In [1]:

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

In [2]:

```
# Read the Data
df = pd.read_csv(r"C:\Users\Admin\Downloads\Electric_Vehicle_Population_Data.csv", engine = 'python', encoding
```

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	KM8K33AGXL	King	Seattle	WA	98103.0	2020	HYUNDAI	KONA	Battery Electric Vehicle (BEV)	Clean Alternative Fuel Vehicle Eligible	258	0	43.0
1	1C4RJYB61N	King	Bothell	WA	98011.0	2022	JEEP	GRAND CHEROKEE	Plug-in Hybrid Electric Vehicle (PHEV)	Not eligible due to low battery range	25	0	1.0
2	1C4RJYD61P	Yakima	Yakima	WA	98908.0	2023	JEEP	GRAND CHEROKEE	Plug-in Hybrid Electric Vehicle (PHEV)	Not eligible due to low battery range	25	0	14.0
3	5YJ3E1EA7J	King	Kirkland	WA	98034.0	2018	TESLA	MODEL 3	Battery Electric Vehicle (BEV)	Clean Alternative Fuel Vehicle Eligible	215	0	45.0
4	WBY7Z8C5XJ	Thurston	Olympia	WA	98501.0	2018	BMW	I3	Plug-in Hybrid Electric Vehicle (PHEV)	Clean Alternative Fuel Vehicle Eligible	97	0	22.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...
150477	WBY43AW05P	Grays Harbor	Montesano	WA	98563.0	2023	BMW	I4	Battery Electric Vehicle (BEV)	Eligibility unknown as battery range has not b...	0	0	19.0
150478	5YJ3E1EB7P	King	Seattle	WA	98104.0	2023	TESLA	MODEL 3	Battery Electric Vehicle (BEV)	Eligibility unknown as battery range has not b...	0	0	43.0
150479	5YJYGDEEXM	King	Seattle	WA	98109.0	2021	TESLA	MODEL Y	Battery Electric Vehicle (BEV)	Eligibility unknown as battery range has not b...	0	0	43.0
150480	5UXTA6C08P	Snohomish	Mountlake Terrace	WA	98043.0	2023	BMW	X5	Plug-in Hybrid Electric Vehicle (PHEV)	Clean Alternative Fuel Vehicle Eligible	30	0	1.0
150481	7SAYGDEF8N	Skagit	Mount Vernon	WA	98273.0	2022	TESLA	MODEL Y	Battery Electric Vehicle (BEV)	Eligibility unknown as battery range has not b...	0	0	40.0

150482 rows × 17 columns

In [4]:

```
# Top data
df.head()
```

Out[4]:

	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	DO Vehicle I
0	KM8K33AGXL	King	Seattle	WA	98103.0	2020	HYUNDAI	KONA	Battery Electric Vehicle (BEV)	Clean Alternative Fuel Vehicle Eligible	258	0	43.0	24967514
1	1C4RJYB61N	King	Bothell	WA	98011.0	2022	JEEP	GRAND CHEROKEE	Plug-in Hybrid Electric Vehicle (PHEV)	Not eligible due to low battery range	25	0	1.0	23392850
2	1C4RJYD61P	Yakima	Yakima	WA	98908.0	2023	JEEP	GRAND CHEROKEE	Plug-in Hybrid Electric Vehicle (PHEV)	Not eligible due to low battery range	25	0	14.0	22967593
3	5YJ3E1EA7J	King	Kirkland	WA	98034.0	2018	TESLA	MODEL 3	Battery Electric Vehicle (BEV)	Clean Alternative Fuel Vehicle Eligible	215	0	45.0	10471446
4	WBY7Z8C5XJ	Thurston	Olympia	WA	98501.0	2018	BMW	I3	Plug-in Hybrid Electric Vehicle (PHEV)	Clean Alternative Fuel Vehicle Eligible	97	0	22.0	18549838

