DSBDAL Mini Project

Use the following covid_vaccine_statewise.csv dataset and perform following analytics on the given dataset https://www.kaggle.com/sudalairajkumar/covid19-in-india?select=covid_vaccine_statewise.csv

a. Describe the dataset

b. Number of persons state wise vaccinated for first dose in India

c. Number of persons state wise vaccinated for second dose in India

d. Number of Males vaccinated

e. Number of females vaccinated

In [1]:

```
# Import necessary libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

Loading the Dataset, checking for null values and preprocessing data

In [2]:

```
# Read the dataset from the specified file path
df = pd.read_csv("/content/drive/MyDrive/TE/Colab Notebooks/Datasets/covid_vaccine_sta
df
```

Out[2]:

	Updated On	State	Total Doses Administered	Sessions	Sites	First Dose Administered	Second Dose Administered	Male (Doses Administered)
0	16/01/2021	India	48276.0	3455.0	2957.0	48276.0	0.0	NaN
1	17/01/2021	India	58604.0	8532.0	4954.0	58604.0	0.0	NaN
2	18/01/2021	India	99449.0	13611.0	6583.0	99449.0	0.0	NaN
3	19/01/2021	India	195525.0	17855.0	7951.0	195525.0	0.0	NaN
4	20/01/2021	India	251280.0	25472.0	10504.0	251280.0	0.0	NaN
...	...	...	...	...	...	...	...	...
7840	11/08/2021	West Bengal	NaN	NaN	NaN	NaN	NaN	NaN
7841	12/08/2021	West Bengal	NaN	NaN	NaN	NaN	NaN	NaN
7842	13/08/2021	West Bengal	NaN	NaN	NaN	NaN	NaN	NaN
7843	14/08/2021	West Bengal	NaN	NaN	NaN	NaN	NaN	NaN
7844	15/08/2021	West Bengal	NaN	NaN	NaN	NaN	NaN	NaN

7845 rows × 24 columns

In [3]:

```python
# Top five rows
print("The top five rows are: ")
df.head()
```

```
The top five rows are:
```

Out[3]:

	Updated On	State	Total Doses Administered	Sessions	Sites	First Dose Administered	Second Dose Administered	Male (Doses Administered)	Ac
0	16/01/2021	India	48276.0	3455.0	2957.0	48276.0	0.0	NaN	
1	17/01/2021	India	58604.0	8532.0	4954.0	58604.0	0.0	NaN	
2	18/01/2021	India	99449.0	13611.0	6583.0	99449.0	0.0	NaN	
3	19/01/2021	India	195525.0	17855.0	7951.0	195525.0	0.0	NaN	
4	20/01/2021	India	251280.0	25472.0	10504.0	251280.0	0.0	NaN	

5 rows × 24 columns

In [4]:

```python
# Last five rows
print("The last five rows are: ")
df.tail()
```

```
The last five rows are:
```

Out[4]:

	Updated On	State	Total Doses Administered	Sessions	Sites	First Dose Administered	Second Dose Administered	Male (Doses Administered)
7840	11/08/2021	West Bengal	NaN	NaN	NaN	NaN	NaN	NaN
7841	12/08/2021	West Bengal	NaN	NaN	NaN	NaN	NaN	NaN
7842	13/08/2021	West Bengal	NaN	NaN	NaN	NaN	NaN	NaN
7843	14/08/2021	West Bengal	NaN	NaN	NaN	NaN	NaN	NaN
7844	15/08/2021	West Bengal	NaN	NaN	NaN	NaN	NaN	NaN

5 rows × 24 columns

In [5]:

```
# Shape of the dataset in the format of (rows, columns)
print("The shape is: ")
df.shape
```

```
The shape is: 
```

Out[5]:

```
(7845, 24)
```

In [6]:

```
# Display information about the DataFrame
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7845 entries, 0 to 7844
Data columns (total 24 columns):
 #   Column                              Non-Null Count  Dtype
---  ------                              --------------  -----
 0   Updated On                          7845 non-null   object
 1   State                               7845 non-null   object
 2   Total Doses Administered            7621 non-null   float64
 3   Sessions                            7621 non-null   float64
 4    Sites                              7621 non-null   float64
 5   First Dose Administered             7621 non-null   float64
 6   Second Dose Administered            7621 non-null   float64
 7   Male (Doses Administered)           7461 non-null   float64
 8   Female (Doses Administered)         7461 non-null   float64
 9   Transgender (Doses Administered)    7461 non-null   float64
 10   Covaxin (Doses Administered)       7621 non-null   float64
 11  CoviShield (Doses Administered)     7621 non-null   float64
 12  Sputnik V (Doses Administered)      2995 non-null   float64
 13  AEFI                                5438 non-null   float64
 14  18-44 Years (Doses Administered)    1702 non-null   float64
 15  45-60 Years (Doses Administered)    1702 non-null   float64
 16  60+ Years (Doses Administered)      1702 non-null   float64
 17  18-44 Years(Individuals Vaccinated) 3733 non-null   float64
 18  45-60 Years(Individuals Vaccinated) 3734 non-null   float64
 19  60+ Years(Individuals Vaccinated)   3734 non-null   float64
 20  Male(Individuals Vaccinated)        160 non-null    float64
 21  Female(Individuals Vaccinated)      160 non-null    float64
 22  Transgender(Individuals Vaccinated) 160 non-null    float64
 23  Total Individuals Vaccinated        5919 non-null   float64
dtypes: float64(22), object(2)
memory usage: 1.4+ MB
```

a. Describe the dataset.

In [7]:

```
# Display descriptive statistics of the  DataFrame
#It gives the output as mean, maximum, minimum, count etc.
df.describe()
```

Out[7]:

	Total Doses Administered	Sessions	Sites	First Dose Administered	Second Dose Administered	Male (Doses Administered)	Admini
count	7.621000e+03	7.621000e+03	7621.000000	7.621000e+03	7.621000e+03	7.461000e+03	7.4610
mean	9.188171e+06	4.792358e+05	2282.872064	7.414415e+06	1.773755e+06	3.620156e+06	3.1684
std	3.746180e+07	1.911511e+06	7275.973730	2.995209e+07	7.570382e+06	1.737938e+07	1.5153
min	7.000000e+00	0.000000e+00	0.000000	7.000000e+00	0.000000e+00	0.000000e+00	2.0000
25%	1.356570e+05	6.004000e+03	69.000000	1.166320e+05	1.283100e+04	5.655500e+04	5.2107
50%	8.182020e+05	4.547000e+04	597.000000	6.614590e+05	1.388180e+05	3.897850e+05	3.3423
75%	6.625243e+06	3.428690e+05	1708.000000	5.387805e+06	1.166434e+06	2.735777e+06	2.5615
max	5.132284e+08	3.501031e+07	73933.000000	4.001504e+08	1.130780e+08	2.701636e+08	2.3951

8 rows × 22 columns

In [8]:

```
df.describe(include='object')
```

Out[8]:

	Updated On	State
count	7845	7845
unique	213	37
top	16/01/2021	Delhi
freq	37	213

In [9]:

```
# Names of columns
print("The columns present in the dataset are: ")
df.columns
```

Out[9]:

```
The columns present in the dataset are:
Index(['Updated On', 'State', 'Total Doses Administered', 'Sessions',
       ' Sites ', 'First Dose Administered', 'Second Dose Administered',
       'Male (Doses Administered)', 'Female (Doses Administered)',
       'Transgender (Doses Administered)', ' Covaxin (Doses Administered)',
       'CoviShield (Doses Administered)', 'Sputnik V (Doses Administered)',
       'AEFI', '18-44 Years (Doses Administered)',
       '45-60 Years (Doses Administered)', '60+ Years (Doses Administered)',
       '18-44 Years(Individuals Vaccinated)',
       '45-60 Years(Individuals Vaccinated)',
       '60+ Years(Individuals Vaccinated)', 'Male(Individuals Vaccinated)',
       'Female(Individuals Vaccinated)', 'Transgender(Individuals Vaccinated)',
       'Total Individuals Vaccinated'],
      dtype='object')
```

In [10]:

```
df.columns.values
```

Out[10]:

```
array(['Updated On', 'State', 'Total Doses Administered', 'Sessions',
       ' Sites ', 'First Dose Administered', 'Second Dose Administered',
       'Male (Doses Administered)', 'Female (Doses Administered)',
       'Transgender (Doses Administered)',
       ' Covaxin (Doses Administered)', 'CoviShield (Doses Administered)',
       'Sputnik V (Doses Administered)', 'AEFI',
       '18-44 Years (Doses Administered)',
       '45-60 Years (Doses Administered)',
       '60+ Years (Doses Administered)',
       '18-44 Years(Individuals Vaccinated)',
       '45-60 Years(Individuals Vaccinated)',
       '60+ Years(Individuals Vaccinated)',
       'Male(Individuals Vaccinated)', 'Female(Individuals Vaccinated)',
       'Transgender(Individuals Vaccinated)',
       'Total Individuals Vaccinated'], dtype=object)
```

In [11]:

```
# specify the datatype of each feature
df.dtypes
```

Out[11]:

```
Updated On                              object
State                                   object
Total Doses Administered               float64
Sessions                               float64
 Sites                                 float64
First Dose Administered                float64
Second Dose Administered               float64
Male (Doses Administered)              float64
Female (Doses Administered)            float64
Transgender (Doses Administered)       float64
 Covaxin (Doses Administered)          float64
CoviShield (Doses Administered)        float64
Sputnik V (Doses Administered)         float64
AEFI                                   float64
18-44 Years (Doses Administered)       float64
45-60 Years (Doses Administered)       float64
60+ Years (Doses Administered)         float64
18-44 Years(Individuals Vaccinated)    float64
45-60 Years(Individuals Vaccinated)    float64
60+ Years(Individuals Vaccinated)      float64
Male(Individuals Vaccinated)           float64
Female(Individuals Vaccinated)         float64
Transgender(Individuals Vaccinated)    float64
Total Individuals Vaccinated           float64
dtype: object
```

In [12]:

```
# to check the missing values
df.isnull().sum()
```

Out[12]:

```
Updated On                                0
State                                     0
Total Doses Administered                224
Sessions                                224
 Sites                                  224
First Dose Administered                 224
Second Dose Administered                224
Male (Doses Administered)               384
Female (Doses Administered)             384
Transgender (Doses Administered)        384
 Covaxin (Doses Administered)           224
CoviShield (Doses Administered)         224
Sputnik V (Doses Administered)         4850
AEFI                                   2407
18-44 Years (Doses Administered)       6143
45-60 Years (Doses Administered)       6143
60+ Years (Doses Administered)         6143
18-44 Years(Individuals Vaccinated)    4112
45-60 Years(Individuals Vaccinated)    4111
60+ Years(Individuals Vaccinated)      4111
Male(Individuals Vaccinated)           7685
Female(Individuals Vaccinated)         7685
Transgender(Individuals Vaccinated)    7685
Total Individuals Vaccinated           1926
dtype: int64
```

Inference: As there are many NULL values present in the given dataset. We need to replace those values by mean(in case of numerical data) or mode(in case of categorical data).

b. Number of persons state wise vaccinated for first dose in India

Here, we need to work on "First Dose Administered". It is of float datatype and, hence we will replace the Nan Values by mean(average).

In [13]:

```
# Average of First Dose Administered
avg_firstdose = df["First Dose Administered"].astype("float").mean(axis = 0)
print("Average of First Dose:", avg_firstdose)
```

```
Average of First Dose: 7414415.300354284
```

In [14]:

```
# Replacing First Dose Administered
df["First Dose Administered"].fillna(value = avg_firstdose, inplace=True)
df
```

Out[14]:

	Updated On	State	Total Doses Administered	Sessions	Sites	First Dose Administered	Second Dose Administered	Male (Dose Administered
0	16/01/2021	India	48276.0	3455.0	2957.0	4.827600e+04	0.0	NaN
1	17/01/2021	India	58604.0	8532.0	4954.0	5.860400e+04	0.0	NaN
2	18/01/2021	India	99449.0	13611.0	6583.0	9.944900e+04	0.0	NaN
3	19/01/2021	India	195525.0	17855.0	7951.0	1.955250e+05	0.0	NaN
4	20/01/2021	India	251280.0	25472.0	10504.0	2.512800e+05	0.0	NaN
...	...	...	...	...	...	...	...	.
7840	11/08/2021	West Bengal	NaN	NaN	NaN	7.414415e+06	NaN	NaN
7841	12/08/2021	West Bengal	NaN	NaN	NaN	7.414415e+06	NaN	NaN
7842	13/08/2021	West Bengal	NaN	NaN	NaN	7.414415e+06	NaN	NaN
7843	14/08/2021	West Bengal	NaN	NaN	NaN	7.414415e+06	NaN	NaN
7844	15/08/2021	West Bengal	NaN	NaN	NaN	7.414415e+06	NaN	NaN

7845 rows × 24 columns

In [15]:

```
# To calculate the Number of persons state wise vaccinated for first dose in India
first_dose = df.groupby('State')[['First Dose Administered']].sum()
first_dose
```

Out[15]:

	First Dose Administered
State	
Andaman and Nicobar Islands	6.091235e+07
Andhra Pradesh	1.277347e+09
Arunachal Pradesh	9.349147e+07
Assam	6.300867e+08
Bihar	1.514989e+09
Chandigarh	8.918960e+07
Chhattisgarh	8.404894e+08
Dadra and Nagar Haveli and Daman and Diu	8.549597e+07
Delhi	6.762404e+08
Goa	1.204779e+08
Gujarat	2.176133e+09
Haryana	8.002848e+08
Himachal Pradesh	3.607805e+08
India	2.830663e+10
Jammu and Kashmir	4.545883e+08
Jharkhand	6.481602e+08
Karnataka	1.917816e+09
Kerala	1.238332e+09
Ladakh	6.229574e+07
Lakshadweep	4.885015e+07
Madhya Pradesh	1.841091e+09
Maharashtra	2.828851e+09
Manipur	1.118961e+08
Meghalaya	1.071025e+08
Mizoram	9.235957e+07
Nagaland	8.689726e+07
Odisha	1.077120e+09
Puducherry	8.583335e+07
Punjab	6.288331e+08
Rajasthan	2.245531e+09
Sikkim	8.146742e+07
Tamil Nadu	1.333019e+09
Telangana	9.248071e+08

	First Dose Administered
State	
Tripura	2.371762e+08
Uttar Pradesh	2.832898e+09
Uttarakhand	4.076779e+08
West Bengal	1.840936e+09

c. Number of persons state wise vaccinated for second dose in India

Here, we need to work on "Second Dose Administered". It is of float datatype and, hence we will replace the Nan Values by mean(average).

In [16]:

```
# Average of Second Dose Administered
avg_seconddose = df["Second Dose Administered"].astype("float").mean(axis = 0)
print("Average of Second Dose:", avg_seconddose)
```

```
Average of Second Dose: 1773755.2436688098
```

In [24]:

```
# Replacing Second Dose Administered
df["Second Dose Administered"].fillna(value = avg_seconddose, inplace = True)
df
```

Out[24]:

	Updated On	State	Total Doses Administered	Sessions	Sites	First Dose Administered	Second Dose Administered	Male (Dose Administered
0	16/01/2021	India	48276.0	3455.0	2957.0	4.827600e+04	0.000000e+00	Na
1	17/01/2021	India	58604.0	8532.0	4954.0	5.860400e+04	0.000000e+00	Na
2	18/01/2021	India	99449.0	13611.0	6583.0	9.944900e+04	0.000000e+00	Na
3	19/01/2021	India	195525.0	17855.0	7951.0	1.955250e+05	0.000000e+00	Na
4	20/01/2021	India	251280.0	25472.0	10504.0	2.512800e+05	0.000000e+00	Na
...	...	...	...	...	...	...	...	.
7840	11/08/2021	West Bengal	NaN	NaN	NaN	7.414415e+06	1.773755e+06	Na
7841	12/08/2021	West Bengal	NaN	NaN	NaN	7.414415e+06	1.773755e+06	Na
7842	13/08/2021	West Bengal	NaN	NaN	NaN	7.414415e+06	1.773755e+06	Na
7843	14/08/2021	West Bengal	NaN	NaN	NaN	7.414415e+06	1.773755e+06	Na
7844	15/08/2021	West Bengal	NaN	NaN	NaN	7.414415e+06	1.773755e+06	Na

7845 rows × 24 columns

In [18]:

```
second_dose = df.groupby('State')[['Second Dose Administered']].sum()
second_dose
```

Out[18]:

	Second Dose Administered
State	
Andaman and Nicobar Islands	1.476109e+07
Andhra Pradesh	3.694601e+08
Arunachal Pradesh	2.257485e+07
Assam	1.414313e+08
Bihar	2.814331e+08
Chandigarh	2.223627e+07
Chhattisgarh	1.827629e+08
Dadra and Nagar Haveli and Daman and Diu	1.701070e+07
Delhi	2.006352e+08
Goa	2.684071e+07
Gujarat	6.110609e+08
Haryana	1.692986e+08
Himachal Pradesh	8.448111e+07
India	6.770264e+09
Jammu and Kashmir	9.659418e+07
Jharkhand	1.327636e+08
Karnataka	4.378297e+08
Kerala	3.746913e+08
Ladakh	1.609629e+07
Lakshadweep	1.169898e+07
Madhya Pradesh	3.275755e+08
Maharashtra	7.235236e+08
Manipur	2.250068e+07
Meghalaya	2.280916e+07
Mizoram	2.064095e+07
Nagaland	1.984717e+07
Odisha	2.619453e+08
Puducherry	1.925139e+07
Punjab	1.317635e+08
Rajasthan	5.023455e+08
Sikkim	2.036617e+07
Tamil Nadu	3.013132e+08
Telangana	2.087955e+08

	Second Dose Administered
State	
Tripura	7.591267e+07
Uttar Pradesh	5.650776e+08
Uttarakhand	1.107276e+08
West Bengal	5.967894e+08

d. Number of Males vaccinated

In [22]:

```
male = df["Male(Individuals Vaccinated)"].sum()
print("The total number of male individuals vaccinated are:", int(male))
```

```
The total number of male individuals vaccinated are: 7138698858
```

e.Number of females vaccinated

In [27]:

```
female = df["Female(Individuals Vaccinated)"].sum()
print("The total number of female individuals vaccinated are:", int(female))
```

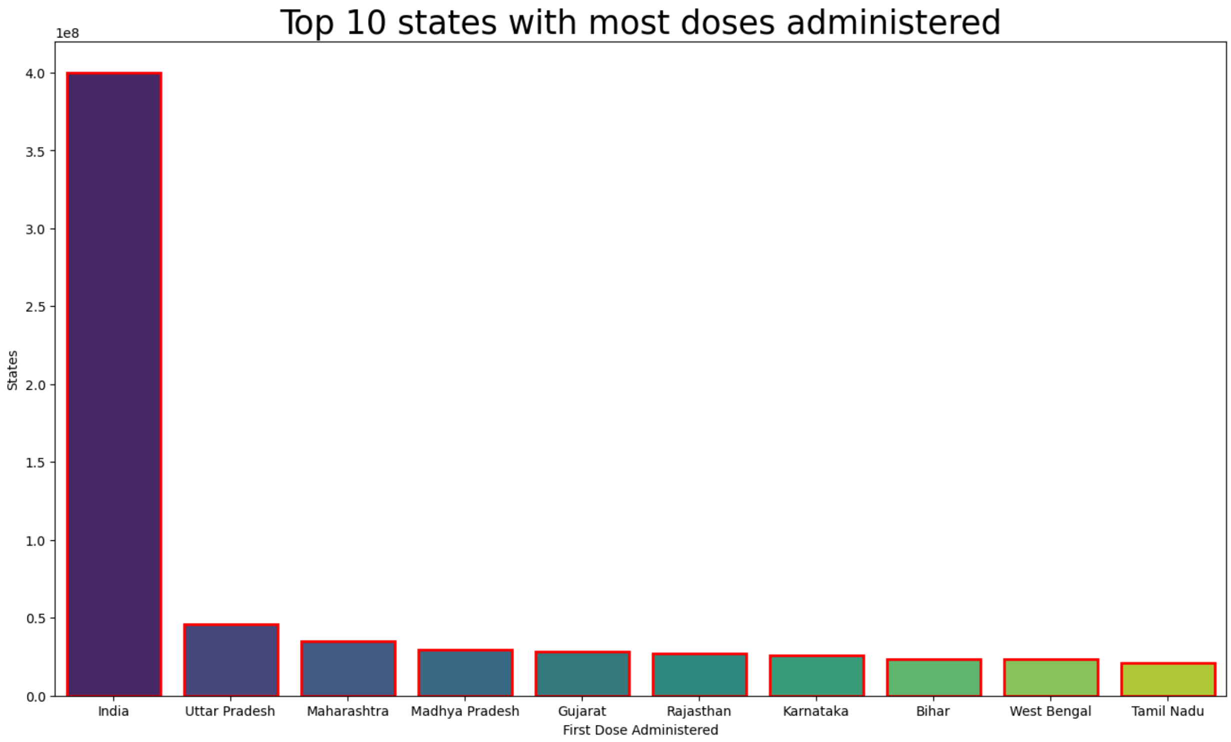
```
The total number of female individuals vaccinated are: 6321628736
```

Data Visualization

In [48]:

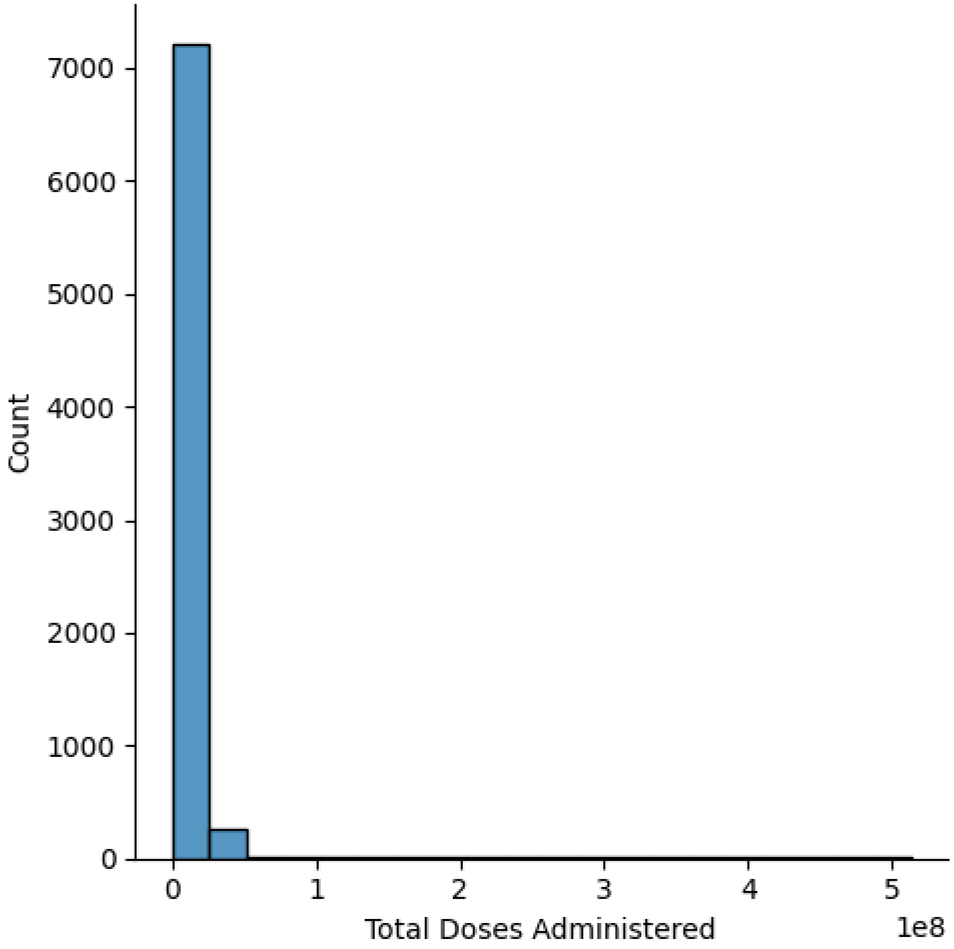
```
import matplotlib.pyplot as plt
import seaborn as sns

# Group by State/UnionTerritory and get the maximum value of "First Dose Administered"
top_10_active_cases = df.groupby(by="State")["First Dose Administered"].max().reset_ir

# Sort the DataFrame by "First Dose Administered" in descending order
top_10_active_cases = top_10_active_cases.sort_values(by="First Dose Administered", as

# Plotting
plt.figure(figsize=(16, 9))
plt.title("Top 10 states with most doses administered", size=25)
ax = sns.barplot(data = top_10_active_cases.iloc[:10],y = "First Dose Administered",x=
#ax = sns.barplot(data = top_10_active_cases, x="First Dose Administered", y="State",
plt.xlabel("First Dose Administered")
plt.ylabel("States")
plt.show()
```

```
<ipython-input-48-f901f589a563>:13: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.
0. Assign the `x` variable to `hue` and set `legend=False` for the same effect.

  ax = sns.barplot(data = top_10_active_cases.iloc[:10],y = "First Dose Administere
d",x= "State",linewidth = 2, edgecolor = "red", palette="viridis")
```



