In [1]:
```
import pandas as pd
```

In [2]:
```
df= pd.read_csv('EV.csv')
df.head()
```

Out[2]:

	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	D Vehicle
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	1989682
1	1G1RD6E45D	Clark	Laughlin	NV	89029	2013	CHEVROLET	VOLT	Plug-in Hybrid Electric Vehicle (PHEV)	Clean Alternative Fuel Vehicle Eligible	38	0	NaN	52044
2	JN1AZ0CP8B	Yakima	Yakima	WA	98901	2011	NISSAN	LEAF	Battery Electric Vehicle (BEV)	Clean Alternative Fuel Vehicle Eligible	73	0	15.0	2189725
3	1G1FW6S08H	Skagit	Concrete	WA	98237	2017	CHEVROLET	BOLT EV	Battery Electric Vehicle (BEV)	Clean Alternative Fuel Vehicle Eligible	238	0	39.0	1867504
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	20067

In [3]:
```
df
```

Out[3]:

	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
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
1	1G1RD6E45D	Clark	Laughlin	NV	89029	2013	CHEVROLET	VOLT	Plug-in Hybrid Electric Vehicle (PHEV)	Clean Alternative Fuel Vehicle Eligible	38	0	NaN
2	JN1AZ0CP8B	Yakima	Yakima	WA	98901	2011	NISSAN	LEAF	Battery Electric Vehicle (BEV)	Clean Alternative Fuel Vehicle Eligible	73	0	15.0
3	1G1FW6S08H	Skagit	Concrete	WA	98237	2017	CHEVROLET	BOLT EV	Battery Electric Vehicle (BEV)	Clean Alternative Fuel Vehicle Eligible	238	0	39.0
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
...	...	...	...	...	...	...	...	...	...	...	...	...	...
112629	7SAYGDEF2N	King	Duvall	WA	98019	2022	TESLA	MODEL Y	Battery Electric Vehicle (BEV)	Eligibility unknown as battery range has not b...	0	0	45.0
112630	1N4BZ1CP7K	San Juan	Friday Harbor	WA	98250	2019	NISSAN	LEAF	Battery Electric Vehicle (BEV)	Clean Alternative Fuel Vehicle Eligible	150	0	40.0
112631	1FMCU0KZ4N	King	Vashon	WA	98070	2022	FORD	ESCAPE	Plug-in Hybrid Electric Vehicle (PHEV)	Clean Alternative Fuel Vehicle Eligible	38	0	34.0
112632	KNDCD3LD4J	King	Covington	WA	98042	2018	KIA	NIRO	Plug-in Hybrid Electric Vehicle (PHEV)	Not eligible due to low battery range	26	0	47.0
112633	YV4BR0CL8N	King	Covington	WA	98042	2022	VOLVO	XC90	Plug-in Hybrid Electric Vehicle (PHEV)	Not eligible due to low battery range	18	0	47.0

112634 rows × 17 columns

In [ ]:

In [4]:
```
df.shape
```

Out[4]:
```
(112634, 17)
```

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

Out[5]:
```
Index(['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', '2020 Census Tract'],
      dtype='object')
```

In [6]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 112634 entries, 0 to 112633
Data columns (total 17 columns):
 #   Column                                             Non-Null Count   Dtype
---  ------                                             --------------   -----
 0   VIN (1-10)                                         112634 non-null  object
 1   County                                             112634 non-null  object
 2   City                                               112634 non-null  object
 3   State                                              112634 non-null  object
 4   Postal Code                                        112634 non-null  int64
 5   Model Year                                         112634 non-null  int64
 6   Make                                               112634 non-null  object
 7   Model                                              112614 non-null  object
 8   Electric Vehicle Type                              112634 non-null  object
 9   Clean Alternative Fuel Vehicle (CAFV) Eligibility  112634 non-null  object
 10  Electric Range                                     112634 non-null  int64
 11  Base MSRP                                          112634 non-null  int64
 12  Legislative District                               112348 non-null  float64
 13  DOL Vehicle ID                                     112634 non-null  int64
 14  Vehicle Location                                   112610 non-null  object
 15  Electric Utility                                   112191 non-null  object
 16  2020 Census Tract                                  112634 non-null  int64
dtypes: float64(1), int64(6), object(10)
memory usage: 14.6+ MB
```

In [7]:
```
df.columns = df.columns.str.replace(' ', '_')
df.columns
```

Out[7]:
```
Index(['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', '2020_Census_Tract'],
      dtype='object')
```

In [8]:
```
df.rename(columns={'Clean_Alternative_Fuel_Vehicle_(CAFV)_Eligibility':'CAFV_Eligibility'}, inplace=True)
df.columns
```

Out[8]:
```
Index(['VIN_(1-10)', 'County', 'City', 'State', 'Postal_Code', 'Model_Year',
       'Make', 'Model', 'Electric_Vehicle_Type', 'CAFV_Eligibility',
       'Electric_Range', 'Base_MSRP', 'Legislative_District', 'DOL_Vehicle_ID',
       'Vehicle_Location', 'Electric_Utility', '2020_Census_Tract'],
      dtype='object')
```

In [9]:
```
df.isnull().sum()
```

Out[9]:
```
VIN_(1-10)                0
County                    0
City                      0
State                     0
Postal_Code               0
Model_Year                0
Make                      0
Model                    20
Electric_Vehicle_Type     0
CAFV_Eligibility          0
Electric_Range            0
Base_MSRP                 0
Legislative_District    286
DOL_Vehicle_ID            0
Vehicle_Location         24
Electric_Utility        443
2020_Census_Tract         0
dtype: int64
```

In [10]:
```
df_dropna = df.dropna()

df_dropna.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 112152 entries, 2 to 112633
Data columns (total 17 columns):
 #   Column                 Non-Null Count   Dtype
---  ------                 --------------   -----
 0   VIN_(1-10)             112152 non-null  object
 1   County                 112152 non-null  object
 2   City                   112152 non-null  object
 3   State                  112152 non-null  object
 4   Postal_Code            112152 non-null  int64
 5   Model_Year             112152 non-null  int64
 6   Make                   112152 non-null  object
 7   Model                  112152 non-null  object
 8   Electric_Vehicle_Type  112152 non-null  object
 9   CAFV_Eligibility       112152 non-null  object
 10  Electric_Range         112152 non-null  int64
 11  Base_MSRP              112152 non-null  int64
 12  Legislative_District   112152 non-null  float64
 13  DOL_Vehicle_ID         112152 non-null  int64
 14  Vehicle_Location       112152 non-null  object
 15  Electric_Utility       112152 non-null  object
 16  2020_Census_Tract      112152 non-null  int64
dtypes: float64(1), int64(6), object(10)
memory usage: 15.4+ MB
```

Univariate Analysis

In [11]:
```
discrete_df = df.select_dtypes(include=['object'])

numerical_df = df.select_dtypes(include=['int64', 'float64'])
```

In [12]:
```
def discrete_univariate_analysis(discrete_data):
    for col_name in discrete_data:
        print("*"*10, col_name, "*"*10)
        print(discrete_data[col_name].agg(['count', 'nunique', 'unique']))
        print('Value Counts: \n', discrete_data[col_name].value_counts())
        print()
```

In [13]:
```
discrete_univariate_analysis(discrete_df)
```