In [5]:

```
# Last Top data
df.tail()
```

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	Vehi
150477	WBY43AW05P	Grays Harbor	Montesano	WA	98563.0	2023	BMW	I4	Battery Electric Vehicle (BEV)	Eligibility unknown as battery range has not b...	0	0	19.0	2512
150478	5YJ3E1EB7P	King	Seattle	WA	98104.0	2023	TESLA	MODEL 3	Battery Electric Vehicle (BEV)	Eligibility unknown as battery range has not b...	0	0	43.0	2413
150479	5YJYGDEEXM	King	Seattle	WA	98109.0	2021	TESLA	MODEL Y	Battery Electric Vehicle (BEV)	Eligibility unknown as battery range has not b...	0	0	43.0	1807
150480	5UXTA6C08P	Snohomish	Mountlake Terrace	WA	98043.0	2023	BMW	X5	Plug-in Hybrid Electric Vehicle (PHEV)	Clean Alternative Fuel Vehicle Eligible	30	0	1.0	2404
150481	7SAYGDEF8N	Skagit	Mount Vernon	WA	98273.0	2022	TESLA	MODEL Y	Battery Electric Vehicle (BEV)	Eligibility unknown as battery range has not b...	0	0	40.0	2076

In [6]:

```
# Describe the size of table
df.shape
```

Out[6]:

```
(150482, 17)
```

In [7]:

```
# Columns Name
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]:

```
# stasticial information
```

```
df.describe()
```

Out[8]:

	Postal Code	Model Year	Electric Range	Base MSRP	Legislative District	DOL Vehicle ID	2020 Census Tract
count	150479.000000	150482.000000	150482.000000	150482.000000	150141.000000	1.504820e+05	1.504790e+05
mean	98168.344154	2020.005436	67.877839	1312.644735	29.343950	2.111122e+08	5.297195e+10
std	2473.612184	3.015209	96.230009	9231.310215	14.824829	8.196388e+07	1.638841e+09
min	1730.000000	1997.000000	0.000000	0.000000	1.000000	4.385000e+03	1.081042e+09
25%	98052.000000	2018.000000	0.000000	0.000000	18.000000	1.693473e+08	5.303301e+10
50%	98122.000000	2021.000000	18.000000	0.000000	33.000000	2.150306e+08	5.303303e+10
75%	98370.000000	2023.000000	97.000000	0.000000	43.000000	2.399119e+08	5.305307e+10
max	99577.000000	2024.000000	337.000000	845000.000000	49.000000	4.792548e+08	5.603300e+10

In [9]:

```
# To get Statistical information
df.describe().transpose()
```

Out[9]:

	count	mean	std	min	25%	50%	75%	max
Postal Code	150479.0	9.816834e+04	2.473612e+03	1.730000e+03	9.805200e+04	9.812200e+04	9.837000e+04	9.957700e+04
Model Year	150482.0	2.020005e+03	3.015209e+00	1.997000e+03	2.018000e+03	2.021000e+03	2.023000e+03	2.024000e+03
Electric Range	150482.0	6.787784e+01	9.623001e+01	0.000000e+00	0.000000e+00	1.800000e+01	9.700000e+01	3.370000e+02
Base MSRP	150482.0	1.312645e+03	9.231310e+03	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	8.450000e+05
Legislative District	150141.0	2.934395e+01	1.482483e+01	1.000000e+00	1.800000e+01	3.300000e+01	4.300000e+01	4.900000e+01
DOL Vehicle ID	150482.0	2.111122e+08	8.196388e+07	4.385000e+03	1.693473e+08	2.150306e+08	2.399119e+08	4.792548e+08
2020 Census Tract	150479.0	5.297195e+10	1.638841e+09	1.081042e+09	5.303301e+10	5.303303e+10	5.305307e+10	5.603300e+10

In [10]:

```
# check Duplicate values
df.duplicated()
```

Out[10]:

```
0         False
1         False
2         False
3         False
4         False
          ...  
150477    False
150478    False
150479    False
150480    False
150481    False
Length: 150482, dtype: bool
```

In [11]:

```
# Get information about data
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150482 entries, 0 to 150481
Data columns (total 17 columns):
 #   Column                                             Non-Null Count   Dtype  
---  ------                                             --------------   -----  
 0   VIN (1-10)                                         150482 non-null  object 
 1   County                                             150479 non-null  object 
 2   City                                               150479 non-null  object 
 3   State                                              150482 non-null  object 
 4   Postal Code                                        150479 non-null  float64
 5   Model Year                                         150482 non-null  int64  
 6   Make                                               150482 non-null  object 
 7   Model                                              150482 non-null  object 
 8   Electric Vehicle Type                              150482 non-null  object 
 9   Clean Alternative Fuel Vehicle (CAFV) Eligibility  150482 non-null  object 
 10  Electric Range                                     150482 non-null  int64  
 11  Base MSRP                                          150482 non-null  int64  
 12  Legislative District                               150141 non-null  float64
 13  DOL Vehicle ID                                     150482 non-null  int64  
 14  Vehicle Location                                   150475 non-null  object 
 15  Electric Utility                                   150479 non-null  object 
 16  2020 Census Tract                                  150479 non-null  float64
dtypes: float64(3), int64(4), object(10)
memory usage: 19.5+ MB
```

In [12]:

```
df.dropna()
```

Out[12]:

	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	KM8K33AGXL	King	Seattle	WA	98103.0	2020	HYUNDAI	KONA	Battery Electric Vehicle (BEV)	Clean Alternative Fuel Vehicle Eligible	258	0	43.0
1	1C4RJYB61N	King	Bothell	WA	98011.0	2022	JEEP	GRAND CHEROKEE	Plug-in Hybrid Electric Vehicle (PHEV)	Not eligible due to low battery range	25	0	1.0
2	1C4RJYD61P	Yakima	Yakima	WA	98908.0	2023	JEEP	GRAND CHEROKEE	Plug-in Hybrid Electric Vehicle (PHEV)	Not eligible due to low battery range	25	0	14.0
3	5YJ3E1EA7J	King	Kirkland	WA	98034.0	2018	TESLA	MODEL 3	Battery Electric Vehicle (BEV)	Clean Alternative Fuel Vehicle Eligible	215	0	45.0
4	WBY7Z8C5XJ	Thurston	Olympia	WA	98501.0	2018	BMW	I3	Plug-in Hybrid Electric Vehicle (PHEV)	Clean Alternative Fuel Vehicle Eligible	97	0	22.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...
150477	WBY43AW05P	Grays Harbor	Montesano	WA	98563.0	2023	BMW	I4	Battery Electric Vehicle (BEV)	Eligibility unknown as battery range has not b...	0	0	19.0
150478	5YJ3E1EB7P	King	Seattle	WA	98104.0	2023	TESLA	MODEL 3	Battery Electric Vehicle (BEV)	Eligibility unknown as battery range has not b...	0	0	43.0
150479	5YJYGDEEXM	King	Seattle	WA	98109.0	2021	TESLA	MODEL Y	Battery Electric Vehicle (BEV)	Eligibility unknown as battery range has not b...	0	0	43.0
150480	5UXTA6C08P	Snohomish	Mountlake Terrace	WA	98043.0	2023	BMW	X5	Plug-in Hybrid Electric Vehicle (PHEV)	Clean Alternative Fuel Vehicle Eligible	30	0	1.0
150481	7SAYGDEF8N	Skagit	Mount Vernon	WA	98273.0	2022	TESLA	MODEL Y	Battery Electric Vehicle (BEV)	Eligibility unknown as battery range has not b...	0	0	40.0

150137 rows × 17 columns

In [13]:

```
# check the null values
df.isnull().sum()
```

Out[13]:

```
VIN (1-10)                                             0
County                                                 3
City                                                   3
State                                                  0
Postal Code                                            3
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                                 341
DOL Vehicle ID                                         0
Vehicle Location                                       7
Electric Utility                                       3
2020 Census Tract                                      3
dtype: int64
```

In [14]:

```
# Check Duplicated Values in Tables
df.duplicated().sum()
```

Out[14]:

```
0
```

In [15]:

```
# Check Duplicated Values in columns
df['County'].duplicated().sum()
```

Out[15]:

```
150298
```

In [16]:

```
# permenet delete the null values
df1 = df.dropna()
df1
```

Out[16]:

	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	KM8K33AGXL	King	Seattle	WA	98103.0	2020	HYUNDAI	KONA	Battery Electric Vehicle (BEV)	Clean Alternative Fuel Vehicle Eligible	258	0	43.0
1	1C4RJYB61N	King	Bothell	WA	98011.0	2022	JEEP	GRAND CHEROKEE	Plug-in Hybrid Electric Vehicle (PHEV)	Not eligible due to low battery range	25	0	1.0
2	1C4RJYD61P	Yakima	Yakima	WA	98908.0	2023	JEEP	GRAND CHEROKEE	Plug-in Hybrid Electric Vehicle (PHEV)	Not eligible due to low battery range	25	0	14.0
3	5YJ3E1EA7J	King	Kirkland	WA	98034.0	2018	TESLA	MODEL 3	Battery Electric Vehicle (BEV)	Clean Alternative Fuel Vehicle Eligible	215	0	45.0
4	WBY7Z8C5XJ	Thurston	Olympia	WA	98501.0	2018	BMW	I3	Plug-in Hybrid Electric Vehicle (PHEV)	Clean Alternative Fuel Vehicle Eligible	97	0	22.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...
150477	WBY43AW05P	Grays Harbor	Montesano	WA	98563.0	2023	BMW	I4	Battery Electric Vehicle (BEV)	Eligibility unknown as battery range has not b...	0	0	19.0
150478	5YJ3E1EB7P	King	Seattle	WA	98104.0	2023	TESLA	MODEL 3	Battery Electric Vehicle (BEV)	Eligibility unknown as battery range has not b...	0	0	43.0
150479	5YJYGDEEXM	King	Seattle	WA	98109.0	2021	TESLA	MODEL Y	Battery Electric Vehicle (BEV)	Eligibility unknown as battery range has not b...	0	0	43.0
150480	5UXTA6C08P	Snohomish	Mountlake Terrace	WA	98043.0	2023	BMW	X5	Plug-in Hybrid Electric Vehicle (PHEV)	Clean Alternative Fuel Vehicle Eligible	30	0	1.0
150481	7SAYGDEF8N	Skagit	Mount Vernon	WA	98273.0	2022	TESLA	MODEL Y	Battery Electric Vehicle (BEV)	Eligibility unknown as battery range has not b...	0	0	40.0

150137 rows × 17 columns

In [17]:

```
# check the null values
df1.isnull().sum()
```

```
Out[17]: 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
```

```
In [18]: df1['County'].unique()
```

```
Out[18]: array(['King', 'Yakima', 'Thurston', 'Snohomish', 'Island', 'Kitsap',
       'Whitman', 'Skagit', 'Kittitas', 'Walla Walla', 'Spokane',
       'Chelan', 'Grant', 'Stevens', 'Clark', 'Douglas', 'Pierce',
       'Jefferson', 'Cowlitz', 'Clallam', 'Benton', 'Lewis', 'Klickitat',
       'Grays Harbor', 'Asotin', 'San Juan', 'Whatcom', 'Franklin',
       'Okanogan', 'Mason', 'Skamania', 'Pacific', 'Pend Oreille',
       'Columbia', 'Wahkiakum', 'Adams', 'Lincoln', 'Ferry', 'Garfield'],
      dtype=object)
```