```
In [20]: sns.displot(df['Total Doses Administered'], bins = 20)
```

```
Out[20]: <seaborn.axisgrid.FacetGrid at 0x7fe1343d3490>
```

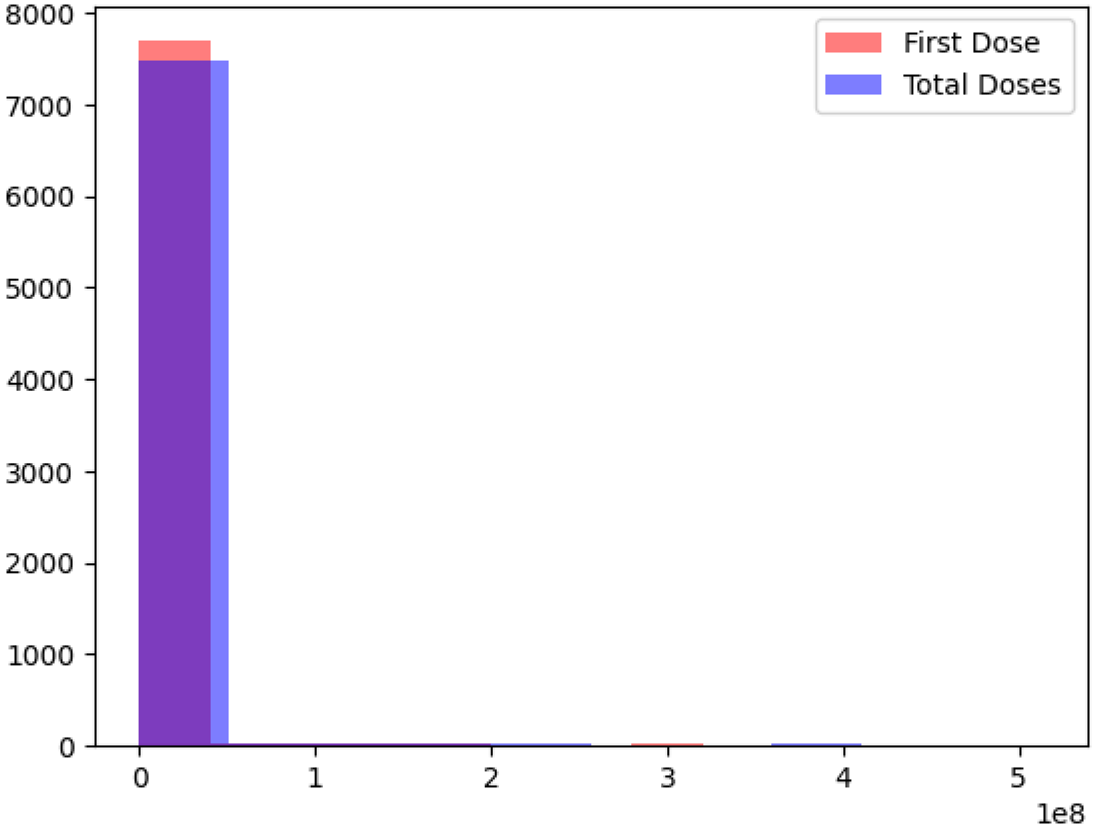


In [36]:

```
plt.hist(df['First Dose Administered'],color='red',label='First Dose',alpha=0.5)
plt.hist(df['Total Doses Administered'],color='blue',label='Total Doses',alpha=0.5)
plt.legend()
```

Out[36]:

```
<matplotlib.legend.Legend at 0x7fe130cfe8f0>
```

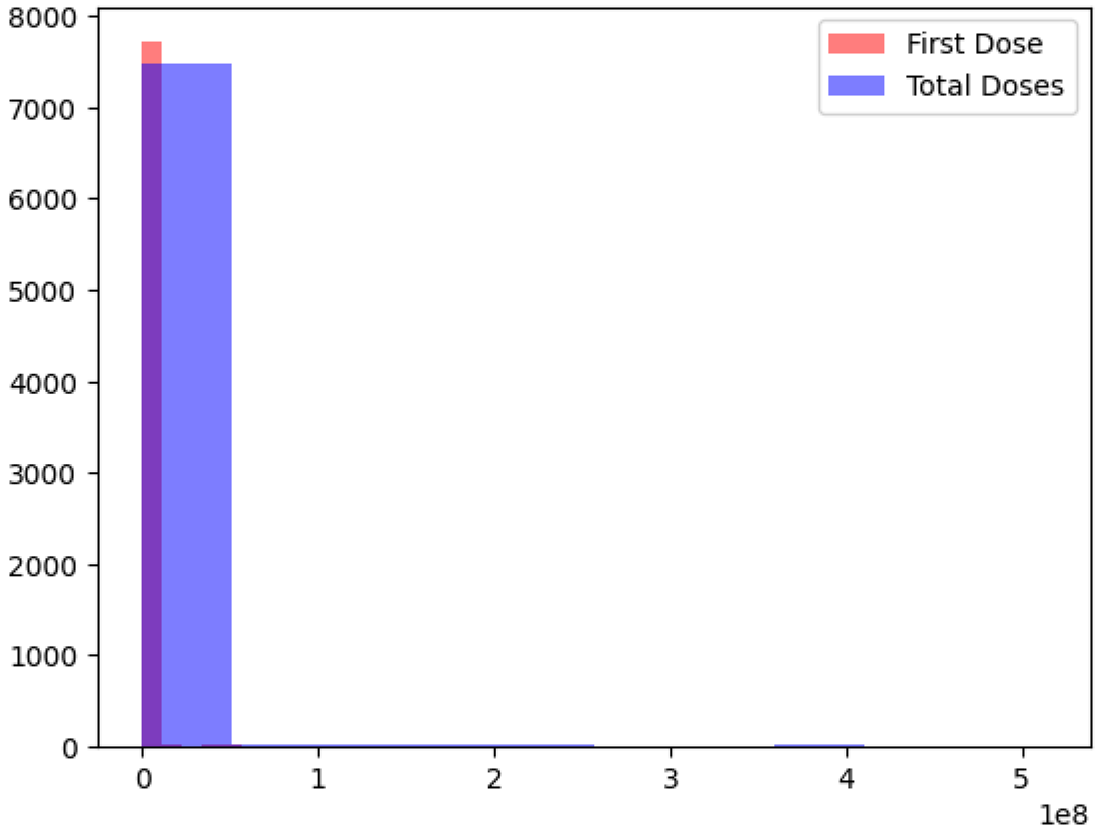


In [35]:

```
plt.hist(df['Second Dose Administered'],color='red',label='First Dose',alpha=0.5)
plt.hist(df['Total Doses Administered'],color='blue',label='Total Doses',alpha=0.5)
plt.legend()
```

Out[35]:

```
<matplotlib.legend.Legend at 0x7fe130d17370>
```

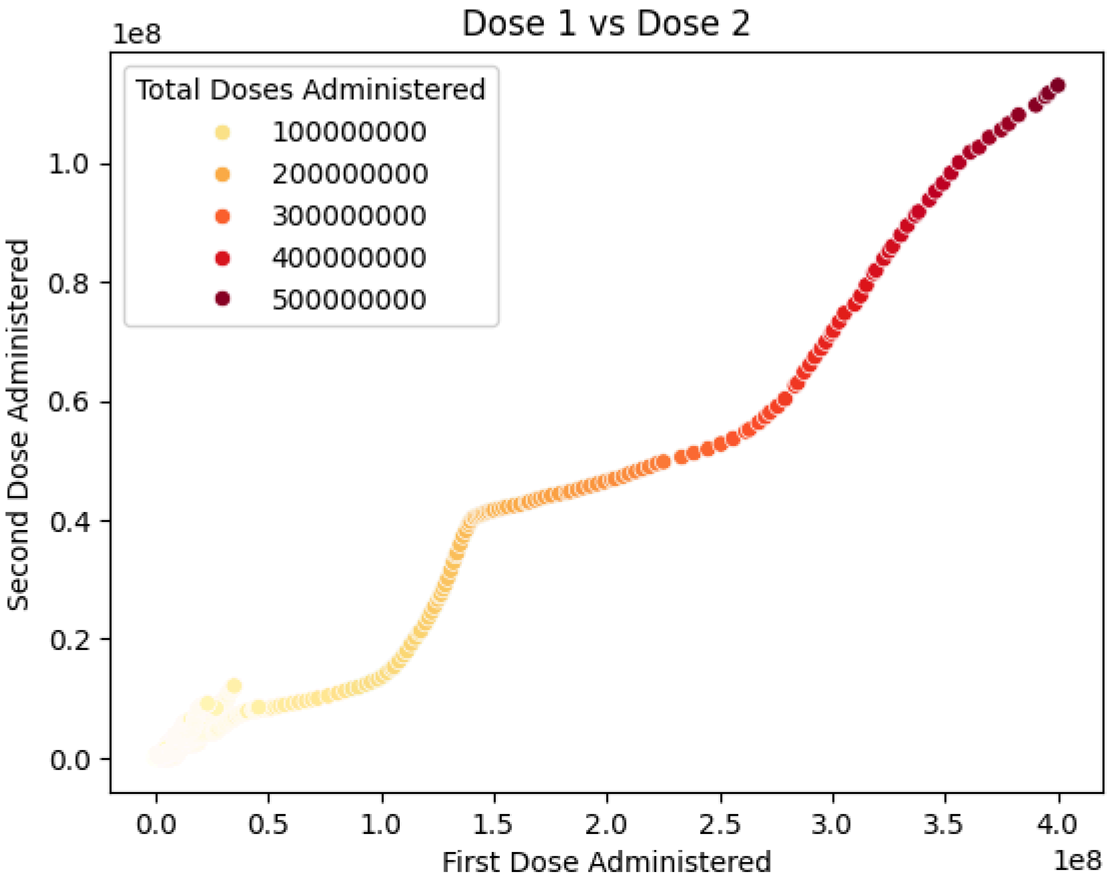


In [32]:

```
sns.scatterplot(x='First Dose Administered', y='Second Dose Administered', hue='Total
plt.title('Dose 1 vs Dose 2')
```

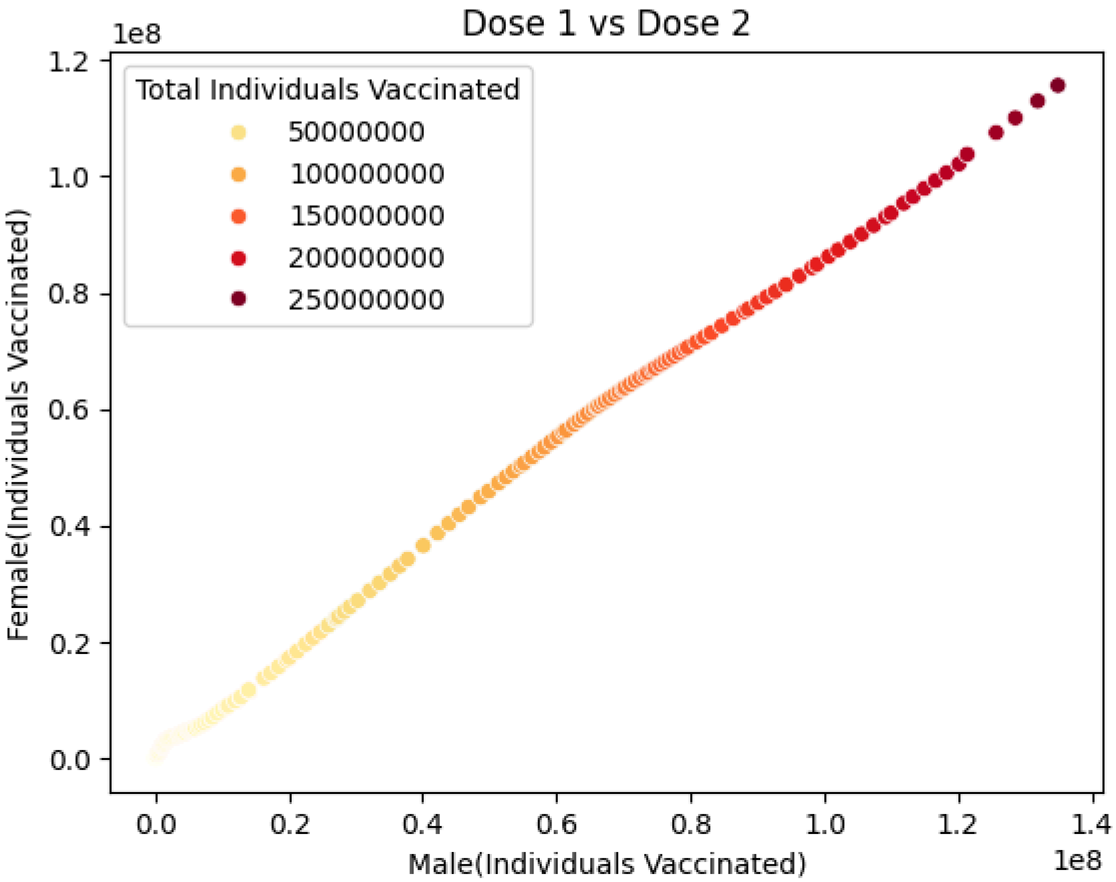
Out[32]:

```
Text(0.5, 1.0, 'Dose 1 vs Dose 2')
```



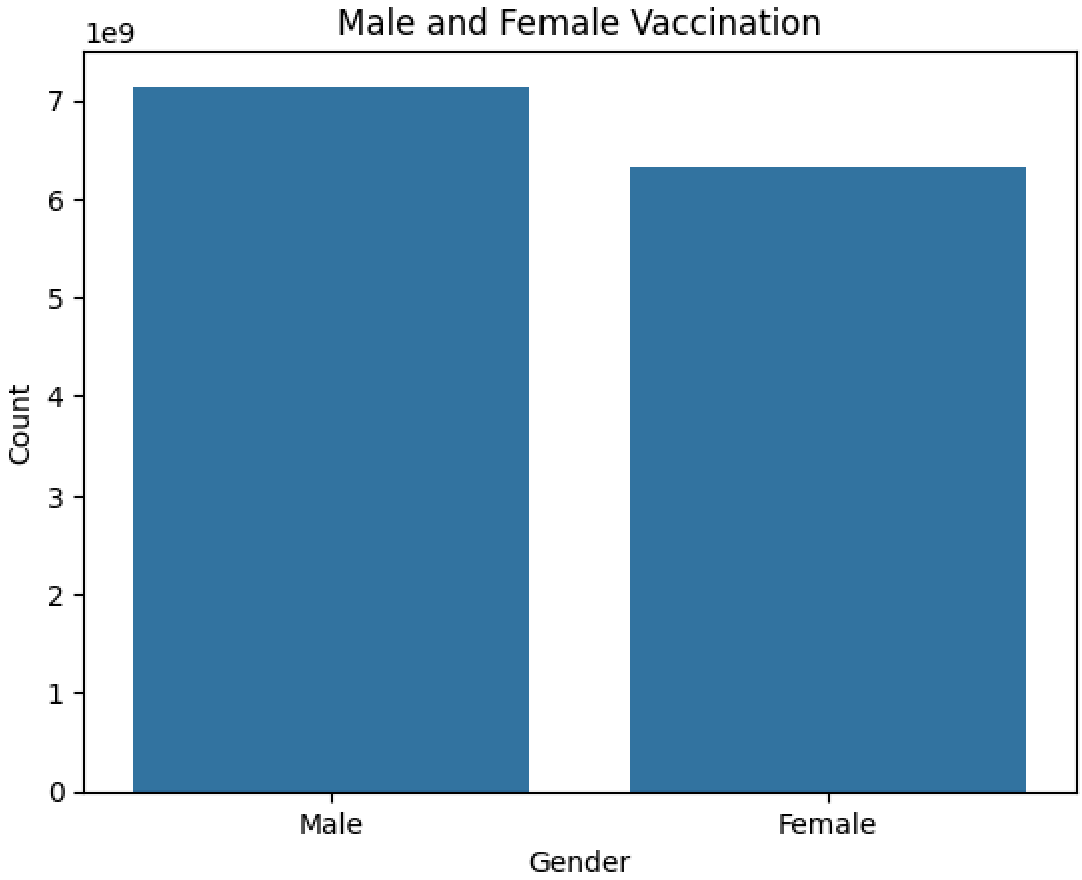
```
In [52]: sns.scatterplot(x='Male(Individuals Vaccinated)', y='Female(Individuals Vaccinated)',
         plt.title('Dose 1 vs Dose 2')
```

```
Out[52]: Text(0.5, 1.0, 'Dose 1 vs Dose 2')
```



In [49]:

```
# Calculate the counts for males and females and Create a count plot
male_count = df["Male(Individuals Vaccinated)"].sum()
female_count = df["Female(Individuals Vaccinated)"].sum()

data = {
    "Gender": ["Male", "Female"],
    "Count": [male_count, female_count]
}

sns.barplot(x="Gender", y="Count", data=data)
plt.title("Male and Female Vaccination")
plt.xlabel("Gender")
plt.ylabel("Count")
plt.show()
```

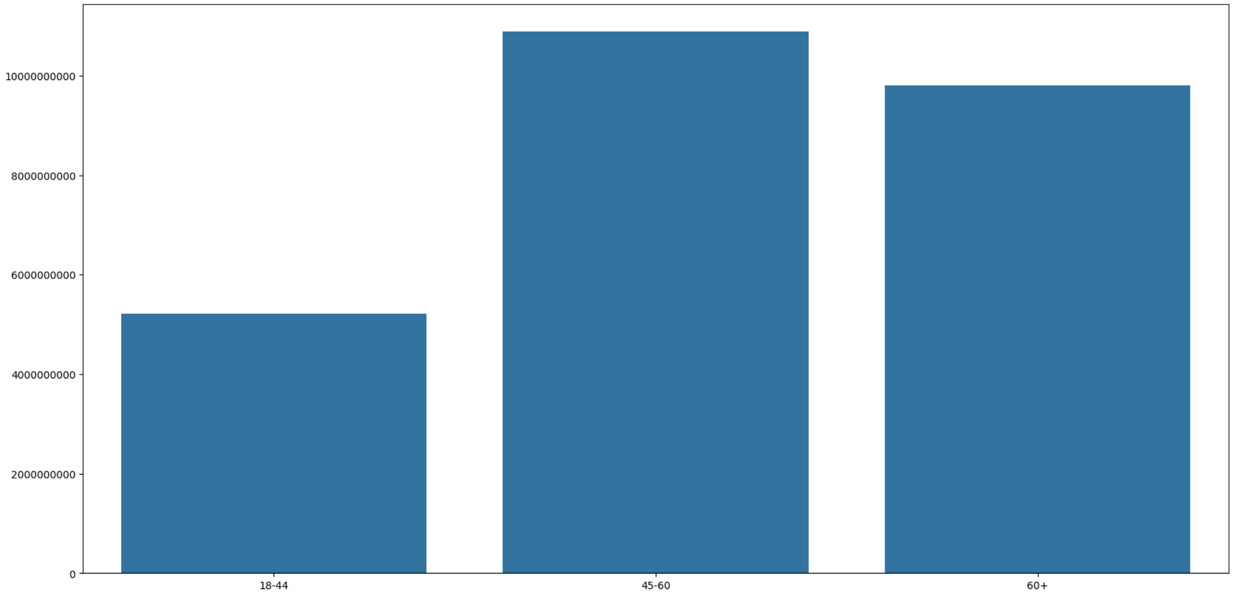


In [61]:

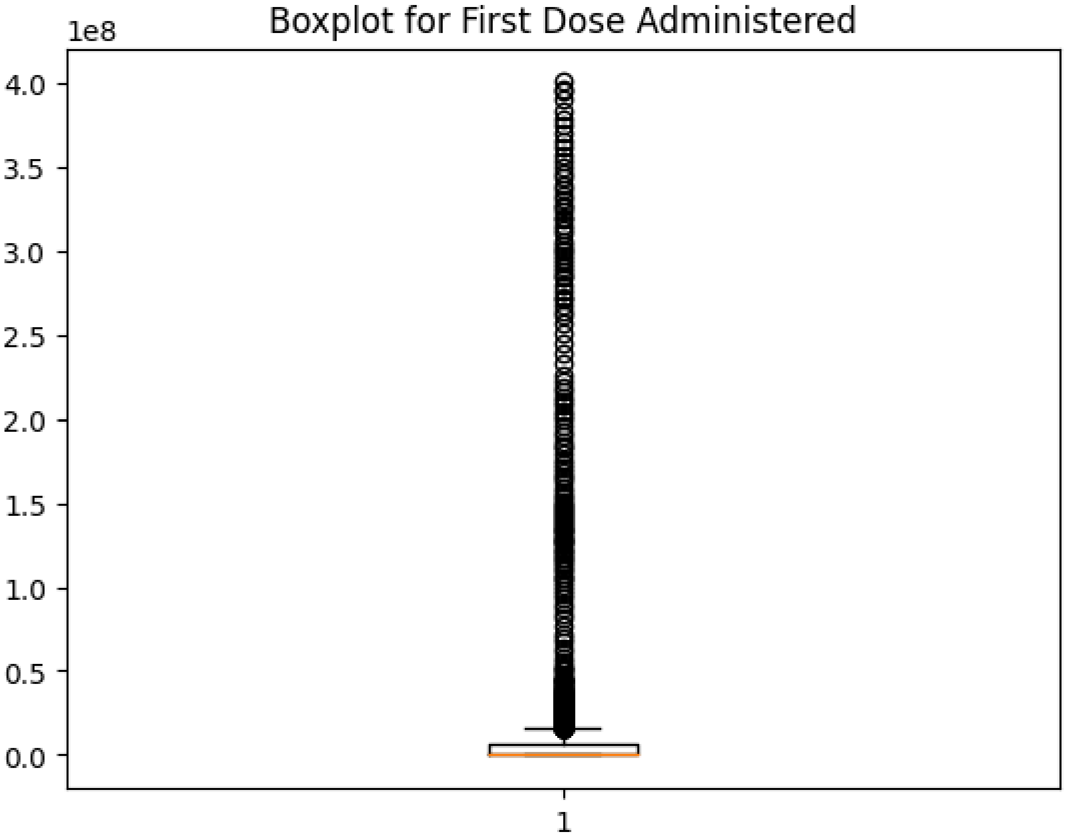
```
fig = plt.figure(figsize=(20,10))