```
********** VIN_(1-10) **********
count                                                112634
nunique                                                7548
unique     [JTMEB3FV6N, 1G1RD6E45D, JN1AZ0CP8B, 1G1FW6S08...
Name: VIN_(1-10), dtype: object
Value Counts:
 5YJYGDEE9M    472
5YJYGDEE0M    465
5YJYGDEE8M    448
5YJYGDEE7M    448
5YJYGDEE2M    437
              ...
WA1LAAGE9M      1
5UXKT0C50H      1
5YJYGAED3M      1
WDC0G5DBXL      1
YV4ED3GM0P      1
Name: VIN_(1-10), Length: 7548, dtype: int64

********** County **********
count                                                112634
nunique                                                 165
unique     [Monroe, Clark, Yakima, Skagit, Snohomish, Isl...
Name: County, dtype: object
Value Counts:
 King          59000
Snohomish     12434
Pierce         8535
Clark          6689
Thurston       4126
              ...
Pinal             1
Elmore            1
Portsmouth        1
Kings             1
Kootenai          1
Name: County, Length: 165, dtype: int64

********** City **********
count                                                112634
nunique                                                 629
unique     [Key West, Laughlin, Yakima, Concrete, Everett...
Name: City, dtype: object
Value Counts:
```

```
 Seattle         20305
Bellevue         5921
Redmond          4201
Vancouver        4013
Kirkland         3598
                ...
Hartline            1
Gaithersburg        1
El Paso             1
Klickitat           1
Worley              1
Name: City, Length: 629, dtype: int64

********** State **********
count                                                  112634
nunique                                                    45
unique     [FL, NV, WA, IL, NY, VA, OK, KS, CA, NE, MD, C...
Name: State, dtype: object
Value Counts:
 WA    112348
CA        76
VA        36
MD        26
TX        14
CO         9
NV         8
GA         7
NC         7
CT         6
DC         6
FL         6
AZ         6
IL         6
SC         5
OR         5
NE         5
HI         4
UT         4
AR         4
NY         4
TN         3
KS         3
MO         3
PA         3
MA         3
LA         3
NJ         3
NH         2
OH         2
WY         2
ID         2
KY         1
RI         1
ME         1
MN         1
SD         1
WI         1
NM         1
AK         1
MS         1
AL         1
DE         1
OK         1
ND         1
Name: State, dtype: int64

********** Make **********
count                                                  112634
nunique                                                    34
unique     [TOYOTA, CHEVROLET, NISSAN, FORD, TESLA, KIA, ...
Name: Make, dtype: object
Value Counts:
 TESLA             52078
NISSAN             12880
CHEVROLET          10182
FORD                5819
BMW                 4680
KIA                 4483
TOYOTA              4405
VOLKSWAGEN          2514
AUDI                2332
VOLVO               2288
CHRYSLER            1794
HYUNDAI             1412
JEEP                1152
RIVIAN               885
FIAT                 822
PORSCHE              818
HONDA                792
```

```
MINI                632
MITSUBISHI          588
POLESTAR            558
MERCEDES-BENZ       506
SMART               273
JAGUAR              219
LINCOLN             168
CADILLAC            108
LUCID MOTORS         65
SUBARU               59
LAND ROVER           38
LEXUS                33
FISKER               20
GENESIS              18
AZURE DYNAMICS        7
TH!NK                 3
BENTLEY               3
Name: Make, dtype: int64

********** Model **********
count                                                112614
nunique                                                 114
unique     [RAV4 PRIME, VOLT, LEAF, BOLT EV, FUSION, MODE...
Name: Model, dtype: object
Value Counts:
 MODEL 3        23135
MODEL Y        17142
LEAF           12880
MODEL S         7377
BOLT EV         4910
               ...
745LE              2
S-10 PICKUP        1
SOLTERRA           1
918                1
FLYING SPUR        1
Name: Model, Length: 114, dtype: int64

********** Electric_Vehicle_Type **********
count                                                112634
nunique                                                   2
unique     [Plug-in Hybrid Electric Vehicle (PHEV), Batte...
Name: Electric_Vehicle_Type, dtype: object
Value Counts:
 Battery Electric Vehicle (BEV)            86044
Plug-in Hybrid Electric Vehicle (PHEV)    26590
Name: Electric_Vehicle_Type, dtype: int64

********** CAFV_Eligibility **********
count                                                112634
nunique                                                   3
unique     [Clean Alternative Fuel Vehicle Eligible, Not ...
Name: CAFV_Eligibility, dtype: object
Value Counts:
 Clean Alternative Fuel Vehicle Eligible                         58639
Eligibility unknown as battery range has not been researched    39236
Not eligible due to low battery range                           14759
Name: CAFV_Eligibility, dtype: int64

********** Vehicle_Location **********
count                                                112610
nunique                                                 758
unique     [POINT (-81.80023 24.5545), POINT (-114.57245 ...
Name: Vehicle_Location, dtype: object
Value Counts:
 POINT (-122.13158 47.67858)    2916
POINT (-122.2066 47.67887)     2059
POINT (-122.1872 47.61001)     2001
POINT (-122.31765 47.70013)    1880
POINT (-122.12096 47.55584)    1852
                               ...
POINT (-124.33152 48.05431)       1
POINT (-77.41203 39.41574)        1
POINT (-123.61022 46.35588)       1
POINT (-112.04165 40.68741)       1
POINT (-116.91895 47.40077)       1
Name: Vehicle_Location, Length: 758, dtype: int64

********** Electric_Utility **********
count                                                112191
nunique                                                  73
unique     [nan, PACIFICORP, PUGET SOUND ENERGY INC, PUD ...
Name: Electric_Utility, dtype: object
Value Counts:
 PUGET SOUND ENERGY INC||CITY OF TACOMA - (WA)                                        40247
PUGET SOUND ENERGY INC                                                               22172
CITY OF SEATTLE - (WA)|CITY OF TACOMA - (WA)                                         21447
BONNEVILLE POWER ADMINISTRATION||PUD NO 1 OF CLARK COUNTY - (WA)                      6522
BONNEVILLE POWER ADMINISTRATION||CITY OF TACOMA - (WA)||PENINSULA LIGHT COMPANY       5053
```

