datasets define vaccination among the country and vaccination by manufacture on day by day. have two different Datasets that is country vaccinations and country vaccination

Problem outline:

The problem is to conduct in-depth analysis of covid 19 vaccine data, focusing on the vaccine efficacy, distribution and adverse effects. The main is to provide hidden insights that aid policymakers and health organizations in optimizing vaccine deployment strategies. It could be useful to their vaccine distribution among the countries.

Design Thinking:

1)Data collection:

- ✓ Datasets were collected on the Kaggle website On the name of covid-19 world vaccination.
- ✓ It contains two different datasets named country vaccination by manufactures, country vaccinations

2)Load the data and required libraries:

In this we need import the required libraries to notebook and Load the downloaded dataset using a variable

3)Data preprocessing:

- ✓ Datasets may contain missing values and many outliers, so we need perform anomaly detection And cleansing on the datasets.

4)Exploratory Data Analysis:

- ✓ Perform the EDA process on the datasets to get some patterns among the data. Exploratory Data Analysis (EDA) is an analysis approach that identifies general patterns in the data. In this data
- ✓ I got some hidden patterns among countries and their total vaccinations and also we can understand the relationship visually by use of libraries matplotlib and seaborn

5)Statistical Analysis:

- ✓ Statistical analysis is the process of collecting and analysing large volumes of data in order to identify trends and develop valuable insights. In this data
- ✓ We can make some statistical analysis to identify the descriptive among the data on both datasets. First of all we convert the date and time into proper format. We can find a good valuable insight to Optimize the vaccination distribution and adverse affects

6)Machine learning Technique:

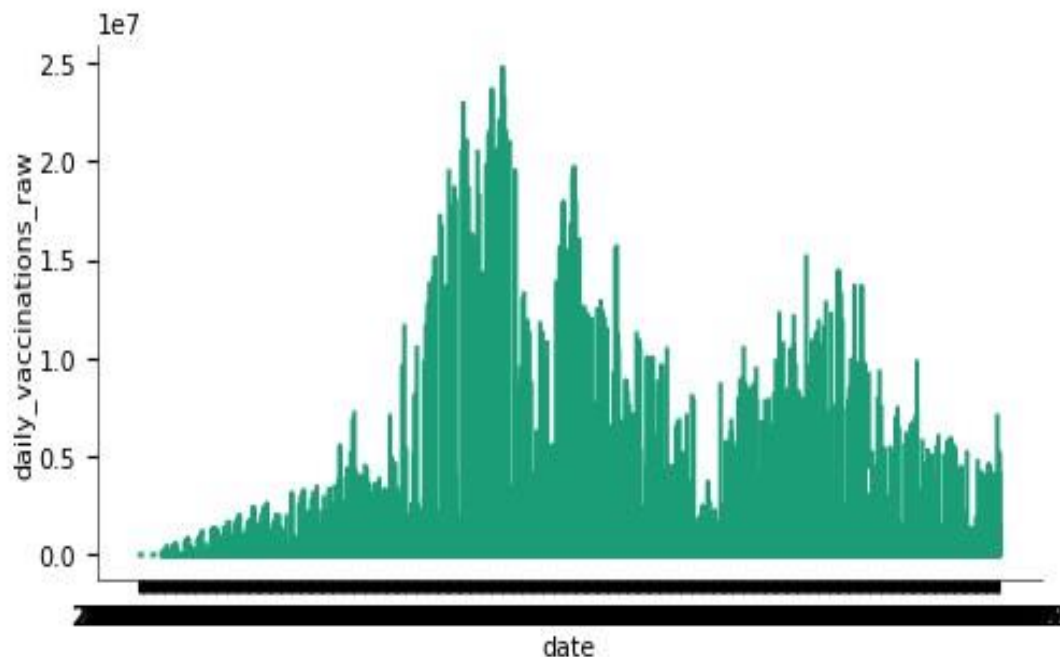
- ✓ Using the machine learning techniques to we can make a model to predict values attribute. Machine Learning Algorithms use full to make a good model And feed the data as per the preprocessed source. On my hope we can make a Good Regression model By this Dataset to we can predict the value.

Ex:-

- ✓ We can use people vaccinated column to predict the people fully vaccinated column for any country in the world.

7)Time series Forecasting:

- ✓ In this Data we seen the date wise vaccinations Country on all over world. So we make the TimeSeries Analysis on the datasets. To make Time Series Forecasting strategy to we plot future Time intervals to find the growth of the Vaccination rate among the Countries



7)Visualization:

Some visualization is good to use to understand the data and their relationship of the attributes .visualize the data easily understand Technique. Some visualization I got online it provide



8)Provide Insights and Recommendation:

- ✓ At last we find some good hidden insights .It could use full for the vaccine distribution on the country so manufacture can focus on the Vaccine effect and country that are vaccinated In least count And also focus on the vaccines that might be used in multiple countries.
- ✓ We give the reccomedations on the vaccine manufacture growth and Vaccination on the countries that are people vaccinated in a least count.so th world organization took care on that countries

Data collection:

It is the process of gathering and analyzing accurate data from various sources to find answers to research problems, trends and probabilities. On our project data is collected at kaggle website.

About the data:-

On project we have two different Datasets that is country vaccinations and country vaccination by manufactures. These two datasets define vaccination among the country and vaccination by manufacture on day by day.

That data was displayed below first five entry with their size

Country vaccinations:

(86512, 15)

	country	iso_code	date	total_vaccinations	people_vaccinated	people_fully_vaccinated	daily_vaccinations_raw	daily_vaccinations	total_vaccinations_per_hundred
94	Afghanistan	AFG	2021-05-27	593313.0	479574.0	113739.0	2859.0	6487.0	1.49
101	Afghanistan	AFG	2021-06-03	630305.0	481800.0	148505.0	4015.0	5285.0	1.58
339	Afghanistan	AFG	2022-01-27	5081064.0	4517380.0	3868832.0	6868.0	9802.0	12.76
433	Albania	ALB	2021-02-18	3049.0	2438.0	611.0	1348.0	254.0	0.11
515	Albania	ALB	2021-05-11	622507.0	440921.0	181586.0	9548.0	12160.0	21.67

Country vaccinations by manufactures:

(35623, 4)

	location	date	vaccine	total_vaccinations
0	Argentina	2020-12-29	Moderna	2
1	Argentina	2020-12-29	Oxford/AstraZeneca	3
2	Argentina	2020-12-29	Sinopharm/Beijing	1
3	Argentina	2020-12-29	Sputnik V	20481
4	Argentina	2020-12-30	Moderna	2

Load the data and required libraries:

Load the required libraries for the data analysis On the dataset to find the hidden details and patterns in the data. major language that is used in the analysis is python. libraries that are used for major analysis is Pandas, Numby, Matplotlib, Seaborn, And for the machine learning process we using the pycaret for model building.