a = df['18-44 Years(Individuals Vaccinated)'].sum()
b = df['45-60 Years(Individuals Vaccinated)'].sum()
c = df['60+ Years(Individuals Vaccinated)'].sum()


print('Total Individuals Vaccinated (18-44) =', a)
print('Total Individuals Vaccinated (45-60) =', b)
print('Total Individuals Vaccinated (60+) =', c)

barplot = sns.barplot(x=['18-44','45-60','60+'],y=[a, b, c])
barplot.set_yticklabels(labels=(barplot.get_yticks()*1).astype(int))

plt.show()
```

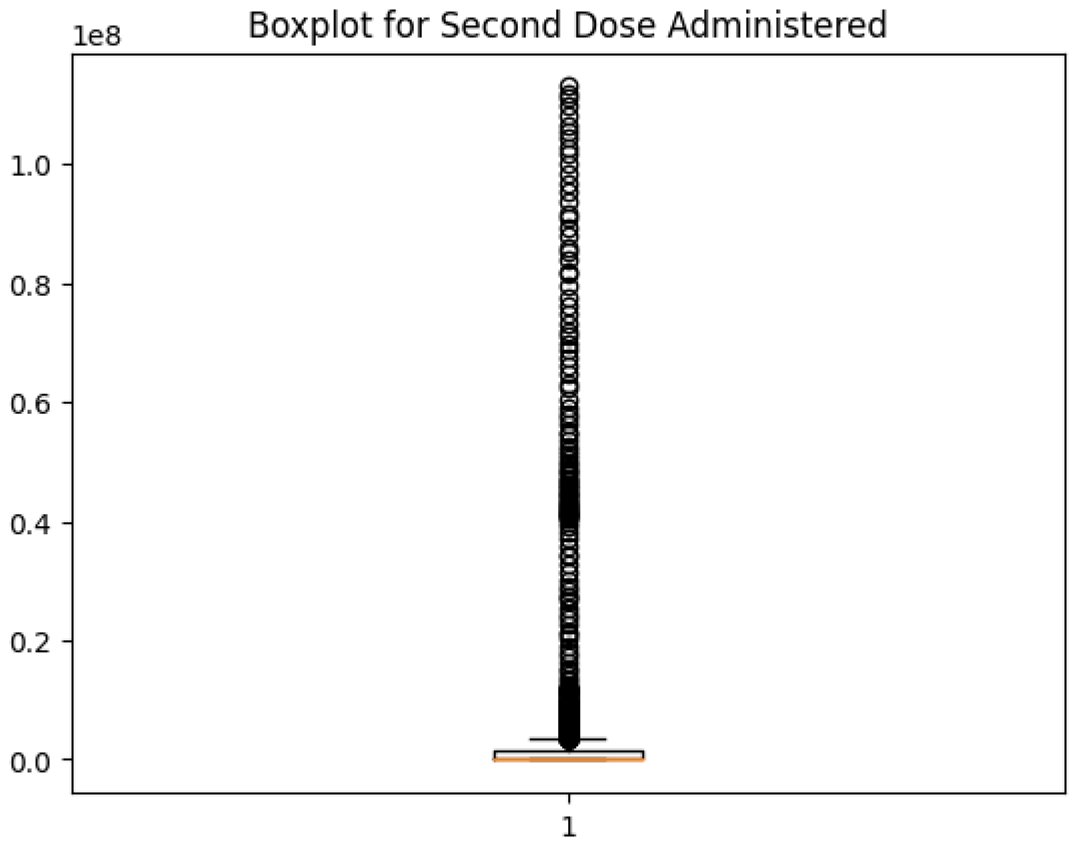
```
Total Individuals Vaccinated (18-44) = 5210874302.0
Total Individuals Vaccinated (45-60) = 10890266225.0
Total Individuals Vaccinated (60+) = 9810876107.0
```

```
<ipython-input-61-659cd06289c9>:13: UserWarning: FixedFormatter should only be used t
ogether with FixedLocator
  barplot.set_yticklabels(labels=(barplot.get_yticks()*1).astype(int))
```



In [70]:
```
columns_to_plot = df[["First Dose Administered", "Second Dose Administered"]]

for column in columns_to_plot.columns:
    plt.boxplot(columns_to_plot[column].dropna())  # Dropping NaN values for each colu
    plt.title(f'Boxplot for {column}')
    plt.show()
```




Boxplot for Second Dose Administered

In [71]:
```
!jupyter nbconvert --to html ('/content/drive/MyDrive/Colab Notebooks/Mini Project.ipy
```

```
/bin/bash: -c: line 1: syntax error near unexpected token `('
/bin/bash: -c: line 1: `jupyter nbconvert --to html ('/content/drive/MyDrive/Colab No
tebooks/Mini Project.ipynb')'
```