```
                                                                                ...
BONNEVILLE POWER ADMINISTRATION||PENINSULA LIGHT COMPANY                                  1
BONNEVILLE POWER ADMINISTRATION||PUD NO 1 OF ASOTIN COUNTY                                1
CITY OF SEATTLE - (WA)                                                                    1
BONNEVILLE POWER ADMINISTRATION||NESPELEM VALLEY ELEC COOP, INC                           1
BONNEVILLE POWER ADMINISTRATION||PUD NO 1 OF CLALLAM COUNTY|PUD NO 1 OF JEFFERSON COUNTY  1
Name: Electric_Utility, Length: 73, dtype: int64
```

In [14]:
```python
def numerical_univariate_analysis(numerical_data):
    for col_name in numerical_data:
        print("*"*10, col_name, "*"*10)
        print(numerical_data[col_name].agg(['min', 'max', 'mean', 'median', 'std']))
        print()
```

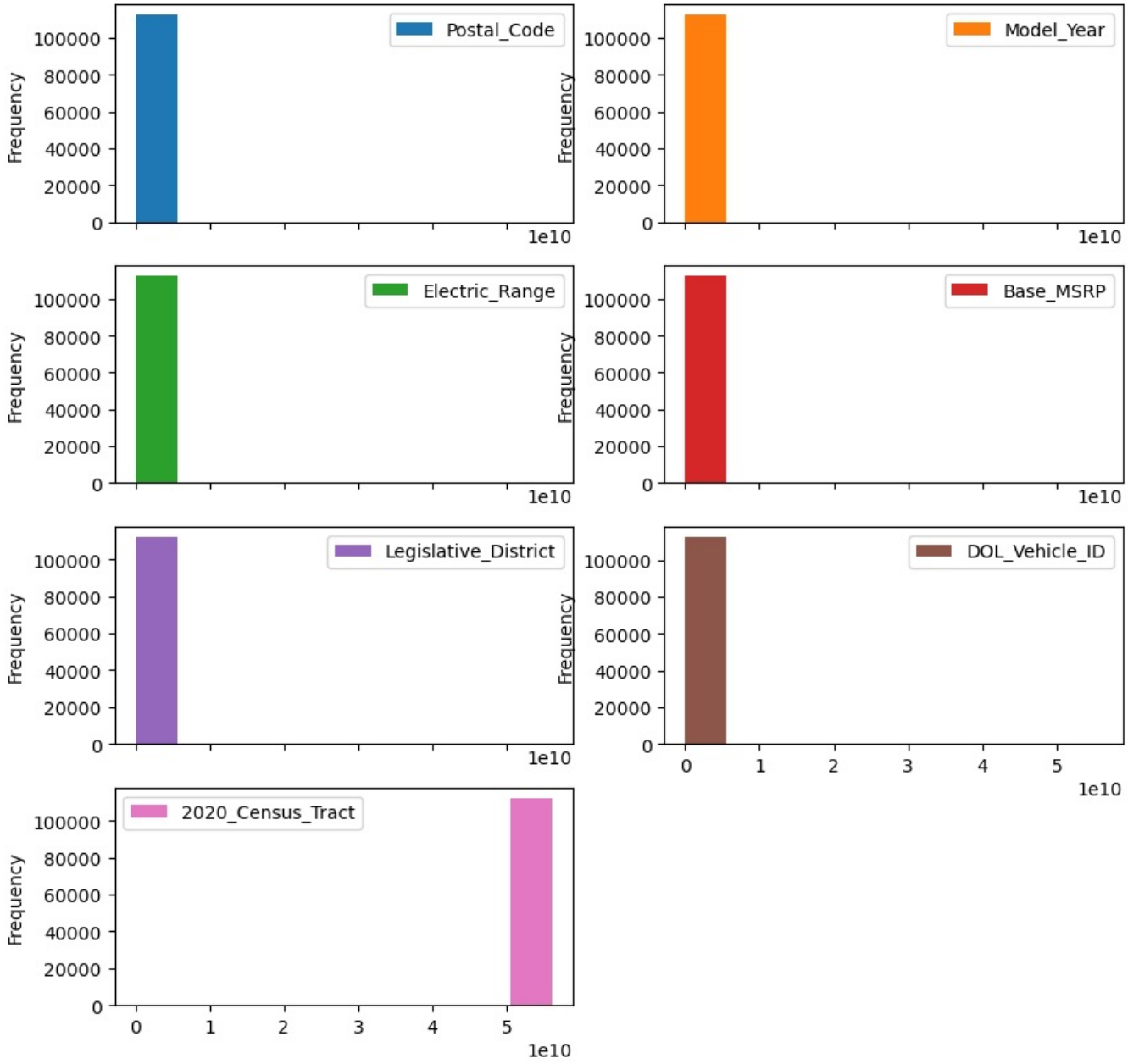
In [15]:
```python
numerical_univariate_analysis(numerical_df)
```

```
********** Postal_Code **********
min        1730.000000
max       99701.000000
mean      98156.226850
median    98119.000000
std        2648.733064
Name: Postal_Code, dtype: float64

********** Model_Year **********
min       1997.000000
max       2023.000000
mean      2019.003365
median    2020.000000
std          2.892364
Name: Model_Year, dtype: float64

********** Electric_Range **********
min         0.000000
max       337.000000
mean       87.812987
median     32.000000
std       102.334216
Name: Electric_Range, dtype: float64

********** Base_MSRP **********
min            0.000000
max       845000.000000
mean        1793.439681
median         0.000000
std        10783.753486
Name: Base_MSRP, dtype: float64

********** Legislative_District **********
min        1.000000
max       49.000000
mean      29.805604
median    34.000000
std       14.700545
Name: Legislative_District, dtype: float64

********** DOL_Vehicle_ID **********
min       4.777000e+03
max       4.792548e+08
mean      1.994567e+08
median    1.923896e+08
std       9.398427e+07
Name: DOL_Vehicle_ID, dtype: float64

