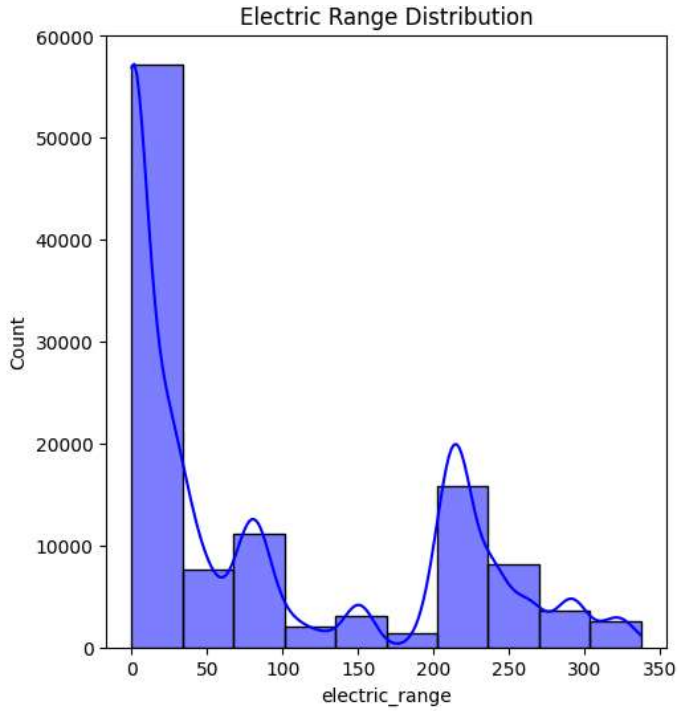
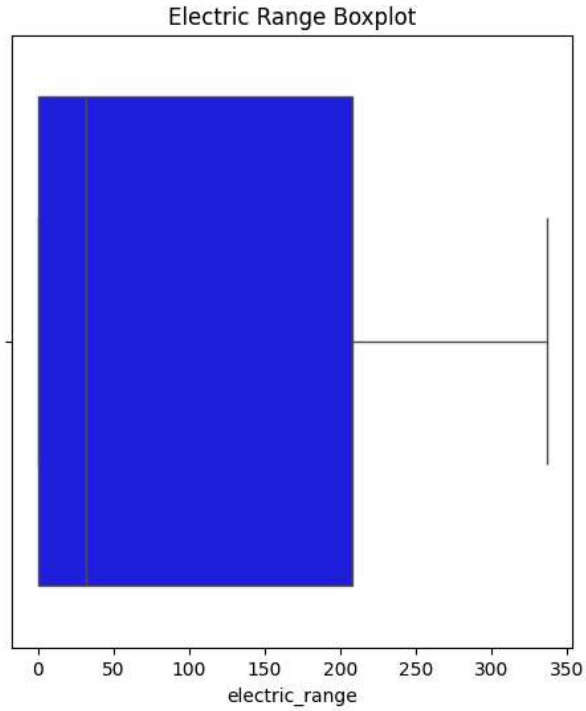
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns


df = pd.read_csv('/content/dataset.csv')


