In [1]:

```
import pandas as pd
import numpy as np
import matplotlib
from pandas_profiling import ProfileReport
import matplotlib.pyplot as plt
import seaborn as sns
from warnings import filterwarnings
%matplotlib inline
from mpl_toolkits.mplot3d import Axes3D
```

In [2]:

```
df = pd.read_csv("covid_vaccine_statewise.csv")
df
```

Out[2]:

	Updated On	State	Total Individuals Vaccinated	Total Sessions Conducted	Total Sites	First Dose Administered	Second Dose Administered	Male(Individuals Vaccinated)	Female(In Va
0	16/01/2021	India	48276.0	3455.0	2957.0	48276.0	0.0	23757.0	
1	17/01/2021	India	58604.0	8532.0	4954.0	58604.0	0.0	27348.0	
2	18/01/2021	India	99449.0	13611.0	6583.0	99449.0	0.0	41361.0	
3	19/01/2021	India	195525.0	17855.0	7951.0	195525.0	0.0	81901.0	
4	20/01/2021	India	251280.0	25472.0	10504.0	251280.0	0.0	98111.0	
...	...	...	...	...	...	...	...	...	
4584	15/05/2021	West Bengal	8955081.0	265500.0	1535.0	8955081.0	3717988.0	4820673.0	4
4585	16/05/2021	West Bengal	8958736.0	72268.0	365.0	8958736.0	3719913.0	4823003.0	4
4586	17/05/2021	West Bengal	9001376.0	379622.0	1730.0	9001376.0	3739934.0	4851954.0	4
4587	18/05/2021	West Bengal	9046876.0	295766.0	1313.0	9046876.0	3758465.0	4882518.0	4
4588	19/05/2021	West Bengal	NaN	NaN	NaN	NaN	NaN	NaN	

4589 rows × 18 columns

In [3]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4589 entries, 0 to 4588
Data columns (total 18 columns):
 #   Column                             Non-Null Count  Dtype
---  ------                             --------------  -----
 0   Updated On                         4589 non-null   object
 1   State                              4589 non-null   object
 2   Total Individuals Vaccinated       4551 non-null   float64
 3   Total Sessions Conducted           4551 non-null   float64
 4   Total Sites                        4551 non-null   float64
 5   First Dose Administered            4551 non-null   float64
 6   Second Dose Administered           4551 non-null   float64
 7   Male(Individuals Vaccinated)       4551 non-null   float64
 8   Female(Individuals Vaccinated)     4551 non-null   float64
```

```
 9   Transgender(Individuals Vaccinated)  4551 non-null   float64
 10  Total Covaxin Administered           4551 non-null   float64
 11  Total CoviShield Administered        4551 non-null   float64
 12  AEFI                                 2368 non-null   float64
 13  18-30 years (Age)                    2368 non-null   float64
 14  30-45 years (Age)                    2368 non-null   float64
 15  45-60 years (Age)                    2368 non-null   float64
 16  60+ years (Age)                      2368 non-null   float64
 17  Total Doses Administered             4589 non-null   int64
dtypes: float64(15), int64(1), object(2)
memory usage: 645.5+ KB
```

In Data Analysis We will Analyze To Find out the below stuff

1. Missing Values

2. All The Numerical Variables

3. Distribution of the Numerical Variables

4. Categorical Variables

5. Outliers

6. Relationship between independent and dependent feature

1. Misiing Values

In [4]:
```
features_with_na = [features for features in df.columns if df[features].isnull().sum() > 1] #list

for feature in features_with_na:
    print(feature, np.round(df[feature].isnull().mean() * 100, 4),  ' % missing values')
```

```
Total Individuals Vaccinated 0.8281  % missing values
Total Sessions Conducted 0.8281  % missing values
Total Sites  0.8281  % missing values
First Dose Administered 0.8281  % missing values
Second Dose Administered 0.8281  % missing values
Male(Individuals Vaccinated) 0.8281  % missing values
Female(Individuals Vaccinated) 0.8281  % missing values
Transgender(Individuals Vaccinated) 0.8281  % missing values
Total Covaxin Administered 0.8281  % missing values
Total CoviShield Administered 0.8281  % missing values
AEFI 48.3983  % missing values
18-30 years (Age) 48.3983  % missing values
30-45 years (Age) 48.3983  % missing values
45-60 years (Age) 48.3983  % missing values
60+ years (Age) 48.3983  % missing values
```

In [5]:
```
df.shape
```

Out[5]: (4589, 18)

In [6]:
```
data = df.copy()
data.head(10)
```

Out[6]:

	Updated On	State	Total Individuals Vaccinated	Total Sessions Conducted	Total Sites	First Dose Administered	Second Dose Administered	Male(Individuals Vaccinated)	Female(Individ Vaccinat
0	16/01/2021	India	48276.0	3455.0	2957.0	48276.0	0.0	23757.0	245
1	17/01/2021	India	58604.0	8532.0	4954.0	58604.0	0.0	27348.0	312
2	18/01/2021	India	99449.0	13611.0	6583.0	99449.0	0.0	41361.0	580
3	19/01/2021	India	195525.0	17855.0	7951.0	195525.0	0.0	81901.0	1136
4	20/01/2021	India	251280.0	25472.0	10504.0	251280.0	0.0	98111.0	1531
5	21/01/2021	India	365965.0	32226.0	12600.0	365965.0	0.0	132784.0	2331
6	22/01/2021	India	549381.0	36988.0	14115.0	549381.0	0.0	193899.0	3554
7	23/01/2021	India	759008.0	43076.0	15605.0	759008.0	0.0	267856.0	4910
8	24/01/2021	India	835058.0	49851.0	18111.0	835058.0	0.0	296283.0	5386
9	25/01/2021	India	1277104.0	55151.0	19682.0	1277104.0	0.0	444137.0	8327

In [7]:
```
data.head(10)
```

Out[7]:

	Updated On	State	Total Individuals Vaccinated	Total Sessions Conducted	Total Sites	First Dose Administered	Second Dose Administered	Male(Individuals Vaccinated)	Female(Individ Vaccinat
0	16/01/2021	India	48276.0	3455.0	2957.0	48276.0	0.0	23757.0	245
1	17/01/2021	India	58604.0	8532.0	4954.0	58604.0	0.0	27348.0	312
2	18/01/2021	India	99449.0	13611.0	6583.0	99449.0	0.0	41361.0	580
3	19/01/2021	India	195525.0	17855.0	7951.0	195525.0	0.0	81901.0	1136
4	20/01/2021	India	251280.0	25472.0	10504.0	251280.0	0.0	98111.0	1531
5	21/01/2021	India	365965.0	32226.0	12600.0	365965.0	0.0	132784.0	2331
6	22/01/2021	India	549381.0	36988.0	14115.0	549381.0	0.0	193899.0	3554
7	23/01/2021	India	759008.0	43076.0	15605.0	759008.0	0.0	267856.0	4910
8	24/01/2021	India	835058.0	49851.0	18111.0	835058.0	0.0	296283.0	5386
9	25/01/2021	India	1277104.0	55151.0	19682.0	1277104.0	0.0	444137.0	8327

In [8]:
```
data.fillna(0) # We won't be deleteing any data, replacing all NA values with 0
```

Out[8]:

	Updated On	State	Total Individuals Vaccinated	Total Sessions Conducted	Total Sites	First Dose Administered	Second Dose Administered	Male(Individuals Vaccinated)	Female(In Va
0	16/01/2021	India	48276.0	3455.0	2957.0	48276.0	0.0	23757.0	
1	17/01/2021	India	58604.0	8532.0	4954.0	58604.0	0.0	27348.0	
2	18/01/2021	India	99449.0	13611.0	6583.0	99449.0	0.0	41361.0	

	Updated On	State	Total Individuals Vaccinated	Total Sessions Conducted	Total Sites	First Dose Administered	Second Dose Administered	Male(Individuals Vaccinated)	Female(In Va
3	19/01/2021	India	195525.0	17855.0	7951.0	195525.0	0.0	81901.0	
4	20/01/2021	India	251280.0	25472.0	10504.0	251280.0	0.0	98111.0	
...	...	...	...	...	...	...	...	...	
4584	15/05/2021	West Bengal	8955081.0	265500.0	1535.0	8955081.0	3717988.0	4820673.0	4
4585	16/05/2021	West Bengal	8958736.0	72268.0	365.0	8958736.0	3719913.0	4823003.0	4
4586	17/05/2021	West Bengal	9001376.0	379622.0	1730.0	9001376.0	3739934.0	4851954.0	4
4587	18/05/2021	West Bengal	9046876.0	295766.0	1313.0	9046876.0	3758465.0	4882518.0	4
4588	19/05/2021	West Bengal	0.0	0.0	0.0	0.0	0.0	0.0	

4589 rows × 18 columns

Converting Date object to Datetime

In [9]:
```
data['Updated On'] =  pd.to_datetime(data['Updated On'], format="%d/%m/%Y")
data['Updated On'] = data['Updated On'].dt.strftime('%m/%y')
data["Updated On"]
```

Out[9]:
```
0       01/21
1       01/21
2       01/21
3       01/21
4       01/21
        ...
4584    05/21
4585    05/21
4586    05/21
4587    05/21
4588    05/21
Name: Updated On, Length: 4589, dtype: object
```

2. All The Numerical Variables

In [10]:
```
data.shape
```

Out[10]:
```
(4589, 18)
```

As India's Overall Data is alos included, it is cretaing high skewness in data, so we need to remove it.

In [11]:
```
data.drop(data.loc[data['State']== "India"].index, inplace=True)
```

In [12]:
```
data.head(10)
```

Out[12]:

	Updated On	State	Total Individuals Vaccinated	Total Sessions Conducted	Total Sites	First Dose Administered	Second Dose Administered	Male(Individuals Vaccinated)	Female(Indivi Vaccin
124	01/21	Andaman and Nicobar Islands	23.0	2.0	2.0	23.0	0.0	12.0	
125	01/21	Andaman and Nicobar Islands	23.0	2.0	2.0	23.0	0.0	12.0	
126	01/21	Andaman and Nicobar Islands	42.0	9.0	2.0	42.0	0.0	29.0	
127	01/21	Andaman and Nicobar Islands	89.0	12.0	2.0	89.0	0.0	53.0	
128	01/21	Andaman and Nicobar Islands	124.0	16.0	3.0	124.0	0.0	67.0	
129	01/21	Andaman and Nicobar Islands	239.0	22.0	6.0	239.0	0.0	110.0	
130	01/21	Andaman and Nicobar Islands	552.0	29.0	6.0	552.0	0.0	231.0	
131	01/21	Andaman and Nicobar Islands	920.0	32.0	9.0	920.0	0.0	342.0	
132	01/21	Andaman and Nicobar Islands	966.0	38.0	9.0	966.0	0.0	357.0	
133	01/21	Andaman and Nicobar Islands	1519.0	43.0	10.0	1519.0	0.0	447.0	1

In [13]:

```
numerical_features = [feature for feature in data.columns if data[feature].dtypes != 'O'] # list c

print('Number of numerical variables: ', len(numerical_features))

# visualise the numerical variables
data[numerical_features].head()
```

```
Number of numerical variables:  16
```

Out[13]:

	Total Individuals Vaccinated	Total Sessions Conducted	Total Sites	First Dose Administered	Second Dose Administered	Male(Individuals Vaccinated)	Female(Individuals Vaccinated)	Transgender(In Va
124	23.0	2.0	2.0	23.0	0.0	12.0	11.0	
125	23.0	2.0	2.0	23.0	0.0	12.0	11.0	
126	42.0	9.0	2.0	42.0	0.0	29.0	13.0	
127	89.0	12.0	2.0	89.0	0.0	53.0	36.0	
128	124.0	16.0	3.0	124.0	0.0	67.0	57.0	

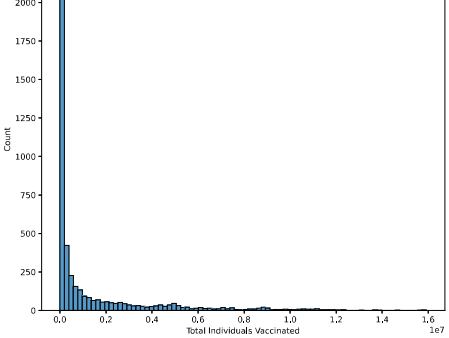
In [14]:
```
data[numerical_features].fillna(0)
```

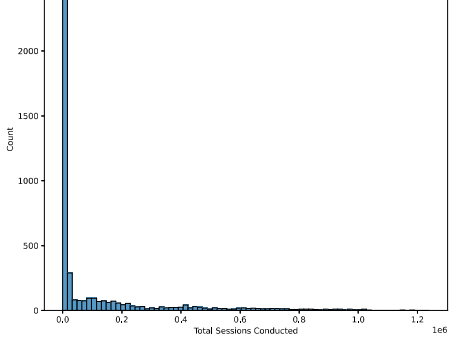
Out[14]:

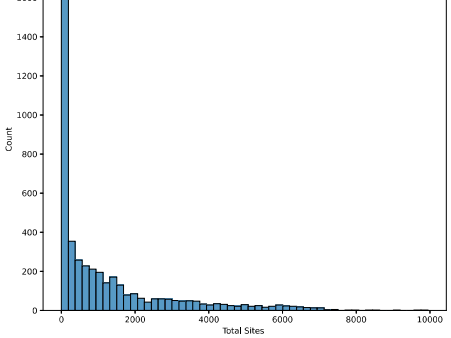
	Total Individuals Vaccinated	Total Sessions Conducted	Total Sites	First Dose Administered	Second Dose Administered	Male(Individuals Vaccinated)	Female(Individuals Vaccinated)	Transgender
124	23.0	2.0	2.0	23.0	0.0	12.0	11.0	
125	23.0	2.0	2.0	23.0	0.0	12.0	11.0	
126	42.0	9.0	2.0	42.0	0.0	29.0	13.0	
127	89.0	12.0	2.0	89.0	0.0	53.0	36.0	
128	124.0	16.0	3.0	124.0	0.0	67.0	57.0	
...	...	...	...	...	...	...	...	
4584	8955081.0	265500.0	1535.0	8955081.0	3717988.0	4820673.0	4133379.0	
4585	8958736.0	72268.0	365.0	8958736.0	3719913.0	4823003.0	4134704.0	
4586	9001376.0	379622.0	1730.0	9001376.0	3739934.0	4851954.0	4148384.0	
4587	9046876.0	295766.0	1313.0	9046876.0	3758465.0	4882518.0	4163312.0	
4588	0.0	0.0	0.0	0.0	0.0	0.0	0.0	

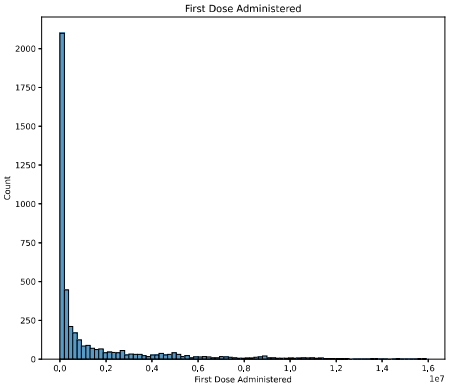
4465 rows × 16 columns

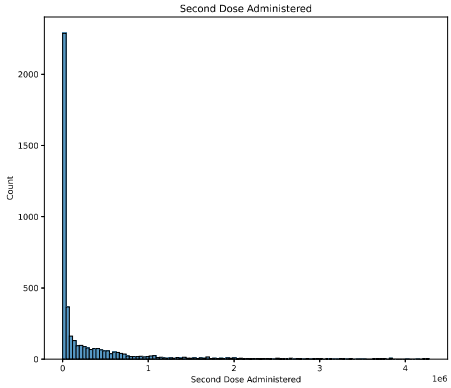
In [15]:
```
a = 9  # number of rows
b = 3  # number of columns
c = 1  # initialize plot counter

fig = plt.figure(figsize=(30,80))

for i in numerical_features:
    plt.subplot(a, b, c)
    plt.title('{}'.format(i))

    sns.histplot(data= data, x= i)

    c = c + 1

plt.show()
```

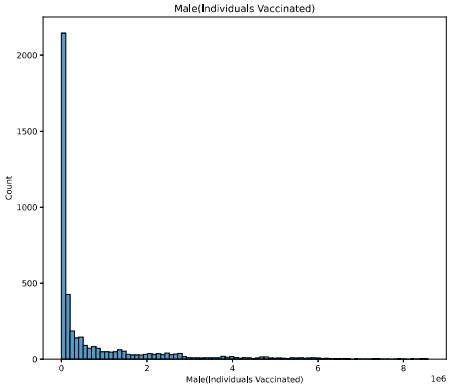

Count
Total Individuals Vaccinated
1e7

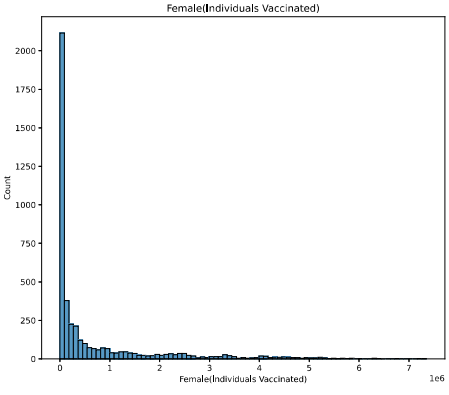

Count
Total Sessions Conducted
1e6

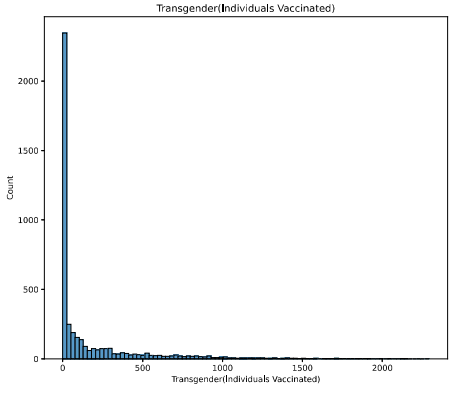

Count
Total Sites

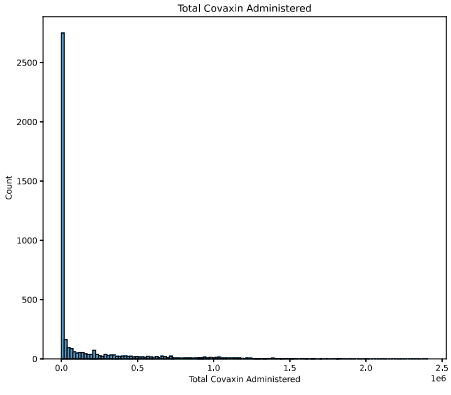

First Dose Administered
Count
First Dose Administered
1e7

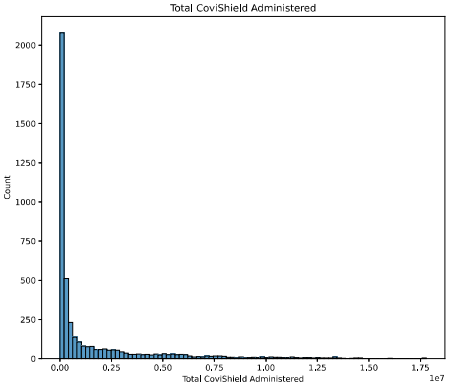

Second Dose Administered
Count
Second Dose Administered
1e6

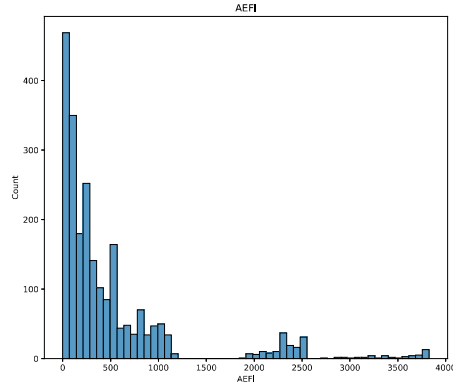

Male(Individuals Vaccinated)
Count
Male(Individuals Vaccinated)
1e6

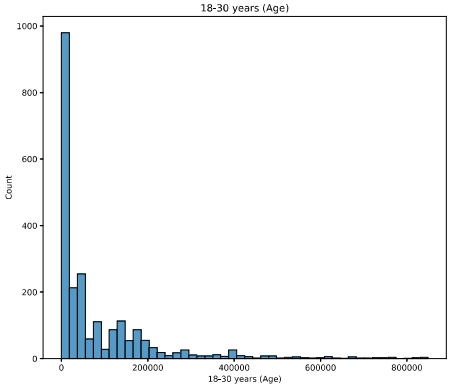

Female(Individuals Vaccinated)
Count
Female(Individuals Vaccinated)
1e6

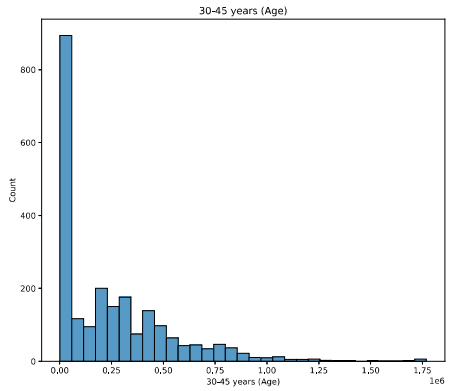

Transgender(Individuals Vaccinated)
Count
Transgender(Individuals Vaccinated)

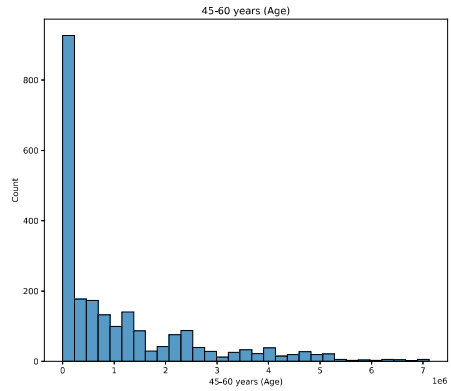

Total Covaxin Administered
Count
Total Covaxin Administered
1e6

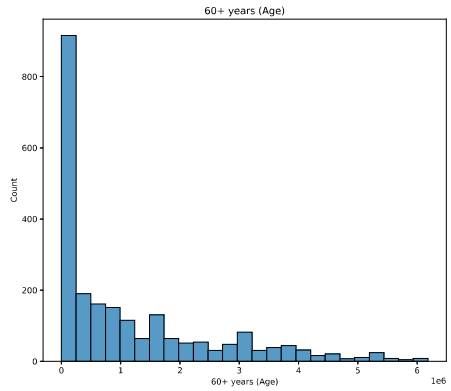

Total CoviShield Administered
Count
Total CoviShield Administered
1e7

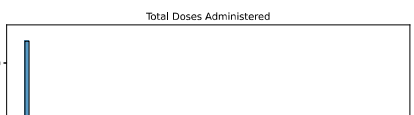

AEFI
Count
AEFI

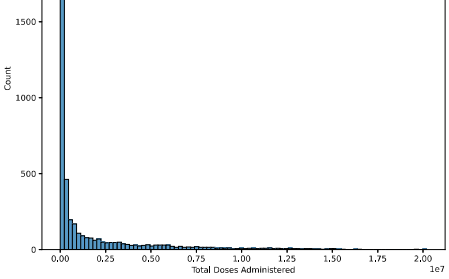

18-30 years (Age)
Count
18-30 years (Age)


30-45 years (Age)
Count
30-45 years (Age)
1e6


45-60 years (Age)
Count
45-60 years (Age)
1e6


60+ years (Age)
Count
60+ years (Age)
1e6


Total Doses Administered



let us check for decreate values

In [16]:
```
## Numerical variables are usually of 2 type
## 1. Continous variable and Discrete Variables