********** 2020_Census_Tract **********
min       1.101001e+09
max       5.603300e+10
mean      5.296650e+10
median    5.303303e+10
std       1.699104e+09
Name: 2020_Census_Tract, dtype: float64
```

Univariate - Visual Analysis

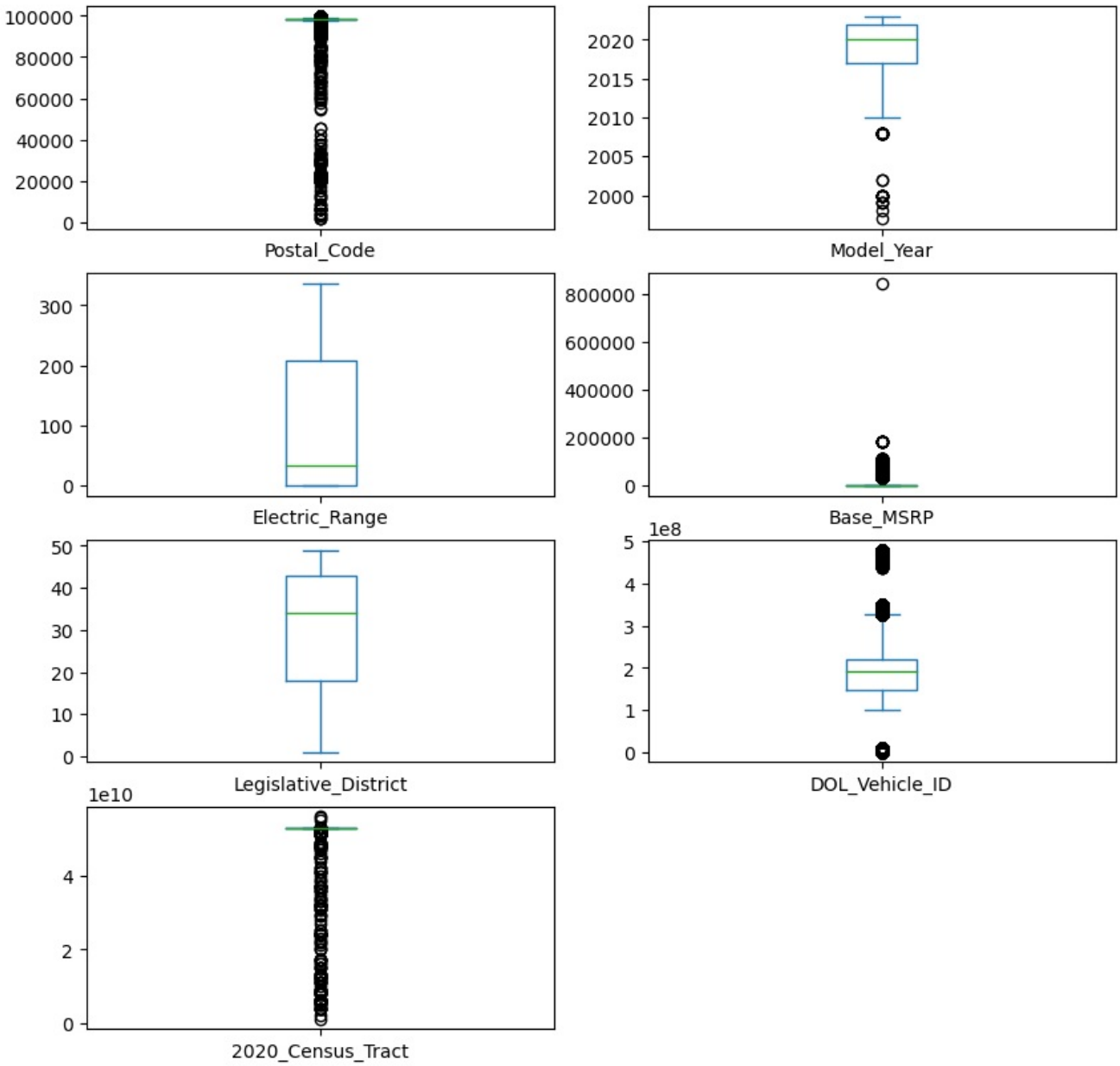
In [16]:
```python
df.plot(kind='hist', subplots=True, layout=(4, 2), figsize=(10, 10))
```

Out[16]:
```
array([[<AxesSubplot:ylabel='Frequency'>,
        <AxesSubplot:ylabel='Frequency'>],
       [<AxesSubplot:ylabel='Frequency'>,
        <AxesSubplot:ylabel='Frequency'>],
       [<AxesSubplot:ylabel='Frequency'>,
        <AxesSubplot:ylabel='Frequency'>],
       [<AxesSubplot:ylabel='Frequency'>,
        <AxesSubplot:ylabel='Frequency'>]], dtype=object)
```



In [17]:
```
df.plot(kind='box', subplots=True, layout=(4, 2), figsize=(10, 10))
```

Out[17]:
```
Postal_Code               AxesSubplot(0.125,0.712609;0.352273x0.167391)
Model_Year              AxesSubplot(0.547727,0.712609;0.352273x0.167391)
Electric_Range            AxesSubplot(0.125,0.511739;0.352273x0.167391)
Base_MSRP               AxesSubplot(0.547727,0.511739;0.352273x0.167391)
Legislative_District        AxesSubplot(0.125,0.31087;0.352273x0.167391)
DOL_Vehicle_ID           AxesSubplot(0.547727,0.31087;0.352273x0.167391)
