COVID-19 World Vaccination Progress

Basic Visualization

1. **Vaccination by Country**
   1.1 Total Vaccinations
   1.2 People Vaccinated
   1.3 People Fully Vaccinated
2. **Vaccination by Country per Hundred**
   2.1 Total Vaccinations
   2.2 People Vaccinated
   2.3 People Fully Vaccinated
3. **Daily Vaccinations**
   3.1 Daily Vaccinations by Country
   3.2 Daily Vaccinations by Country per Million

Advanced Visualization

1. **Total Vaccination & 30-day Rolling**
2. **Daily Vaccination**
   2.1 Day of Week
   2.2 Month
3. **Total Vaccination Status Across Countries**

Import packages

In [1]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
```

Pandas default settings

In [2]:
```
# pd.set_option('display.max_rows', 500)
pd.set_option('display.max_columns', 30)
pd.set_option('display.float_format', '{:,.2f}'.format)
```

In [3]:
```
#Load Dataset

df_vaccination = pd.read_csv('../input/covid-world-vaccination-progress/country_vacci
nations.csv')

#data is from : https://gpreda/covid-world-vaccination-progress
```

Exploring the dataset

In [4]:
```
#Display first 5 rows
```

```
df_vaccination.head()
```

Out[4]:

	country	iso_code	date	total_vaccinations	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	vaccines	source_name	source_website
0	Afghanistan	AFG	2021-02-22	0.00	0.00	NaN	NaN	NaN	0.00	0.00	NaN	NaN	Johnson&Johnson, Oxford/AstraZeneca, Pfizer/Bi...	World Health Organization	https://covid19.who.int/
1	Afghanistan	AFG	2021-02-23	NaN	NaN	NaN	NaN	1,367.00	NaN	NaN	NaN	34.00	Johnson&Johnson, Oxford/AstraZeneca, Pfizer/Bi...	World Health Organization	https://covid19.who.int/
2	Afghanistan	AFG	2021-02-24	NaN	NaN	NaN	NaN	1,367.00	NaN	NaN	NaN	34.00	Johnson&Johnson, Oxford/AstraZeneca, Pfizer/Bi..	World Health Organization	https://covid19.who.int/

	country	iso_code	date	total_vaccinations	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	vaccines	source_name	source_website
													.		
3	Afghanistan	AFG	2021-02-25	NaN	NaN	NaN	NaN	1,367.00	NaN	NaN	NaN	34.00	Johnson&Johnson, Oxford/AstraZeneca, Pfizer/Bi...	World Health Organization	https://covid19.who.int/
4	Afghanistan	AFG	2021-02-26	NaN	NaN	NaN	NaN	1,367.00	NaN	NaN	NaN	34.00	Johnson&Johnson, Oxford/AstraZeneca, Pfizer/Bi...	World Health Organization	https://covid19.who.int/

In [5]:
```
df_vaccination.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 77709 entries, 0 to 77708
Data columns (total 15 columns):
 #   Column                               Non-Null Count  Dtype
---  ------                               --------------  -----
 0   country                              77709 non-null  object
 1   iso_code                             77709 non-null  object
 2   date                                 77709 non-null  object
 3   total_vaccinations                   40048 non-null  float64
 4   people_vaccinated                    37945 non-null  float64
```

```
 5   people_fully_vaccinated              35465 non-null  float64
 6   daily_vaccinations_raw               32591 non-null  float64
 7   daily_vaccinations                   77429 non-null  float64
 8   total_vaccinations_per_hundred       40048 non-null  float64
 9   people_vaccinated_per_hundred        37945 non-null  float64
 10  people_fully_vaccinated_per_hundred  35465 non-null  float64
 11  daily_vaccinations_per_million       77429 non-null  float64
 12  vaccines                             77709 non-null  object
 13  source_name                          77709 non-null  object
 14  source_website                       77709 non-null  object
dtypes: float64(9), object(6)
memory usage: 8.9+ MB
```

Content

The data (country vaccinations) contains the following information:

- **Country**- this is the country for which the vaccination information is provided;
- **Country ISO Code** - ISO code for the country;
- **Date** - date for the data entry; for some of the dates we have only the daily vaccinations, for others, only the (cumulative) total;
- **Total number of vaccinations** - this is the absolute number of total immunizations in the country;
- **Total number of people vaccinated** - a person, depending on the immunization scheme, will receive one or more (typically 2) vaccines; at a certain moment, the number of vaccination might be larger than the number of people;
- **Total number of people fully vaccinated** - this is the number of people that received the entire set of immunization according to the immunization scheme (typically 2); at a certain moment in time, there might be a certain number of people that received one vaccine and another number (smaller) of people that received all vaccines in the scheme;
- **Daily vaccinations (raw)** - for a certain data entry, the number of vaccination for that date/country;
- **Daily vaccinations** - for a certain data entry, the number of vaccination for that date/country;
- **Total vaccinations per hundred** - ratio (in percent) between vaccination number and total population up to the date in the country;
- **Total number of people vaccinated per hundred** - ratio (in percent) between population immunized and total population up to the date in the country;
- **Total number of people fully vaccinated per hundred** - ratio (in percent) between population fully immunized and total population up to the date in the country;
- **Daily vaccinations per million** - ratio (in ppm) between vaccination number and total population for the current date in the country;
- **Vaccines used in the country** - total number of vaccines used in the country (up to date);
- **Source name** - source of the information (national authority, international organization, local organization etc.);
- **Source website** - website of the source of information;

In [6]:

```
#Find the number or rows and columns

df_vaccination.shape
```

```
#There are 76095 rows and 15 columns
```

Out[6]:

```
(77709, 15)
```

In [7]:

```
df_vaccination.isnull().sum()

#There are no empty rows for country, iso_code or date columns.
```

Out[7]:

```
country                                  0
iso_code                                 0
date                                     0
total_vaccinations                   37661
people_vaccinated                    39764
people_fully_vaccinated              42244
daily_vaccinations_raw               45118
daily_vaccinations                     280
total_vaccinations_per_hundred       37661
people_vaccinated_per_hundred        39764
people_fully_vaccinated_per_hundred  42244
daily_vaccinations_per_million         280
vaccines                                 0
source_name                              0
source_website                           0
dtype: int64
```

In [8]:

```
# General Overview of the calculations in data

df_vaccination.describe()
```

Out[8]:

	total_vaccinations	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
count	40,048.00	37,945.00	35,465.00	32,591.00	77,429.00	40,048.00	37,945.00	35,465.00	77,429.00
mean	40,384,563.30	15,903,249.33	12,278,486.43	276,490.71	135,957.05	73.11	38.60	32.81	3,417.77
st	202,533,898.	63,330,	49,156,65	1,245,21	797,86	63.46	28.74	27.62	4,028.63

	total_vaccinations	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
d	56	416.02	1.29	0.53	8.94				
min	0.00	0.00	1.00	0.00	0.00	0.00	0.00	0.00	0.00
25%	472,141.25	314,538.00	211,664.00	5,240.00	972.00	13.31	9.72	5.58	704.00
50%	3,141,270.00	1,939,970.00	1,442,791.00	26,278.00	7,869.00	59.37	37.40	27.31	2,253.00
75%	15,091,858.25	8,102,781.00	6,399,728.00	129,159.00	45,644.00	123.68	65.33	58.32	4,933.00
max	3,063,391,000.00	1,266,426,000.00	1,228,340,000.00	24,741,000.00	22,424,286.00	333.76	123.75	121.14	117,497.00

Data Preparation

In [9]:

```
#drop the source_name,source_website and vaccine columns

df_vaccine_country = df_vaccination.drop(['source_name','source_website','vaccines'],
axis=1)
df_vaccine_country.head()
```

Out[9]:

	country	iso_code	date	total_vaccinations	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

	country	iso_code	date	total_vaccinations	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
0	Afghanistan	AFG	2021-02-22	0.00	0.00	NaN	NaN	NaN	0.00	0.00	NaN	NaN
1	Afghanistan	AFG	2021-02-23	NaN	NaN	NaN	NaN	1,367.00	NaN	NaN	NaN	34.00
2	Afghanistan	AFG	2021-02-24	NaN	NaN	NaN	NaN	1,367.00	NaN	NaN	NaN	34.00
3	Afghanistan	AFG	2021-02-25	NaN	NaN	NaN	NaN	1,367.00	NaN	NaN	NaN	34.00

	country	iso_code	date	total_vaccinations	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
4	Afghanistan	AFG	2021-02-26	NaN	NaN	NaN	NaN	1,367.00	NaN	NaN	NaN	34.00

In [10]:

```
# convert Date column to date type and fill na values with 0 for calculation

df_vaccine_country["date"] = pd.to_datetime(df_vaccine_country["date"], format = '%Y-%m-%d')

df_vaccine_country = df_vaccine_country.replace([np.inf, -np.inf], np.nan)
df_vaccine_country = df_vaccine_country.fillna(0)
df_vaccine_country.isnull().sum()
```

Out[10]:

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

In [11]:

```
#Function to find total, avergae, maximum  and minimum of different vaccinations status by country
def vaccination_country(col_name,func_name):

    '''
    Function that requires vaccination column name, and sum/mean/max/min function name as string arguments.
    '''
    if func_name == 'sum':
```

```
        return  (df_vaccine_country[['country',col_name]].groupby(by='country')
                                    .sum()
                                    .sort_values(by=col_name,ascending= False)
                                    .reset_index()
                                )

    elif func_name == 'mean':

        return  (df_vaccine_country[['country',col_name]].groupby(by='country')
                                    .mean()
                                    .sort_values(by=col_name,ascending= False)
                                    .reset_index()
                                   )
    elif func_name == 'max':

        return  (df_vaccine_country[['country',col_name]].groupby(by='country')
                                    .max()
                                    .sort_values(by=col_name,ascending= False)
                                    .reset_index()
                                   )
    elif func_name == 'min':

        return  (df_vaccine_country[['country',col_name]].groupby(by='country')
                                    .min()
                                    .sort_values(by=col_name,ascending= False)
                                    .reset_index()
                                   )
```

In [12]:

```
# Calculating different vaccinations for visualizations
max_total_vaccinations = vaccination_country('total_vaccinations','max')

sum_people_vaccinated = vaccination_country('people_vaccinated','sum')
sum_people_fully_vaccinated = vaccination_country('people_fully_vaccinated','sum')

avg_total_vaccinations = vaccination_country('total_vaccinations_per_hundred','mean')
avg_people_vaccinated = vaccination_country('people_vaccinated_per_hundred','mean')
avg_people_fully_vaccinated = vaccination_country('people_fully_vaccinated_per_hundre
d','mean')
avg_daily_vaccinations = vaccination_country('daily_vaccinations_per_million','mean')
```

In [13]:

```
#Function for Country with maximum and minimum daily vaccinations
def daily_vaccination_country(col_name,func_name):

    '''
    A function that requires daily_vaccination column and max/min function name as st
ring arguments.
    '''

    daily_vaccination = (df_vaccine_country
                                    .pivot_table(index='country',columns='date',values=c
ol_name)
                                       )
    if func_name == 'max':
```

```
        daily_vaccination['Highest Daily Vaccination'] = daily_vaccination.max(axis=1)
        daily_vaccination['Date - Highest Daily Vaccination'] = daily_vaccination.idx
max(axis=1)
        daily_vaccination.sort_values(by='Highest Daily Vaccination',ascending=False,
inplace=True)
        daily_vaccination.rename_axis('',axis=1,inplace=True)

        return daily_vaccination[['Highest Daily Vaccination','Date - Highest Daily V
accination']].reset_index()

    elif func_name == 'min':

        daily_vaccination.replace(0.00,np.nan,inplace=True)
        daily_vaccination['Lowest Daily Vaccination'] = daily_vaccination.min(axis=1)
        daily_vaccination['Date - Lowest Daily Vaccination'] = daily_vaccination.idxm
in(axis=1)
        daily_vaccination.sort_values(by='Lowest Daily Vaccination',ascending=False,i
nplace=True)
        daily_vaccination.rename_axis('',axis=1,inplace=True)

        return daily_vaccination[['Lowest Daily Vaccination','Date - Lowest Daily Vac
cination']].reset_index()
```

In [14]:

```
#Calculating highest and lowest daily vaccination and the respective dates.
highest_daily_vaccination = daily_vaccination_country('daily_vaccinations','max')
lowest_daily_vaccination = daily_vaccination_country('daily_vaccinations','min')
```

Data Visualization

1.1 Top & Bottom 5 Countries in terms of Total Vaccination

In [15]:

```
#Set sns theme and default figsize for all the sns visualizations.
sns.set_theme(style='whitegrid')
sns.set(rc={'figure.figsize' : (12,5)})

fig, axes  = plt.subplots(2,1)

sns.barplot(x='country',y='total_vaccinations',data=max_total_vaccinations.head(),ax=
axes[0])
axes[0].set(xlabel = '', ylabel = 'Total Vaccinations', title ='Top 5 Countries in te
rms of total vaccinations!')

sns.barplot(x='country',y='total_vaccinations',data=max_total_vaccinations.tail(),ax=
axes[1])
axes[1].set(xlabel = '', ylabel = 'Total Vaccinations', title ='Bottom 5 Countries in
 terms of total vaccinations!')