discrete_feature=[feature for feature in numerical_features if len(data[feature].unique())]
print("Discrete Variables Count: {}".format(len(discrete_feature)))
```

```
Discrete Variables Count: 16
```

0 discrete values

Continous Features

In [17]:
```
continuous_feature=[feature for feature in numerical_features if feature not in discrete_feature ]
print("Continuous feature Count {}".format(len(continuous_feature)))
```

```
Continuous feature Count 0
```

Plotting discrete features against tota individual vaccinated

In [18]:
```
discrete_feature
```

Out[18]:
```
['Total Individuals Vaccinated',
 'Total Sessions Conducted',
 'Total Sites ',
 'First Dose Administered',
 'Second Dose Administered',
 'Male(Individuals Vaccinated)',
 'Female(Individuals Vaccinated)',
 'Transgender(Individuals Vaccinated)',
 'Total Covaxin Administered',
 'Total CoviShield Administered',
 'AEFI',
 '18-30 years (Age)',
 '30-45 years (Age)',
 '45-60 years (Age)',
 '60+ years (Age)',
 'Total Doses Administered']
```

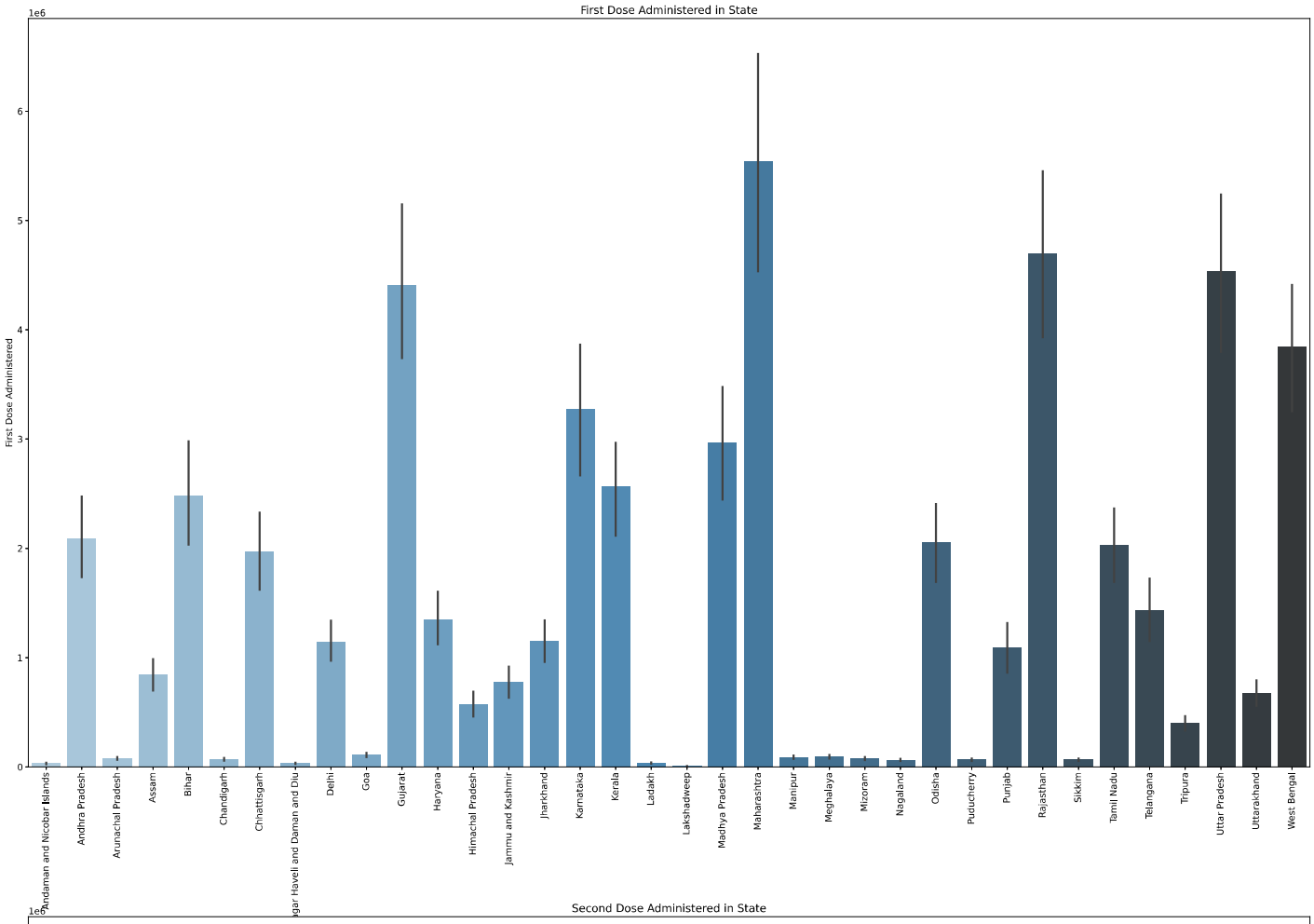
breaking down Discrete feature list to sub categories for easy of plotting

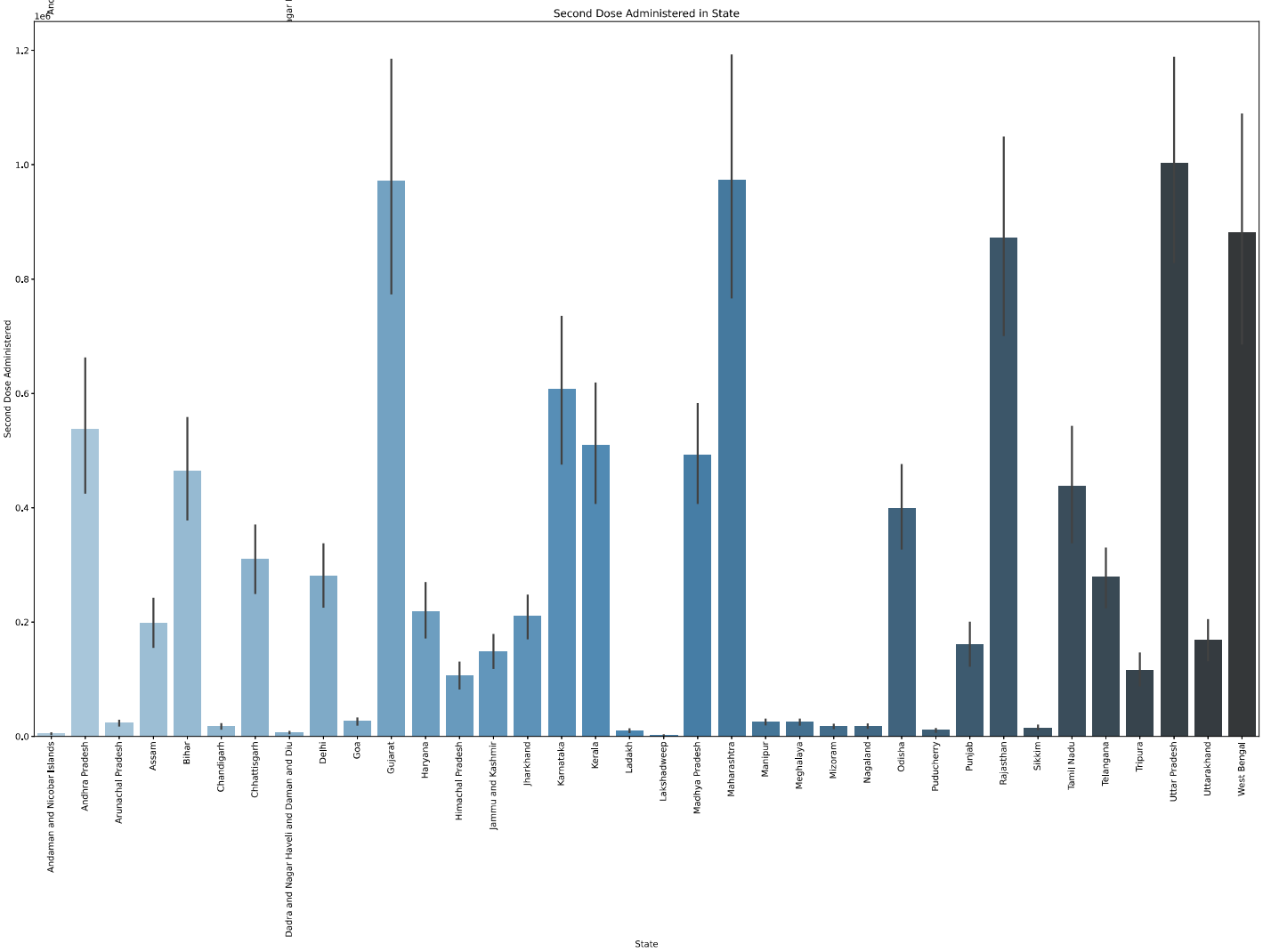
In [19]:
```
Covid_dose = ['First Dose Administered','Second Dose Administered']
Vaccinated = [ 'Male(Individuals Vaccinated)','Female(Individuals Vaccinated)','Transgender(Indivi
Vaccines_available = ['Total Covaxin Administered','Total CoviShield Administered']
Age = ['18-30 years (Age)','30-45 years (Age)','45-60 years (Age)','60+ years (Age)']
```

Bar plot to determine number of covid dose administered State Wise

In [20]:

```
a = 3  # number of rows
b = 1  # number of columns
c = 1  # initialize plot counter

fig = plt.figure(figsize=(30, 60))
font = {'size': 12}

# using rc function
plt.rc('font', **font)

for i in Covid_dose:
    plt.subplot(a, b, c)
    plt.title('{} in State'.format(i))
    sns.barplot(x= "State", y=  i, data= data, palette="Blues_d")
    plt.xticks(rotation = 90)
    c = c + 1

plt.show()
```

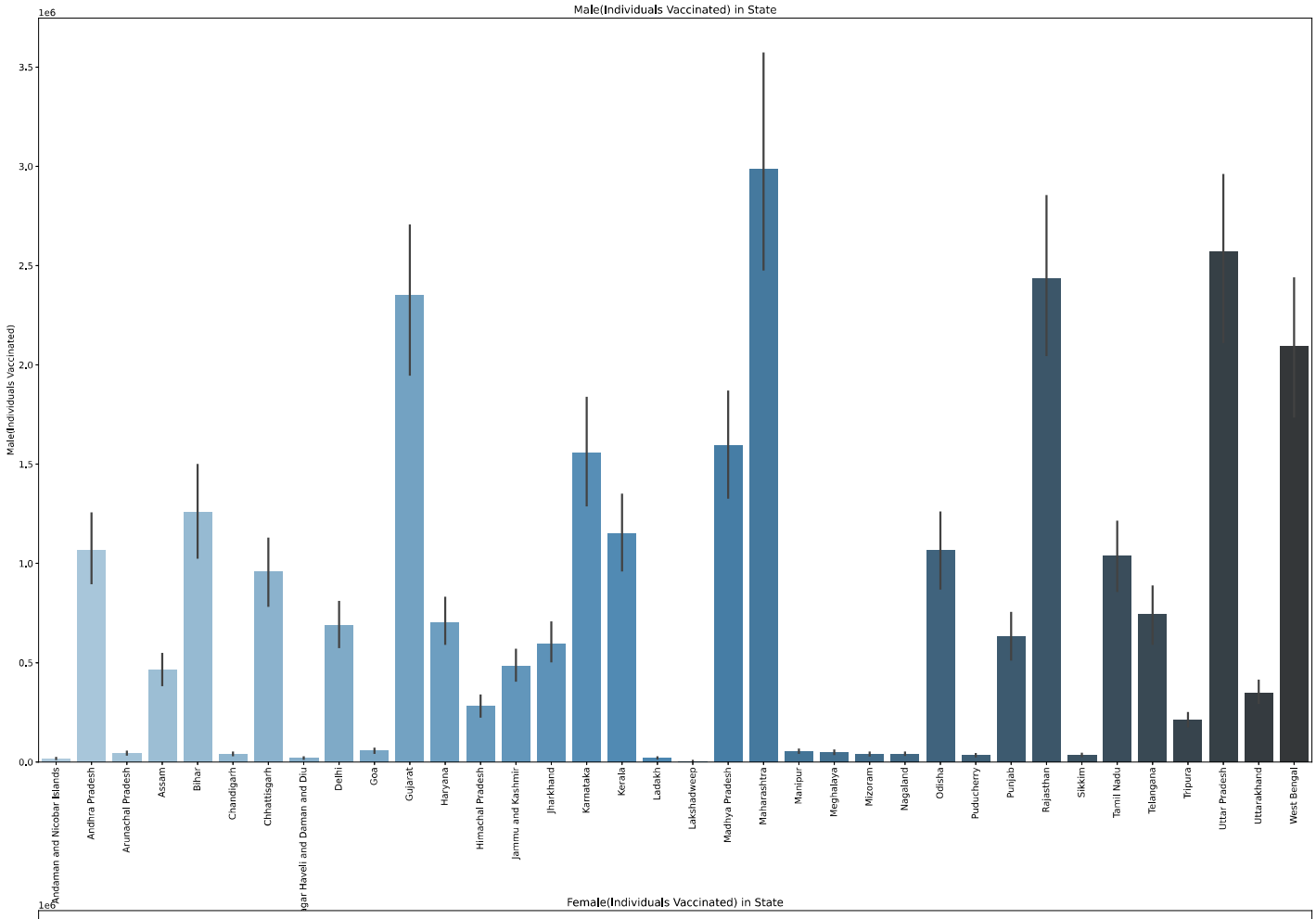

First Dose Administered in State
1e6
First Dose Administered
0
1
2
3
4
5
6
Andaman and Nicobar Islands
Andhra Pradesh
Arunachal Pradesh
Assam
Bihar
Chandigarh
Chhattisgarh
Dadra and Nagar Haveli and Daman and Diu
Delhi
Goa
Gujarat
Haryana
Himachal Pradesh
Jammu and Kashmir
Jharkhand
Karnataka
Kerala
Ladakh
Lakshadweep
Madhya Pradesh
Maharashtra
Manipur
Meghalaya
Mizoram
Nagaland
Odisha
Puducherry
Punjab
Rajasthan
Sikkim
Tamil Nadu
Telangana
Tripura
Uttar Pradesh
Uttarakhand
West Bengal

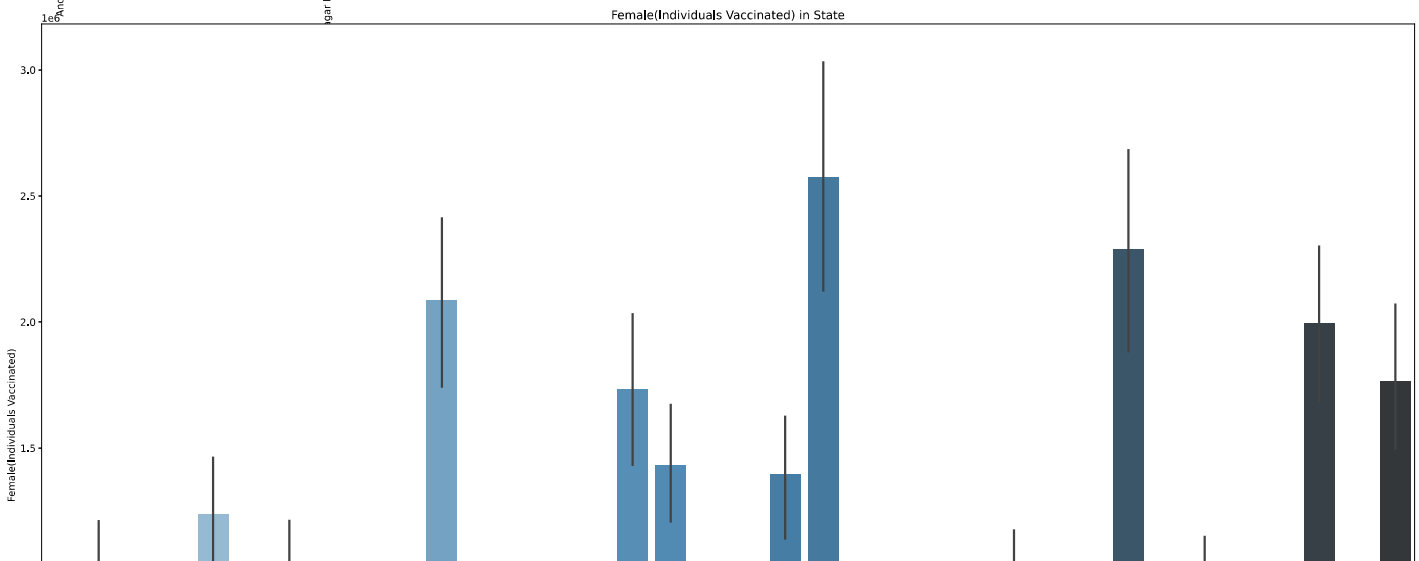

Second Dose Administered in State
1e6
Second Dose Administered
0.0
0.2
0.4
0.6
0.8
1.0
1.2
Andaman and Nicobar Islands
Andhra Pradesh
Arunachal Pradesh
Assam
Bihar
Chandigarh
Chhattisgarh
Dadra and Nagar Haveli and Daman and Diu
Delhi
Goa
Gujarat
Haryana
Himachal Pradesh
Jammu and Kashmir
Jharkhand
Karnataka
Kerala
Ladakh
Lakshadweep
Madhya Pradesh
Maharashtra
Manipur
Meghalaya
Mizoram
Nagaland
Odisha
Puducherry
Punjab
Rajasthan
Sikkim
Tamil Nadu
Telangana
Tripura
Uttar Pradesh
Uttarakhand
West Bengal
State

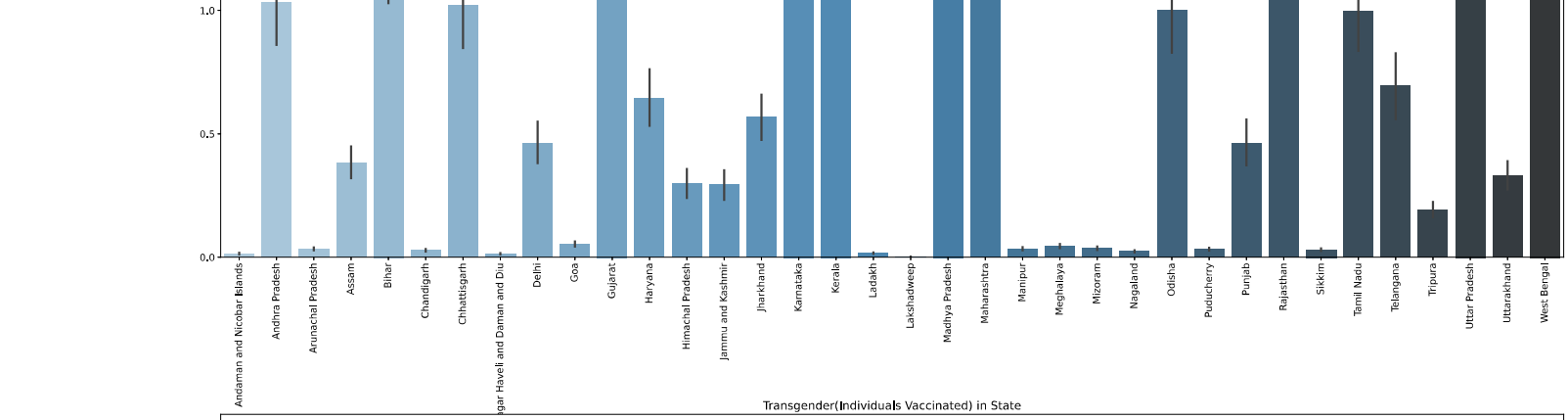
Bar plot to determine number of People Vaccinated State Wise

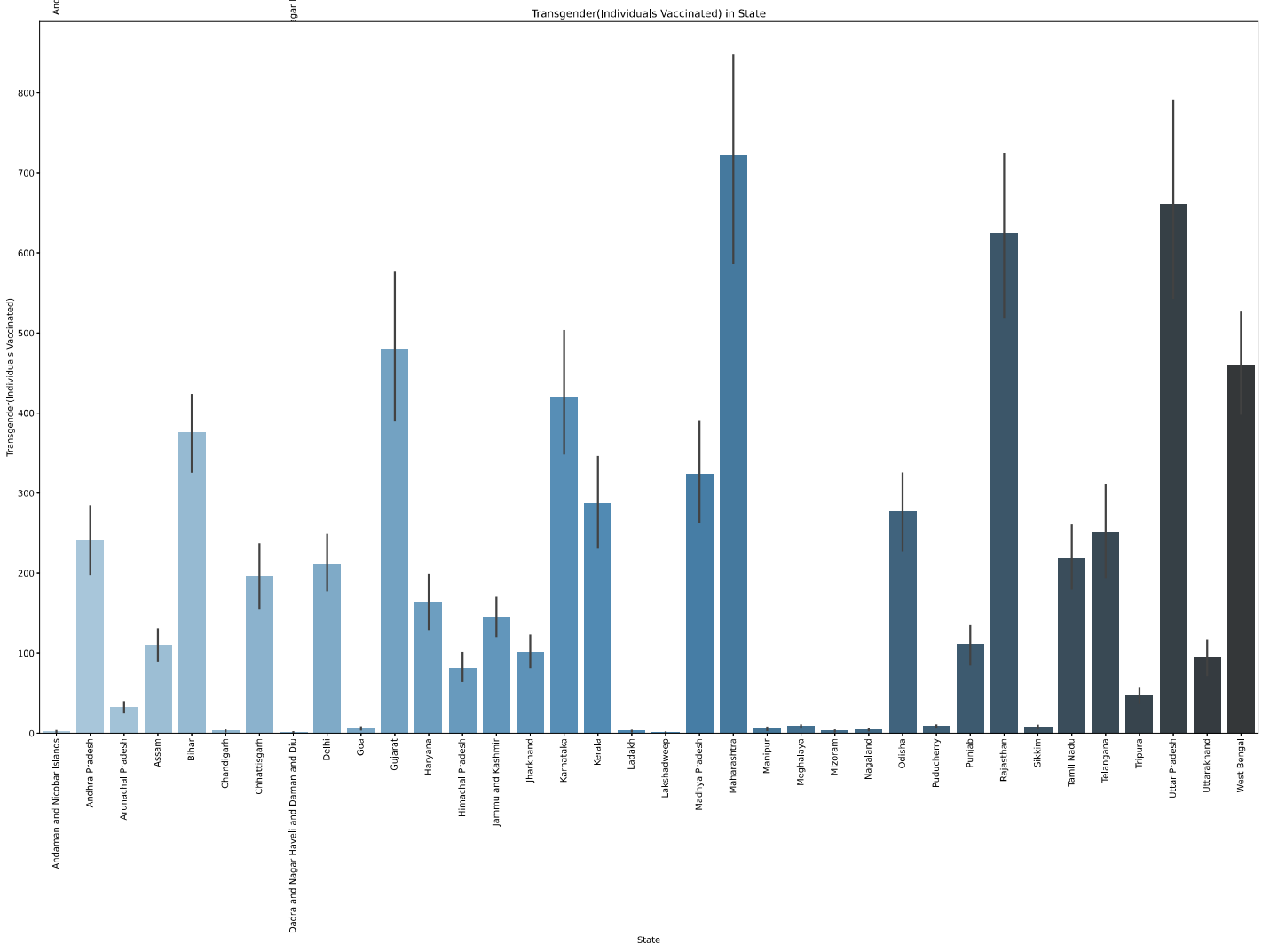
In [21]:

```
a = 3  # number of rows
b = 1  # number of columns
c = 1  # initialize plot counter

fig = plt.figure(figsize=(30, 60))

for i in Vaccinated:
    plt.subplot(a, b, c)
    plt.title('{} in State'.format(i))
    sns.barplot(x="State", y= i, data= data, palette="Blues_d")
    plt.xticks(rotation = 90)
    c = c + 1