df.head()
```

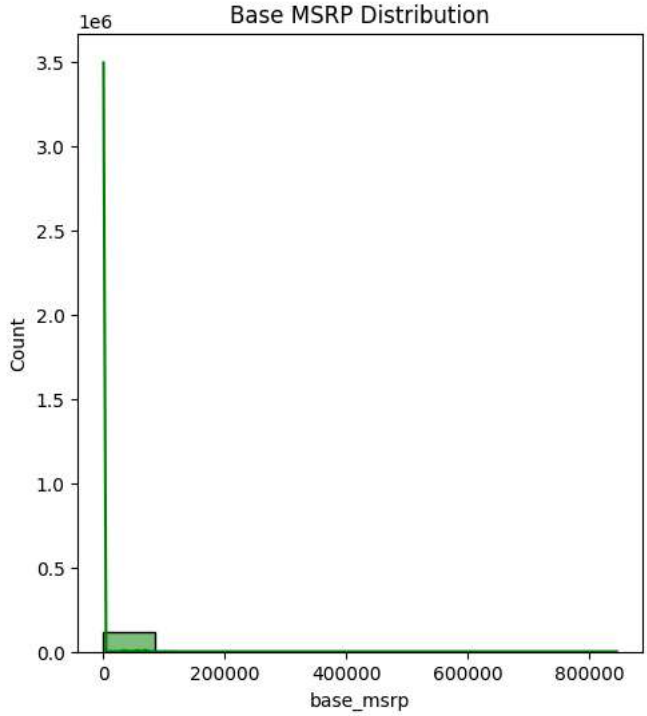
	VIN (1-10)	County	City	State	Postal Code	Model Year	Make	Model	Electric Vehicle Type	Clean Alternative Fuel Vehicle (CAFV) Eligibility	Electric Range	Base MSRP	Legislative District	Ve
0	JTMEB3FV6N	Monroe	Key West	FL	33040	2022	TOYOTA	RAV4 PRIME	Plug-in Hybrid Electric Vehicle (PHEV)	Clean Alternative Fuel Vehicle Eligible	42	0	NaN	1989
1	1G1RD6E45D	Clark	Laughlin	NV	89029	2013	CHEVROLET	VOLT	Plug-in Hybrid Electric Vehicle (PHEV)	Clean Alternative Fuel Vehicle Eligible	38	0	NaN	52
2	JN1AZ0CP8B	Yakima	Yakima	WA	98901	2011	NISSAN	LEAF	Battery Electric Vehicle (BEV)	Clean Alternative Fuel Vehicle Eligible	73	0	15.0	2189
3	1G1FW6S08H	Skagit	Concrete	WA	98237	2017	CHEVROLET	BOLT EV	Battery Electric Vehicle (BEV)	Clean Alternative Fuel Vehicle Eligible	238	0	39.0	1867
4	3FA6P0SU1K	Snohomish	Everett	WA	98201	2019	FORD	FUSION	Plug-in Hybrid Electric Vehicle (PHEV)	Not eligible due to low battery range	26	0	38.0	20

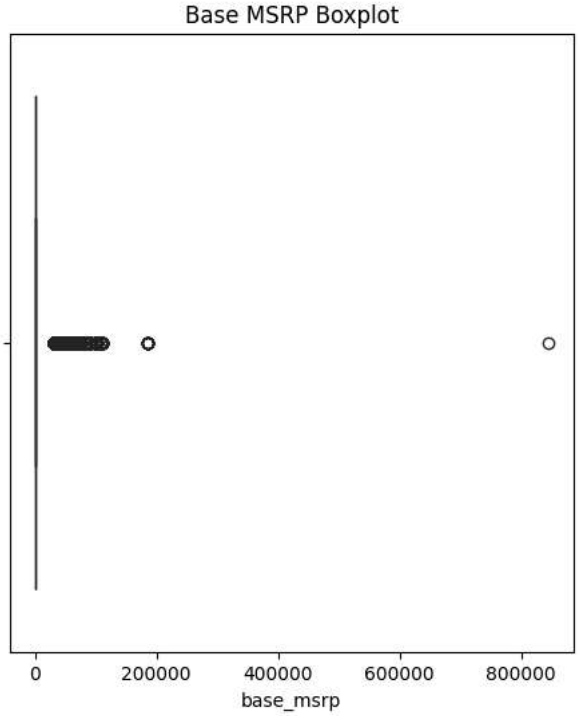
```
df.columns = df.columns.str.strip()
df.columns = df.columns.str.lower()
df.columns = df.columns.str.replace(' ', '_')
df.columns = df.columns.str.replace('[()/-]', '', regex=True)


