

```
# import libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
```

```
plt.rc('font', size=10)
%matplotlib inline
```

```
# import dataset from CSV
vac = '../country_vaccinations.csv'
manu = '../country_vaccinations_by_manufacturer.csv'
df_vac = pd.read_csv(vac, parse_dates= ['date'])
df_manu = pd.read_csv(manu, parse_dates = [])
df_manu.info()
```

```
<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
```

```
df_vac.tail(5)
```

	country	iso_code	date	total_vaccinations	people_vaccinated	people_fully_v
86507	Zimbabwe	ZWE	2022-03-25	8691642.0	4814582.0	
86508	Zimbabwe	ZWE	2022-03-26	8791728.0	4886242.0	
86509	Zimbabwe	ZWE	2022-03-27	8845039.0	4918147.0	
86510	Zimbabwe	ZWE	2022-03-28	8934360.0	4975433.0	
86511	Zimbabwe	ZWE	2022-03-29	9039729.0	5053114.0	

```
df_manu.head()
```

	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	

```
# the most popular vaccine
most_vac = df_manu.groupby(['vaccine'])[['location', 'date', 'total_vaccinations']].sum().sort_values(by = 'total_vaccinations', ascending = False)
most_vac['Total_vac_per_million'] = round(most_vac['total_vaccinations']/1000000,2)
most_vac['Percent_of_total_vac'] = round(100* most_vac['total_vaccinations']/most_vac['total_vaccinations'].sum(),2)
most_vac.reset_index(inplace = True)
most_vac
```

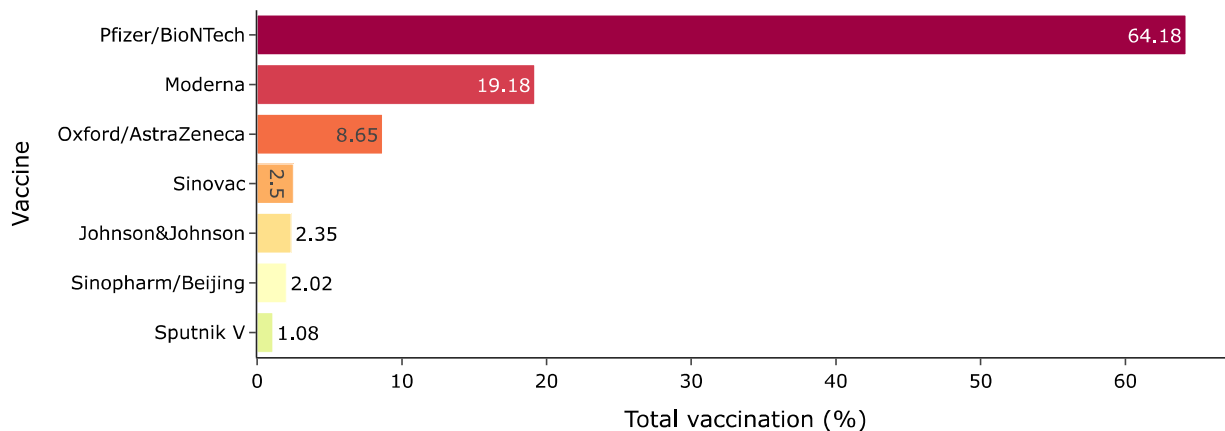
```
<ipython-input-5-ba5b9c3bd4be>:2: FutureWarning: The default value of numeric_only in DataFrameGroupBy.sum is deprecated. In a future ve
most_vac = df_manu.groupby(['vaccine'])[['location', 'date', 'total_vaccinations']].sum().sort_values(by = 'total_vaccinations', ascend
```

	vaccine	total_vaccinations	Total_vac_per_million	Percent_of_total_vac	
0	Pfizer/BioNTech	344835955037	344835.96	64.18	
1	Moderna	103072147621	103072.15	19.18	
2	Oxford/AstraZeneca	46451509497	46451.51	8.65	
3	Sinovac	13407163275	13407.16	2.50	
4	Johnson&Johnson	12611375881	12611.38	2.35	
5	Sinopharm/Beijing	10877006517	10877.01	2.02	
6	Sputnik V	5787343199	5787.34	1.08	
7	CanSino	271397675	271.40	0.05	
8	Novavax	8268113	8.27	0.00	
9	Covaxin	3572	0.00	0.00	

```
# Let's plot this for easy visualization
```

```
fig = px.bar(most_vac[:7], x="Percent_of_total_vac", y="vaccine", template = 'simple_white',
            width=1000, height=400, orientation = 'h', color = "vaccine",
            color_discrete_sequence=px.colors.diverging.Spectral, text_auto=True,
            labels=dict(Percent_of_total_vac = "Total vaccination (%)", vaccine="Vaccine")).update_xaxes(categoryorder = "total descending")
fig.update_layout(
    title="<b>The world most popular vaccine</b>",
    font=dict(
        size=14,
        color="black"), showlegend = False
)
fig.show()
```

