

Import Required libraries

```
In [209...]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
EV_data = pd.read_csv("E:\Innomatics\EV\dataset.csv")
import warnings
warnings.filterwarnings("ignore")
EV_data.head()
```

Out[209...]

	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
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	198968248 (-1)
1	1G1RD6E45D	Clark	Lahrglin	NV	89029	2013	CHEVROLET	VOLT	Plug-in Hybrid Electric Vehicle (PHEV)	Clean Alternative Fuel Vehicle Eligible	38	0	NaN	5204412 (-1)
2	JN1AZ0CP8B	Yakima	Yakima	WA	98901	2011	NISSAN	LEAF	Battery Electric Vehicle (BEV)	Clean Alternative Fuel Vehicle Eligible	73	0	15.0	218972519 (-1)
3	1G1FW6S08H	Skagit	Concrete	WA	98237	2017	CHEVROLET	BOLT EV	Battery Electric Vehicle (BEV)	Clean Alternative Fuel Vehicle Eligible	238	0	39.0	186750406 (-1)
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	2006714 (-1)

◀ ▶

In [2]: EV_data.columns #All columns present in dataset

```
Out[2]: 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 [3]: EV_data.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 [4]: print(EV_data.isnull())      # Shows True where there are missing values
print(EV_data.isnull().sum())    # Total count of missing values in each column
```

```
VIN (1-10)  County  City  State  Postal Code  Model Year  Make \
0          False   False  False  False        False      False  False
1          False   False  False  False        False      False  False
2          False   False  False  False        False      False  False
3          False   False  False  False        False      False  False
4          False   False  False  False        False      False  False
...        ...     ...    ...    ...       ...     ...    ...
112629    False   False  False  False        False      False  False
112630    False   False  False  False        False      False  False
112631    False   False  False  False        False      False  False
112632    False   False  False  False        False      False  False
112633    False   False  False  False        False      False  False
```

```
Model  Electric Vehicle Type \
0    False           False
1    False           False
2    False           False
3    False           False
4    False           False
...   ...
112629 False          False
112630 False          False
112631 False          False
112632 False          False
112633 False          False
```

```
Clean Alternative Fuel Vehicle (CAFV) Eligibility  Electric Range \
0          False      False
1          False      False
2          False      False
3          False      False
4          False      False
...        ...
112629    False      False
112630    False      False
112631    False      False
112632    False      False
112633    False      False
```

```
Base MSRP  Legislative District  DOL Vehicle ID  Vehicle Location \
0          False      True    False      False
1          False      True    False      False
2          False     False   False      False
3          False     False   False      False
4          False     False   False      False
...        ...
112629    False     False   False      False
112630    False     False   False      False
112631    False     False   False      False
112632    False     False   False      False
112633    False     False   False      False
```

```
Electric Utility  2020 Census Tract
0            True    False
1            True    False
2            False   False
3            False   False
4            False   False
...        ...
112629    False   False
112630    False   False
112631    False   False
112632    False   False
112633    False   False
```

[112634 rows x 17 columns]

	0
VIN (1-10)	0
County	0
City	0
State	0
Postal Code	0
Model Year	0
Make	0
Model	20
Electric Vehicle Type	0
Clean Alternative Fuel Vehicle (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 [5]: EV_data=EV_data.dropna() # Drop rows with a missing values
EV_data
```

Out[5]:

	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	Vehicle
2	JN1AZ0CP8B	Yakima	Yakima	WA	98901	2011	NISSAN	LEAF	Battery Electric Vehicle (BEV)	Clean Alternative Fuel Vehicle Eligible	73	0	15.0	218972
3	1G1FW6S08H	Skagit	Concrete	WA	98237	2017	CHEVROLET	BOLT EV	Battery Electric Vehicle (BEV)	Clean Alternative Fuel Vehicle Eligible	238	0	39.0	186750
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	2006
5	5YJ3E1EB5J	Snohomish	Bothell	WA	98021	2018	TESLA	MODEL 3	Battery Electric Vehicle (BEV)	Clean Alternative Fuel Vehicle Eligible	215	0	1.0	475635
6	1N4AZ0CP4D	Snohomish	Everett	WA	98203	2013	NISSAN	LEAF	Battery Electric Vehicle (BEV)	Clean Alternative Fuel Vehicle Eligible	75	0	38.0	253546
...
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	217955
112630	1N4BZ1CP7K	San Juan	Friday Harbor	WA	98250	2019	NISSAN	LEAF	Battery Electric Vehicle (BEV)	Clean Alternative Fuel Vehicle Eligible	150	0	40.0	103663
112631	1FMCU0KZ4N	King	Vashon	WA	98070	2022	FORD	ESCAPE	Plug-in Hybrid Electric Vehicle (PHEV)	Clean Alternative Fuel Vehicle Eligible	38	0	34.0	193878
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	125039
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	194673

112152 rows × 17 columns

```
In [6]: EV_data.isnull().sum()
```

```
Out[6]: VIN (1-10)          0
County                   0
City                     0
State                     0
Postal Code               0
Model Year                0
Make                     0
Model                     0
Electric Vehicle Type     0
Clean Alternative Fuel Vehicle (CAFV) Eligibility 0
Electric Range             0
Base MSRP                  0
Legislative District        0
DOL Vehicle ID             0
Vehicle Location            0
Electric Utility            0
2020 Census Tract           0
dtype: int64
```

Change Data type

```
In [7]: EV_data['Postal Code'] = EV_data['Postal Code'].astype(int)
EV_data['Model Year'] = pd.to_datetime(EV_data['Model Year'], format='%Y')
EV_data['Electric Range'] = EV_data['Electric Range'].astype(int)
EV_data['Base MSRP'] = EV_data['Base MSRP'].astype(int)
EV_data['Legislative District'] = EV_data['Legislative District'].astype(int)
EV_data['DOL Vehicle ID'] = EV_data['DOL Vehicle ID'].astype(int)
#EV_data['Vehicle Location'] = EV_data['Vehicle Location'].astype(float)
EV_data['2020 Census Tract'] = EV_data['2020 Census Tract'].astype(int)
EV_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 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   int32  
 5   Model Year       112152 non-null   datetime64[ns]
 6   Make              112152 non-null   object 
 7   Model             112152 non-null   object 
 8   Electric Vehicle Type 112152 non-null   object 
 9   Clean Alternative Fuel Vehicle (CAFV) Eligibility 112152 non-null   object 
 10  Electric Range    112152 non-null   int32  
 11  Base MSRP         112152 non-null   int32  
 12  Legislative District 112152 non-null   int32  
 13  DOL Vehicle ID    112152 non-null   int32  
 14  Vehicle Location   112152 non-null   object 
 15  Electric Utility   112152 non-null   object 
 16  2020 Census Tract 112152 non-null   int32 

dtypes: datetime64[ns](1), int32(6), object(10)
memory usage: 12.8+ MB
```

```
In [8]: EV_data[['Longitude', 'Latitude']] = EV_data['Vehicle Location'].str.extract(r'POINT \((([-\d\.\.]+) ([-\d\.\.]+)\))' ) # Extract coordinate
EV_data['Longitude'] = EV_data['Longitude'].astype(float)
EV_data['Latitude'] = EV_data['Latitude'].astype(float)
```