df.columns = ['vin_1_10', 'county', 'city', 'state', 'postal_code', 'model_year',
              'make', 'model', 'electric_vehicle_type',
              'clean_alternative_fuel_vehicle_cafv_eligibility', 'electric_range',
              'base_msrp', 'legislative_district', 'dol_vehicle_id',
              'vehicle_location', 'electric_utility', 'census_tract_2020']


plt.figure(figsize=(12, 6))
plt.subplot(1, 2, 1)
sns.histplot(df['electric_range'], kde=True, bins=10, color='blue')
plt.title('Electric Range Distribution')

plt.subplot(1, 2, 2)
sns.boxplot(x=df['electric_range'], color='blue')
plt.title('Electric Range Boxplot')
plt.show()

# Base MSRP
plt.figure(figsize=(12, 6))
plt.subplot(1, 2, 1)
sns.histplot(df['base_msrp'], kde=True, bins=10, color='green')
plt.title('Base MSRP Distribution')

plt.subplot(1, 2, 2)
sns.boxplot(x=df['base_msrp'], color='green')
plt.title('Base MSRP Boxplot')
```

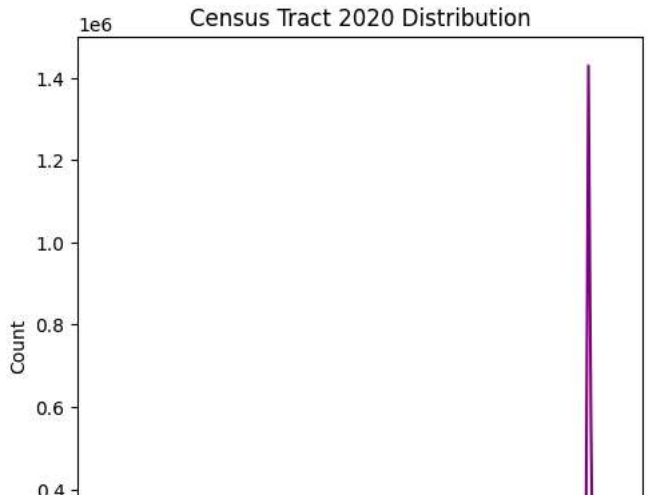
```
plt.show()

# Census Tract 2020
plt.figure(figsize=(12, 6))
plt.subplot(1, 2, 1)
sns.histplot(df['census_tract_2020'], kde=True, bins=10, color='purple')
plt.title('Census Tract 2020 Distribution')