fig.tight_layout()
plt.show()
```

1.2 Top & Bottom 5 Countries in terms of People Vaccinated

In [16]:
```
fig, axes = plt.subplots(2,1)

sns.barplot(x='country',y='people_vaccinated',data=sum_people_vaccinated.head(),ax=ax
es[0])
axes[0].set(xlabel = '', ylabel = 'People Vaccinated', title ='Top 5 Countries in ter
ms of people vaccinated!')

sns.barplot(x='country', y='people_vaccinated',data=sum_people_vaccinated.tail(),ax=a
xes[1])
axes[1].set(xlabel = '', ylabel = 'People Vaccinated', title ='Bottom 5 Countries in
terms of people vaccinated!')


fig.tight_layout()
plt.show()
```

1.3 Top & Bottom 5 Countries in terms of People Fully Vaccinated

In [17]:
```
fig, axes = plt.subplots(2,1)

sns.barplot(x='country',y='people_fully_vaccinated',data=sum_people_fully_vaccinated.
head(),ax=axes[0])
axes[0].set(xlabel = '', ylabel = 'People Fully Vaccinated', title ='Top 5 Countries
in terms of people fully vaccinated!')

sns.barplot(x='country',y='people_fully_vaccinated',data=sum_people_fully_vaccinated.
tail(),ax=axes[1])
axes[1].set(xlabel = '', ylabel = 'People Fully Vaccinated', title ='Bottom 5 Countri
es in terms of people fully vaccinated!')


# plt.ticklabel_format(style='plain', axis='y') #Uncomment if y label needs to displa
y accurate values

fig.tight_layout()
plt.show()
```

2.1 Top & Bottom 5 Countries in terms of Total Vaccinations per Hundred

In [18]:
```
fig, axes = plt.subplots(2,1)

sns.barplot(x='country', y='total_vaccinations_per_hundred',data=avg_total_vaccinatio
ns.head(),ax=axes[0])
axes[0].set(xlabel='', ylabel='Average Vaccinations per 100', title='Top 5 Countries
in terms of average vaccinations per hundred!')

sns.barplot(x='country', y='total_vaccinations_per_hundred',data=avg_total_vaccinatio
ns.tail(),ax=axes[1])
axes[1].set(xlabel='', ylabel='Average Vaccinations per 100', title='Bottom 5 Countri
es in terms of average vaccinations per hundred!')
```

```
fig.tight_layout(h_pad=3)
plt.show()
```

2.2 Top & Bottom 5 Countries in terms of People Vaccinated per Hundred

In [19]:
```
fig, axes = plt.subplots(2,1)

sns.barplot(x='country', y='people_vaccinated_per_hundred',data=avg_people_vaccinated.
head(),ax=axes[0])
axes[0].set(xlabel='', ylabel='People Vaccinated per 100', title='Top 5 Countries in
terms of average people vaccinated per hundred!')

sns.barplot(x='country', y='people_vaccinated_per_hundred',data=avg_people_vaccinated.
tail(),ax=axes[1])
axes[1].set(xlabel='', ylabel='People Vaccinated per 100', title='Bottom 5 Countries
in terms of average people vaccinated per hundred!')

fig.tight_layout()
plt.show()
```

2.3 Top & Bottom 5 Countries in terms of People Fully Vaccinated per Hundred

In [20]:
```
fig, axes = plt.subplots(2,1)

sns.barplot(x='country', y='people_fully_vaccinated_per_hundred',data=avg_people_full
y_vaccinated.head(),ax=axes[0])
axes[0].set(xlabel='', ylabel='People Fully Vaccinated per 100', title='Top 5 Countri
es in terms of average people fully vaccinated per hundred!')

sns.barplot(x='country', y='people_fully_vaccinated_per_hundred',data=avg_people_full
y_vaccinated.tail(),ax=axes[1])
axes[1].set(xlabel='', ylabel='People Fully Vaccinated per 100', title='Bottom 5 Coun
tries in terms of average people fully vaccinated per hundred!')

fig.tight_layout(h_pad=3)
plt.show()
```

3.1 Highest & Lowest 5 Daily Vaccination by Country

unfold_moreShow hidden code

In [22]:
```
fig, axes = plt.subplots(1,2)

sns.barplot(data=daily_top5_highest,x="country", y="Highest Daily Vaccination",ax=axe
s[0],hue='Date - Highest Daily Vaccination')
axes[0].set(xlabel='',ylabel='Daily Vaccination',title='Highest Daily Vaccination by
Country')
```

```
sns.barplot(data=daily_top5_lowest,x="country", y="Lowest Daily Vaccination",ax=axes
[1],hue='Date - Lowest Daily Vaccination')
axes[1].set(xlabel='',ylabel='Daily Vaccination',title='Lowest Daily Vaccination by C
ountry')