```
In [9]: EV_data.info() # Cross checked
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 112152 entries, 2 to 112633
Data columns (total 19 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   int32  
 5   Model Year     112152 non-null   datetime64[ns] 
 6   Make            112152 non-null   object  
 7   Model           112152 non-null   object  
 8   Electric Vehicle Type 112152 non-null   object  
 9   Clean Alternative Fuel Vehicle (CAFV) Eligibility 112152 non-null   object  
 10  Electric Range 112152 non-null   int32  
 11  Base MSRP       112152 non-null   int32  
 12  Legislative District 112152 non-null   int32  
 13  DOL Vehicle ID 112152 non-null   int32  
 14  Vehicle Location 112152 non-null   object  
 15  Electric Utility 112152 non-null   object  
 16  2020 Census Tract 112152 non-null   int32  
 17  Longitude        112152 non-null   float64 
 18  Latitude         112152 non-null   float64 

dtypes: datetime64[ns](1), float64(2), int32(6), object(10)
memory usage: 14.5+ MB
```

```
In [10]: unique_values = pd.unique(EV_data.values.ravel())      # Found unique values across the entire dataset
print(unique_values)
```

```
['JN1AZ0CP8B' 'Yakima' 'WA' ... 193878387 125039043 194673692]
```

```
In [11]: unique_value_counts = {col: EV_data[col].value_counts() for col in EV_data.columns}  # Get unique values with counts for each column
print(unique_value_counts)
```

```

{'VIN (1-10)': VIN (1-10)
5YJYGDDE9M 471
5YJYGDDE0M 463
5YJYGDDE7M 447
5YJYGDDE8M 446
5YJYGDDE2M 435
...
YV4BR0DL8M 1
JJHKCFZSN 1
WA1J2BFZ3N 1
KNDC4DLC5P 1
WA1LAAGE5M 1
Name: count, Length: 7522, dtype: int64, 'County': County
King 58980
Snohomish 12412
Pierce 8525
Clark 6681
Thurston 4109
Kitsap 3828
Whatcom 2839
Spokane 2785
Benton 1376
Island 1298
Skagit 1228
Clallam 728
San Juan 717
Jefferson 698
Chelan 654
Yakima 617
Cowlitz 569
Mason 547
Lewis 431
Grays Harbor 402
Kittitas 392
Franklin 365
Grant 335
Walla Walla 312
Douglas 221
Whitman 177
Klickitat 175
Okanogan 149
Pacific 145
Skamania 139
Stevens 91
Asotin 48
Wahkiakum 39
Adams 34
Pend Oreille 32
Lincoln 30
Ferry 27
Columbia 13
Garfield 4
Name: count, dtype: int64, 'City': City
Seattle 20295
Bellevue 5919
Redmond 4199
Vancouver 4013
Kirkland 3598
...
Walla Walla Co 1
Clallam Bay 1
Malott 1
Rockport 1
Uniontown 1
Name: count, Length: 435, dtype: int64, 'State': State
WA 112152
Name: count, dtype: int64, 'Postal Code': Postal Code
98052 2914
98033 2059
98004 2001
98115 1878
98006 1851
...
98283 1
98530 1
98535 1
98243 1
99179 1
Name: count, Length: 516, dtype: int64, 'Model Year': Model Year
2022-01-01 26455
2021-01-01 18277
2018-01-01 14190
2020-01-01 10998
2019-01-01 10216

```

```

2017-01-01    8598
2016-01-01    5709
2015-01-01    4918
2013-01-01    4669
2014-01-01    3665
2023-01-01    1863
2012-01-01    1695
2011-01-01    835
2010-01-01     24
2008-01-01     23
2000-01-01     10
1999-01-01      3
2002-01-01      2
1997-01-01      1
1998-01-01      1
Name: count, dtype: int64, 'Make': Make
TESLA          51883
NISSAN         12846
CHEVROLET      10140
FORD            5780
BMW             4660
KIA             4469
TOYOTA          4368
VOLKSWAGEN     2507
AUDI            2320
VOLVO           2256
CHRYSLER        1780
HYUNDAI         1407
JEEP             1143
RIVIAN           883
FIAT             820
PORSCHE          817
HONDA            788
MINI             631
MITSUBISHI      585
POLESTAR         557
MERCEDES-BENZ   503
SMART            271
JAGUAR           218
LINCOLN          167
CADILLAC         108
LUCID MOTORS     65
SUBARU            59
LAND ROVER       38
LEXUS             33
FISKER            19
GENESIS           18
AZURE DYNAMICS    7
TH!NK              3
BENTLEY           3
Name: count, dtype: int64, 'Model': Model
MODEL 3          23042
MODEL Y          17086
LEAF             12846
MODEL S          7346
BOLT EV          4895
...
745LE            2
S-10 PICKUP      1
SOLTERRA         1
918              1
FLYING SPUR      1
Name: count, Length: 114, dtype: int64, 'Electric Vehicle Type': Electric Vehicle Type
Battery Electric Vehicle (BEV)      85732
Plug-in Hybrid Electric Vehicle (PHEV)  26420
Name: count, dtype: int64, 'Clean Alternative Fuel Vehicle (CAFV) Eligibility': Clean Alternative Fuel Vehicle (CAFV) Eligibility
Clean Alternative Fuel Vehicle Eligible  58395
Eligibility unknown as battery range has not been researched  39097
Not eligible due to low battery range    14660
Name: count, dtype: int64, 'Electric Range': Electric Range
0                39097
215              6282
84               4188
220              4012
238              3461
...
11                3
95                2
57                1
39                1
59                1
Name: count, Length: 101, dtype: int64, 'Base MSRP': Base MSRP
0                 108654
69900            1493

```

```
31950      406
52900      213
32250      158
54950      135
59900      134
39995      118
36900      100
44100      95
64950      82
33950      78
45600      76
52650      67
34995      58
36800      50
55700      47
53400      28
110950     24
98950      23
81100      19
102000     19
90700      18
75095      16
184400     12
43700      10
109000     7
89100      7
91250      4
845000     1
Name: count, dtype: int64, 'Legislative District': Legislative District
41      7602
45      7112
48      6460
36      5251
46      4721
1       4714
5       4691
43      4620
37      3554
34      3477
18      3024
22      2772
32      2707
11      2702
44      2670
23      2614
21      2613
40      2599
26      2260
33      2112
10      2051
31      1908
17      1907
47      1875
24      1661
27      1651
42      1625
35      1613
39      1574
49      1573
28      1447
30      1267
2       1225
8       1157
38      1065
25      1049
6       1041
12      1004
20      973
4       845
13      748
14      720
29      692
19      669
16      611
9       606
3       557
7       486
15      277
Name: count, dtype: int64, 'DOL Vehicle ID': DOL Vehicle ID
218972519    1
200615726    1
172852179    1
218915427    1
121879635    1
```