## The world most popular vaccine



```
# The list of SEA countries which have the highest percentage of fully vaccinated people
```

```
sea = ['Brunei', 'India', 'Indonesia', 'Laos', 'Malaysia', 'Myanmar', 'Philippines', 'Singapore', 'Thailand', 'Vietnam']
df_vac_sea = df_vac[df_vac['country'].isin(sea)]
df_vac_sea_group = df_vac_sea.groupby(['country'])[['date', 'people_fully_vaccinated_per_hundred']].max().sort_values(by = 'people_fully_vacci
df_vac_sea_group
```

```

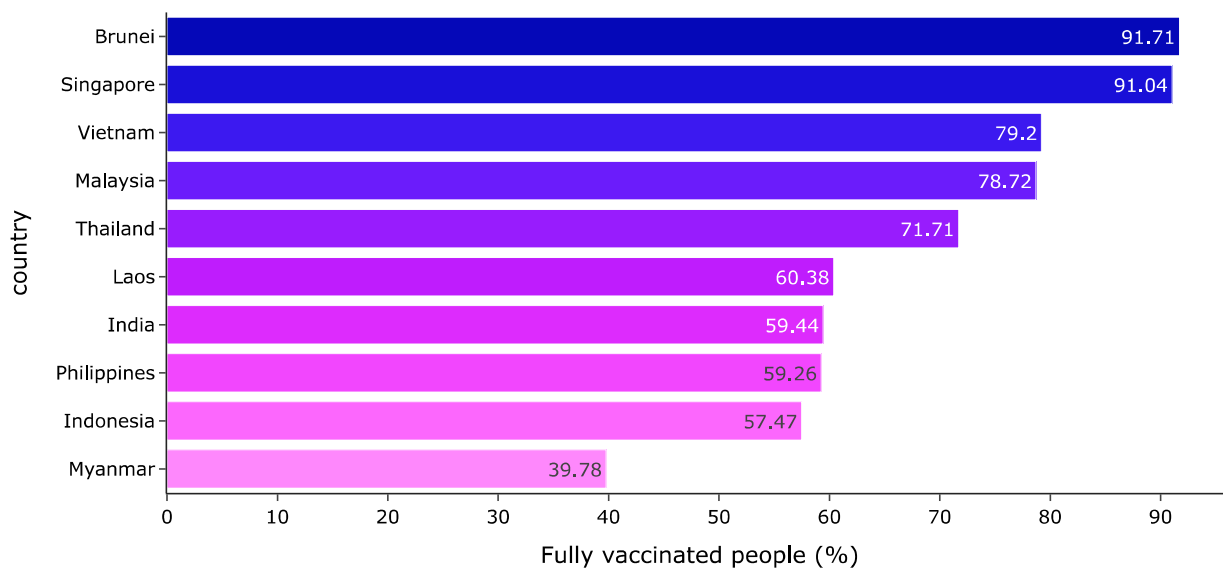
    date  people_fully_vaccinated_per_hundred
country
Brunei   2022-03-18                        91.71

df_vac_sea_group['iso_alpha'] = ["BRN","SGP","VNM", "MYS","THA","LAO","IND", "PHL", "IDN","MMR"]
vietnam   2022-03-22                        79.20

fig = px.bar(df_vac_sea_group, x= "people_fully_vaccinated_per_hundred", y= df_vac_sea_group.index, template = 'simple_white',
             width=1000, height=500 , orientation = 'h', color = df_vac_sea_group.index,
             color_discrete_sequence=px.colors.sequential.Plotly3, text_auto=True,
             labels=dict(people_fully_vaccinated_per_hundred = "Fully vaccinated people (%)")).update_xaxes(categoryorder = "total descending")
fig.update_layout(
    title="<b>SEA total fully vaccinated people (%)</b>",
    font=dict(
        size=14,
        color="black"),
    showlegend = False
)
fig.show()

