Q1

In [1]:

```
import pandas as pd
import numpy as np
import plotly.express as px
```

In [2]:

```
pd.set_option('display.max_columns', None)
```

In [3]:

```
df = pd.read_csv('Delhi-Electricity-SubStations.csv')
```

In [4]:

```
df.sample(5)
```

Out[4]:

	_id	Substations	Telephone Numbers	Address	Voltage Class	Coordinates
28	29	Pragati	23378308	Pragti Power Station Complex, IP Estate, Ring ...	220 kV	28.61508, 77.25007
36	37	Shalimar Bagh	27854692	Sector-19, Rohini, Opp. Rohini District, Jail,...	220 kV	28.72746, 77.14096
32	33	Rohini-I	27573965	Near F-2 Pocket, Rohini, Delhi-110085	220 kV	28.73254, 77.10797
16	17	Masjid Moth	26241262	Masjid Moth, Outer Ring Road, G.K.-I Near Chir...	220 kV	28.54181, 77.23242
10	11	Gopalpur	27652190	Near Burari, Outer Ring Road, Delhi-110009	220 kV	28.72189, 77.21667

In [5]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 43 entries, 0 to 42
Data columns (total 6 columns):
 #   Column             Non-Null Count  Dtype 
---  ------             --------------  ----- 
 0   _id                43 non-null     int64 
 1   Substations        43 non-null     object
 2   Telephone Numbers  40 non-null     object
 3   Address            43 non-null     object
 4   Voltage Class      43 non-null     object
 5   Coordinates        41 non-null     object
dtypes: int64(1), object(5)
memory usage: 2.1+ KB
```

In [6]:

```
df.isnull().sum()
```

Out[6]:

```
_id                  0
Substations          0
Telephone Numbers    3
Address              0
Voltage Class        0
Coordinates          2
dtype: int64
```

In [7]:

```
df.duplicated().sum()
```

Out[7]:

```
0
```

Q1.1 In Telephone Numbers there are multiple numbers. Create separate columns for these Telephone1, Telephone2 etc.

In [8]:

```
# replacing , with / in telephone number column for id with 28
df.loc[27, 'Telephone Numbers'] = df.loc[27, 'Telephone Numbers'].replace(',', '/').rstrip
```

In [9]:

```
df.iloc[27]
```

Out[9]:

```
_id                                                                28
Substations                                                Peeragarhi
Telephone Numbers                                  25276576/ 25276578
Address              A-1 Block Near Milansaar Apptt. Opp. A-1/266, ...
Voltage Class                                                  220 kV
Coordinates                                        28.67118, 77.09605
Name: 27, dtype: object
```

In [10]:

```
df['Telephone Numbers'].isnull().sum()
```

Out[10]:

```
3
```

In [11]:

```
df['Telephone Numbers'].fillna('Not available',inplace=True)
```

In [12]:

```
df['Telephone1']= df['Telephone Numbers'].apply(lambda x : x.split('/',1)[0])
df['Telephone2'] =  df['Telephone Numbers'].apply(lambda x : x.split('/',1)[1] if len(x.sp
```

In [13]:

```
# removing  / in telephone2  column for id with 1 ,  I think Its a typing misktake
df.loc[0,'Telephone2'] = df.loc[0, 'Telephone2'].replace('/','')
```

In [14]:

```
# Final output
df
```

Out[14]:

	_id	Substations	Telephone Numbers	Address	Voltage Class	Coordinates	Telephone1	Telephone2
0	1	Bawana	27791190/1193/1210	400kV Sub-Station Bawana, Sector-5, DSIIDC Baw...	400 kV	28.79568, 77.0723	27791190	11931210
1	2	Bamnauli	25314199/25314204	Village-Bamnauli, P.O. Dhul Sirus, Near Chhawa...	400 kV	28.5447, 77.03269	25314199	25314204
2	3	Harsh Vihar	0120-6500138	Harsh Vihar, Loni Road (Near Bhopura Chowk), D...	400 kV	28.71185, 77.29044	0120-6500138	Not available
3	4	Tikri Kalan(Mundka)	65108444/7290010893	Neewala Village Road, Near Vaishno Devi Mandir...	400 kV	28.67671, 76.98639	65108444	7290010893
4	5	BTPS	26948637	Badarpur Thermal Power Station Complex, Badarp...	220 kV	28.50765, 77.30015	26948637	Not available
5	6	DSIDC Bawana	65005603/7290013476	Near H Block, J.J. colony, Bawana Hanuman Mand...	220 kV	28.80361, 77.05284	65005603	7290013476
6	7	DIAL	25655090	Near IGI Airport, Delhi	220 kV	28.55616, 77.09995	25655090	Not available
7	8	Electric Lane	23329790	HCM Lane, Behind BSNL Building, Janpath, New D...	220 kV	28.62174, 77.21969	23329790	Not available
8	9	Geeta Colony	22044528	Near Sai Memorial Public School, Geeta Colony,...	220 kV	28.65226, 77.26774	22044528	Not available
9	10	Gazipur	22770009	Village Gazipur, Near DDA Janta Flat, Delhi-11...	220 kV	28.62961, 77.31898	22770009	Not available
10	11	Gopalpur	27652190	Near Burari, Outer Ring Road, Delhi-110009	220 kV	28.72189, 77.21667	27652190	Not available
11	12	Indraprastha	23370325	IP Station, Delhi-110002	220 kV	28.62711, 77.24726	23370325	Not available
12	13	Khanjawala	25951677/25953185	Near Police Station & DTC Depott., Ghewra Road...	220 kV	28.69137, 76.99758	25951677	25953185
13	14	Kashmere Gate	23866577	Kudasiya Ghat, Opp, I.S.B.T, Kashmeri Gate, De...	220 kV	28.66657, 77.22906	23866577	Not available
14	15	Lodhi Road	24362750	Near CGO Complex, Lodhi Road, New Delhi-110003	220 kV	28.59262, 77.23759	24362750	Not available
15	16	Maharani Bagh	26841323/26341324	Maharani Bagh, Opp. Sarai Kale Khan ISBT, Behl...	220 kV	28.58704, 77.26738	26841323	26341324
16	17	Masjid Moth	26241262	Masjid Moth, Outer Ring Road, G.K.-I Near Chir...	220 kV	28.54181, 77.23242	26241262	Not available
17	18	Mehrauli	26892360	Near Vasant Kunj, D-2,, Kishangarh Village, Ne...	220 kV	28.51996, 77.16834	26892360	Not available
18	19	Najafgarh	69999237/69999238	Opp. Delhi Jal Board Office, New Delhi-110043	220 kV	28.61277, 76.99618	69999237	69999238
19	20	Naraina	25699563	Behind COD Bus Stand, Ring Road, Naraina, Delh...	220 kV	28.61292, 77.13663	25699563	Not available
20	21	Narela	27781434/22781771	Piyo Maniyari Road, Near Kanya Gurkul, Narela,...	220 kV	28.86308, 77.09617	27781434	22781771

	_id	Substations	Telephone Numbers	Address	Voltage Class	Coordinates	Telephone1	Telephone2
21	22	Okhla	29982007/29985044	Opp. Pocket A-13, Kalkaji Extension, New Delhi...	220 kV	28.52724, 77.26601	29982007	29985044
22	23	Pappankalan-I	25085415/25080740	Sector-2, Pappankalan, Phase-I, Delhi-110045	220 kV	28.6083, 77.05726	25085415	25080740
23	24	Papankalan-II	24506081	Near Metro Rail Station, Sector-14, Dwarka, De...	220 kV	28.60205, 77.02661	24506081	Not available
24	25	Papankalan-III	Not available	Sector-19 B, Dwarka, New Delhi-110075	220 kV	28.57641, 77.04328	Not available	Not available
25	26	Park Street	23744910/23364916	Opp. Talkatora Stadium, Near RML Hospital, Par...	220 kV	28.62529, 77.19523	23744910	23364916
26	27	Patparganj	22476611/22476241	Near Mother Diary, Mandawali, Fazalpur, Delhi-...	220 kV	28.62779, 77.2942	22476611	22476241
27	28	Peeragarhi	25276576/ 25276578	A-1 Block Near Milansaar Apptt. Opp. A-1/266, ...	220 kV	28.67118, 77.09605	25276576	25276578
28	29	Pragati	23378308	Pragti Power Station Complex, IP Estate, Ring ...	220 kV	28.61508, 77.25007	23378308	Not available
29	30	Preet Vihar	22372310	Preet Vihar, Near Kadkadi Mod, Behind (Deepak ...	220 kV	28.63854, 77.29483	22372310	Not available
30	31	Rajghat Power House	26895886	Control Room, Inside Rajghat Power House, New ...	220 kV	28.64062, 77.2556	26895886	Not available
31	32	Ridge Valley	24112187	Dhaula Kaun, Round About, New Delhi-110021	220 kV	NaN	24112187	Not available
32	33	Rohini-I	27573965	Near F-2 Pocket, Rohini, Delhi-110085	220 kV	28.73254, 77.10797	27573965	Not available
33	34	Rohini-II	7428366440	Phase-IV, Rohini Sector-29, Landmark Opp. CNG ...	220 kV	28.72751, 77.1428	7428366440	Not available
34	35	R. K. Puram	Not available	Sector-4, R. K. Puram, Near Sultani Masjid, Ne...	220 kV	28.55539, 77.17773	Not available	Not available
35	36	Sarita Vihar	26940624/9643740060	Opp. Pocket - E, Near Priyanka Complex, Sarita...	220 kV	28.52751, 77.303	26940624	9643740060
36	37	Shalimar Bagh	27854692	Sector-19, Rohini, Opp. Rohini District, Jail,...	220 kV	28.72746, 77.14096	27854692	Not available
37	38	South of Wazirabad	22855028/22566216	Opp. Viil. Garhi, Near Khajuri Khas Chowk, Del...	220 kV	NaN	22855028	22566216
38	39	Subzi Mandi	23855513	Kabir Basti Road, Near Barafkhana Chowk,Subzi ...	220 kV	28.66661, 77.19888	23855513	Not available
39	40	Trauma Centre	32973334	GIS, Trauma, IMS Center, Ansari Nagar, West, N...	220 kV	28.56818, 77.20068	32973334	Not available
40	41	Tughlakabad	Not available	Behind Air Force Residential Quarter, Karni si...	220 kV	28.50568, 77.26196	Not available	Not available
41	42	Vasant Kunj	26895886	C-9 Vasant Kunj, Delhi-110070.	220 kV	28.53959, 77.14348	26895886	Not available

	_id	Substations	Telephone Numbers	Address	Voltage Class	Coordinates	Telephone1	Telephone2
42	43	Wazirpur	27373358/27373359	A Block, Wazirpur Industrial Area, New Delhi -...	220 kV	28.69892, 77.16073	27373358	27373359

FINDING LONGITUDE AND LATTITUDE USING PGEOCODE LIBRARY

In [15]:

```python
import pgeocode
```

In [16]:

```python
geo = pgeocode.Nominatim('IN')
```

In [17]:

```python
def get_cordinates(address):
    address = address[-6:]
    my_list = ''
    try:
        # Get location information based on the address
        location_info = geo.query_postal_code(address)

        if not location_info.empty:
            latitude = location_info['latitude']
            longitude = location_info['longitude']
            my_list += str(latitude) +','
            my_list += str(longitude)
#             print(f"Latitude: {latitude}, Longitude: {longitude}")
#             print(my_list)
        else:
            print("Address not found.")
    except Exception as e:
        print(f"Error: {str(e)}")

    return my_list

```

In [18]:

```python
df['new_Cordinates'] = df['Address'].apply(get_cordinates)
```

In [19]:

```python
df
```

	_id	Substations	Telephone Numbers	Address	Voltage Class	Coordinates	Telephone1	Telephone2	new_
0	1	Bawana	27791190/1193/1210	400kV Sub-Station Bawana, Sector-5, DSIIDC Baw...	400 kV	28.79568, 77.0723	27791190	11931210	28.804991666666663,77.0308
1	2	Bamnauli	25314199/25314204	Village-Bamnauli, P.O. Dhul Sirus, Near Chhawa...	400 kV	28.5447, 77.03269	25314199	25314204	28.5
2	3	Harsh Vihar	0120-6500138	Harsh Vihar, Loni Road (Near Bhopura Chowk), D...	400 kV	28.71185, 77.29044	0120-6500138	Not available	28.6

In [20]:

```
df['Coordinates'] = df['Coordinates'].fillna('0,0')
df['new_Cordinates'] = df['new_Cordinates'].fillna('0,0')
```

In [21]:

```
df[['Latitude', 'Longitude']] = df['Coordinates'].str.split(',', expand=True)
df[['Latitude', 'Longitude']] = df[['Latitude', 'Longitude']].astype(float)
```

In [22]:

```
df[['new_Latitude', 'new_Longitude']] = df['new_Cordinates'].str.split(',',expand=True)
df['new_Latitude'] = df['new_Latitude'].apply(lambda x: 0 if x=='nan' else x)
df['new_Longitude'] = df['new_Longitude'].apply(lambda x: 0 if x=='nan' else x)
df[['new_Latitude', 'new_Longitude']] = df[['new_Latitude', 'new_Longitude']].astype(float
```

In [23]:

```
df['distance'] = df.apply(lambda row: (
                (row['Latitude'] - row['new_Latitude']) ** 2 +
                (row['Longitude'] - row['new_Longitude']) ** 2)
                          ** 0.5, axis=1)
```

In [24]:

```
def calculate_accuracy(distance):
    if distance == 0:
        return 100.0
    else:
        accuracy = max(0, 100 - (distance / 10))
        return accuracy
```

In [25]:

```
df['GeoAccuracy'] = df['distance'].apply(calculate_accuracy)
```

In [26]:

```python
df['GeoAccuracy']
```

Out[26]:

```
0     99.995756
1     99.995428
2     99.989925
3     99.997223
4     99.999038
5     99.997801
6     91.778165
7     91.764659
8     99.999213
9     99.997237
10    99.998552
11    99.998173
12    99.996117
13    99.997323
14    99.998637
15    99.998670
16    91.766239
17    99.997118
18    99.998125
19    99.987074
20    99.997763
21    99.987944
22    99.996750
23    91.783446
24    99.995341
25    99.997402
26    99.998515
27    99.982677
28    99.996945
29    99.998049
30    99.998896
31    91.769834
32    99.998649
33    99.995922
34    99.995070
35    99.997461
36    99.993023
37    91.757386
38    99.998702
39    91.768302
40    99.995346
41    91.774658
42    99.991189
Name: GeoAccuracy, dtype: float64
```

In [ ]:

```python

```

Q2

1. Merge the data of each year (2000-2001, 2001-2002,….) to a single period (2000-2018) by the region and sub region.

In [27]:

```python
df1 = pd.read_csv('MH-Vehicle-Reg.csv',sep=';')
```

In [28]:

```
df1.head(5)
```

Out[28]:

	_id	Sr No.	Year	Region	Sub Region	Motor Cycles	Scooters	Moped	Cars	Jeeps	Stn. Wagons	Taxis meter fited	Luxury /Turist Cabs/	Auto-rikshaws
0	1	1	2000-2001	Greater Mumbai	Mumbai(C)	84289	62444	7289	164758	12727	2705	40451	0	0
1	2	2	2000-2001	Greater Mumbai	Mumbai(W)	80320	96297	15230	110397	5465	962	11460	0	60128
2	3	3	2000-2001	Greater Mumbai	Mumbai(E)	39930	44932	9786	42945	4439	472	10536	0	41786
3	4	4	2000-2001	Greater Mumbai	Borivali	0	0	0	0	0	0	0	0	0
4	5	5	2000-2001	Thane Region	Thane	130448	104028	9304	96933	23755	730	3320	0	54827

In [29]:

```
# making the year column only a singular value ,
# eg. if 2000 -2001 then I will consider it year 2000
```

In [30]:

```
df1['Year']= df1['Year'].apply(lambda x : x.split("-")[0])
```

In [31]:

```
df1.sample(5)
```

Out[31]:

	_id	Sr No.	Year	Region	Sub Region	Motor Cycles	Scooters	Moped	Cars	Jeeps	Stn. Wagons	Taxis meter fited	Luxury /Turist Cabs/	Au riksha
520	521	521	2010	Pune Region	Akluj	77475	6	14	8691	2025	13	0	233	10
821	822	822	2016	Nashik Region	Nashik	815599	220350	54882	155647	26442	94	2100	3656	222
833	834	834	2016	Latur Region	Ambejogai	77217	6305	3789	5717	3447	0	73	340	31
416	417	417	2008	Pune Region	Pune	807767	310134	194312	217283	39063	950	0	12923	606
540	541	541	2010	Amrawati Region	Akola	100487	34532	31096	10787	3839	8	42	1865	98

In [32]:

```
# groupping
grouped_df =df1.groupby(["Year",'Region','Sub Region']).sum(numeric_only=True).iloc[:,2:]
```

In [33]:

```
grouped_df
```

Out[33]:

			Motor Cycles	Scooters	Moped	Cars	Jeeps	Stn. Wagons	Taxis meter fited	Luxury /Turist Cabs/	Auto-rikshaws	Stage carriages
Year	**Region**	**Sub Region**										
2000	**Amrawati Region**	**Akola**	41444	21900	30166	3997	4127	12	639	0	5823	430
		Amrawati	33351	29795	50414	4178	3100	319	322	0	8260	493
		Buldhana	27600	6496	13003	1347	2131	180	465	0	3245	438
		Washim	0	0	0	0	0	0	0	0	0	0
		Yawatmal	24564	13259	24593	2002	2397	48	357	0	3623	399
...	**...**	**...**	...	...	...	...	...	...	...	...	...	...
2017	**Pune Region**	**Solapur**	568356	81072	72939	51192	15861	192	20	2409	11938	292
	Thane Region	**Kalyan**	601959	76014	9148	123090	4487	129	793	7471	57077	320
		Thane	843715	297194	13608	433402	45414	5469	3398	38546	107157	1058
		Vasai	200000	68550	815	70689	276	72	175	5186	17393	791
		Vashi Navi Mumbai	199046	11253	5	115245	4073	570	1641	12570	22324	304

900 rows × 22 columns

2. Write a function that would take year as parameter and plot a graph. Classify the vehicles into 3 categories Private, Commercial and Others. Commercial vehicles will have 2 subcategories Light Motor Vehicles (LMV) and Heavy Motor Vehicles (HMV). Use different colors for each type of vehicle and also show the legend. The graph should show both the values and percentages.

1. Consider Motorcycles, Scooters, Moped, Cars, Jeeps, Stn. Wagons and Private Service Vehicles as Private and rest as Commercial.
2. Bus, Truck, Trailer, Tractor, Tanker, Stage/Contract carriage are all HMV and rest LMV.

In [34]:

```python
new_df = df1.groupby('Year').sum(numeric_only=True).iloc[:,2:]
new_df
```

Out[34]:

	Motor Cycles	Scooters	Moped	Cars	Jeeps	Stn. Wagons	Taxis meter fited	Luxury /Turist Cabs/	Auto-rikshaws	Stage carriages	Contract carriages /Mini Bus	Scho Buse
Year												
2000	2151944	1394472	972935	692212	216998	12718	89308	0	413828	27338	13910	174
2001	2540274	1542122	1040730	757229	230427	12662	97081	0	440018	26439	11850	189
2002	2988428	1497876	1101358	831261	244025	12599	94920	0	463550	26991	12355	200
2003	3493179	1562751	1160864	924006	262741	12609	102475	0	493142	27576	12904	225
2004	4072397	1640891	1213288	1039800	283016	12613	113380	0	515249	29024	18043	291
2005	4733612	1715355	1242889	1165365	300023	14489	122389	0	534535	29506	19208	321
2006	4731165	1873302	1161013	1289206	269977	24131	124213	89372	472241	24830	20281	517
2007	6185422	1876020	1302966	1458554	338253	18100	60946	88417	564445	30530	24049	412
2008	6887857	1962276	1362207	1603728	356986	18477	63534	94382	598013	30570	26783	469
2009	7726106	2054666	1400990	1790259	373958	18752	63970	104337	626332	31823	28564	522
2010	8799451	2190658	1438902	2027080	394647	18677	57047	111449	640700	34134	30415	741
2011	10093662	2353201	1474900	2307841	423305	19021	59216	116581	640040	32645	29663	1778
2012	11416537	2526109	1514527	2592565	453380	19414	59518	124548	655299	35133	33257	1984
2013	12585191	2779318	1545886	2834847	471795	19201	60635	128152	658977	36780	37052	2365
2014	13960692	3069600	1573543	3113773	496255	19352	64068	129322	695619	37087	40191	1838
2015	15268420	3481296	1606109	3406872	517239	19244	67304	169702	703030	37443	45317	2232
2016	16637560	3840235	1624581	3705828	537386	19346	78909	226797	753373	38612	54755	2529
2017	18335895	4078720	1680421	4063866	550000	18812	80895	256274	863196	37351	59398	2680

In [35]:

```python
new_df['Private'] = (new_df['Motor Cycles'] +
                     new_df['Scooters'] +
                     new_df['Moped'] +
                     new_df['Cars'] +
                     new_df['Jeeps']+
                     new_df['Stn. Wagons'] +
                     new_df['Private Service Vehicles']) #7
```

In [36]:

```python
new_df['Commercial (HMV)'] = (new_df['School Buses'] +
                              new_df['Contract carriages /Mini Bus'] +
                              new_df['Trucks & Lorries'] +
                              new_df['Tanker'] +
                              new_df['Stage carriages']+
                              new_df['Tractors'] +
                              new_df['Trailors']) #7 Heavy Motor Vehicles
```

In [37]:

```python
new_df['Commercial (LMV)'] = (new_df['Taxis meter fited'] +
                              new_df['Luxury /Turist Cabs/'] +
                              new_df['Auto-rikshaws'] +
                              new_df['Ambulances']+
                              new_df['Articulated/Multi.']+
                              new_df['Delivery Van (4 wheelers)']+
                              new_df['Delivery Van (3 wheelers)']) #8 Light Moto
```

In [38]:

```
final= new_df.iloc[:,-4:]
```

In [39]:

```
final
```

Out[39]:

	Others	Private	Commercial (HMV)	Commercial (LMV)
Year				
2000	10058	5447527	603946	697943
2001	10199	6129543	626152	747677
2002	10678	6681544	650554	791061
2003	11217	7422511	673794	861211
2004	12444	8269610	718328	935583
2005	14957	9180227	762160	1009090
2006	17576	9357030	746291	1168525
2007	21320	11189433	893566	1171422
2008	24080	12200293	964651	1261864
2009	27066	13373834	1023309	1344212
2010	27188	14877741	1105272	1423898
2011	32557	16681381	1206343	1512080
2012	35734	18532626	1291711	1628081
2013	41695	20246970	1402934	1702177
2014	44538	22244562	1487789	1815286
2015	54407	24310802	1572948	1931709
2016	58452	26377706	1670246	2120443
2017	63290	28739823	1768382	2317648

In [40]:

```
temp= final[final.index== '2017'].reset_index()
```

In [41]:

```
temp
```

Out[41]:

	Year	Others	Private	Commercial (HMV)	Commercial (LMV)
0	2017	63290	28739823	1768382	2317648

In [42]:

```
temp= temp.melt(id_vars=['Year'], var_name='Category', value_name='Count')
```

In [43]:

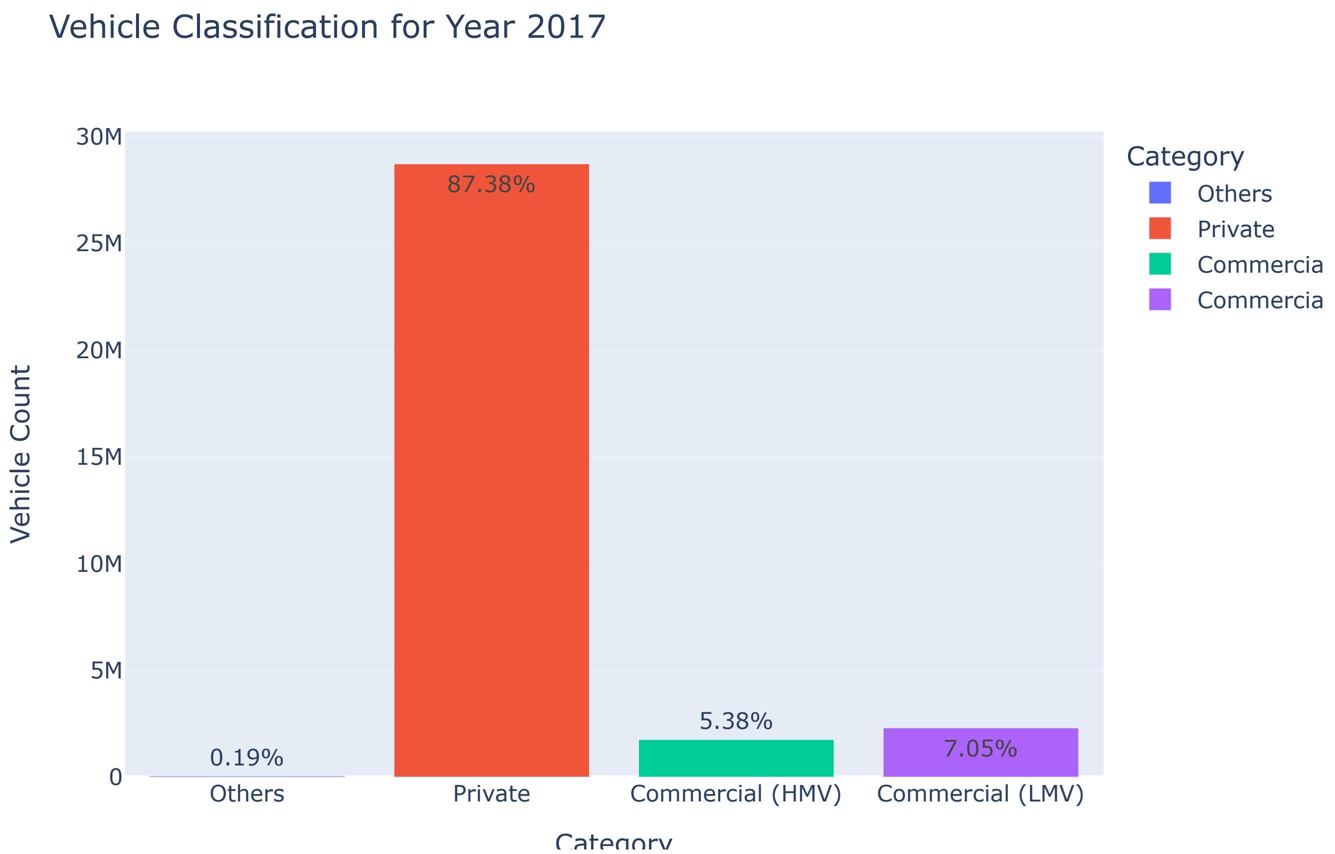
```
temp['Percentage'] =(temp['Count'] / temp['Count'].sum()) * 100
temp['Percentage'] = np.round(temp['Percentage'], 2)
temp['Percentage']=temp['Percentage'].apply(lambda x : str(x) + "%")
temp
```

Out[43]:

	Year	Category	Count	Percentage
0	2017	Others	63290	0.19%
1	2017	Private	28739823	87.38%
2	2017	Commercial (HMV)	1768382	5.38%
3	2017	Commercial (LMV)	2317648	7.05%

In [44]:

```
fig = px.bar(
        temp,
        x='Category',
        y='Count',
        color='Category',
        labels={'Count': 'Vehicle Count'},
        title=f'Vehicle Classification for Year {2017}',
        text='Percentage'  # Display count values on the bars
    )
fig.show()
```

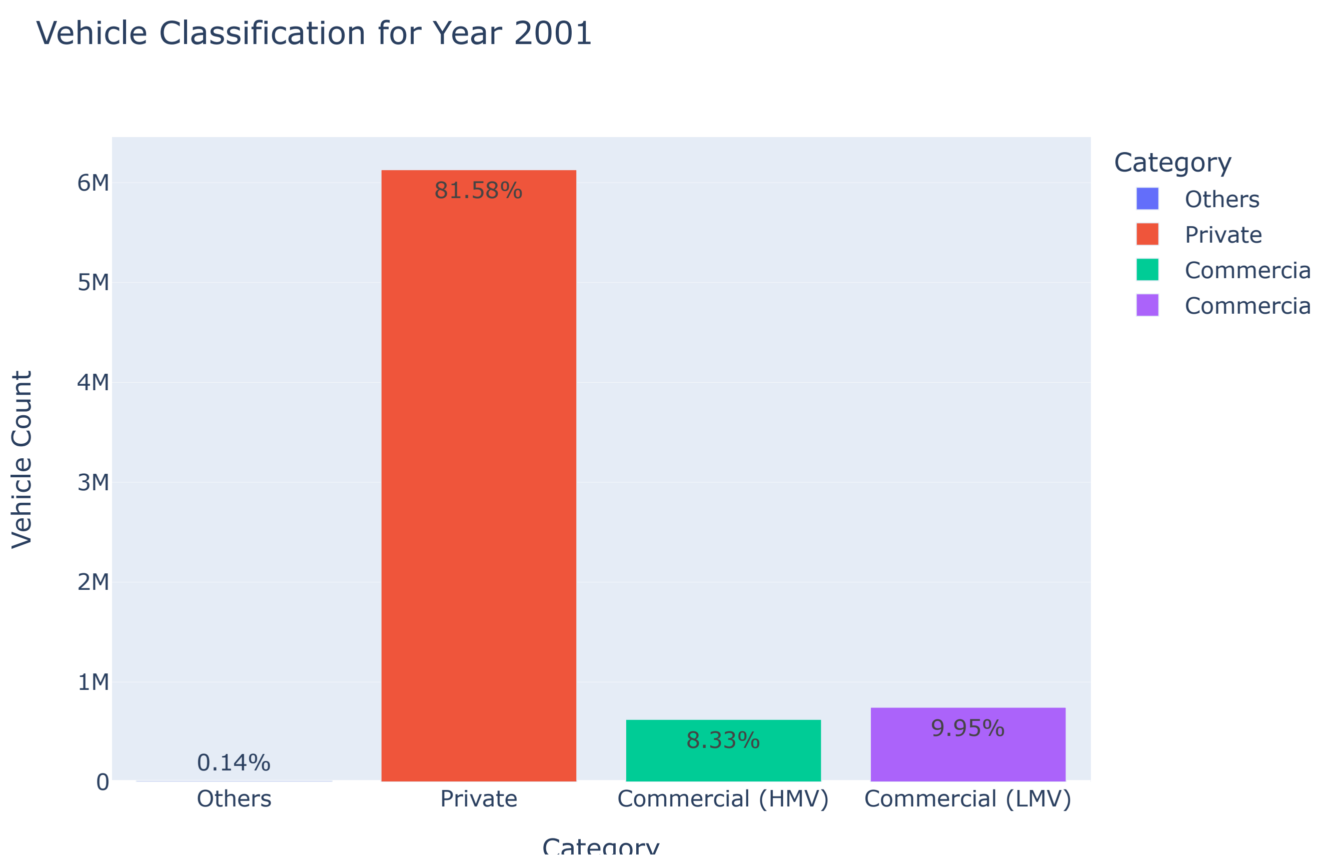


In [45]:

```
def generate_plot(year):
    temp= final[final.index== str(year)].reset_index()
    temp= temp.melt(id_vars=['Year'], var_name='Category', value_name='Count')
    temp['Percentage'] =(temp['Count'] / temp['Count'].sum()) * 100
    temp['Percentage'] = np.round(temp['Percentage'], 2)
    temp['Percentage']=temp['Percentage'].apply(lambda x : str(x) + "%")

    fig = px.bar(
        temp,
        x='Category',
        y='Count',
        color='Category',
        labels={'Count': 'Vehicle Count'},
        title=f'Vehicle Classification for Year {year}',
        text='Percentage'  # Display count values on the bars
    )
    fig.show()
```

```
In [46]: generate_plot(2001)
```



```
In [ ]:
```