```

223766105    1
183253458    1
104299002    1
477394753    1
194673692    1
Name: count, Length: 112152, dtype: int64, 'Vehicle Location': Vehicle Location
POINT (-122.13158 47.67858)    2914
POINT (-122.2066 47.67887)    2059
POINT (-122.1872 47.61001)    2001
POINT (-122.31765 47.70013)   1878
POINT (-122.12096 47.55584)   1851
...
POINT (-121.59274 48.48758)   1
POINT (27.25316 67.01865)    1
POINT (-124.16705 47.11487)  1
POINT (-123.00026 48.61989)  1
POINT (-117.08742 46.53906)  1
Name: count, Length: 516, dtype: int64, 'Electric Utility': Electric Utility
PUGET SOUND ENERGY INC||CITY OF TACOMA - (WA)          40231
PUGET SOUND ENERGY INC                                22166
CITY OF SEATTLE - (WA)||CITY OF TACOMA - (WA)         21439
BONNEVILLE POWER ADMINISTRATION||PUD NO 1 OF CLARK COUNTY - (WA) 6522
BONNEVILLE POWER ADMINISTRATION||CITY OF TACOMA - (WA)||PENINSULA LIGHT COMPANY 5049
...
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: count, Length: 73, dtype: int64, '2020 Census Tract': 2020 Census Tract
1493420948    583
1493424769    550
1493400248    418
1493416548    401
1493398048    394
...
1481412851    1
1482372548    1
1537393748    1
1538332455    1
1535392548    1
Name: count, Length: 1760, dtype: int64, 'Longitude': Longitude
-122.13158    2914
-122.20660    2059
-122.18720    2001
-122.31765    1878
-122.12096    1851
...
-121.59274    1
27.25316     1
-124.16705    1
-123.00026    1
-117.08742    1
Name: count, Length: 516, dtype: int64, 'Latitude': Latitude
47.67858     2914
47.67887     2059
47.61001     2001
47.70013     1878
47.55584     1851
...
48.48758     1
67.01865     1
47.11487     1
48.61989     1
46.53906     1
Name: count, Length: 516, dtype: int64}

```

```
In [12]: duplicate_rows = EV_data[EV_data.duplicated()]
print(duplicate_rows)
```

```
Empty DataFrame
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, 2020 Census Tract, Longitude, Latitude]
Index: []
```

Univariate Analysis = Min, Max, Mean, Median, Std, Skew, Kurt

```
In [13]: 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','skew','kurt']))
print()

```

In [14]: numerical_univariate_analysis(EV_data[['Electric Range', 'Base MSRP', 'Legislative District', 'DOL Vehicle ID']])

```

***** Electric Range *****
min      0.000000
max     337.000000
mean     87.829651
median    32.000000
std      102.336645
skew      0.816322
kurt     -0.877834
Name: Electric Range, dtype: float64

***** Base MSRP *****
min      0.000000
max    845000.000000
mean    1793.882320
median   0.000000
std     10785.259118
skew     10.113620
kurt    372.907751
Name: Base MSRP, dtype: float64

***** Legislative District *****
min      1.000000
max     49.000000
mean    29.817703
median   34.000000
std     14.698726
skew    -0.543619
kurt    -0.978869
Name: Legislative District, dtype: float64

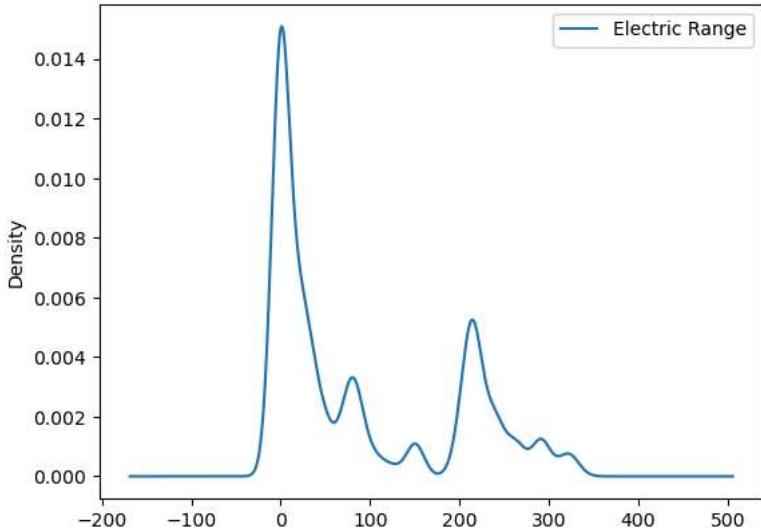
***** DOL Vehicle ID *****
min    4.777000e+03
max    4.792548e+08
mean   1.994712e+08
median  1.923916e+08
std    9.401842e+07
skew   1.149405e+00
kurt   2.464145e+00
Name: DOL Vehicle ID, dtype: float64

```

Graphical Representation of data

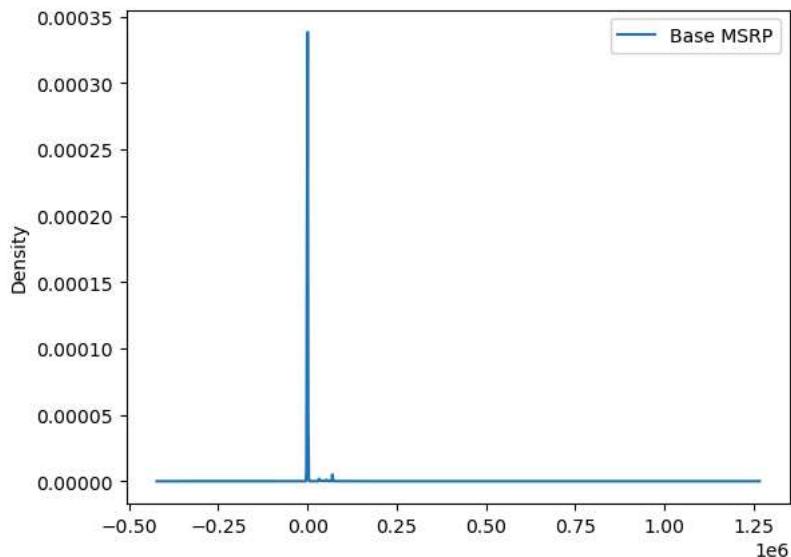
In [15]: EV_data[['Electric Range']].plot(kind='kde')

Out[15]: <Axes: ylabel='Density'>



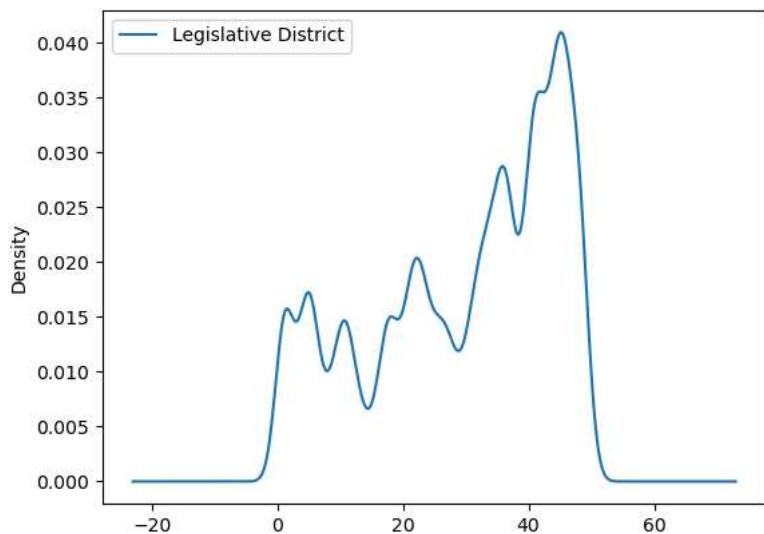
In [16]: EV_data[['Base MSRP']].plot(kind='kde')

Out[16]: <Axes: ylabel='Density'>



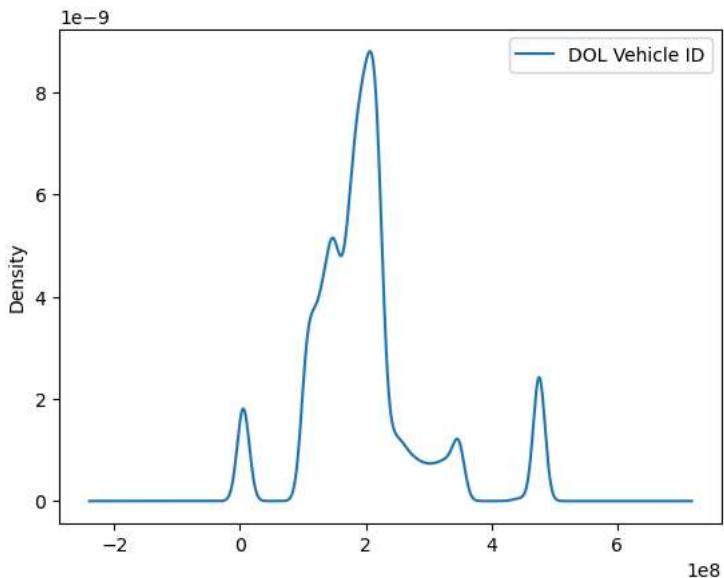
```
In [17]: EV_data[['Legislative District']].plot(kind='kde')
```