plt.show()
```





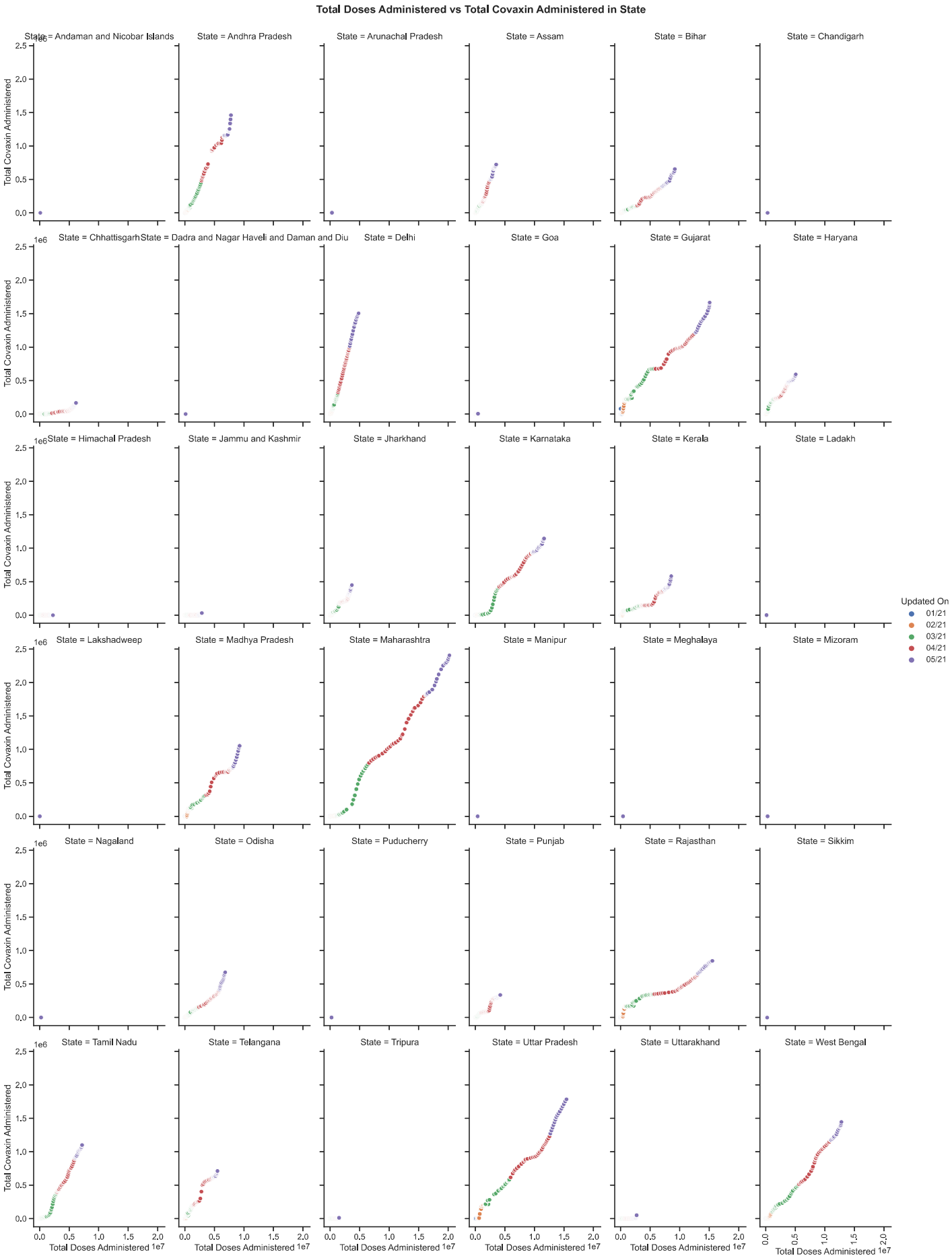


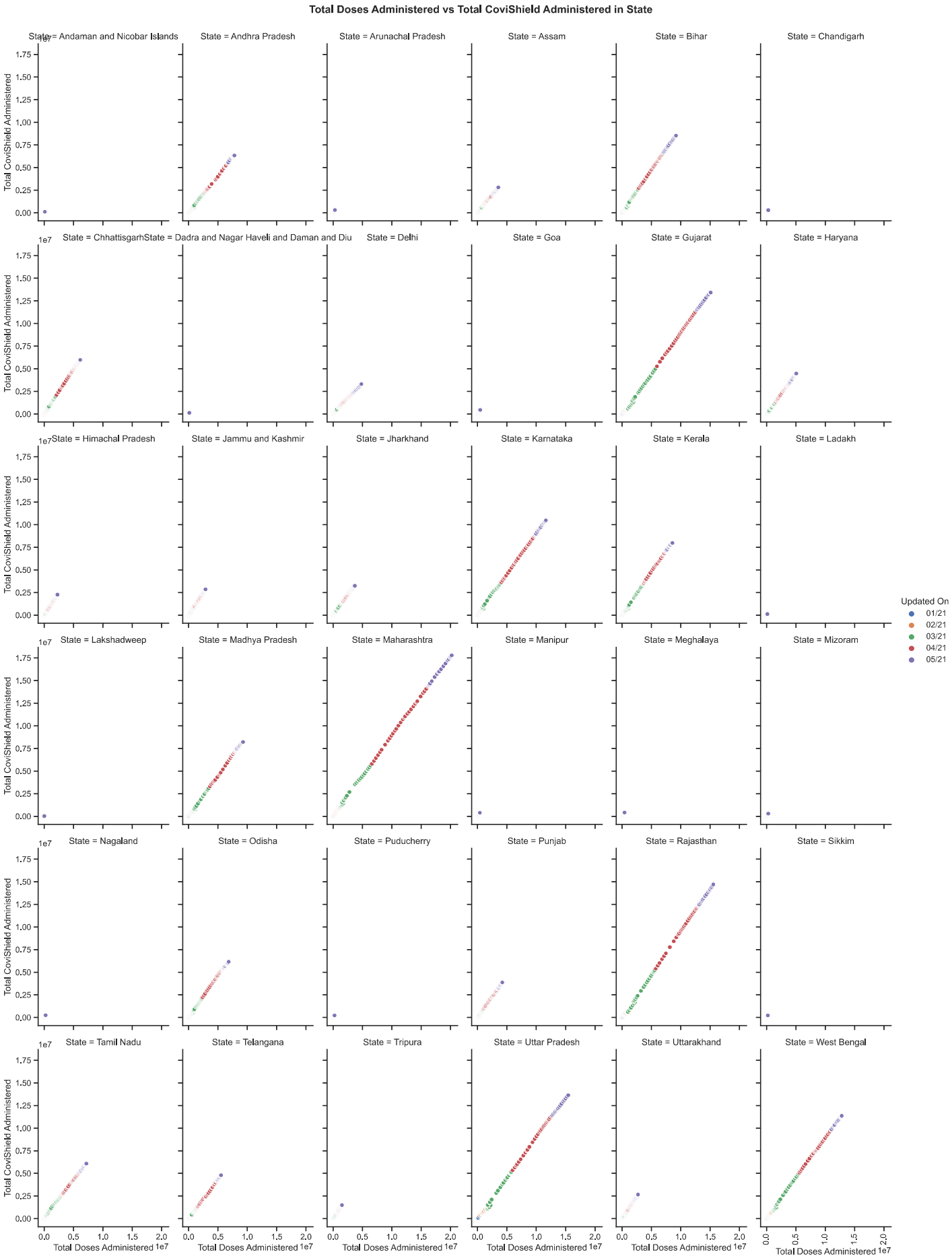


Scatterplot of individual states to Determine what Vaccines are given among total number of vaccines available.

In [22]:

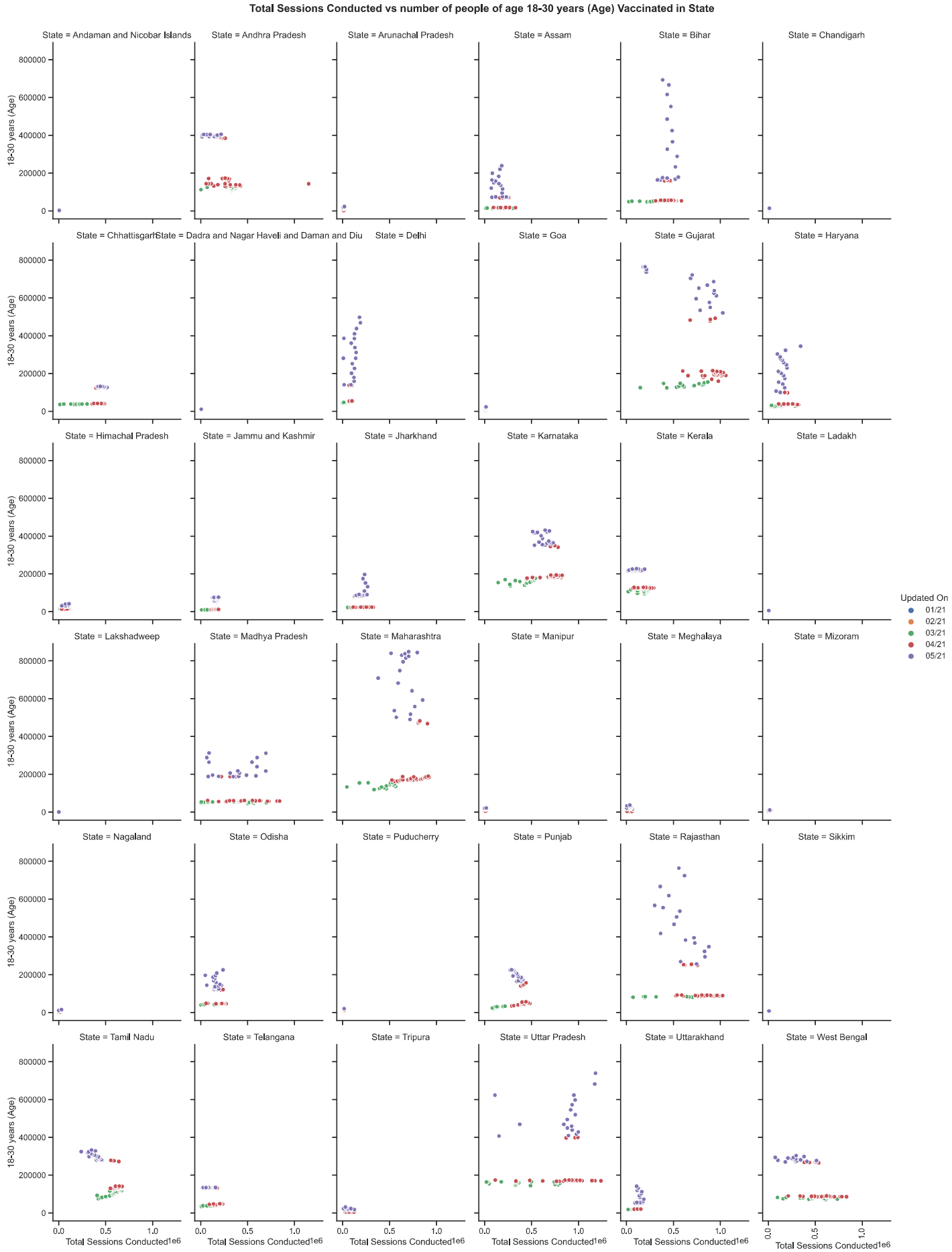
```
sns.set_theme(style="ticks")
for i in Vaccines_available:
    g = sns.relplot(data=data, x="Total Doses Administered", y=i, hue="Updated On",col= data.State
    g.fig.suptitle(f" Total Doses Administered vs {i} in State", fontweight ="bold")
    plt.subplots_adjust(top=.95)
    plt.xticks(rotation = 90)
plt.show()
```

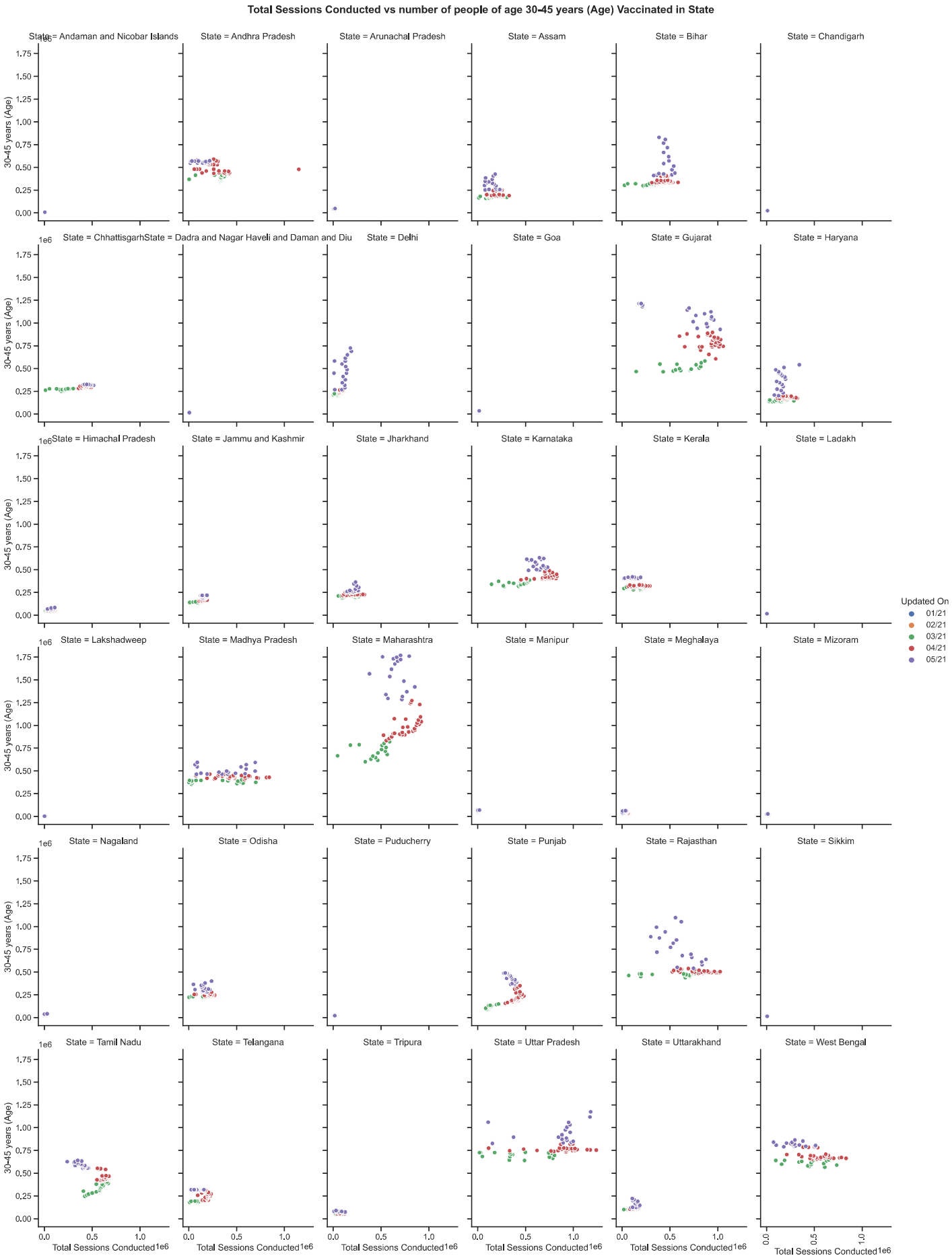

Total Doses Administered vs Total Covaxin Administered in State
State = Andaman and Nicobar Islands
State = Andhra Pradesh
State = Arunachal Pradesh
State = Assam
State = Bihar
State = Chandigarh
State = Chhattisgarh
State = Dadra and Nagar Haveli and Daman and Diu
State = Delhi
State = Goa
State = Gujarat
State = Haryana
State = Himachal Pradesh
State = Jammu and Kashmir
State = Jharkhand
State = Karnataka
State = Kerala
State = Ladakh
State = Lakshadweep
State = Madhya Pradesh
State = Maharashtra
State = Manipur
State = Meghalaya
State = Mizoram
State = Nagaland
State = Odisha
State = Puducherry
State = Punjab
State = Rajasthan
State = Sikkim
State = Tamil Nadu
State = Telangana
State = Tripura
State = Uttar Pradesh
State = Uttarakhand