```
In [19]: df1['City'].unique()
```

```
Out[19]: array(['Seattle', 'Bothell', 'Yakima', 'Kirkland', 'Olympia',
       'Marysville', 'Kent', 'Woodinville', 'Coupeville', 'Bellevue',
       'Port Orchard', 'Mukilteo', 'Redmond', 'Rochester', 'Bremerton',
       'Poulsbo', 'Lake Stevens', 'Silverdale', 'Kingston', 'Enumclaw',
       'Auburn', 'Moxee', 'Tieton', 'Renton', 'Tukwila', 'Seatac',
       'Tumwater', 'Suquamish', 'Oak Harbor', 'Shoreline', 'Seabeck',
       'Monroe', 'Bainbridge Island', 'Yelm', 'Kenmore', 'Stanwood',
       'Medina', 'Snohomish', 'Toppenish', 'Everett', 'Lynnwood',
       'Issaquah', 'Lacey', 'Selah', 'Edmonds', 'Hansville', 'Tenino',
       'Sammamish', 'Woodway', 'Pullman', 'Rainier', 'Mount Vernon',
       'Cle Elum', 'Burlington', 'Burien', 'Anacortes', 'La Conner',
       'Walla Walla', 'Spokane', 'Concrete', 'Mill Creek', 'Freeland',
       'Leavenworth', 'Wenatchee', 'Moses Lake', 'Coulee City',
       'Sedro-Woolley', 'Bow', 'Ellensburg', 'Des Moines', 'Rice',
       'Chewelah', 'Mountlake Terrace', 'Clinton', 'Vashon',
       'South Cle Elum', 'Langley', 'Cashmere', 'Entiat', 'Vancouver',
       'Granite Falls', 'Brier', 'Colville', 'Waterville', 'La Center',
       'Lakewood', 'Brush Prairie', 'Port Townsend', 'North Bend',
       'Normandy Park', 'Woodland', 'Olalla', 'Sequim', 'Pacific',
       'Ariel', 'Longview', 'Nine Mile Falls', 'Brinnon', 'Kalama',
       'Lake Forest Park', 'Washougal', 'Newcastle', 'Camas',
       'Federal Way', 'Duvall', 'Battle Ground', 'Beaux Arts',
       'Port Ludlow', 'Covington', 'Arlington', 'University Place',
       'Mercer Island', 'Silverlake', 'Kelso', 'Ridgefield', 'Tacoma',
       'Steilacoom', 'Cheney', 'Richland', 'Clyde Hill', 'Lakebay',
       'Fall City', 'Sunnyside', 'Castle Rock', 'Chehalis', 'Naches',
       'White Salmon', 'Grandview', 'Ocean Shores', 'Yacolt', 'Amboy',
       'Quilcene', 'Puyallup', 'Aberdeen', 'Sumner', 'Centralia',
       'Darrington', 'Kennewick', 'Nordland', 'Bingen', 'Gig Harbor',
       'Clarkston', 'Milton', 'Ravensdale', 'Camano Island',
       'Bonney Lake', 'Tulalip', 'Lake Tapps', 'Friday Harbor',
       'Port Angeles', 'Indianola', 'Bellingham', 'Granger',
       'Maple Valley', 'Pasco', 'Sultan', 'Fox Island', 'Port Hadlock',
       'Keyport', 'Ephrata', 'Spokane Valley', 'Deer Park', 'Algona',
       'Omak', 'Shelton', 'Joint Base Lewis Mcchord', 'Carnation',
       'Klickitat', 'Mabton', 'Wapato', 'Zillah', 'Amanda Park',
       'Lummi Island', 'Forks', 'Deming', 'Ferndale', 'Snoqualmie',
       'Graham', 'Bucoda', 'Black Diamond', 'Gold Bar', 'Chimacum',
       'Toutle', 'Outlook', 'Cowiche', 'Hunts Point', 'Goldendale',
       'Yarrow Point', 'Skykomish', 'Union Gap', 'Greenbank', 'Belfair',
       'South Hill', 'Mattawa', 'Liberty Lake', 'Roy', 'Mazama', 'Lynden',
       'Fircrest', 'Twisp', 'Winthrop', 'Chelan', 'Warden', 'Naselle',
       'Long Beach', 'East Wenatchee', 'Rockford', 'Winlock',
       'Newman Lake', 'Orting', 'Othello', 'Lopez Island', 'Edgewood',
       'Ocean Park', 'Clearlake', 'Buckley', 'Oroville', 'Neah Bay',
       'West Richland', 'Newport', 'Anderson Island', 'Greenacres',
       'Manson', 'Dayton', 'Cathlamet', 'Eastsound', 'Allyn', 'Mica',