```
Out[17]: <Axes: ylabel='Density'>
```



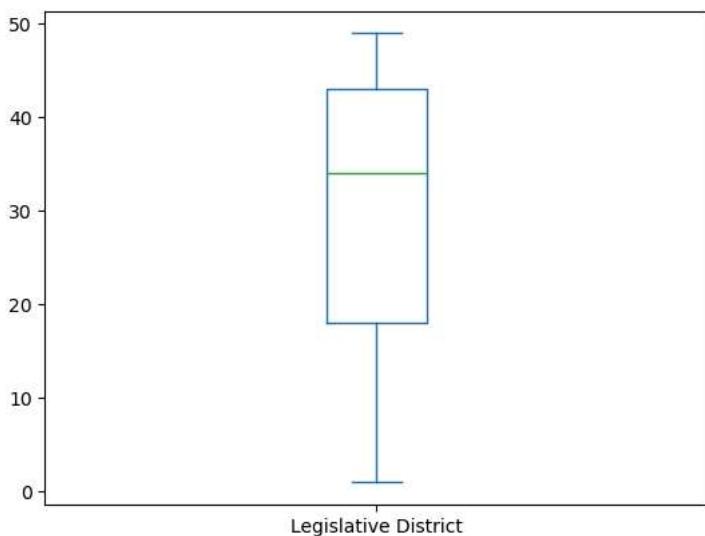
```
In [18]: EV_data[['DOL Vehicle ID']].plot(kind='kde')
```

```
Out[18]: <Axes: ylabel='Density'>
```



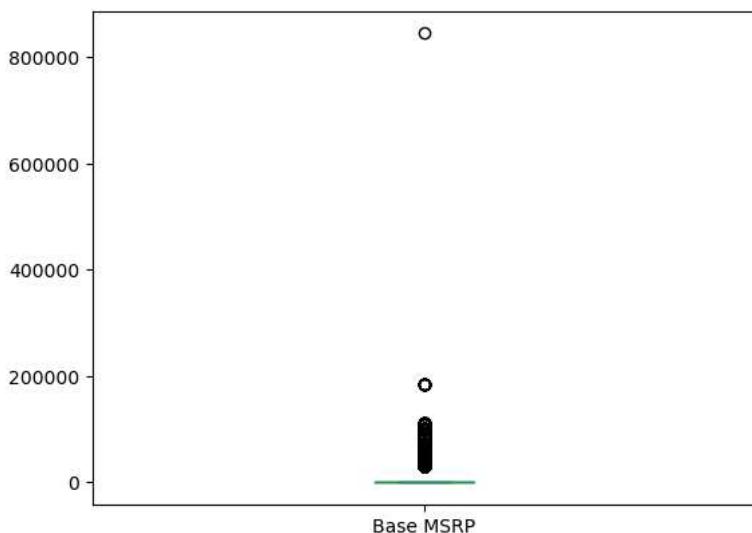
```
In [19]: EV_data['Legislative District'].plot(kind="box")
```

```
Out[19]: <Axes: >
```



```
In [20]: EV_data['Base MSRP'].plot(kind="box")
```

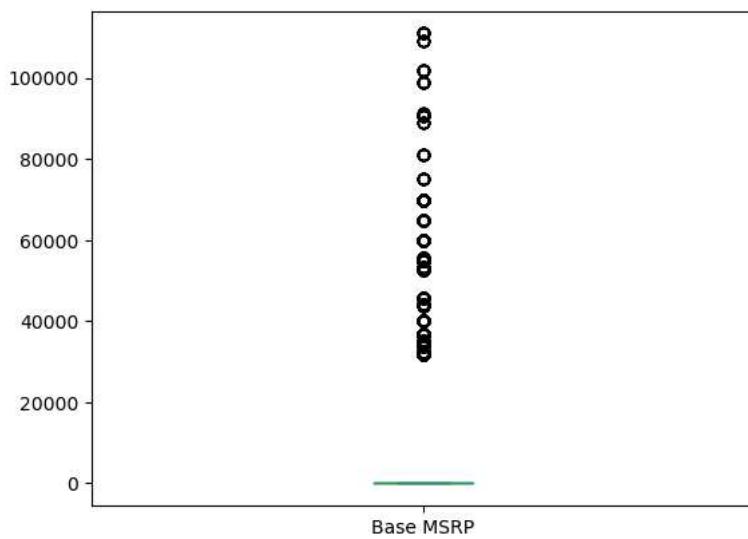
```
Out[20]: <Axes: >
```



```
In [21]: EV_data = EV_data[EV_data["Base MSRP"] <= 125000]
```

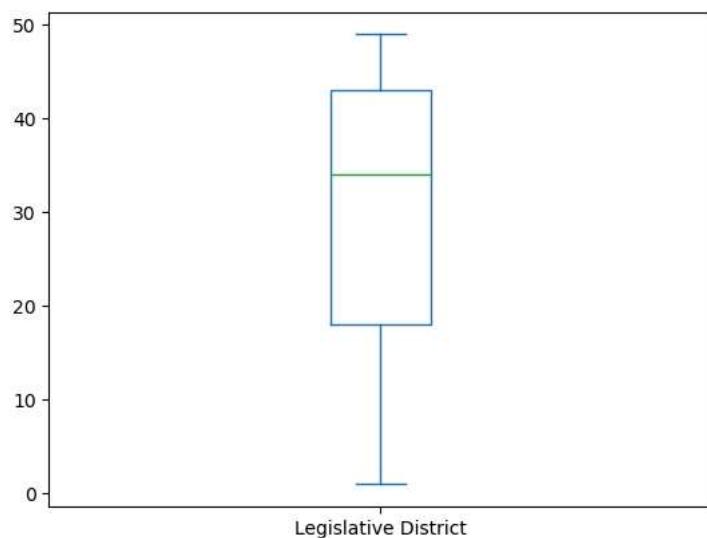
```
In [22]: EV_data['Base MSRP'].plot(kind="box")
```