State = West Bengal
Total Covaxin Administered
Total Doses Administered
1e6
1e7
Updated On
01/21
02/21
03/21
04/21
05/21

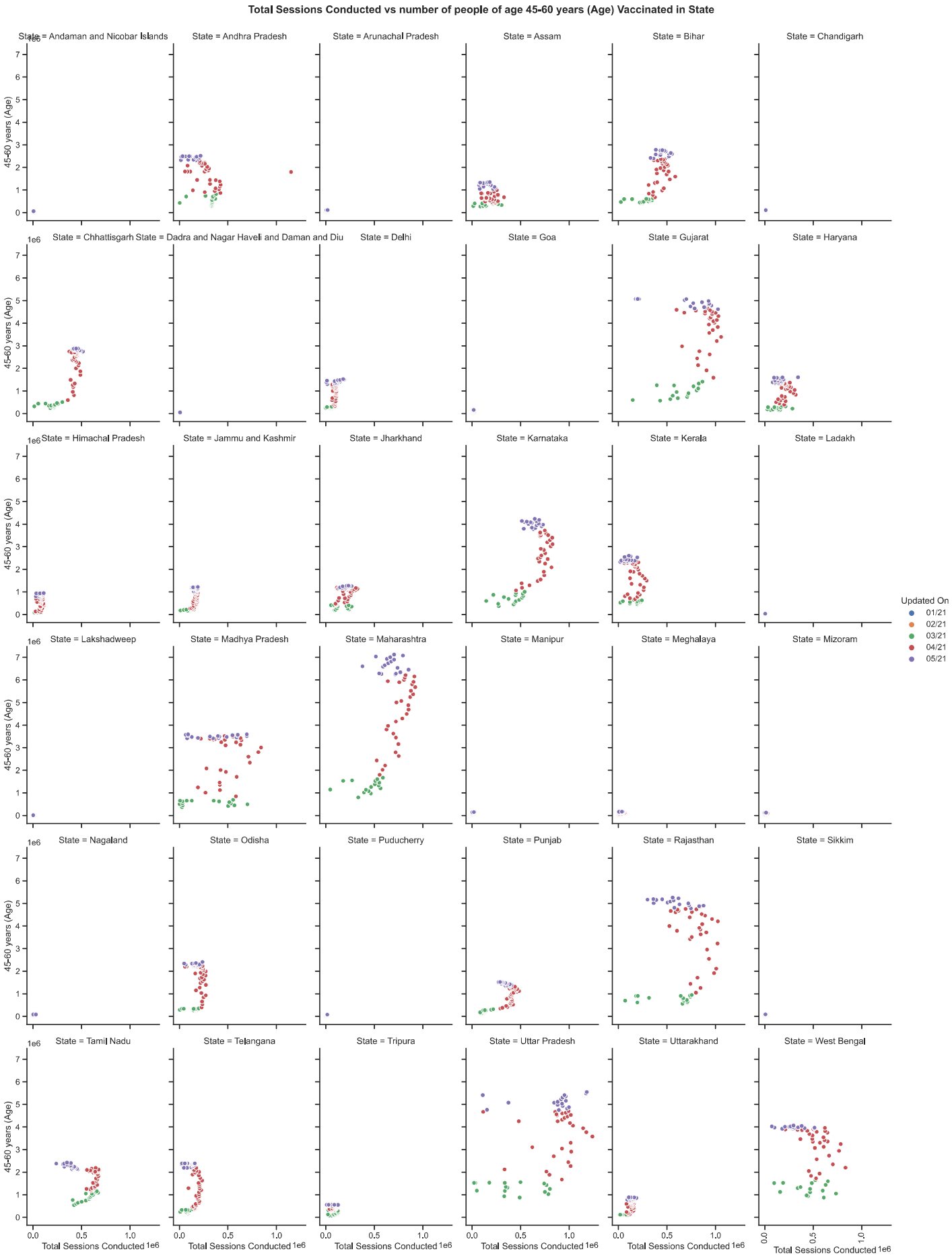


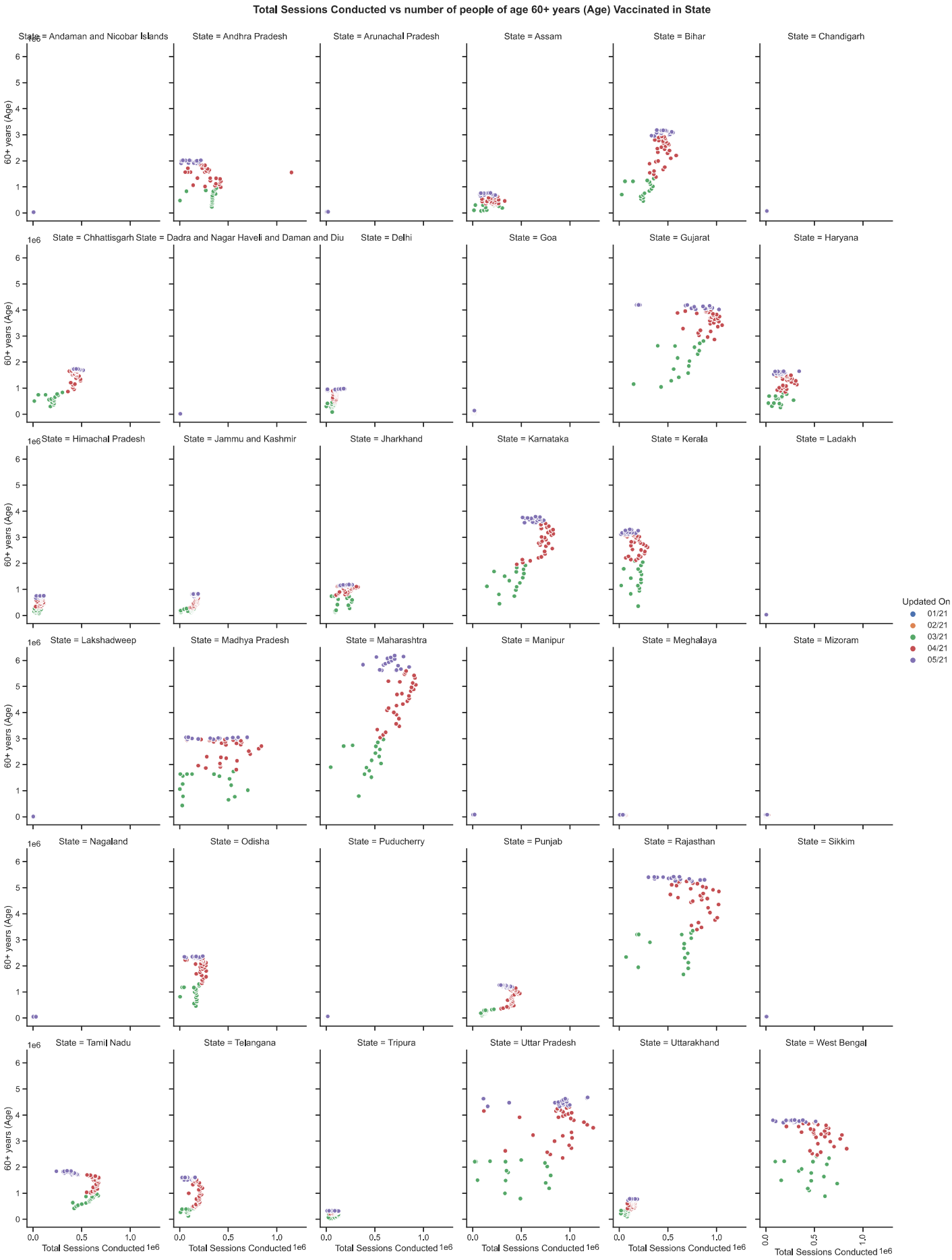
Scatterplot of individual states to Determine How many indivuals are vacinnated Age Wise in Total Number of Vaccine Sessions conducted

In [23]:
```
for i in Age:
    g = sns.relplot(data=data, x= data["Total Sessions Conducted"], y=i,col= data.State, col_wrap=
    g.fig.suptitle(f" Total Sessions Conducted vs number of people of age {i} Vaccinated in State"
    plt.subplots_adjust(top=.95)
    plt.xticks(rotation = 90)
plt.show()
```




Total Sessions Conducted vs number of people of age 30-45 years (Age) Vaccinated in State
State = Andaman and Nicobar Islands
State = Andhra Pradesh
State = Arunachal Pradesh
State = Assam
State = Bihar
State = Chandigarh
State = Chhattisgarh
State = Dadra and Nagar Haveli and Daman and Diu
State = Delhi
State = Goa
State = Gujarat
State = Haryana
State = Himachal Pradesh
State = Jammu and Kashmir
State = Jharkhand
State = Karnataka
State = Kerala
State = Ladakh
State = Lakshadweep
State = Madhya Pradesh
State = Maharashtra
State = Manipur
State = Meghalaya
State = Mizoram
State = Nagaland
State = Odisha
State = Puducherry
State = Punjab
State = Rajasthan
State = Sikkim
State = Tamil Nadu
State = Telangana
State = Tripura
State = Uttar Pradesh