2020_Census_Tract             AxesSubplot(0.125,0.11;0.352273x0.167391)
dtype: object
```



Bivariate Analysis a. Continuous vs Continuous Numerical Data

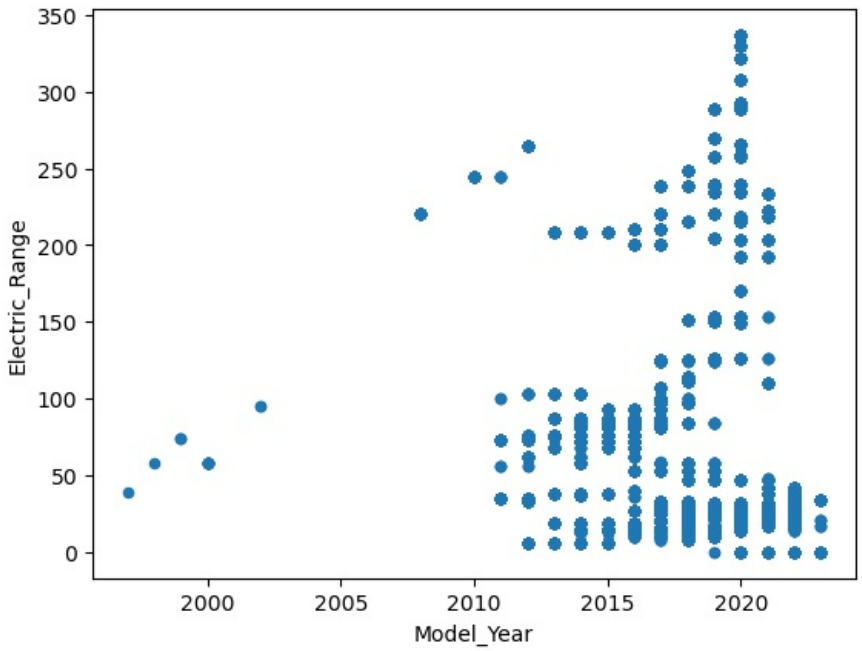
In [18]:
```
numerical_df.corr()
```

Out[18]:

	Postal_Code	Model_Year	Electric_Range	Base_MSRP	Legislative_District	DOL_Vehicle_ID	2020_Census_Tract
Postal_Code	1.000000	-0.004485	0.000385	0.001151	-0.433405	0.003365	0.501170
Model_Year	-0.004485	1.000000	-0.288433	-0.229130	0.010439	-0.068295	0.000714
Electric_Range	0.000385	-0.288433	1.000000	0.085025	0.024387	0.009682	0.000722
Base_MSRP	0.001151	-0.229130	0.085025	1.000000	0.012426	0.000504	0.000979
Legislative_District	-0.433405	0.010439	0.024387	0.012426	1.000000	-0.001671	-0.111991
DOL_Vehicle_ID	0.003365	-0.068295	0.009682	0.000504	-0.001671	1.000000	0.002754
2020_Census_Tract	0.501170	0.000714	0.000722	0.000979	-0.111991	0.002754	1.000000

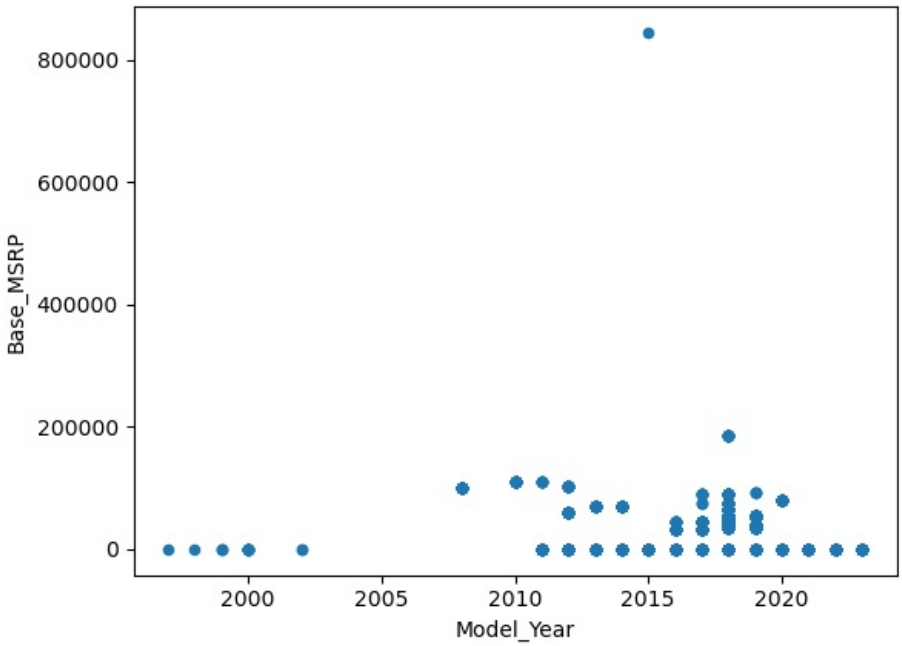
In [19]:
```
df.plot(kind='scatter', x='Model_Year',y='Electric_Range')
```

Out[19]:
```
<AxesSubplot:xlabel='Model_Year', ylabel='Electric_Range'>
```



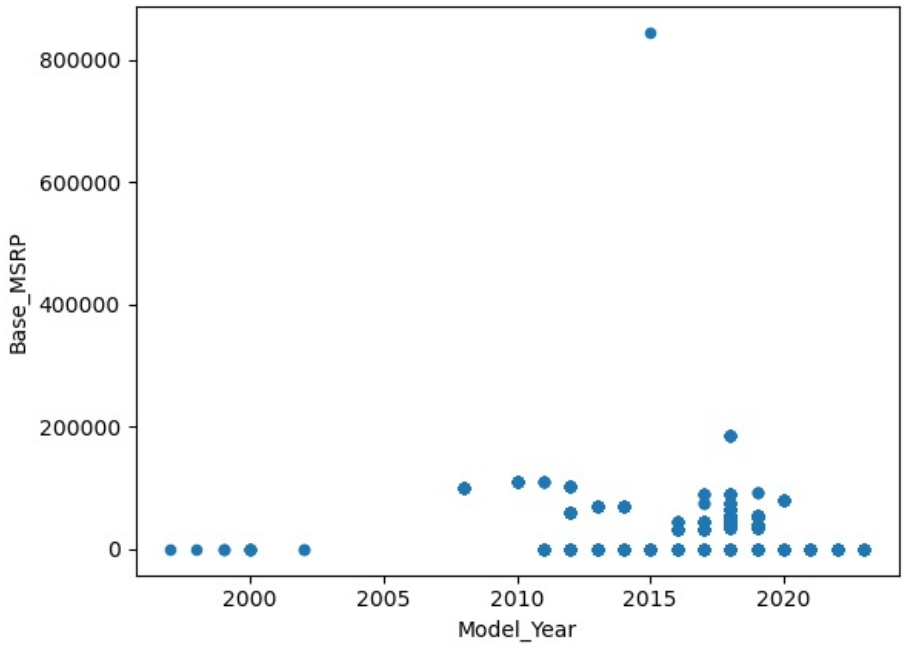
In [20]:
```
df.plot(kind='scatter', x='Model_Year',y='Base_MSRP')
```

Out[20]:
```
<AxesSubplot:xlabel='Model_Year', ylabel='Base_MSRP'>
```



In [21]:
```
f = df.loc[(df['Base_MSRP'] < 80000)]