```

### SEA total fully vaccinated people (%)



```

fig = px.choropleth(df_vac_sea_group, locations="iso_alpha",
                    color="people_fully_vaccinated_per_hundred",
                    width=900, height=600,
                    hover_name=df_vac_sea_group.index, # column to add to hover information
                    color_continuous_scale=px.colors.sequential.Plotly3[::-1],
                    labels=dict(people_fully_vaccinated_per_hundred = "Total fully vaccinated people(%)"))

fig.update_geos(fitbounds="locations", visible=True)
fig.update_layout(height=400,margin={"r":0,"t":0,"l":0,"b":0})
fig.show()

```

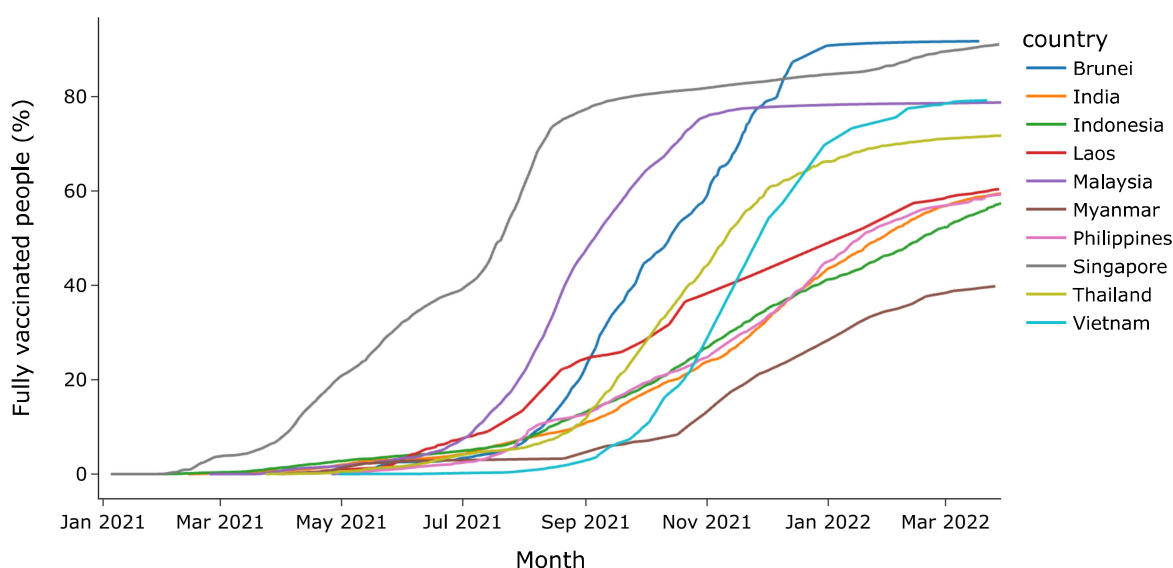
Total fully vaccinated people(%)



```
fig = px.line(df_vac_sea, x = 'date', y='people_fully_vaccinated_per_hundred', color = 'country', template="simple_white",
              width = 900, height = 500)
```

```
fig.update_layout(
    title="<b>Vaccination rate in SEA countries (%)</b>",
    xaxis_title="Month",
    yaxis_title="Fully vaccinated people (%)",
    font=dict(
        size=14,
        color="black"
    )
)
fig.update_traces(connectgaps=True)
fig.show()
```

### Vaccination rate in SEA countries (%)



```
# check NULL values in daily_vaccination data
df_vac_daily = df_vac_sea[['country', 'date', 'daily_vaccinations_per_million']]
top = ['Brunei', 'Singapore', 'Vietnam']
df_vac_daily = df_vac_daily[df_vac_daily['country'].isin(top)]
df_vac_daily.daily_vaccinations_per_million.isna().sum()
df_vac_daily[df_vac_daily['daily_vaccinations_per_million'].isna()]
```

	country	date	daily_vaccinations_per_million	
11395	Brunei	2021-04-02	NaN	
69775	Singapore	2020-12-30	NaN	
84250	Vietnam	2021-03-07	NaN	

```
# Fill NULL values with back values close to that NULL
df_vac_daily['daily_vaccinations_per_million'].fillna(method = 'bfill', inplace = True)
df_vac_daily.daily_vaccinations_per_million.isna().sum()
```