State = Uttarakhand
State = West Bengal
30-45 years (Age)
Total Sessions Conducted
1e6
Updated On
01/21
02/21
03/21
04/21
05/21


Total Sessions Conducted vs number of people of age 45-60 years (Age) Vaccinated in State
State = Andaman and Nicobar Islands
State = Andhra Pradesh
State = Arunachal Pradesh
State = Assam
State = Bihar
State = Chandigarh
State = Chhattisgarh
State = Dadra and Nagar Haveli and Daman and Diu
State = Delhi
State = Goa
State = Gujarat
State = Haryana
State = Himachal Pradesh
State = Jammu and Kashmir
State = Jharkhand
State = Karnataka
State = Kerala
State = Ladakh
State = Lakshadweep
State = Madhya Pradesh
State = Maharashtra
State = Manipur
State = Meghalaya
State = Mizoram
State = Nagaland
State = Odisha
State = Puducherry
State = Punjab
State = Rajasthan
State = Sikkim
State = Tamil Nadu
State = Telangana
State = Tripura
State = Uttar Pradesh
State = Uttarakhand
State = West Bengal
45-60 years (Age)
Total Sessions Conducted
1e6
Updated On
01/21
02/21
03/21
04/21
05/21



categorical vairables

In [24]:

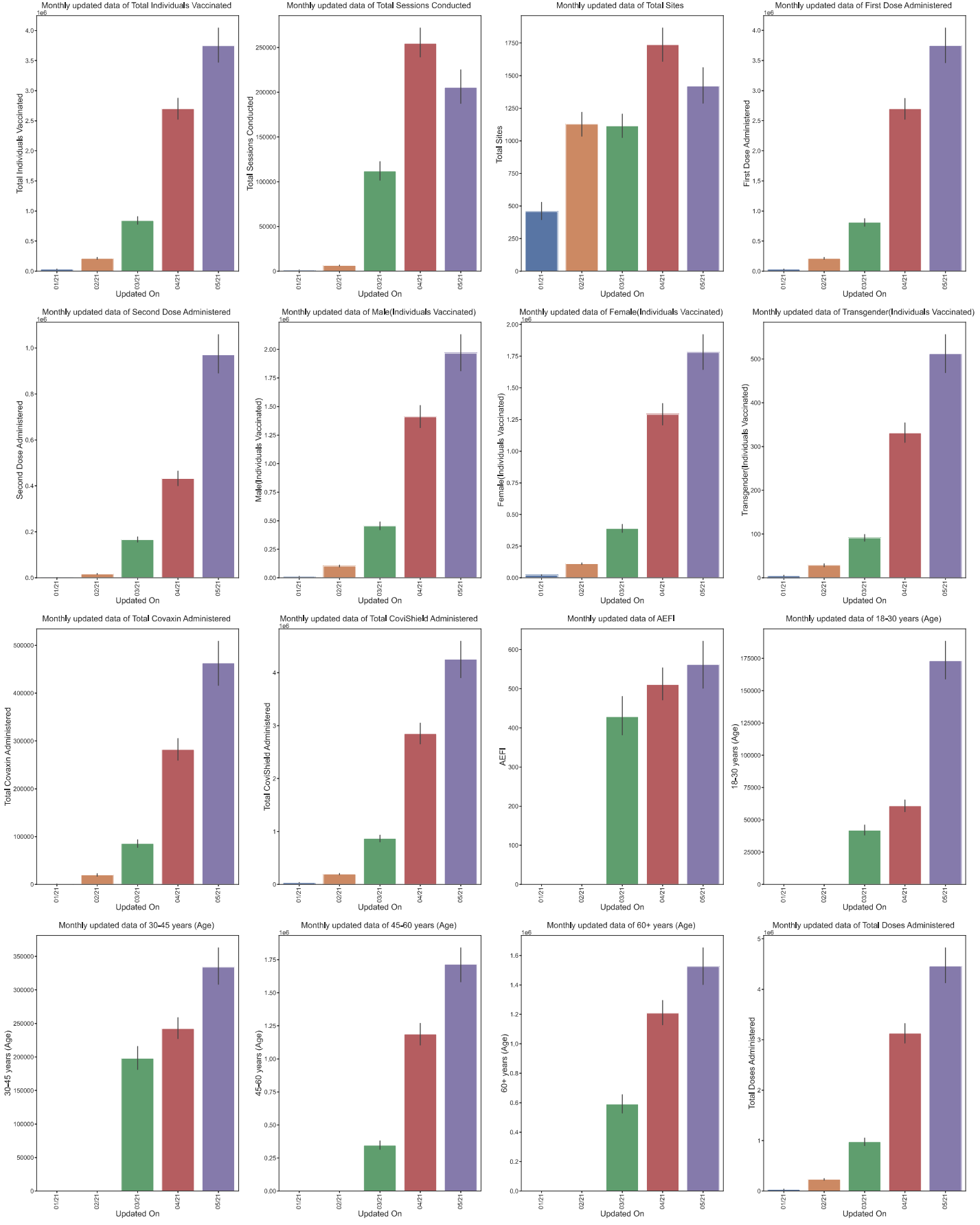
```
categorical_features=[feature for feature in df.columns if df[feature].dtypes=='O']
categorical_features
```

Out[24]: ['Updated On', 'State']

In [25]:
```
for feature in categorical_features:
    print('The feature is {} and number of categories are {}'.format(feature,len(df[feature].uniqu
```