```
Out[22]: <Axes: >
```



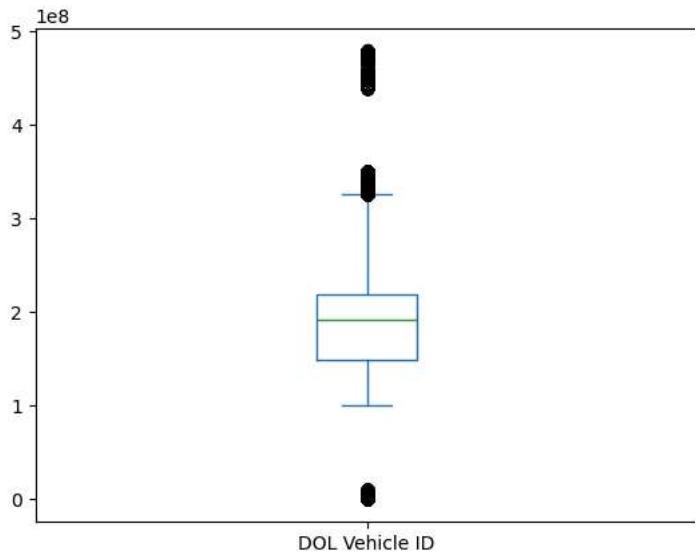
```
In [23]: EV_data['Legislative District'].plot(kind="box")
```

```
Out[23]: <Axes: >
```



```
In [24]: EV_data['DOL Vehicle ID'].plot(kind="box")
```

```
Out[24]: <Axes: >
```



```
In [25]: 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(normalize=True))
        print()
```

```
In [26]: EV_data.info()

<class 'pandas.core.frame.DataFrame'>
Index: 112139 entries, 2 to 112633
Data columns (total 19 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   VIN (1-10)      112139 non-null   object 
 1   County          112139 non-null   object 
 2   City            112139 non-null   object 
 3   State           112139 non-null   object 
 4   Postal Code     112139 non-null   int32  
 5   Model Year      112139 non-null   datetime64[ns]
 6   Make            112139 non-null   object 
 7   Model           112139 non-null   object 
 8   Electric Vehicle Type 112139 non-null   object 
 9   Clean Alternative Fuel Vehicle (CAFV) Eligibility 112139 non-null   object 
 10  Electric Range  112139 non-null   int32  
 11  Base MSRP       112139 non-null   int32  
 12  Legislative District 112139 non-null   int32  
 13  DOL Vehicle ID 112139 non-null   int32  
 14  Vehicle Location 112139 non-null   object 
 15  Electric Utility 112139 non-null   object 
 16  2020 Census Tract 112139 non-null   int32  
 17  Longitude        112139 non-null   float64
 18  Latitude         112139 non-null   float64
dtypes: datetime64[ns](1), float64(2), int32(6), object(10)
memory usage: 14.5+ MB
```

```
In [27]: discrete_univariate_analysis(EV_data[['VIN (1-10)', 'County', 'City', 'State', 'Make', 'Model', 'Model Year', 'Electric Vehicle Type']]
```

```
***** VIN (1-10) *****
count 112139
nunique 7512
unique [JN1AZ0CP8B, 1G1FW6S08H, 3FA6P0SU1K, 5YJ3E1EB5...
Name: VIN (1-10), dtype: object
value counts: |n VIN (1-10)
5YJYGDEE9M 0.004200
5YJYGDEE0M 0.004129
5YJYGDEE7M 0.003986
5YJYGDEE8M 0.003977
5YJYGDEE2M 0.003879
...
KMHC65LDXJ 0.000009
WAUSPBFF0H 0.000009
1FTZR081XY 0.000009
WMEFK9BA8H 0.000009
WA1LAAGE5M 0.000009
Name: proportion, Length: 7512, dtype: float64

***** County *****
count 112139
nunique 39
unique [Yakima, Skagit, Snohomish, Island, Thurston, ...
Name: County, dtype: object
value counts: |n County
King 0.525901
Snohomish 0.110675
Pierce 0.076013
Clark 0.059551
Thurston 0.036633
Kitsap 0.034136
Whatcom 0.025317
Spokane 0.024835
Benton 0.012270
Island 0.011575
Skagit 0.010942
Clallam 0.006492
San Juan 0.006394
Jefferson 0.006224
Chelan 0.005832
Yakima 0.005502
Cowlitz 0.005074
Mason 0.004878
Lewis 0.003843
Grays Harbor 0.003585
Kittitas 0.003496
Franklin 0.003255
Grant 0.002987
Walla Walla 0.002782
Douglas 0.001971
Whitman 0.001578
Klickitat 0.001561
Okanogan 0.001329
Pacific 0.001293
Skamania 0.001240
Stevens 0.000811
Asotin 0.000428
Wahkiakum 0.000348
Adams 0.000303
Pend Oreille 0.000285
Lincoln 0.000268
Ferry 0.000241
Columbia 0.000116
Garfield 0.000036
Name: proportion, dtype: float64

***** City *****
count 112139
nunique 435
unique [Yakima, Concrete, Everett, Bothell, Mukilteo,...
Name: City, dtype: object
value counts: |n City
Seattle 0.180972
Bellevue 0.052774
Redmond 0.037445
Vancouver 0.035768
Kirkland 0.032085
...
Walla Walla Co 0.000009
Clallam Bay 0.000009
Malott 0.000009
Rockport 0.000009
Uniontown 0.000009
Name: proportion, Length: 435, dtype: float64
```

```
***** State *****
count      112139
nunique      1
unique      [WA]
Name: State, dtype: object
value counts: |n State
WA      1.0
Name: proportion, dtype: float64

***** Make *****
count                  112139
nunique                 34
unique      [NISSAN, CHEVROLET, FORD, TESLA, KIA, AUDI, BM...
Name: Make, dtype: object
value counts: |n Make
TESLA      0.462667
NISSAN      0.114554
CHEVROLET    0.090423
FORD        0.051543
BMW         0.041556
KIA          0.039852
TOYOTA       0.038952
VOLKSWAGEN   0.022356
AUDI         0.020689
VOLVO        0.020118
CHRYSLER     0.015873
HYUNDAI      0.012547
JEEP          0.010193
RIVIAN        0.007874
FIAT          0.007312
PORSCHE       0.007170
HONDA         0.007027
MINI          0.005627
MITSUBISHI   0.005217
POLESTAR      0.004967
MERCEDES-BENZ 0.004486
SMART         0.002417
JAGUAR        0.001944
LINCOLN       0.001489
CADILLAC      0.000963
LUCID MOTORS  0.000580
SUBARU        0.000526
LAND ROVER    0.000339
LEXUS          0.000294
FISKER        0.000169
GENESIS        0.000161
AZURE DYNAMICS 0.000062
TH!NK          0.000027
BENTLEY       0.000027
Name: proportion, dtype: float64

***** Model *****
count                  112139
nunique                 113
unique      [LEAF, BOLT EV, FUSION, MODEL 3, SOUL, Q5 E, M...
Name: Model, dtype: object
value counts: |n Model
MODEL 3      0.205477
MODEL Y      0.152364
LEAF          0.114554
MODEL S      0.065508
BOLT EV      0.043651
...
BENTAYGA     0.000018
745LE         0.000018
S-10 PICKUP   0.000009
SOLTERRA      0.000009
FLYING SPUR   0.000009
Name: proportion, Length: 113, dtype: float64