plt.subplot(1, 2, 2)
sns.boxplot(x=df['census_tract_2020'], color='purple')
plt.title('Census Tract 2020 Boxplot')
plt.show()
```

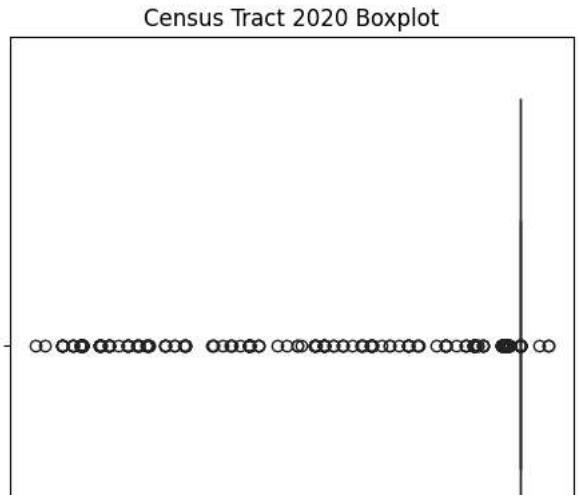

Electric Range Distribution
Count
electric_range
0
10000
20000
30000
40000
50000
60000
0
50
100
150
200
250
300
350

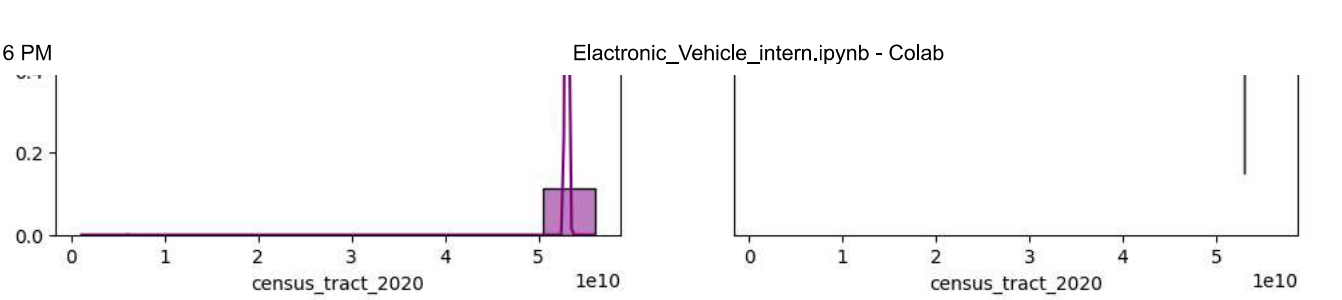

Electric Range Boxplot
electric_range
0
50
100
150