0

```
# Creating the Figure instance
fig = px.line(df_vac_daily, x= 'date' , y= 'daily_vaccinations_per_million', color = 'country', template="simple_white",
              width = 900, height = 500)

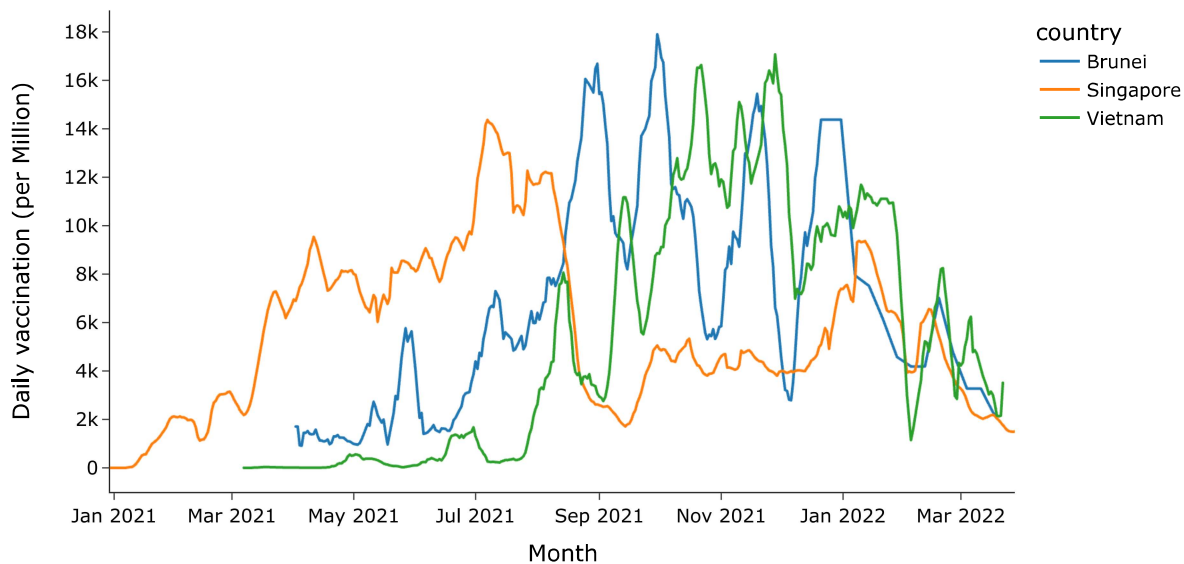
fig.update_layout(
    title="<b>Interactive daily vaccination rate</b>",
```

```

axis_title="Month",
yaxis_title="Daily vaccination (per Million)",
font=dict(
    size=14,
    color="black")
)
fig.show()

```

### Interactive daily vaccination rate



```
!pip install --upgrade pip
```

```

Requirement already satisfied: pip in /usr/local/lib/python3.10/dist-packages (23.1.2)
Collecting pip
  Downloading pip-23.3-py3-none-any.whl (2.1 MB)
    ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 2.1/2.1 MB 12.8 MB/s eta 0:00:00
Installing collected packages: pip
  Attempting uninstall: pip
    Found existing installation: pip 23.1.2
    Uninstalling pip-23.1.2:
      Successfully uninstalled pip-23.1.2
  Successfully installed pip-23.3

```

```
!pip install pystan~=2.14
```

```

Requirement already satisfied: pystan~=2.14 in /usr/local/lib/python3.10/dist-packages (2.19.1.1)
Requirement already satisfied: Cython!=0.25.1,>=0.22 in /usr/local/lib/python3.10/dist-packages (from pystan~=2.14) (3.0.3)
Requirement already satisfied: numpy>=1.7 in /usr/local/lib/python3.10/dist-packages (from pystan~=2.14) (1.23.5)
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system package manager. It is

```

```

# Fill NULL using interpolate
df_in = df_vac_sea[df_vac_sea['country'] == 'Indonesia'][['date', 'people_fully_vaccinated_per_hundred']]
df_in['people_fully_vaccinated_per_hundred'] = df_in['people_fully_vaccinated_per_hundred'].interpolate()
df_in = df_in.rename(columns={'people_fully_vaccinated_per_hundred': 'y', 'date': 'ds'})

```

## New Section