***** Model Year *****
count                  112139
nunique                 20
unique      [2011-01-01 00:00:00, 2017-01-01 00:00:00, 201...
Name: Model Year, dtype: object
value counts: |n Model Year
2022-01-01    0.235913
2021-01-01    0.162985
2018-01-01    0.126432
2020-01-01    0.098075
2019-01-01    0.091101
2017-01-01    0.076673
2016-01-01    0.050910
```

```

2015-01-01  0.043847
2013-01-01  0.041636
2014-01-01  0.032683
2023-01-01  0.016613
2012-01-01  0.015115
2011-01-01  0.007446
2010-01-01  0.000214
2008-01-01  0.000205
2000-01-01  0.000089
1999-01-01  0.000027
2002-01-01  0.000018
1997-01-01  0.000009
1998-01-01  0.000009
Name: proportion, dtype: float64

```

```

***** Electric Vehicle Type *****
count                               112139
nunique                             2
unique      [Battery Electric Vehicle (BEV), Plug-in Hybri...
Name: Electric Vehicle Type, dtype: object
value counts: |n Electric Vehicle Type
Battery Electric Vehicle (BEV)        0.764515
Plug-in Hybrid Electric Vehicle (PHEV) 0.235485
Name: proportion, dtype: float64

```

```

***** Clean Alternative Fuel Vehicle (CAFV) Eligibility *****
count                               112139
nunique                             3
unique      [Clean Alternative Fuel Vehicle Eligible, Not ...
Name: Clean Alternative Fuel Vehicle (CAFV) Eligibility, dtype: object
value counts: |n Clean Alternative Fuel Vehicle (CAFV) Eligibility
Clean Alternative Fuel Vehicle Eligible          0.520738
Eligibility unknown as battery range has not been researched 0.348648
Not eligible due to low battery range           0.130615
Name: proportion, dtype: float64

```

```

***** Electric Utility *****
count                               112139
nunique                             73
unique      [PACIFICORP, PUGET SOUND ENERGY INC, PUD NO 2 ...
Name: Electric Utility, dtype: object
value counts: |n Electric Utility
PUGET SOUND ENERGY INC||CITY OF TACOMA - (WA)            0.358716
PUGET SOUND ENERGY INC                                0.197639
CITY OF SEATTLE - (WA)||CITY OF TACOMA - (WA)            0.191173
BONNEVILLE POWER ADMINISTRATION||PUD NO 1 OF CLARK COUNTY - (WA) 0.058133
BONNEVILLE POWER ADMINISTRATION||CITY OF TACOMA - (WA)||PENINSULA LIGHT COMPANY 0.045016
                                         ...
BONNEVILLE POWER ADMINISTRATION||PENINSULA LIGHT COMPANY    0.000009
BONNEVILLE POWER ADMINISTRATION||PUD NO 1 OF ASOTIN COUNTY 0.000009
CITY OF SEATTLE - (WA)                                    0.000009
BONNEVILLE POWER ADMINISTRATION||NESPELEM VALLEY ELEC COOP, INC 0.000009
BONNEVILLE POWER ADMINISTRATION||PUD NO 1 OF CLALLAM COUNTY||PUD NO 1 OF JEFFERSON COUNTY 0.000009
Name: proportion, Length: 73, dtype: float64

```

```

***** Vehicle Location *****
count                               112139
nunique                             516
unique      [POINT (-120.50721 46.60448), POINT (-121.7515...
Name: Vehicle Location, dtype: object
value counts: |n Vehicle Location
POINT (-122.13158 47.67858)     0.025986
POINT (-122.2066 47.67887)      0.018361
POINT (-122.1872 47.61001)      0.017835
POINT (-122.31765 47.70013)     0.016747
POINT (-122.12096 47.55584)     0.016497
                                         ...
POINT (-121.59274 48.48758)     0.000009
POINT (27.25316 67.01865)       0.000009
POINT (-124.16705 47.11487)     0.000009
POINT (-123.00026 48.61989)     0.000009
POINT (-117.08742 46.53906)     0.000009
Name: proportion, Length: 516, dtype: float64

```

```

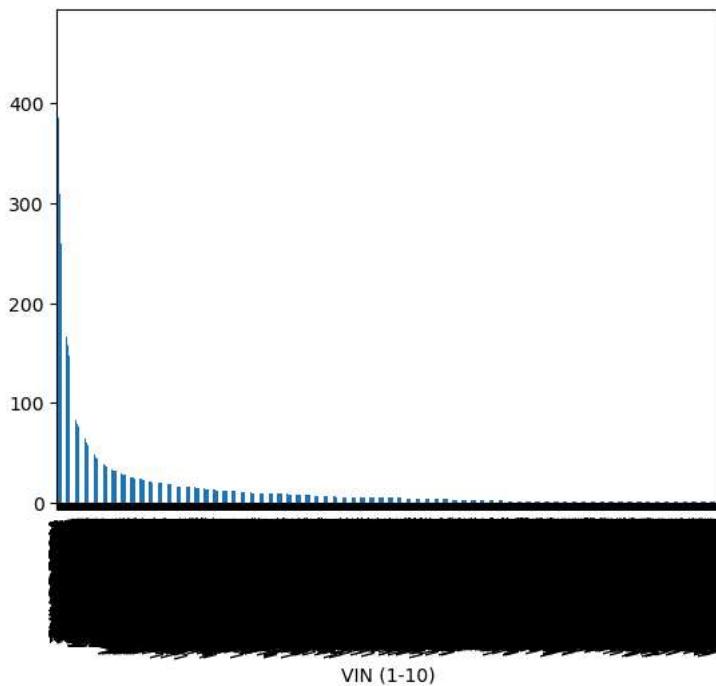
***** 2020 Census Tract *****
count                               112139
nunique                             1760
unique      [1537394050, 1518343549, 1521433948, 152144436...
Name: 2020 Census Tract, dtype: object
value counts: |n 2020 Census Tract
1493420948  0.005199
1493424769  0.004905
1493400248  0.003728
1493416548  0.003567

```