200
250
300
350


Base MSRP Distribution
1e6
Count
base_msrp
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
0
200000
400000
600000
800000


Base MSRP Boxplot
base_msrp
0
200000
400000
600000
800000

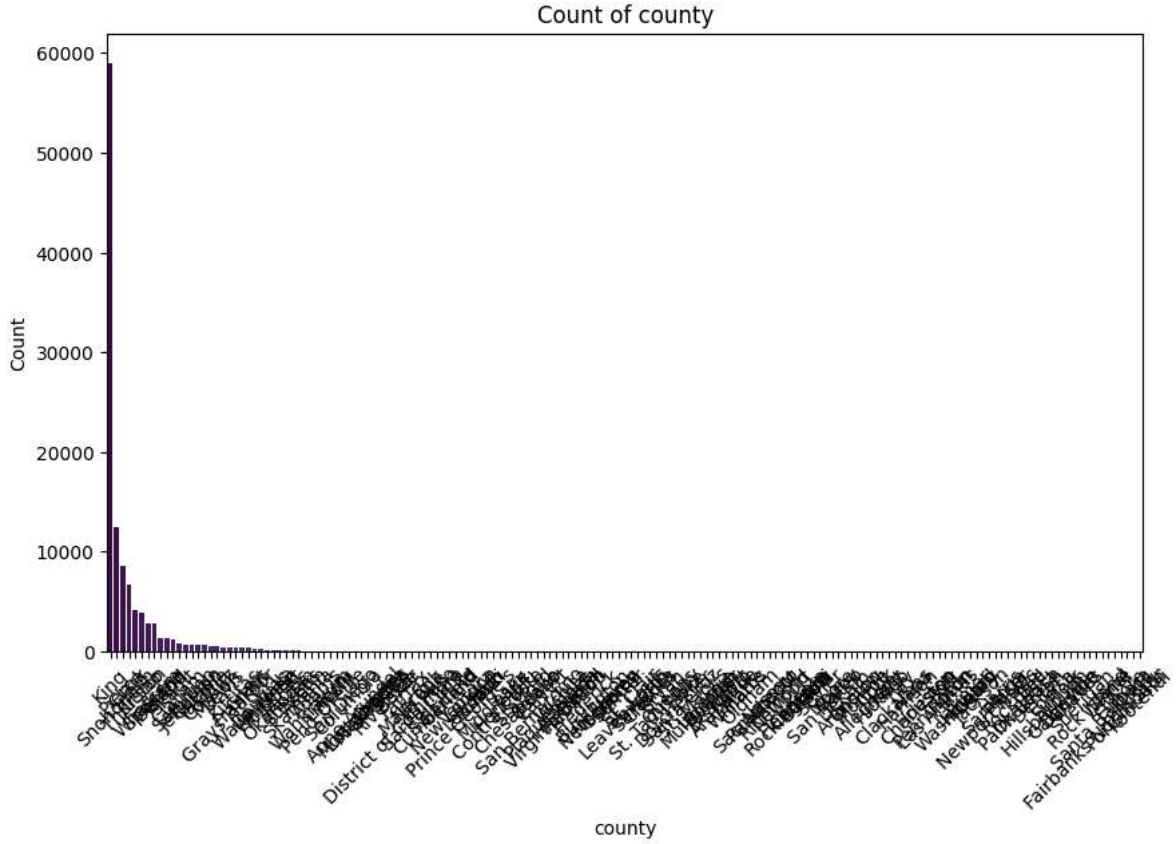

Census Tract 2020 Distribution
1e6
Count
0.4
0.6
0.8
1.0
1.2
1.4


Census Tract 2020 Boxplot

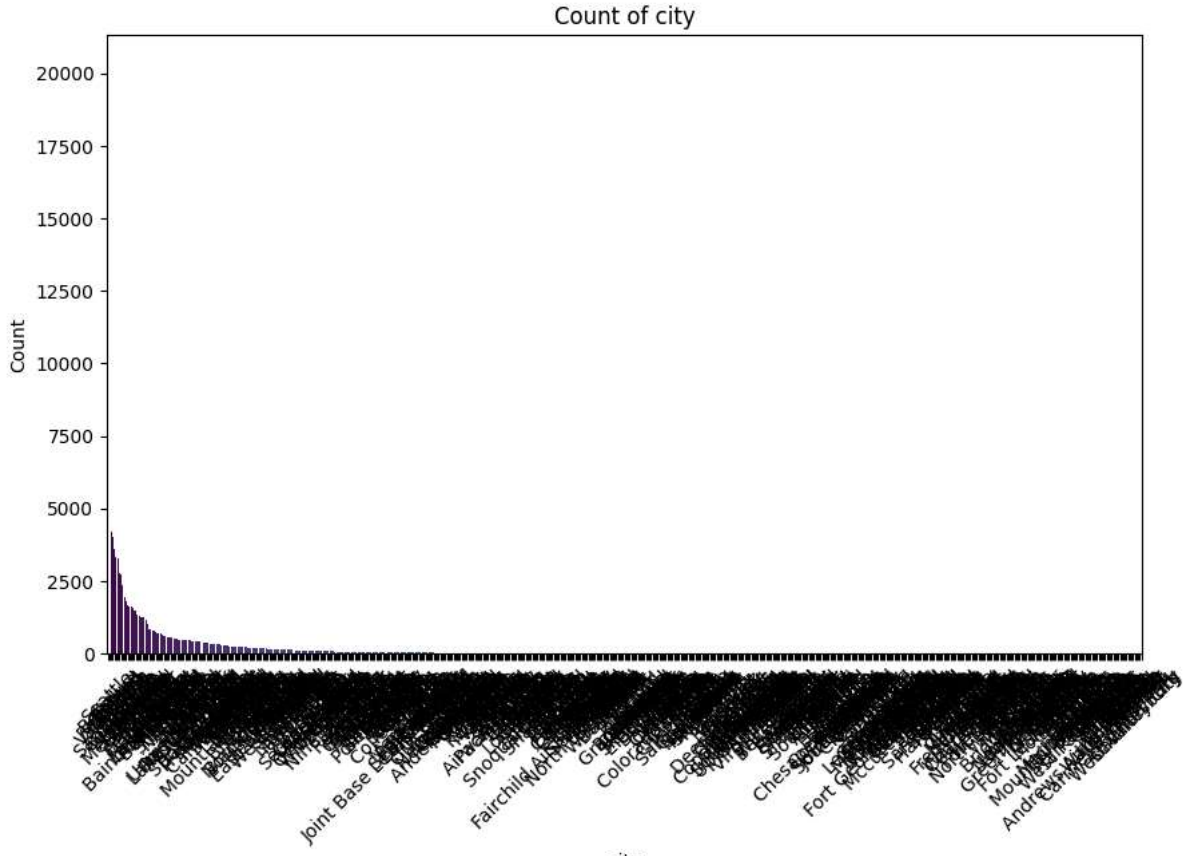