df.plot(kind='scatter', x='Model_Year',y='Base_MSRP')
```

Out[21]:
```
<AxesSubplot:xlabel='Model_Year', ylabel='Base_MSRP'>
```



In [22]:
```
pd.crosstab(df['Electric_Utility'],df['Electric_Vehicle_Type'])
```

Out[22]:

Electric_Vehicle_Type	Battery Electric Vehicle (BEV)	Plug-in Hybrid Electric Vehicle (PHEV)
Electric_Utility		
AVISTA CORP	151	84
BONNEVILLE POWER ADMINISTRATION||AVISTA CORP||BIG BEND ELECTRIC COOP, INC	21	11
BONNEVILLE POWER ADMINISTRATION||AVISTA CORP||INLAND POWER & LIGHT COMPANY	1151	528
BONNEVILLE POWER ADMINISTRATION||AVISTA CORP||PUD NO 1 OF ASOTIN COUNTY	28	17
BONNEVILLE POWER ADMINISTRATION||BENTON RURAL ELECTRIC ASSN	0	3
...	...	...
PUD NO 1 OF WHATCOM COUNTY	22	13
PUD NO 2 OF GRANT COUNTY	227	108
PUGET SOUND ENERGY INC	16701	5471
PUGET SOUND ENERGY INC||CITY OF TACOMA - (WA)	32594	7653
PUGET SOUND ENERGY INC||PUD NO 1 OF WHATCOM COUNTY	1936	735

73 rows × 2 columns

In [23]:
```
pd.crosstab(df['Model_Year'],df['Electric_Vehicle_Type'])
```

Out[23]:

Electric_Vehicle_Type	Battery Electric Vehicle (BEV)	Plug-in Hybrid Electric Vehicle (PHEV)
Model_Year		
1997	1	0
1998	1	0
1999	3	0
2000	10	0
2002	2	0
2008	23	0
2010	24	0
2011	769	71
2012	814	891
2013	3018	1673
2014	1864	1821
2015	3625	1315
2016	3938	1797
2017	4498	4146
2018	9902	4344
2019	8440	1826
2020	9444	1594
2021	14873	3491
2022	22960	3570
2023	1835	51

In [24]:
```
pd.crosstab(df['CAFV_Eligibility'],df['Electric_Vehicle_Type'])
```

Out[24]:

Electric_Vehicle_Type	Battery Electric Vehicle (BEV)	Plug-in Hybrid Electric Vehicle (PHEV)
CAFV_Eligibility		
Clean Alternative Fuel Vehicle Eligible	46799	11840
Eligibility unknown as battery range has not been researched	39236	0
Not eligible due to low battery range	9	14750

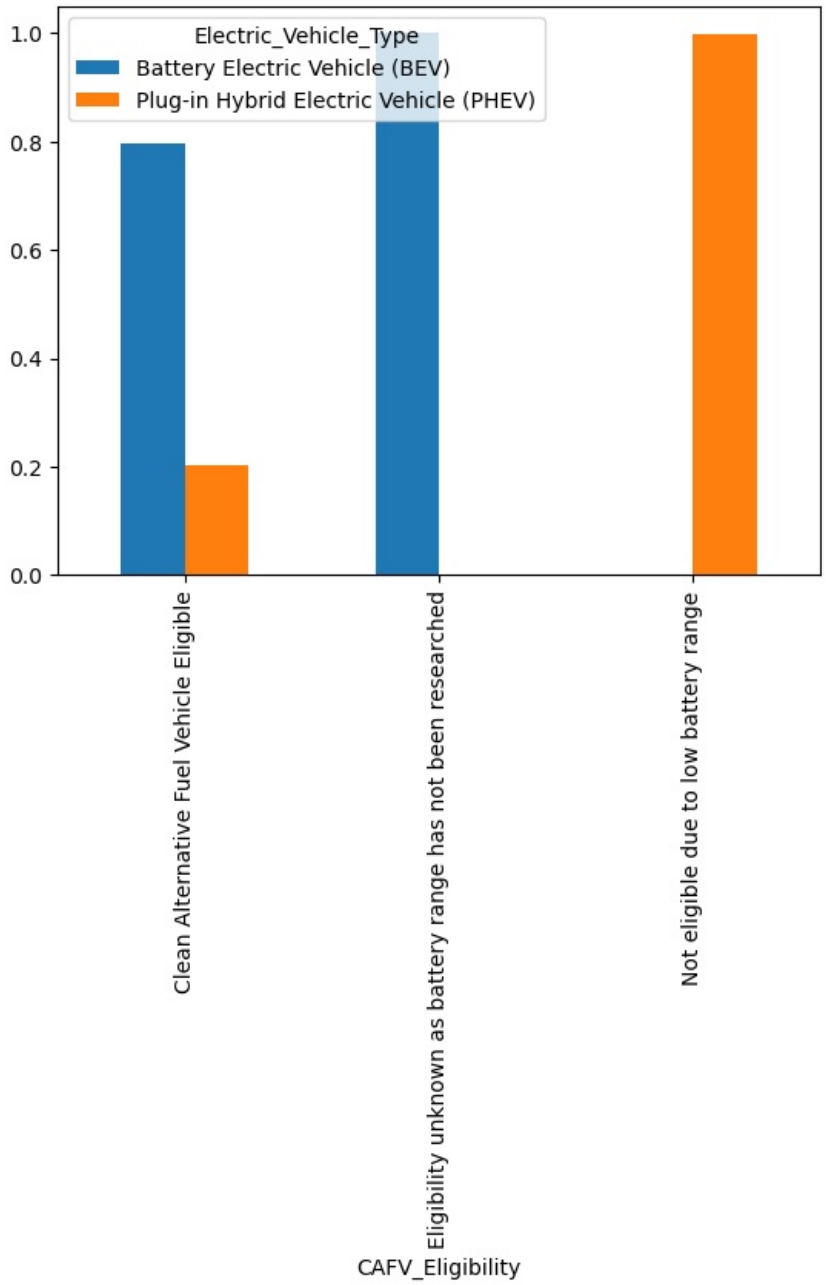
In [25]:
```
pd.crosstab(df['Model_Year'],df['CAFV_Eligibility'])
```

Out[25]:

CAFV_Eligibility	Clean Alternative Fuel Vehicle Eligible	Eligibility unknown as battery range has not been researched	Not eligible due to low battery range
Model_Year			
1997	1	0	0
1998	1	0	0
1999	3	0	0
2000	10	0	0
2002	2	0	0
2008	23	0	0
2010	24	0	0
2011	840	0	0
2012	1330	0	375
2013	3836	0	855
2014	2896	0	789
2015	4262	0	678
2016	4330	0	1405
2017	6348	0	2296
2018	11921	0	2325
2019	8844	2	1420
2020	9784	53	1201
2021	2099	14386	1879
2022	2042	22960	1528
2023	43	1835	8

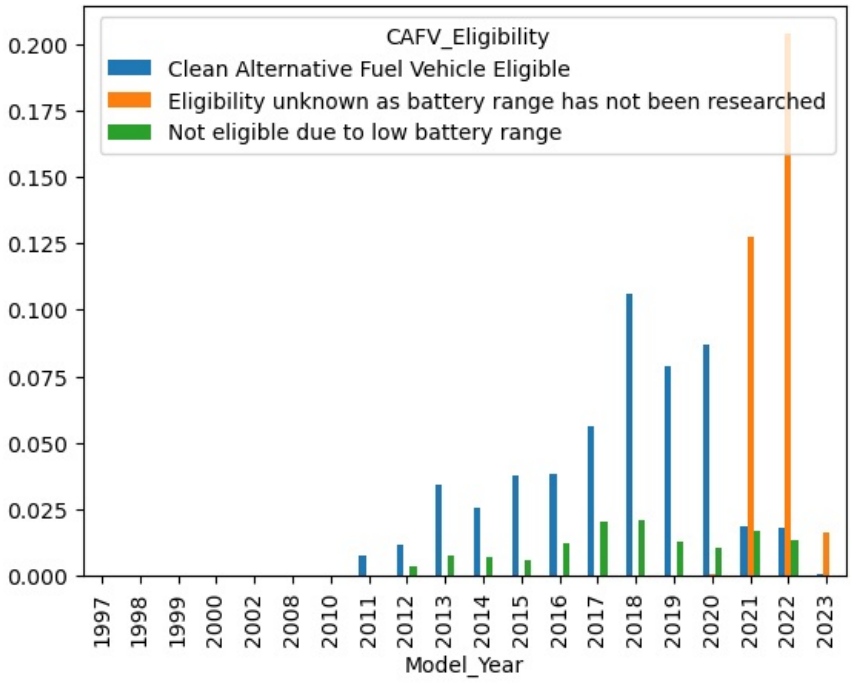
In [26]:
```
tab = pd.crosstab(df['CAFV_Eligibility'], df['Electric_Vehicle_Type'], normalize='index')