```
1493398048  0.003513
...
1481412851  0.000009
1482372548  0.000009
1537393748  0.000009
1538332455  0.000009
1535392548  0.000009
Name: proportion, Length: 1760, dtype: float64
```

In [28]: `EV_data["VIN (1-10)"].value_counts().plot(kind="bar")`

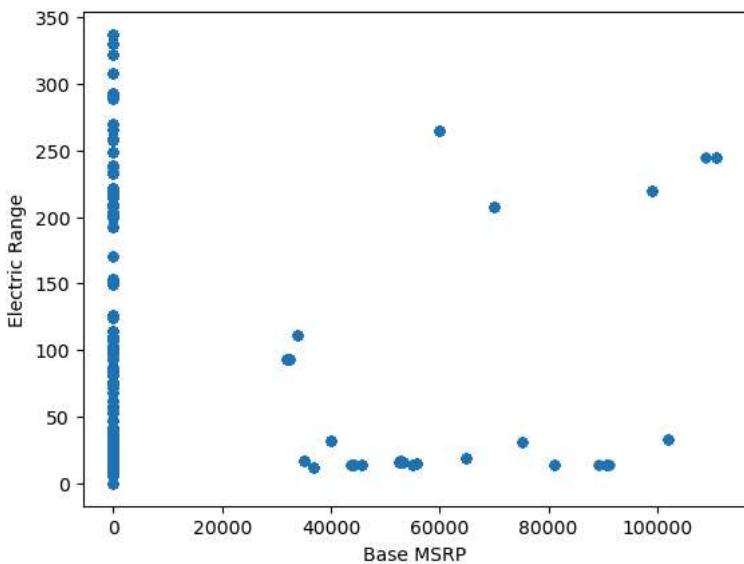
Out[28]: <Axes: xlabel='VIN (1-10)'>



Bivariate analysis

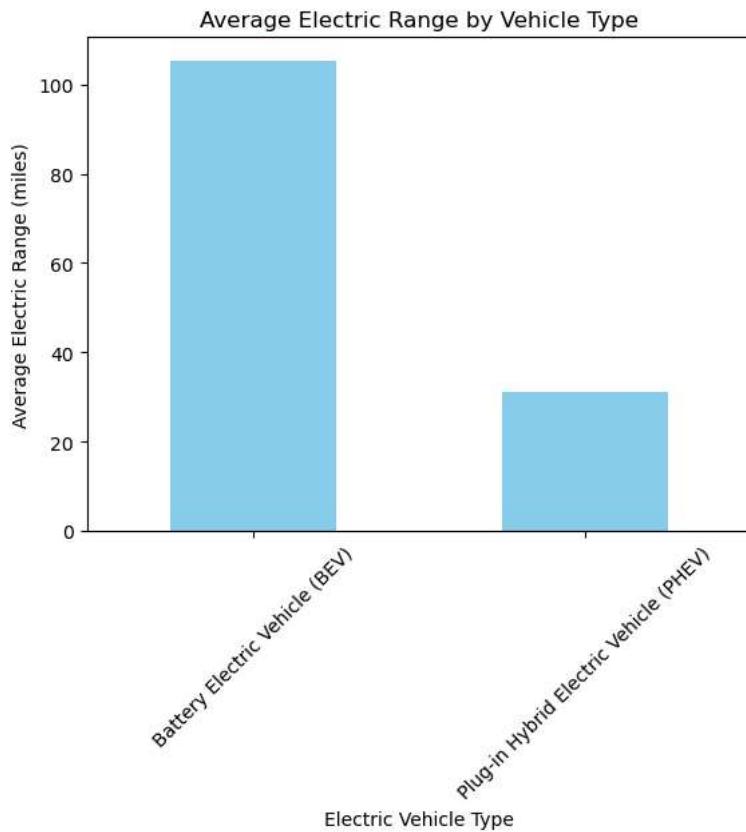
In [30]: `EV_data.plot(kind='scatter', x= 'Base MSRP', y='Electric Range')`

Out[30]: <Axes: xlabel='Base MSRP', ylabel='Electric Range'>



In [31]: `grouped_data = EV_data.groupby('Electric Vehicle Type')['Electric Range'].mean()
Plotting the bar chart
grouped_data.plot(kind='bar', color='skyblue')
plt.title('Average Electric Range by Vehicle Type') # Add titles and labels
plt.xlabel('Electric Vehicle Type')
plt.ylabel('Average Electric Range (miles)')`

```
plt.xticks(rotation=45)  
plt.show()
```



```
In [32]: !pip install plotly  
import plotly.express as px
```

```
Requirement already satisfied: plotly in c:\users\admin\anaconda3\lib\site-packages (5.22.0)  
Requirement already satisfied: tenacity>=6.2.0 in c:\users\admin\anaconda3\lib\site-packages (from plotly) (8.2.2)  
Requirement already satisfied: packaging in c:\users\admin\anaconda3\lib\site-packages (from plotly) (23.2)
```

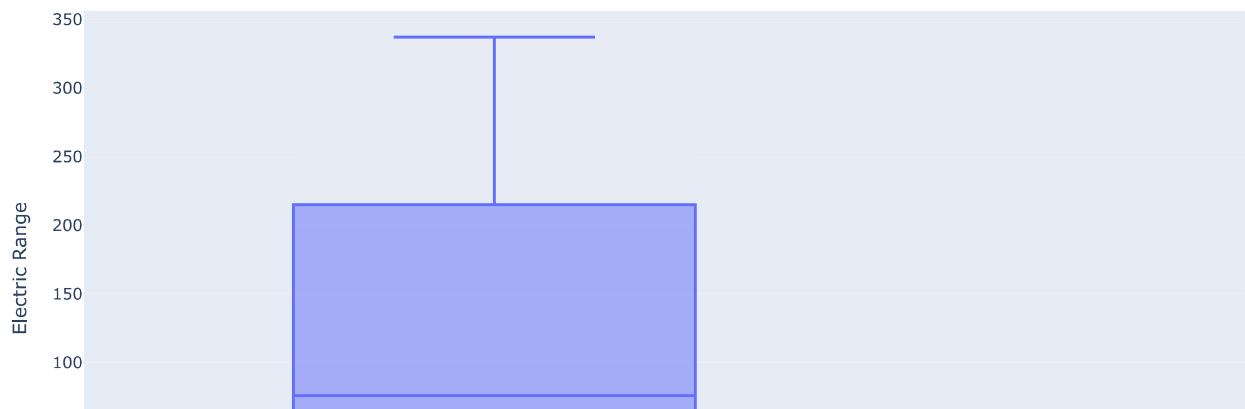
Scatter Plot using plotly.express

```
In [34]: px.scatter(EV_data, x= 'Base MSRP',y='Electric Range')
```



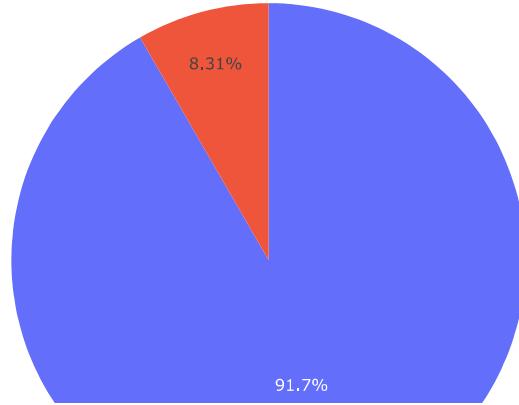
Box Plot using plotly.express

```
In [36]: px.box(EV_data, x='Electric Vehicle Type', y='Electric Range')
```



Pie Chart Plot using plotly.express

```
In [76]: px.pie(EV_data, names = 'Electric Vehicle Type', values = 'Electric Range')
```



Choropleth Plot using plotly.express

```
In [90]: EV_data.columns
```

```
Out[90]: 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',
       'Longitude', 'Latitude'],
      dtype='object')
```

```
In [ ]: # fig = px.choropleth(temp_EV_data,
#                           locations = "County",
#                           color = "Num",
#                           hover_name = "Confirmed",
#                           locationmode="County Names")
# fig.show()
```

```
In [94]: EV_data["State"].unique()
```

```
Out[94]: array(['WA'], dtype=object)
```

```
In [102... grouped_EV_data = EV_data.groupby("State").agg({"Electric Range": "mean"})
grouped_EV_data
```

```
Out[102... Electric Range
```

State	
WA	87.838228

```
In [110... EV_Counts = EV_data .groupby('2020 Census Tract')['VIN (1-10)'].count().reset_index()
EV_Counts
```

Out[110...]

	2020 Census Tract	VIN (1-10)
0	1462342548	10
1	1462342648	3
2	1462342749	4
3	1462342750	1
4	1462342751	4
...
1755	1538332450	6
1756	1538332451	6
1757	1538332453	1
1758	1538332454	3
1759	1538332455	1

1760 rows × 2 columns

In [123...]