```
categorical_columns = ['county', 'city', 'state', 'make', 'model',
                       'electric_vehicle_type', 'clean_alternative_fuel_vehicle_cafv_eligibility',
                       'vehicle_location', 'electric_utility']

def plot_categorical_counts(column_name):
    plt.figure(figsize=(10, 6))
    sns.countplot(data=df, x=column_name, order=df[column_name].value_counts().index, palette='viridis')
    plt.title(f'Count of {column_name}')
    plt.xticks(rotation=45)
    plt.ylabel('Count')
    plt.xlabel(column_name)
    plt.show()

# Plot counts for each categorical column
for column in categorical_columns:
    plot_categorical_counts(column)
```

```
<ipython-input-7-1ec858ffa379>:7: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `leg

  sns.countplot(data=df, x=column_name, order=df[column_name].value_counts().index, palette='viridis')
```

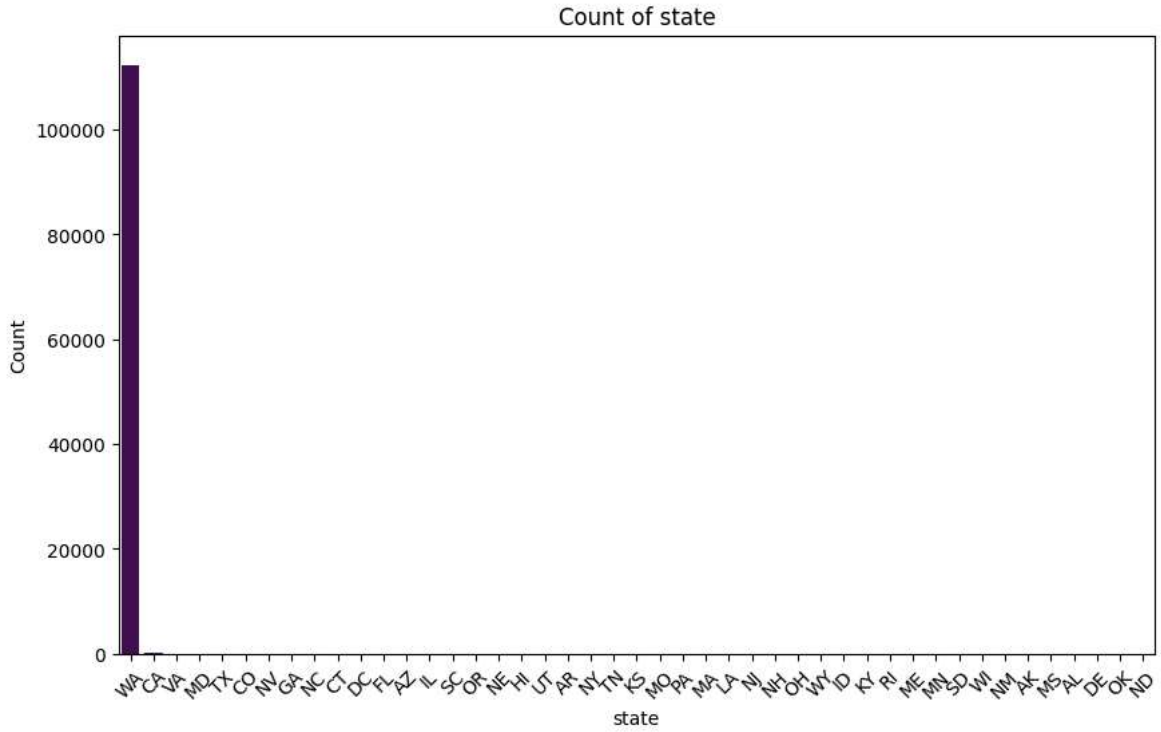