```
The feature is Updated On and number of categories are 125
The feature is State and number of categories are 37
```

In [26]:
```
a = 4  # number of rows
b = 4  # number of columns
c = 1  # initialize plot counter

fig = plt.figure(figsize=(55, 70))
# plt.rcParams.update({'font.size': 25})
size=27
params = {'legend.fontsize': 'large',
          'axes.labelsize': size,
          'axes.titlesize': size,
          'xtick.labelsize': size*0.75,
          'ytick.labelsize': size*0.75,
          'axes.titlepad': 25}
plt.rcParams.update(params)

for i in discrete_feature:
    plt.subplot(a, b, c)
    plt.title('Monthly updated data of {} '.format(i))
    sns.barplot(x=  "Updated On", y=  i, data= data ,palette="deep")
    plt.xticks(rotation = 90)
    c = c + 1

plt.show()
```



In [27]:

```
a = 3  # number of rows
b = 1  # number of columns
c = 1  # initialize plot counter

size=22
params = {'legend.fontsize': 'large',
          'axes.labelsize': size,
          'axes.titlesize': size,
```

```
          'xtick.labelsize': size*0.75,
          'ytick.labelsize': size*0.75,
          'axes.titlepad': 25}
plt.rcParams.update(params)

fig = plt.figure(figsize=(20, 25))
plt.style.use("ggplot")

for i in categorical_features[1:]:
    plt.subplot(a, b, c)
    plt.title('{} '.format(i))
    sns.barplot(x = data[i], y = data["Total Sessions Conducted"], palette="deep")
    plt.xticks(rotation=90)


    c = c + 1

plt.show()
```