tab.plot(kind='bar')
```

Out[26]:
```
<AxesSubplot:xlabel='CAFV_Eligibility'>
```



In [27]:
```
tab = pd.crosstab(df['Model_Year'],df['CAFV_Eligibility'], normalize= True)

tab.plot(kind='bar')
```

Out[27]:
```
<AxesSubplot:xlabel='Model_Year'>
```



Continuous Numerical vs Discrete Data

In [28]:
```
group = df.groupby('Electric_Vehicle_Type')
group['Electric_Range'].agg(['min', 'max', 'mean', 'median'])
```

Out[28]:

Electric_Vehicle_Type	min	max	mean	median
Battery Electric Vehicle (BEV)	0	337	105.369671	76.0
Plug-in Hybrid Electric Vehicle (PHEV)	6	153	31.000376	25.0

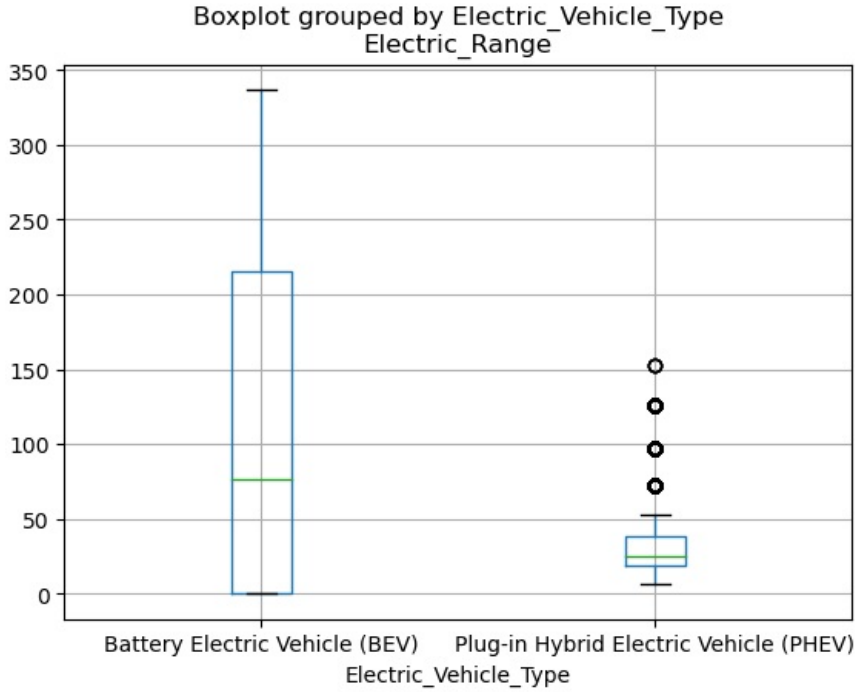
In [29]:
```
group = df.groupby('CAFV_Eligibility')
group['Electric_Range'].agg(['min', 'max', 'mean', 'median'])
```

Out[29]:

CAFV_Eligibility	min	max	mean	median
Clean Alternative Fuel Vehicle Eligible	30	337	163.797558	208.0
Eligibility unknown as battery range has not been researched	0	0	0.000000	0.0
Not eligible due to low battery range	6	29	19.364659	19.0

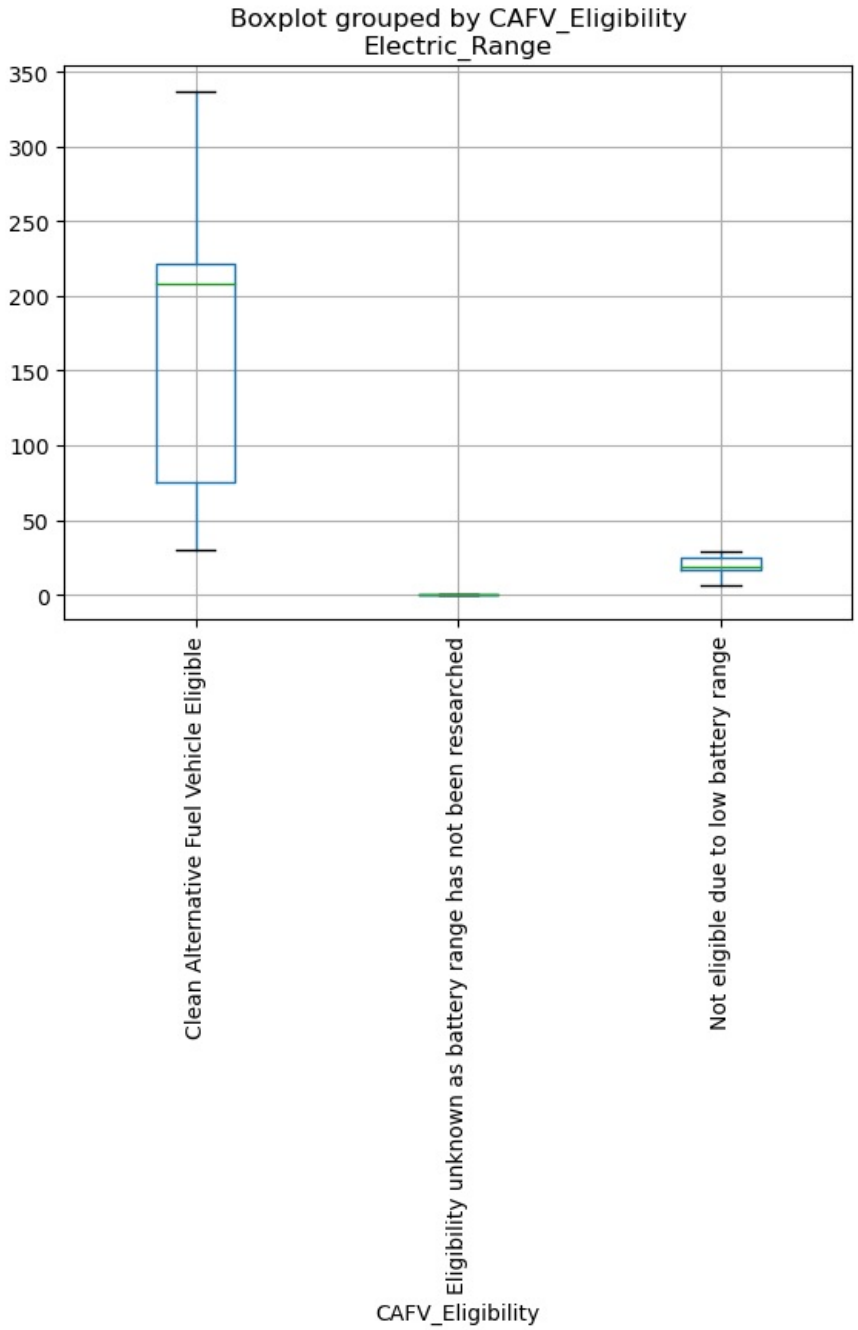
In [30]:
```
df.boxplot(by="Electric_Vehicle_Type",column=['Electric_Range'])
```

Out[30]:
```
<AxesSubplot:title={'center':'Electric_Range'}, xlabel='Electric_Vehicle_Type'>
```



In [31]:

```
import pandas as pd
import matplotlib.pyplot as plt

# Assuming 'df' is your DataFrame
df.boxplot(by="CAFV_Eligibility", column=['Electric_Range'])

# Rotate x-axis labels by 90 degrees
plt.xticks(rotation=90)

# Show the plot
plt.show()
```



Task 2: Create a Choropleth using plotly.express to display the number of EV vehicles based on location

In [32]:
```
! pip install plotly
```

```
Requirement already satisfied: plotly in c:\users\shanm\anaconda3\lib\site-packages (5.9.0)
Requirement already satisfied: tenacity>=6.2.0 in c:\users\shanm\anaconda3\lib\site-packages (from plotly) (8.4
.2)
```

In [33]:
```
import plotly.express as px
```

In [34]:
```
ev_count_by_state = df.groupby('State').size().reset_index(name='Number_of_EV_Vehicles')
ev_count_by_state
```

Out[34]:

	State	Number_of_EV_Vehicles
0	AK	1
1	AL	1
2	AR	4
3	AZ	6
4	CA	76
5	CO	9
6	CT	6
7	DC	6
8	DE	1
9	FL	6
10	GA	7
11	HI	4
12	ID	2
13	IL	6
14	KS	3
15	KY	1
16	LA	3
17	MA	3
18	MD	26
19	ME	1
20	MN	1
21	MO	3
22	MS	1
23	NC	7
24	ND	1
25	NE	5
26	NH	2
27	NJ	3
28	NM	1
29	NV	8
30	NY	4
31	OH	2
32	OK	1
33	OR	5
34	PA	3
35	RI	1
36	SC	5
37	SD	1
38	TN	3
39	TX	14
40	UT	4
41	VA	36
42	WA	112348
43	WI	1
44	WY	2

In [35]:

```
fig = px.choropleth(ev_count_by_state,
                    locations='State',
                    locationmode="USA-states",
                    color='Number_of_EV_Vehicles',
                    scope="usa",
                    color_continuous_scale="Viridis",
                    labels={'Number_of_EV_Vehicles': 'EV Vehicles'},
                    title='Number of EV Vehicles by State')

# Show the map
fig.show()
```

TASK 3

In [50]:
```
ev_make_by_year = df.groupby(['Make', 'Model Year']).size().reset_index(name='Number_of_Vehicles')

# Display the resulting DataFrame for verification
print(ev_make_by_year)
```

```
      Make  Model Year  Number_of_Vehicles
0     AUDI        2016                 214
1     AUDI        2017                 187
2     AUDI        2018                 174
3     AUDI        2019                 392
4     AUDI        2020                 224
..     ...         ...                 ...
204  VOLVO        2019                 190
205  VOLVO        2020                 162
206  VOLVO        2021                 580
207  VOLVO        2022                 882
208  VOLVO        2023                  21

[209 rows x 3 columns]
```

In [53]:
```
import plotly.express as px

# Check your DataFrame structure
print(ev_make_by_year.head())
print(ev_make_by_year.columns)

# Create the animated racing bar plot with annotations
fig = px.bar(ev_make_by_year,
             y='Make',  # Place Make on y-axis
             x='Number_of_Vehicles',  # Place the count of EV vehicles on the x-axis
             color='Make',  # Color each make differently
             animation_frame='Model Year',  # Corrected animation frame to match DataFrame
             orientation='h',  # Horizontal bar chart
             title='EV Makes and their Count Over the Years',
             labels={'Number_of_Vehicles': 'Number of EV Vehicles'},
             range_x=[0, 3000]
            )

# Update traces for displaying values
fig.update_traces(texttemplate='%{x}',  # Display the actual x-axis values (Number_of_Vehicles)
                  textposition='outside',  # Place the text outside the bars
                  textfont_size=16)  # Adjust the font size for better readability

# Adjust the layout for improved visibility and emphasis on movement
fig.update_layout(
    xaxis=dict(showgrid=True, gridcolor='LightGray'),  # Show grid for better visibility
    yaxis_title='EV Makes',
    xaxis_title='Number of EV Vehicles',
    showlegend=False,  # Hide legend as it's not necessary for this chart
```

```
    title_x=0.5,  # Center title
    title_font=dict(size=20),  # Increase title font size
    margin=dict(l=50, r=50, t=50, b=50),  # Adjust margins
    width=800,  # Set a fixed width
    height=600  # Set a fixed height
)

# Show the plot
fig.show()
```

```
   Make  Model Year  Number_of_Vehicles
0  AUDI        2016                 214
1  AUDI        2017                 187
2  AUDI        2018                 174
3  AUDI        2019                 392
4  AUDI        2020                 224
Index(['Make', 'Model Year', 'Number_of_Vehicles'], dtype='object')
```