# plt.ticklabel_format(style='plain', axis='y')
fig.tight_layout()
plt.show()
```

3.2 Top & Bottom 5 Daily Vaccination by Country per Million

In [23]:
```
fig, axes = plt.subplots(2,1)

sns.barplot(x='country', y='daily_vaccinations_per_million',data=avg_daily_vaccinatio
ns.head(),ax=axes[0])
axes[0].set(xlabel='', ylabel='Daily Vaccinations per Million', title='Top 5 Countrie
s in terms of daily vaccinations per million!')

sns.barplot(x='country', y='daily_vaccinations_per_million',data=avg_daily_vaccinatio
ns.tail(),ax=axes[1])
axes[1].set(xlabel='', ylabel='Daily Vaccinations per Million', title='Bottom 5 Count
ries in terms of daily vaccinations per million!')

fig.tight_layout(h_pad=3)
plt.show()
```

Advanced Data Visualization

Import Plotly Library

In [24]:
```
from plotly.offline import init_notebook_mode
import plotly.express as px
init_notebook_mode(connected=True)
```

1. Total Vaccination & 30-day Rolling by Top 5 Country

In [25]:
```
#Top 5 country with highest total vaccinations
list(max_total_vaccinations['country'].head())
```

Out[25]:
```
['China', 'India', 'United States', 'Brazil', 'Indonesia']
```

In [26]:
```
# Filter the top 5 countries and find their 30 day rolling average of total_vaccinati
ons
top5_country_total = ['China', 'India', 'United States', 'Brazil', 'Indonesia']
top5_country_total_day = df_vaccine_country[df_vaccine_country['country'].isin(top5_c
ountry_total)].copy()
top5_country_total_day['30 - Day Rolling'] = top5_country_total_day['total_vaccinatio
ns'].rolling(window=30).mean()
```

In [27]:
```
fig = px.line(top5_country_total_day,x="date",y="total_vaccinations",color='country',
                labels={"country" : 'Top 5 Country', 'date' : 'Date', 'total_vaccina
tions' : "Total Vaccinations"},
              title="Total Vaccination Progress - Top 5 Country",template='plotly_d
ark')

for country in top5_country_total_day['country'].unique():
    fig.add_scatter(x=top5_country_total_day[top5_country_total_day['country'] == cou
ntry]['date']
                    ,y=top5_country_total_day[top5_country_total_day['country'] == co
untry]['30 - Day Rolling']
                    ,mode="lines",name='30 Day Rolling Vaccination ' + country)
```

2.1 Daily Vaccination by Day of Week by Top 5 Country

In [28]:
```
# Get the name of the day of vaccinations
top5_country_total_day['Day of Week'] = top5_country_total_day['date'].apply(lambda x:
x.day_name())
top5_country_total_day['Day of Week'].unique()
```

Out[28]:
```
array(['Sunday', 'Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday',
       'Saturday'], dtype=object)
```

In [29]:
```
fig = px.box(top5_country_total_day,x='Day of Week',y='daily_vaccinations',color='cou
ntry',
             labels={"country" : 'Top 5 Country', 'daily_vaccinations' : "Daily Vacci
nation"},
             title="Dailly Vaccination by Day of Week - Top 5 Country",template='p
lotly_dark')
```

2.2 Daily Vaccination by Month by Top 5 Country

In [30]:
```
# Get the name of the month of vaccinations
top5_country_total_day['Month'] = top5_country_total_day['date'].apply(lambda x:x.mon
th_name())
top5_country_total_day['Month'].unique()
```

Out[30]:
```
array(['January', 'February', 'March', 'April', 'May', 'June', 'July',
       'August', 'September', 'October', 'November', 'December'],
      dtype=object)
```

In [31]:
linkcode
```
fig = px.bar(top5_country_total_day,x='Month',y='daily_vaccinations',color='country',
             labels={"country" : 'Top 5 Country', 'daily_vaccinations' : "Daily Vacci
nation"},
             title="Dailly Vaccination by Month - Top 5 Country",template='plotly_
dark')
```

3. Total Vaccination Status Across Countries

In [32]:

```
fig = px.choropleth(max_total_vaccinations,locations='country',locationmode='country
names',
                    color='total_vaccinations',hover_name="country", template='plotly_
dark',
                    title= 'Total Vaccination Status Across Countries',projection='nat
ural earth',
                    labels={'country' : 'Country','total_vaccinations' : 'Total Vaccin
ations'})
```