```
<ipython-input-7-1ec858ffa379>:7: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `leg

  sns.countplot(data=df, x=column_name, order=df[column_name].value_counts().index, palette='viridis')
```

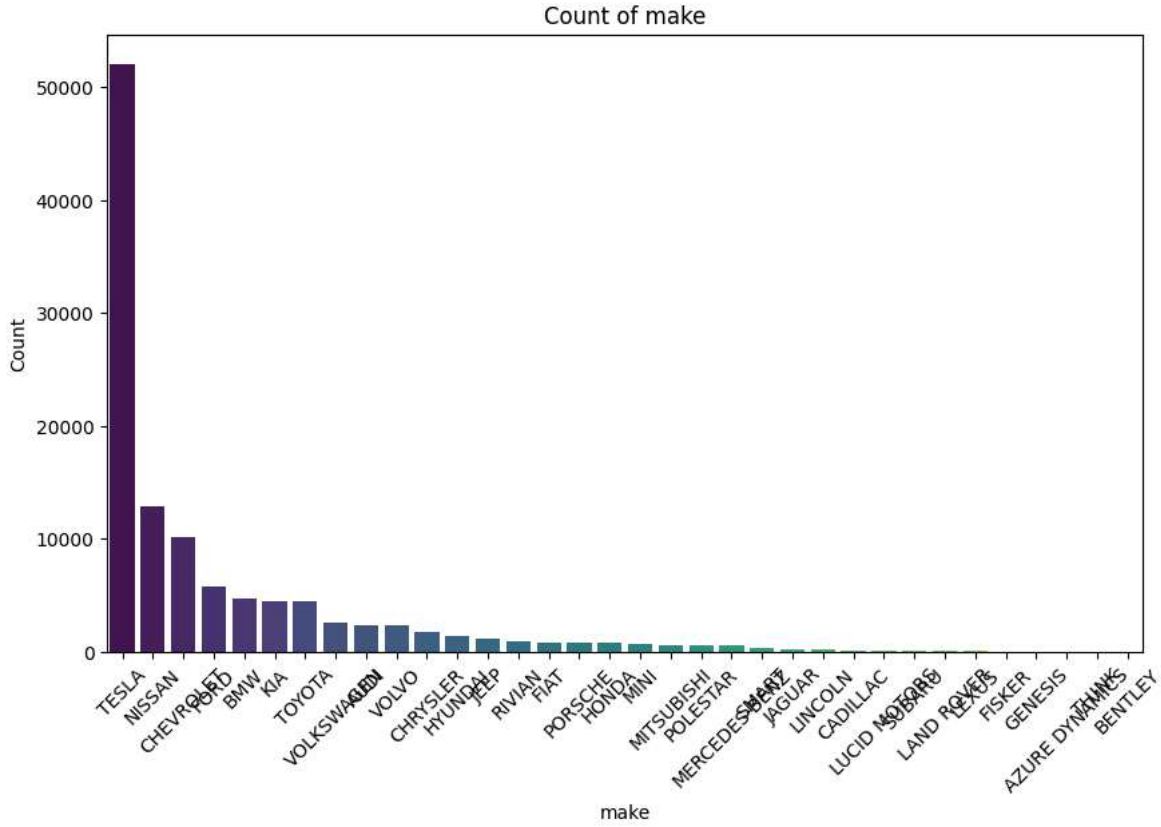


city

```
<ipython-input-7-1ec858ffa379>:7: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `leg

  sns.countplot(data=df, x=column_name, order=df[column_name].value_counts().index, palette='viridis')
```




```
<ipython-input-7-1ec858ffa379>:7: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `leg

  sns.countplot(data=df, x=column_name, order=df[column_name].value_counts().index, palette='viridis')
```




```
<ipython-input-7-1ec858ffa379>:7: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and set `leg

  sns.countplot(data=df, x=column_name, order=df[column_name].value_counts().index, palette='viridis')
```