As per code

import libraries

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings("ignore")
```

load the datasets

```
data=pd.read_csv("/content/drive/MyDrive/country_vaccinations.csv")
data_manu=pd.read_csv("country_vaccinations_by_manufacturer.csv")
print(data.shape)
print(data.head(2))
print(data_manu.shape)
print(data_manu.head(4))
```

output:

(86512, 15)

	country	iso_code	date	total_vaccinations	people_vaccinated	people_fully_vaccinated	daily_vaccinations_raw	daily_vaccinations	total_vaccinations_per_hundred	people
0	Afghanistan	AFG	2021-02-22	0.0	0.0	NaN	NaN	NaN	0.0	
1	Afghanistan	AFG	2021-02-23	NaN	NaN	NaN	NaN	1367.0	NaN	

(35623, 4)

	location	date	vaccine	total_vaccinations
0	Argentina	2020-12-29	Moderna	2
1	Argentina	2020-12-29	Oxford/AstraZeneca	3
2	Argentina	2020-12-29	Sinopharm/Beijing	1
3	Argentina	2020-12-29	Sputnik V	2048

Data preprocessing:

Data preprocessing can refer to manipulation or dropping of data before it is used in order to ensure or enhance performance,^[1] and is an important step in the data analysis process.

Handling missing values:

Find how many values are missed in datasets

Using codes

#using info we can understand the datatypes of column

```
print(data.info())
print(data_manu.info())
print(data.isnull().sum())
print(data_manu.isnull().sum())
```

output:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 86512 entries, 0 to 86511
Data columns (total 15 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   country                               86512 non-null  object
1   iso_code                              86512 non-null  object
2   date                                  86512 non-null  object
3   total_vaccinations                   43607 non-null  float64
4   people_vaccinated                    41294 non-null  float64
5   people_fully_vaccinated              38802 non-null  float64
6   daily_vaccinations_raw               35362 non-null  float64
7   daily_vaccinations                   86213 non-null  float64
8   total_vaccinations_per_hundred       43607 non-null  float64
9   people_vaccinated_per_hundred        41294 non-null  float64
10  people_fully_vaccinated_per_hundred  38802 non-null  float64
11  daily_vaccinations_per_million       86213 non-null  float64
12  vaccines                             86512 non-null  object
13  source_name                          86512 non-null  object
14  source_website                       86512 non-null  object
dtypes: float64(9), object(6)
memory usage: 9.9+ M
```

```
<class 'pandas.core.frame.DataFrame'>
```

```
RangeIndex: 35623 entries, 0 to 35622
Data columns (total 4 columns):
#   Column                Non-Null Count  Dtype
---  -
0   location              35623 non-null  object
1   date                  35623 non-null  object
2   vaccine               35623 non-null  object
3   total_vaccinations    35623 non-null  int64
dtypes: int64(1), object(3)
memory usage: 1.1+ MB
```

country

0

country	0
iso_code	0
date	0
total_vaccinations	42905
people_vaccinated	45218
people_fully_vaccinated	47710
daily_vaccinations_raw	51150
daily_vaccinations	299
total_vaccinations_per_hundred	42905
people_vaccinated_per_hundred	45218
people_fully_vaccinated_per_hundred	47710
daily_vaccinations_per_million	299
vaccines	0
source_name	0
source_website	0

location	0
date	0
vaccine	0
total_vaccinations	0

on the above data there is no missing value in data_manu but data contain many missing values so we need handle the missing values in the data.half of the data contain null values so the best option just drop it.After dropping the missing values data become null free.after checking the size it confirm it(because it reduced).

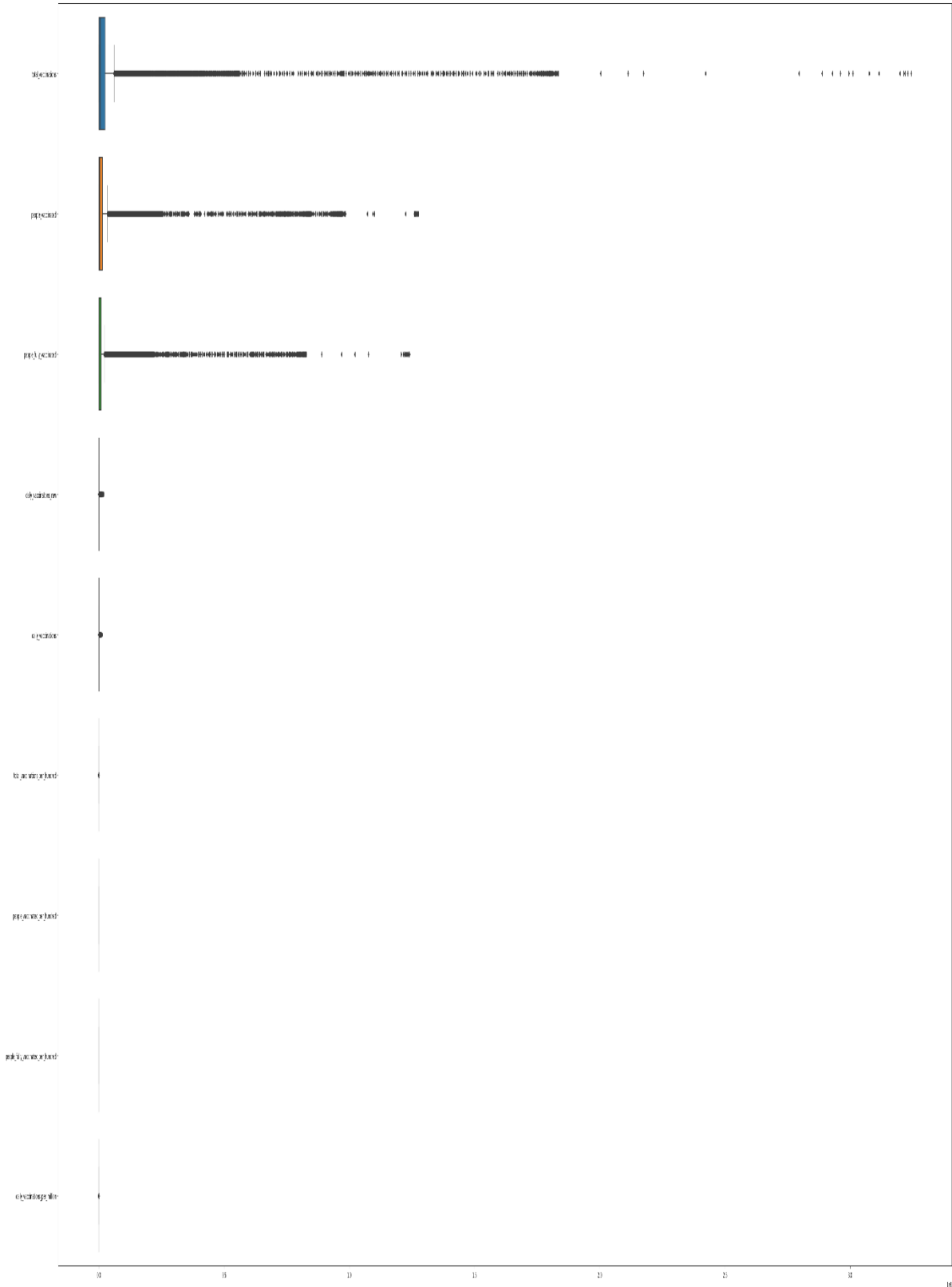
```
data.dropna(axis=0,inplace=True)
data.shape
output:
(30847, 15)
```

Outlier handling:

Outlier is a point that differs significantly from the Observation.The interquartile range method is one of the most commonly used methods to identify outlier.