       'Kittitas', 'Seaview', 'Parkland', 'Mineral', 'Eatonville',
       'Orondo', 'Husum', 'Burbank', 'Medical Lake', 'Montesano',
       'Ilwaco', 'Garfield', 'Underwood', 'Mead', 'Blaine', 'Endicott',
       'Roosevelt', 'Carson', 'Toledo', 'Dupont', 'Sedro Woolley',
       'Quincy', 'Spanaway', 'Glacier', 'Colbert', 'Vaughn',
       'College Place', 'Snoqualmie Pass', 'Ashford', 'Vantage', 'Union',
       'Connell', 'Olga', 'Peshastin', 'Murdock', 'Prosser', 'Hoquiam',
       'Vader', 'Palouse', 'South Prairie', 'Bz Corner', 'Lind', 'Index',
       'Ritzville', 'Raymond', 'North Bonneville', 'Mossyrock',
       'Tonasket', 'Packwood', 'Royal City', 'Maple Falls', 'Riverside',
       'Easton', 'Fife', 'Cosmopolis', 'Asotin', 'Grapeview',
       'Valleyford', 'Malaga', 'Reardan', 'Morton', 'Elma',
       'Point Roberts', 'Grayland', 'Silver Creek', 'Brewster',
       'Chattaroy', 'Ethel', 'Airway Heights', 'Rosalia', 'Westport',
       'Everson', 'Stevenson', 'Trout Lake', 'Methow', 'Rock Island',
       'Clallam Bay', 'Kettle Falls', 'Veradale', 'Elk', 'Mccleary',
       'Harrington', 'Mesa', 'Electric City', 'Tahuya', 'Ronald',
       'Glenoma', 'Evans', 'Benton City', 'Rockport', 'Edwall',
       'Hoodsport', 'Roslyn', 'Lilliwaup', 'Fairchild Air Force Base',
       'Onalaska', 'Eltopia', 'Oakville', 'Nooksack', 'Okanogan',
       'Uniontown', 'Pacific Beach', 'Colfax', 'Taholah', 'Lopez Is',
       'Kapowsin', 'Waldron', 'North Cove', 'Wishram', 'Hatton', 'Curtis',
       'Cusick', 'Cinebar', 'Shaw Island', 'Salkum', 'Sumas', 'Waitsburg',
       'Wilbur', 'Snowden', 'Silver Lake', 'Ione', 'Carlton', 'Lamont',
       'Lyle', 'Spangle', 'Otis Orchards', 'Carbonado', 'Orcas Is',
       'Davenport', 'Ruston', 'Baring', 'South Bend', 'Republic',
       'Custer', 'Oysterville', 'Dallesport', 'Usk', 'Surfside',
       'Nahcotta', 'Touchet', 'Soap Lake', 'Seven Bays', 'Pomeroy',
       'Loon Lake', 'Springdale', 'Pateros', 'Nespelem', 'Frances',
       'Menlo', 'Glenwood', 'Randle', 'Copalis Beach', 'Danville',
       'Odessa', 'Valley', 'Tokeland', 'Bridgeport Bar', 'Pe Ell',
       'Lincoln', 'Latah', 'Inchelium', 'Maryhill', 'Hunters',
       'Mansfield', 'Ford', 'Sekiu', 'Curlew', 'Acme', 'Palisades',
       'Coulee Dam', 'Tekoa', 'Chinook', 'Preston', 'Malott', 'Clayton',
       'Marblemount', 'Grand Coulee', 'Deer Harbor', 'Thorp', 'Lyman',
       'Marlin', 'Waverly', 'Centerville', 'McCleary', 'Hartline', 'Addy',
       'Cougar', 'Bell Hill', 'Deer Meadows', 'Wahkiacus', 'Wilkeson',
       'Beaver', 'Satsop', 'Rosburg', 'Startup', 'Alderdale', 'Napavine',
       'Longbranch', 'Pros', 'Orchards', 'Copalis Crossing', 'Skamokawa',
       'Grays River', 'Fruitland', 'Moclips', 'Creston', 'Bay Center',
       'Rich', 'White Swan', 'Sprague', 'Elbe', 'Kenn', 'Lebam',
       'Artondale', 'Stratford', 'St John', 'Port Gamble', 'Bridgeport',
       'Walla Walla Co', 'Lakeview', 'Mcchord Afb', 'Plymouth',
       'Lacrosse', 'Keller', 'Orcas', 'Gifford', 'Quinault', 'Dryden',
       'Incorporated', 'Tumtum', 'Washtucna', 'Colton', 'Hamilton'],
      dtype=object)
```

In [20]:
```
df1['State'].unique()
```

Out[20]:
```
array(['WA'], dtype=object)
```

In [21]:
```
df1['Postal Code'].unique()
```

Out[21]:

```
array([98103., 98011., 98908., 98034., 98501., 98271., 98042., 98072.,
       98239., 98004., 98033., 98367., 98366., 98275., 98052., 98579.,
       98112., 98125., 98311., 98370., 98312., 98108., 98258., 98383.,
       98346., 98506., 98022., 98109., 98006., 98133., 98002., 98936.,
       98947., 98057., 98008., 98188., 98392., 98277., 98012., 98177.,
       98380., 98058., 98272., 98110., 98155., 98597., 98028., 98502.,
       98168., 98292., 98056., 98039., 98122., 98021., 98296., 98948.,
       98144., 98204., 98036., 98029., 98310., 98107., 98117., 98092.,
       98007., 98503., 98903., 98942., 98199., 98146., 98105., 98101.,
       98902., 98516., 98513., 98512., 98115., 98020., 98340., 98119.,
       98178., 98026., 98589., 98074., 98031., 98270., 99163., 98576.,
       98273., 98922., 98059., 98233., 98208., 98166., 98221., 98102.,
       98121., 98257., 99362., 99205., 98087., 98005., 98274., 98203.,
       98237., 98901., 98037., 98249., 98826., 98148., 98801., 98837.,
       99207., 99115., 98284., 98116., 98232., 98926., 98315., 98055.,
       98198., 99167., 99109., 98043., 98236., 98070., 98201., 98943.,
       98260., 98815., 98822., 98683., 98030., 98252., 99114., 98682.,
       98858., 98629., 98498., 98606., 98368., 98662., 98045., 98685.,
       98686., 98674., 98359., 98106., 98382., 98047., 98603., 98001.,
       98632., 99026., 98104., 98075., 98027., 98320., 98625., 98671.,
       98663., 98607., 98023., 98019., 98604., 98684., 98660., 98365.,
       98223., 98664., 98467., 98337., 98040., 98661., 98645., 98003.,
       98626., 98642., 98290., 98407., 98053., 98409., 98388., 98439.,
       98665., 99004., 99354., 99223., 98349., 98024., 98944., 98134.,
       98466., 98611., 98532., 98937., 99212., 98672., 98930., 98569.,
       98675., 98601., 98376., 98371., 98520., 98390., 98531., 98241.,
       99336., 98358., 98605., 98374., 98335., 99403., 98354., 98051.,
       98282., 98391., 98250., 98362., 98342., 98329., 98225., 98932.,
       98038., 99301., 98226., 98294., 98333., 98339., 98404., 98363.,
       98345., 98823., 99027., 98332., 98405., 99006., 99037., 98032.,
       98841., 98584., 98433., 98014., 98635., 98935., 98951., 98953.,
       99201., 98406., 98526., 99208., 98262., 98499., 98136., 98331.,
       98244., 98248., 98065., 98338., 98530., 98010., 98126., 98251.,
       98325., 98649., 98938., 98118., 98923., 98620., 98077., 98288.,
       98253., 98528., 99202., 98229., 99352., 99349., 99216., 99019.,
       98580., 98833., 99338., 98264., 98445., 98856., 98862., 98816.,
       98857., 98638., 98631., 98802., 98444., 99218., 99030., 98596.,
       98373., 99025., 98360., 99344., 98261., 98372., 99206., 98640.,