```
EV_Count_By_Pincode = EV_data.groupby(['Postal Code', 'Model Year', 'State']).size().reset_index(name='Number_of_EV_Vehicles')
state_data = EV_Count_By_Pincode[EV_Count_By_Pincode['State'] == 'WA']
state_data
```

Out[123...]

	Postal Code	Model Year	State	Number_of_EV_Vehicles
0	98001	2011-01-01	WA	1
1	98001	2012-01-01	WA	5
2	98001	2013-01-01	WA	23
3	98001	2014-01-01	WA	23
4	98001	2015-01-01	WA	19
...
4731	99403	2019-01-01	WA	1
4732	99403	2020-01-01	WA	4
4733	99403	2021-01-01	WA	6
4734	99403	2022-01-01	WA	5
4735	99403	2023-01-01	WA	1

4736 rows × 4 columns

In [153...]

```
fig = px.choropleth_mapbox(state_data,
                            locations='Postal Code',
                            color='Number_of_EV_Vehicles',
                            hover_name='Number_of_EV_Vehicles',
                            animation_frame='Model Year', # Correct the typo here
                            mapbox_style='carto-positron',
                            zoom=5,
                            center={"lat": 37.0902, "lon": -95.7129}) # Adjust center and zoom as per your data
fig.show()
```

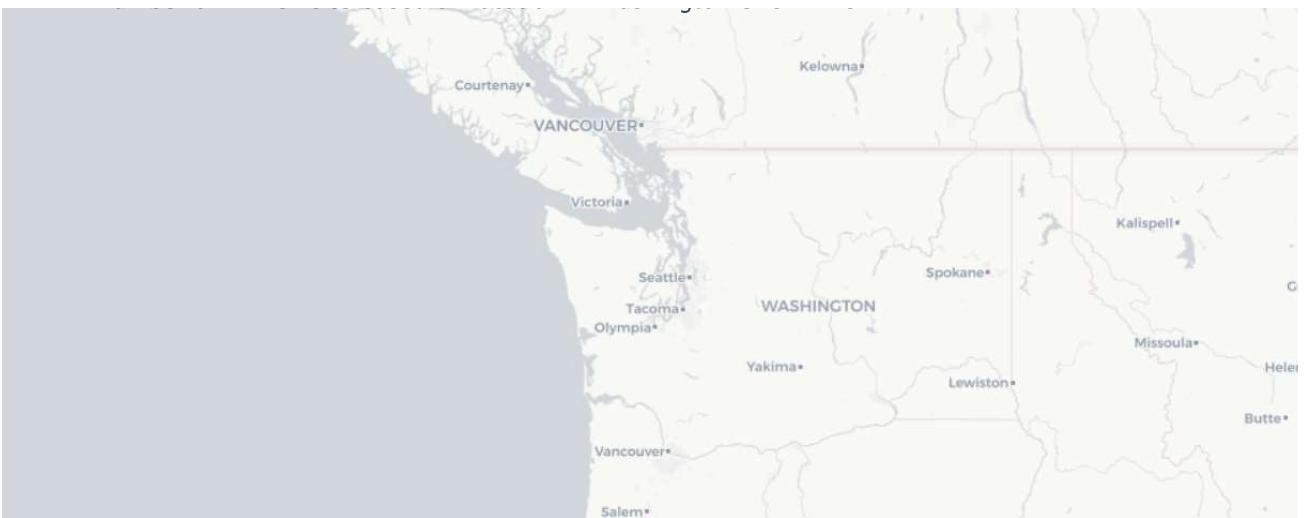


```
In [159]: import plotly.express as px

fig = px.choropleth_mapbox(state_data,
                           geojson="https://raw.githubusercontent.com/OpenDataDE/State-zip-code-GeoJSON/master/wa_washington_zip_code.json",
                           locations='Postal Code',
                           color='Number_of_EV_Vehicles',
                           featureidkey="properties.ZCTA5CE10",
                           mapbox_style="carto-positron",
                           zoom=5, # Adjust zoom level
                           center={"lat": 47.7511, "lon": -120.7401},
                           title="Number of EV vehicles based on location in Washington Over Time",
                           animation_frame="Model Year",
                           color_continuous_scale="Viridis",
                           hover_data=['Number_of_EV_Vehicles']
                           )

# Update Layout for better fit and aesthetics
fig.update_layout(margin={"r": 0, "t": 0, "l": 0, "b": 0})

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



Racing Bar Plot

In [162...]

```
!pip install bar-chart-race
Collecting bar-chart-race
  Downloading bar_chart_race-0.1.0-py3-none-any.whl.metadata (4.2 kB)
Requirement already satisfied: pandas>=0.24 in c:\users\admin\anaconda3\lib\site-packages (from bar-chart-race) (2.2.2)
Requirement already satisfied: matplotlib>=3.1 in c:\users\admin\anaconda3\lib\site-packages (from bar-chart-race) (3.8.4)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\admin\anaconda3\lib\site-packages (from matplotlib>=3.1->bar-chart-race) (1.2.0)
Requirement already satisfied: cycler>=0.10 in c:\users\admin\anaconda3\lib\site-packages (from matplotlib>=3.1->bar-chart-race) (0.1.0)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\admin\anaconda3\lib\site-packages (from matplotlib>=3.1->bar-chart-race) (4.51.0)
Requirement already satisfied: kiwisolver>=1.3.1 in c:\users\admin\anaconda3\lib\site-packages (from matplotlib>=3.1->bar-chart-race) (1.4.4)
Requirement already satisfied: numpy>=1.21 in c:\users\admin\anaconda3\lib\site-packages (from matplotlib>=3.1->bar-chart-race) (1.26.4)
Requirement already satisfied: packaging>=20.0 in c:\users\admin\anaconda3\lib\site-packages (from matplotlib>=3.1->bar-chart-race) (2.3.2)
Requirement already satisfied: pillow>=8 in c:\users\admin\anaconda3\lib\site-packages (from matplotlib>=3.1->bar-chart-race) (10.3.0)
Requirement already satisfied: pyparsing>=2.3.1 in c:\users\admin\anaconda3\lib\site-packages (from matplotlib>=3.1->bar-chart-race) (3.0.9)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\admin\anaconda3\lib\site-packages (from matplotlib>=3.1->bar-chart-race) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in c:\users\admin\anaconda3\lib\site-packages (from pandas>=0.24->bar-chart-race) (2024.1)
Requirement already satisfied: tzdata>=2022.7 in c:\users\admin\anaconda3\lib\site-packages (from pandas>=0.24->bar-chart-race) (2023.3)
Requirement already satisfied: six>=1.5 in c:\users\admin\anaconda3\lib\site-packages (from python-dateutil>=2.7->matplotlib>=3.1->bar-chart-race) (1.16.0)
  Downloading bar_chart_race-0.1.0-py3-none-any.whl (156 kB)
    ----- 0.0/156.8 kB ? eta -----
    ----- 156.8/156.8 kB 4.7 MB/s eta 0:00:00
Installing collected packages: bar-chart-race
Successfully installed bar-chart-race-0.1.0
```

In [174...]

```
import plotly.express as px

df = EV_data.groupby(['Make', 'Model Year']).size().reset_index(name='Number_of_Vehicles')

fig = px.bar(df, # Create the animated racing bar plot with annotations
             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', # Create animation by year
             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]
)
```

```

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

fig.update_layout( # Adjust the layout for improved visibility and emphasis on movement
    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
)
fig.show()

```

EV Makes and their Count Over the Years