Boxplot is very useful visualize the outliers on covid vaccinations

```
plt.figure(figsize=(60,30))
sns.boxplot(data,orient='h')
plt.show()
```



Taking the count of an outliers

Outlier on total_vaccinations is 4407

Outlier on people_vaccinated is 4384

Outlier on people_fully_vaccinated is 4826

Outlier on daily_vaccinations_raw is 4091

Outlier on daily_vaccinations is 4004

Outlier on total_vaccinations_per_hundred is 26

Outlier on daily_vaccinations_per_million is 769

After using interquatile method again to clear the outliers in data ,data become

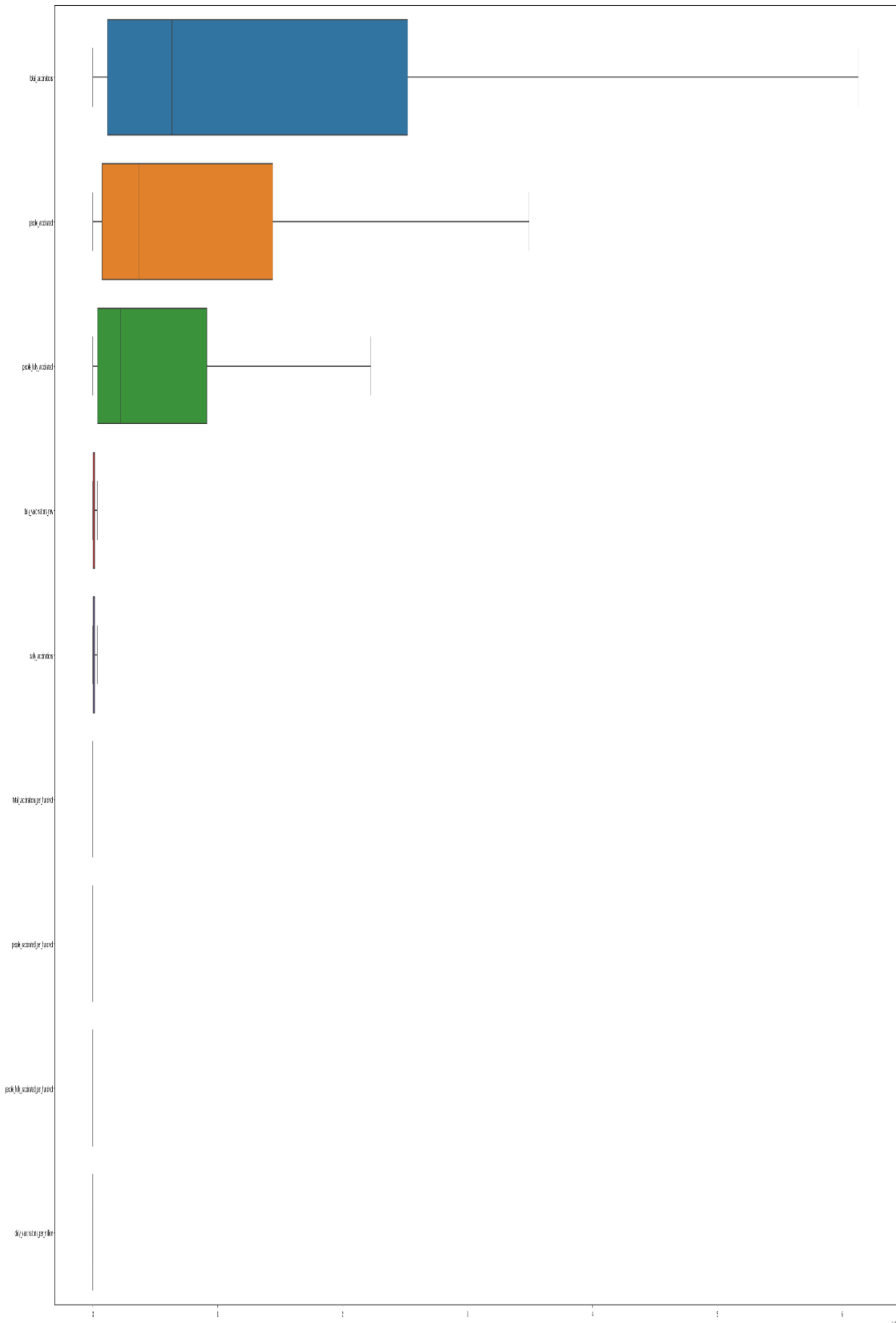
```
list1=['total_vaccinations','people_vaccinated','people_fully_vaccinated','daily_vaccinations_raw','daily_vaccinations','total_vaccinations_per_hundred','daily_vaccinations_per_million']
for c in list1:
    col=data[c]
    q1=col.quantile(0.25)
    q3=col.quantile(0.75)
    iqr=q3-q1
    lower=q1 - 1.5 *iqr
    upper=q3 + 1.5*iqr
    col[col<lower]=lower
    col[col>upper]=upper
    print(" outlier handling completed")
```

output:

```
outlier handling completed
outlier handling completed
outlier handling completed
outlier handling completed
outlier handling completed
outlier handling completed
outlier handling completed
```

After outlier handling

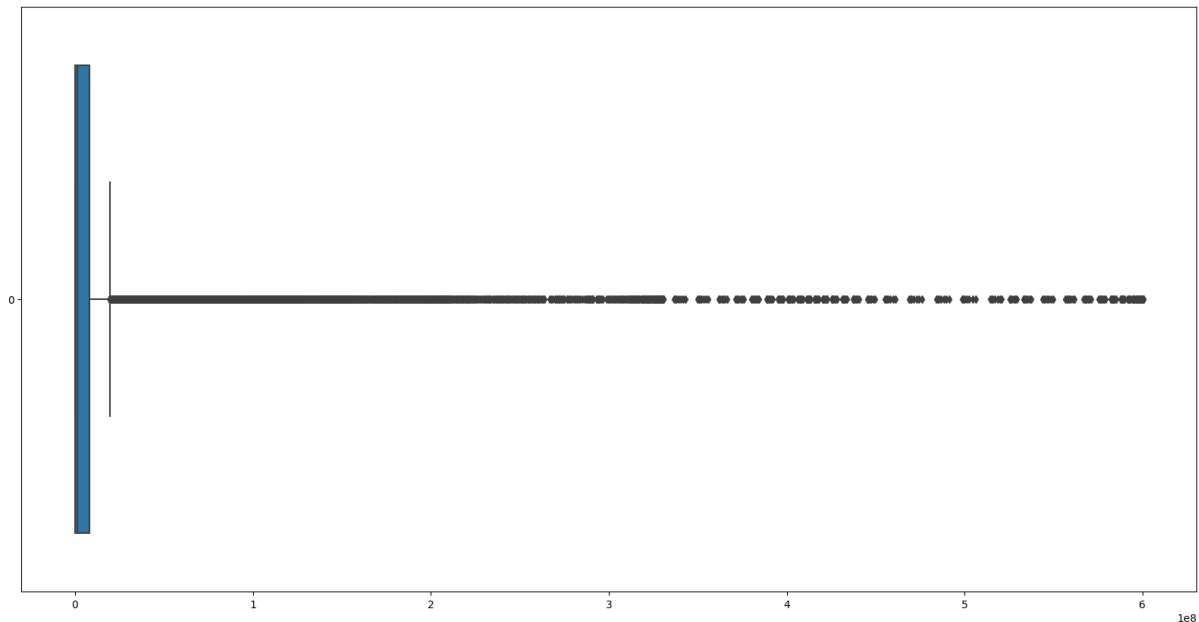
Box polt become



Making same method on the covid vaccinations by manufactures:

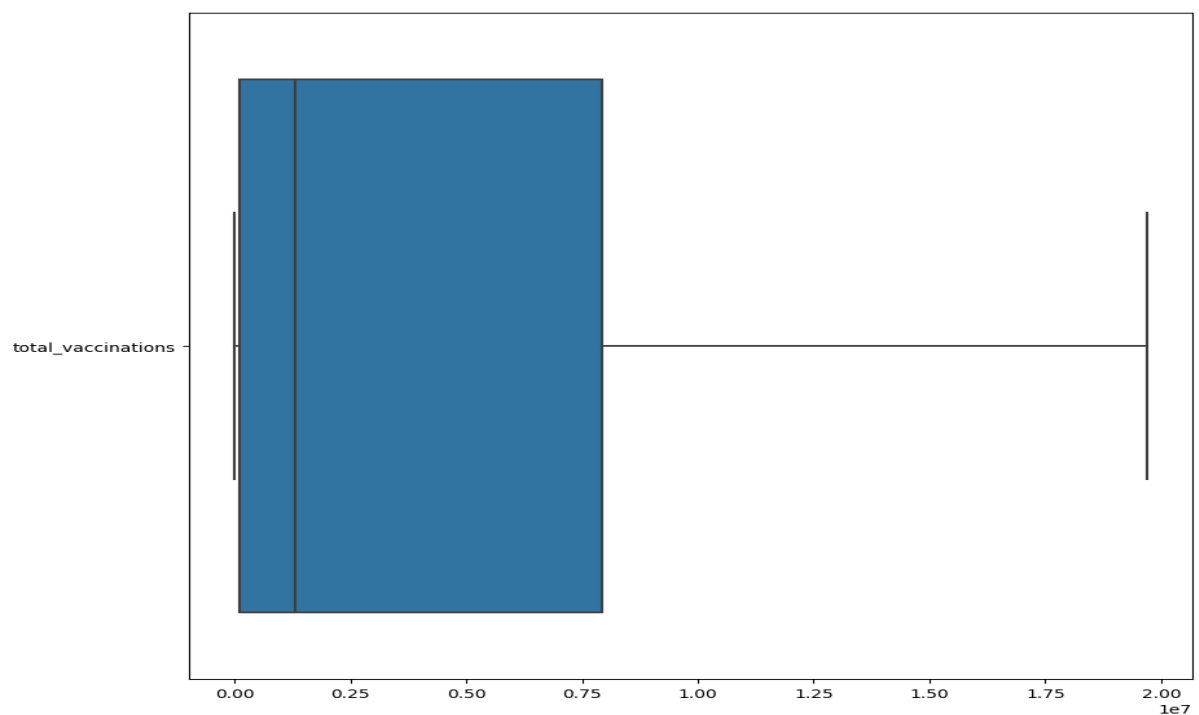
But in this dataset have only one numerical column.

Boxplot for the total vaccination:



Outlier on total_vaccinations is 4544

After using same method to Handle the outlier and Box plot become



Outlier on total_vaccinations is 0

Exploratory Data Analysis:

On covid vaccinations dataset

Number of countries that present in covid vaccinations dataset:**169**

Without date all countries total vaccinations people fully vaccinated, people vaccinated dataframe

	country	total_vaccinations	People_Vaccinated	people_Fully_vaccinated
0	Afghanistan	5.304682e+06	5.478754e+06	4.131076e+06
1	Albania	1.748274e+08	9.592519e+07	7.658774e+07
2	Algeria	2.432556e+07	1.357837e+07	1.070525e+07
3	Andorra	1.526900e+04	9.781000e+03	4.484000e+03
4	Antigua and Barbuda	6.160890e+05	3.551400e+05	2.609490e+05
...
164	Uzbekistan	2.648153e+08	1.442560e+08	6.025293e+07
165	Vietnam	3.551691e+09	2.521270e+09	8.917782e+08
166	Wales	1.805534e+09	8.840849e+08	6.909660e+08
167	Zambia	1.662901e+07	1.146332e+07	5.165692e+06
168	Zimbabwe	1.534183e+09	8.799751e+08	6.423882e+08

169 rows × 4 columns

There are 74 group vaccines used all over countries.

That vaccine groups used countries and its total vaccination

	vaccines	countries name	No_of_countries	total_vaccinations
0	Johnson&Johnson, Oxford/AstraZeneca, Pfizer/Bi...	[Afghanistan, Belize, Cameroon, Namibia, Trini...	5	3.356344e+08
1	Oxford/AstraZeneca, Pfizer/BioNTech, Sinovac, ...	[Albania, Azerbaijan, Bosnia and Herzegovina, ...	4	2.263555e+09
2	Oxford/AstraZeneca, Sinopharm/Beijing, Sinovac...	[Algeria, Zimbabwe]	2	1.558509e+09
3	Moderna, Oxford/AstraZeneca, Pfizer/BioNTech	[Andorra, Australia, England, Finland, Guernse...	12	6.971054e+10
4	Oxford/AstraZeneca, Pfizer/BioNTech, Sputnik V	[Antigua and Barbuda]	1	6.160890e+05
...
69	Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, ...	[Thailand]	1	1.269902e+10
70	Pfizer/BioNTech, Sinovac, Turkovac	[Turkey]	1	1.934375e+10
71	Oxford/AstraZeneca, Pfizer/BioNTech, Sinopharm...	[United Arab Emirates]	1	1.973593e+09
72	Moderna, Oxford/AstraZeneca, Pfizer/BioNTech, ...	[Uzbekistan]	1	2.648153e+08
73	Abdala, Moderna, Oxford/AstraZeneca, Pfizer/Bi...	[Vietnam]	1	3.551691e+09

74 rows × 4 columns

EDA on the manufacture vaccinations

Number of countries in Data:43

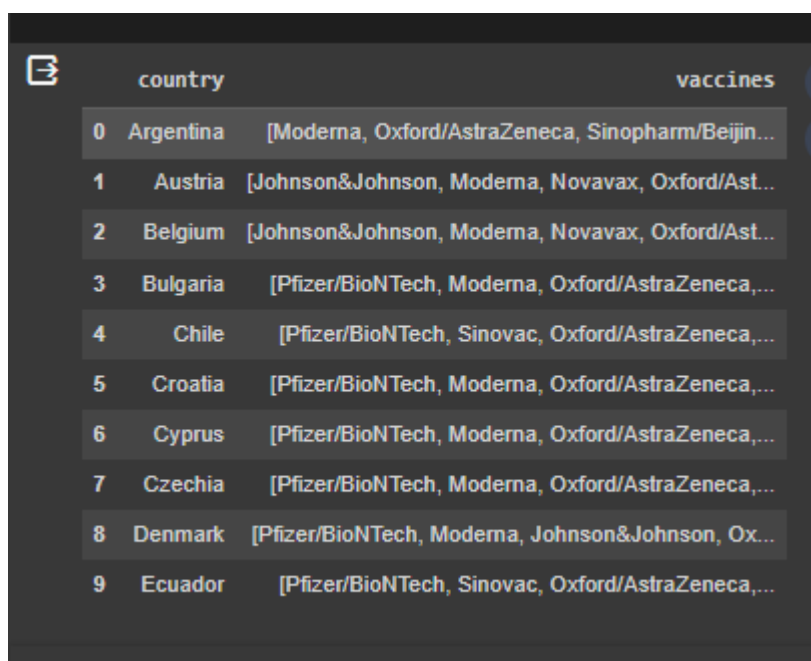
Number of specified in dataset:10

Countries total vaccinations first five entries,



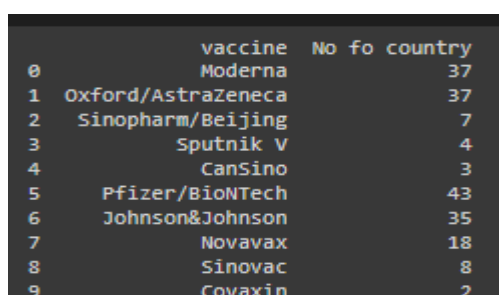
	location	total_vaccinations
0	Argentina	1.711444e+10
1	Austria	5.965148e+08
2	Belgium	8.343959e+08
3	Bulgaria	1.342383e+08
4	Chile	9.170587e+09

Countries and its used vaccinations(first 10 entry):



	country	vaccines
0	Argentina	[Moderna, Oxford/AstraZeneca, Sinopharm/Beijin...
1	Austria	[Johnson&Johnson, Moderna, Novavax, Oxford/Ast...
2	Belgium	[Johnson&Johnson, Moderna, Novavax, Oxford/Ast...
3	Bulgaria	[Pfizer/BioNTech, Moderna, Oxford/AstraZeneca,...
4	Chile	[Pfizer/BioNTech, Sinovac, Oxford/AstraZeneca,...
5	Croatia	[Pfizer/BioNTech, Moderna, Oxford/AstraZeneca,...
6	Cyprus	[Pfizer/BioNTech, Moderna, Oxford/AstraZeneca,...
7	Czechia	[Pfizer/BioNTech, Moderna, Oxford/AstraZeneca,...
8	Denmark	[Pfizer/BioNTech, Moderna, Johnson&Johnson, Ox...
9	Ecuador	[Pfizer/BioNTech, Sinovac, Oxford/AstraZeneca,...

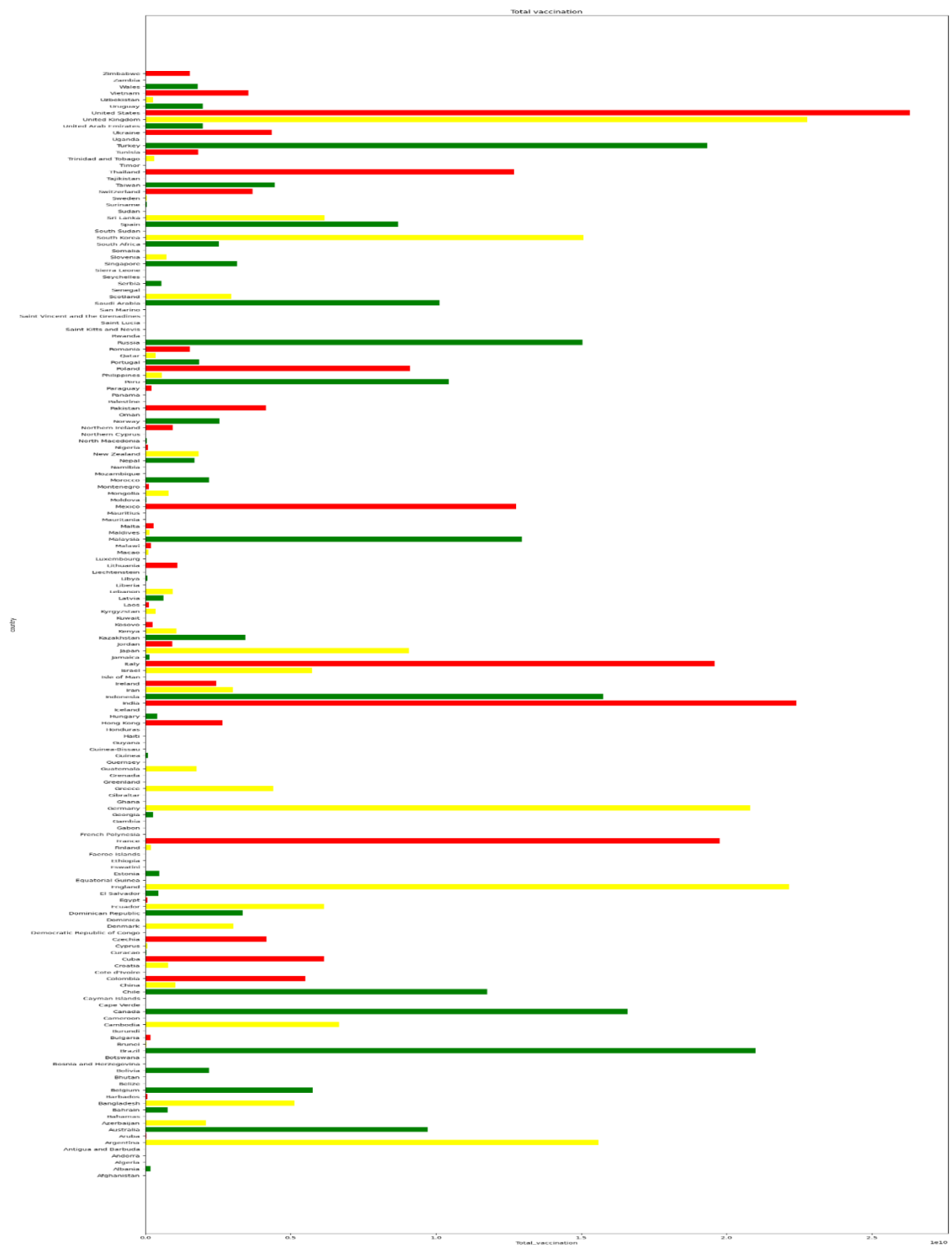
Vaccines and its number of countries used count:



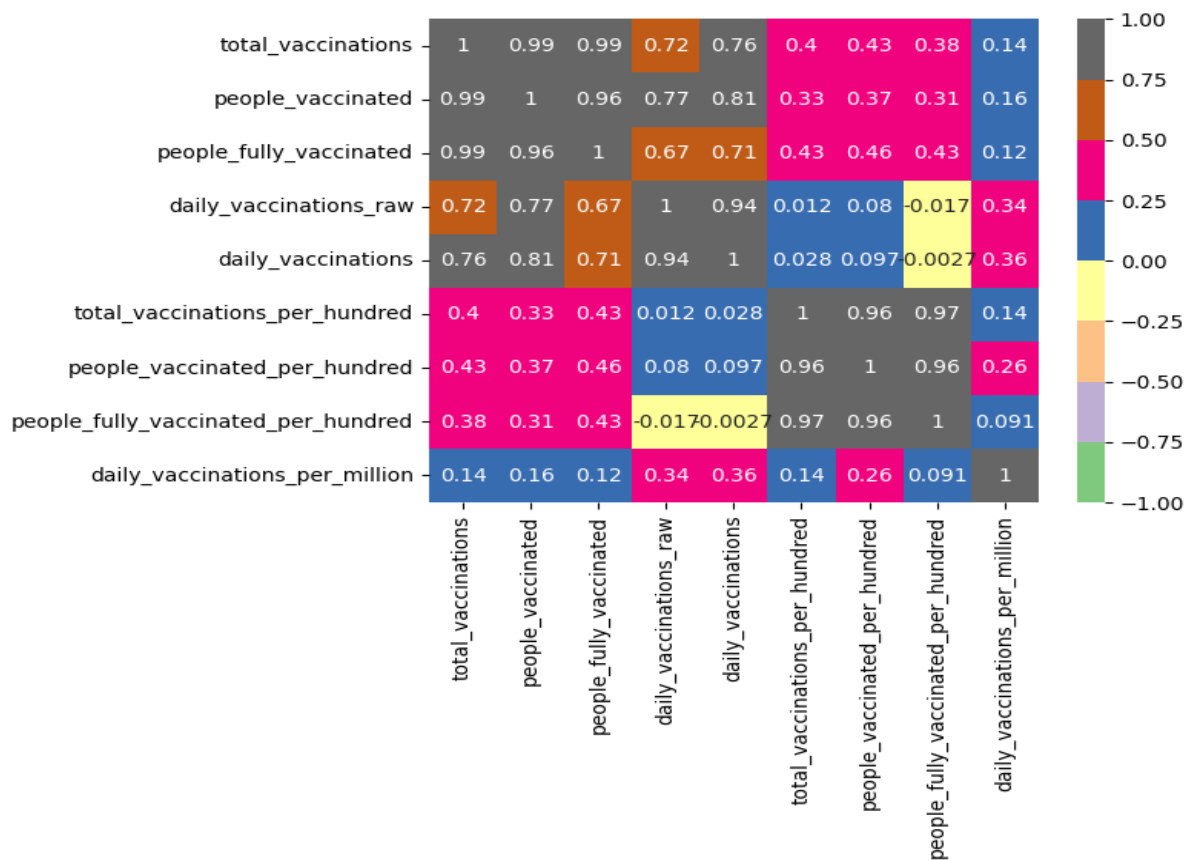
	vaccine	No fo country
0	Moderna	37
1	Oxford/AstraZeneca	37
2	Sinopharm/Beijing	7
3	Sputnik V	4
4	CanSino	3
5	Pfizer/BioNTech	43
6	Johnson&Johnson	35
7	Novavax	18
8	Sinovac	8
9	Covaxin	2

Visualization:

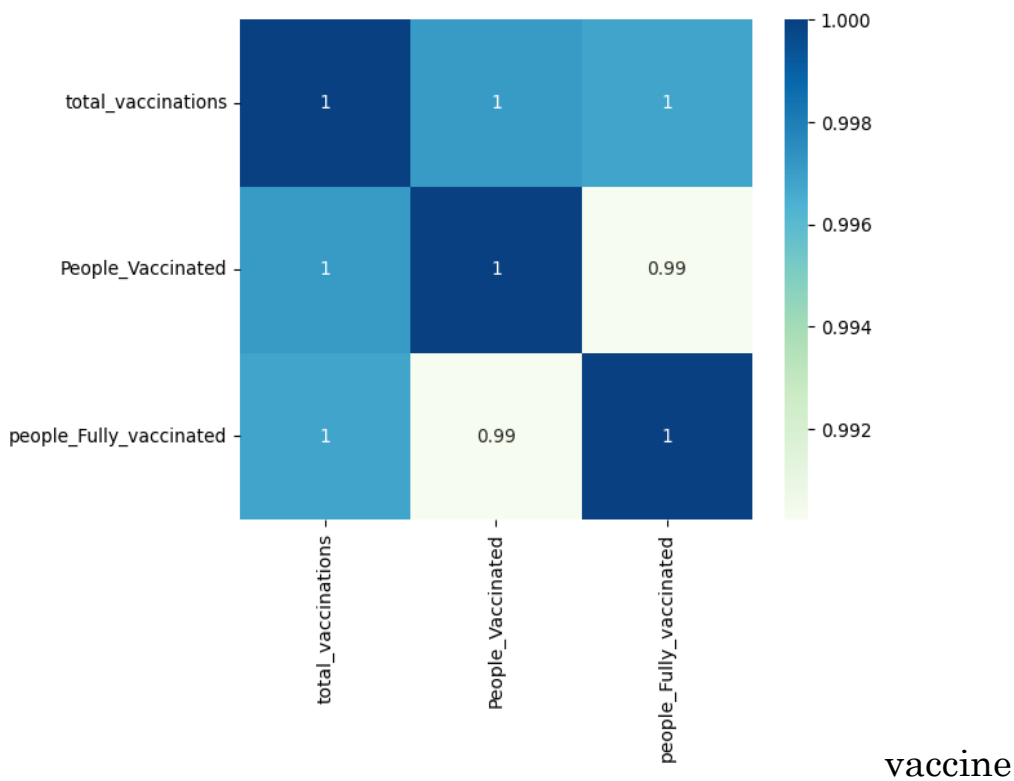
Country wise total vaccinations:



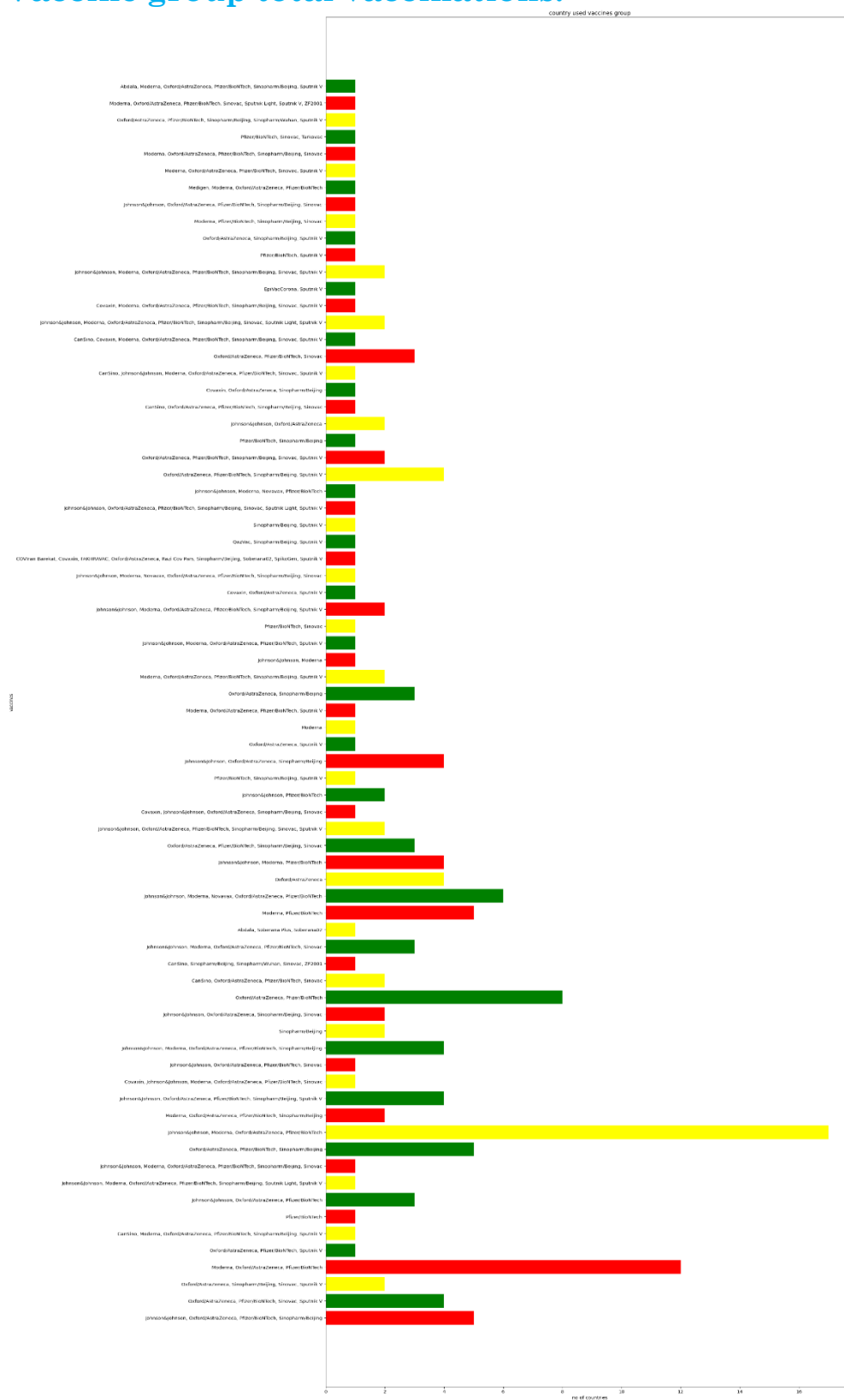
Country vacciantions Heatmap:



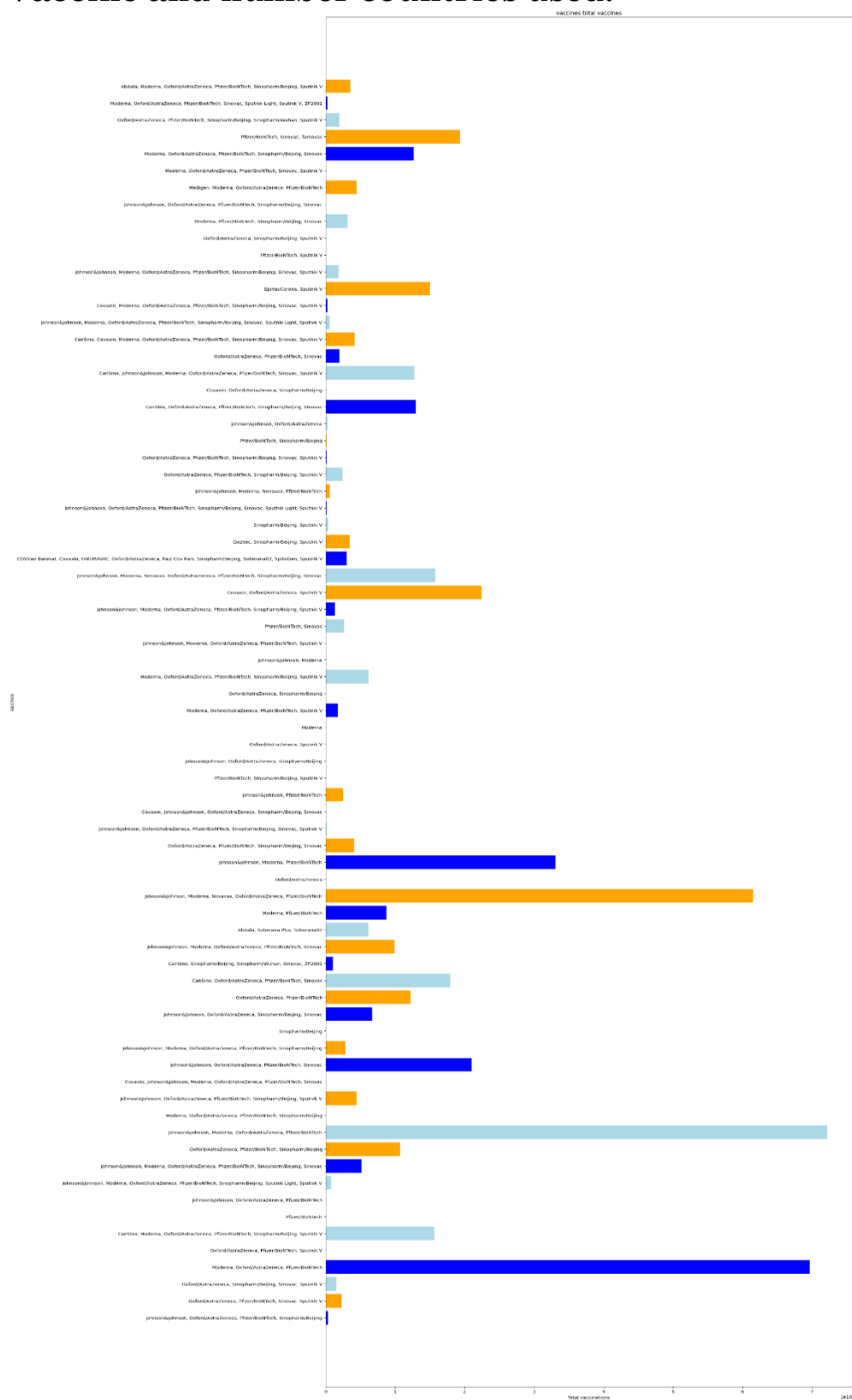
Heatmap for without date finded new dataframe



Vaccine group total vaccinations:



Vaccine and number countries used:



Machine Learning Technique:

Description	Value	
0	Session id	1265
1	Target	total_vaccinations
2	Target type	Regression
3	Original data shape	(30847, 15)
4	Transformed data shape	(30847, 15)
5	Transformed train set shape	(21592, 15)
6	Transformed test set shape	(9255, 15)
7	Numeric features	8
8	Categorical features	6
9	Preprocess	True
10	Imputation type	simple
11	Numeric imputation	mean
12	Categorical imputation	mode
13	Maximum one-hot encoding	25
14	Encoding method	None
15	Fold Generator	KFold
16	Fold Number	10
17	CPU Jobs	-1
18	Use GPU	False
19	Log Experiment	False
20	Experiment Name	reg-default-name
21	USI	19e4

	Model	MAE	MSE	RMSE	R2	RMS LE	MA PE	TT (Sec)
et	Extra Trees Regres sor	34493.37 69	24069774864.4 843	149981.5 199	1.00 00	0.018 1	0.00 45	6.46 00
rf	Rando m Forest Regres sor	71519.20 35	70872907863.2 902	262394.3 854	0.99 99	0.024 7	0.00 83	19.0 150
lightg bm	Light Gradie nt Boosti ng Machin e	157914.9 927	117269374154. 8219	340029.8 134	0.99 98	0.260 2	0.16 78	1.63 90
xgboo st	Extrem e Gradie nt Boosti ng	129275.2 648	90804408729.6 000	298858.0 938	0.99 98	0.195 3	0.07 58	0.66 10
dt	Decisio n Tree Regres sor	107110.1 125	148027334504. 1575	376612.9 728	0.99 97	0.034 9	0.01 39	0.43 80
gbr	Gradie nt Boosti ng Regres sor	356324.2 147	500508546030. 8246	706325.6 443	0.99 90	0.376 7	0.41 21	7.50 50
knn	K Neighb ors Regres sor	357565.5 188	611235836723. 2000	775262.3 375	0.99 87	0.374 0	0.77 09	0.47 00
br	Bayesi an Ridge	741982.3 081	1773742198117 .8313	1330690. 6318	0.99 63	0.727 7	1.25 03	0.23 40
en	Elastic Net	741609.0 001	1773813552992 .6714	1330715. 3628	0.99 63	0.733 3	1.26 17	0.51 80
ridge	Ridge Regres sion	742051.8 977	1773742366365 .3560	1330691. 0420	0.99 63	0.725 7	1.24 84	0.22 90

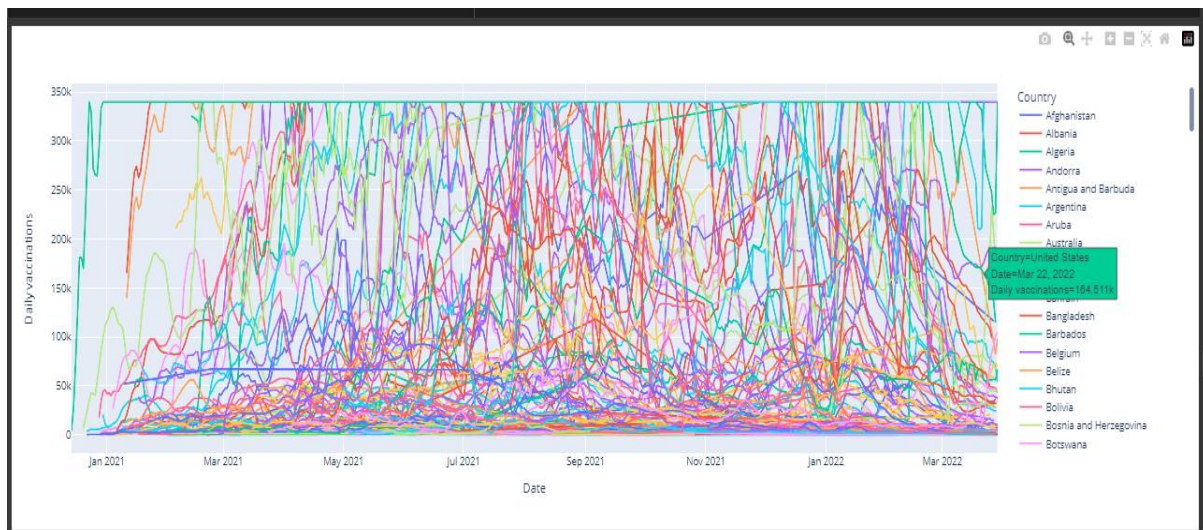
lasso	Lasso Regres sion	742051.9 349	1773742368075 .9712	1330691. 0428	0.99 63	0.725 7	1.24 84	0.85 00
lr	Linear Regres sion	742131.6 837	1773858148285 .1355	1330736. 3468	0.99 63	0.727 2	1.24 84	0.73 70
huber	Huber Regres sor	772739.8 265	2948895942392 .7075	1715874. 3694	0.99 39	0.403 8	0.18 58	0.60 10
par	Passive Aggres sive Regres sor	1250606. 7672	4579474547023 .5488	2076977. 2645	0.99 05	0.834 0	2.73 61	0.25 20
ada	AdaBo ost Regres sor	1832934. 0717	4750308947411 .0488	2177918. 6896	0.99 01	1.398 8	12.2 781	2.57 50
llar	Lasso Least Angle Regres sion	3754807. 9698	4805184705546 8.2344	5819314. 5749	0.90 00	1.315 3	14.9 825	0.45 40
lar	Least Angle Regres sion	7926486. 0272	1434535917475 71.4688	11440649 .6386	0.70 23	1.877 5	36.7 547	0.32 70
omp	Orthog onal Matchi ng Pursuit	8221054. 3718	1647713439436 52.4688	12828726 .6462	0.65 72	1.517 7	24.3 647	0.21 90
dum my	Dumm y Regres sor	18021419 .6000	4811366313492 48.0000	21932706 .8000	- 0.00 06	2.547 0	86.8 031	0.24 30

	MAE	MSE	RMSE	R2	RMSLE	MAPE
Fold						
0	29156.9509	13191665333.4341	114854.9752	1.0000	0.0160	0.0041
1	35629.6076	19401552298.0424	139289.4551	1.0000	0.0132	0.0042
2	42664.8630	65554078398.4964	256035.3069	0.9999	0.0182	0.0042
3	30525.3944	13048857808.6591	114231.5972	1.0000	0.0120	0.0041
4	35627.9274	21052843240.0043	145095.9794	1.0000	0.0155	0.0042
5	33808.5197	21612112593.8392	147010.5867	1.0000	0.0286	0.0056
6	33927.8836	23633581306.4297	153732.1739	1.0000	0.0357	0.0059
7	37107.5519	31043860933.5644	176192.6813	0.9999	0.0093	0.0037
8	32244.3700	17471985552.7957	132181.6385	1.0000	0.0198	0.0045
9	34240.7007	14687211179.5777	121190.8048	1.0000	0.0126	0.0044
Mean	34493.3769	24069774864.4843	149981.5199	1.0000	0.0181	0.0045
Std	3554.1291	14758275875.0401	39690.2828	0.0000	0.0078	0.0007

Final saved model

```
Transformation Pipeline and Model Successfully Saved
(Pipeline(memory=Memory(location=None),
  steps=[('numerical_imputer',
    TransformerWrapper(include=['people_vaccinated',
      'people_fully_vaccinated',
      'daily_vaccinations_raw',
      'daily_vaccinations',
      'total_vaccinations_per_hundred',
      'people_vaccinated_per_hundred',
      'people_fully_vaccinated_per_hundred',
      'daily_vaccinations_per_million'],
    transformer=SimpleImputer()...
    transformer=SimpleImputer(strategy='most_frequent'))),
  ('rest_encoding',
    TransformerWrapper(include=['country', 'iso_code', 'date',
      'vaccines', 'source_name',
      'source_website'],
    transformer=TargetEncoder(cols=['country',
      'iso_code',
      'date',
      'vaccines',
      'source_name',
      'source_website'],
    handle_missing='return_nan'))),
  ('trained_model',
    ExtraTreesRegressor(n_jobs=-1, random_state=8192))]),
```


Time Series Forecasting:



Providing Hidden Insights:

Some hidden insight I got,

- ✓ There among 225 country India has maximum People is fully vaccinated.
- ✓ Pitcarin has lowest vaccinated people
So mainly focus on that country
- ✓ In the start of covid Johnson & Johnson, Moderna, Pfizer has maximum vaccination all over the world
- ✓ But Covaxine plays major role in the covid-19 Vaccination in field.
- ✓ That was produced by our Indian government.

It could be assist for the pharmaceutical scientists
In the medicine field.

Conclusion:

This project could help vaccine distributors and countries on their vaccination. It should be more versatile when we create a model for prediction and analysis of the data. At last I find hidden insights should be valuable for workers on the field and manufacturers.

OUR TEAM MEMBERS:

K.MATHAN KUMAR

952421104035

https://github.com/MATHAN190/DAC_Phase1.git

R.KARUPPASAMY

952421104030

https://github.com/Karuppasamy200/DAC_Phase1.git

E.DINESH KANAGU

952421104021

<https://github.com/dineshsri4/ProjectDA.git>

P.ESAKKI DURAI

952421104022

https://github.com/Esaidurai/DAC_Phase1.git