       98235., 98321., 98844., 99224., 98357., 99353., 99156., 98303.,
       99016., 98831., 99328., 98612., 99203., 98245., 98403., 98524.,
       99204., 99023., 98934., 98644., 98446., 98355., 98328., 98843.,
       99217., 99323., 98563., 98402., 98408., 99022., 98624., 99130.,
       98651., 99021., 98230., 99337., 99125., 99356., 98610., 98591.,
       98327., 98848., 98387., 99005., 98394., 99324., 98068., 98304.,
       98950., 98592., 99326., 98279., 98847., 98465., 98617., 99350.,
       98550., 98593., 99161., 98401., 98385., 98164., 99341., 98256.,
       99169., 98422., 98577., 98639., 98564., 99036., 98855., 98361.,
       99357., 98266., 98849., 98925., 98424., 98537., 99402., 98546.,
       98828., 99029., 98356., 98421., 98541., 98281., 98547., 98585.,
       98812., 99003., 98542., 99001., 98375., 99170., 98595., 98247.,
       98648., 98650., 98834., 98850., 98326., 99141., 99009., 98557.,
       98418., 99134., 99343., 99123., 98588., 98940., 99126., 99320.,
       98283., 99008., 98548., 98941., 98555., 99011., 98438., 98570.,
       99330., 98568., 98314., 98276., 98840., 99113., 98571., 99111.,
       98587., 98344., 98297., 98673., 98538., 99119., 98533., 98286.,
       98582., 98295., 99361., 99185., 99139., 98814., 99017., 99031.,
       98323., 99122., 98224., 98586., 99166., 98240., 98443., 99180.,
       99360., 98158., 98851., 98206., 99347., 99148., 99173., 98846.,
       99155., 98619., 98377., 98641., 98535., 99121., 99159., 99181.,
       98590., 98572., 98336., 99147., 99018., 99138., 99137., 98830.,
       99013., 98381., 99118., 98220., 98845., 99116., 99033., 98614.,
       99110., 98267., 99133., 98243., 98946., 98050., 98263., 98832.,
       99039., 98613., 99135., 99101., 98616., 98670., 98396., 98305.,
       98583., 98643., 98565., 98351., 98536., 98647., 98621., 99179.,
       99129., 98562., 99117., 98527., 98952., 99032., 98504., 98330.,
       98507., 98124., 98853., 99171., 98227., 98813., 99346., 99143.,
       99140., 98280., 99131., 98575., 98411., 98821., 99034., 99371.,
       98255.])
```

In [22]:

```
df1['Model Year'].unique()
```

Out[22]:

```
array([2020, 2022, 2023, 2018, 2017, 2016, 2019, 2013, 2015, 2014, 2021,
       2011, 2012, 2024, 2010, 2008, 2002, 2000, 1998, 2003, 1999, 1997],
      dtype=int64)
```

In [23]:

```
df1['Model'].unique()
```

Out[23]:
```
array(['KONA', 'GRAND CHEROKEE', 'MODEL 3', 'I3', 'PACIFICA', 'MODEL Y',
       'FUSION', 'PRIUS PRIME', 'E-TRON', 'LEAF', 'NIRO', 'X5', 'MODEL S',
       'VOLT', 'RAV4', 'C-MAX', 'E-GOLF', '500', 'BOLT EV', 'ID.4',
       '330E', 'A3', 'FOCUS', 'SORENTO', 'TUCSON', 'MODEL X', 'SPORTAGE',
       'HARDTOP', 'WRANGLER', 'EQ FORTWO', 'X3', 'Q5 E', 'R1S', 'Q5',
       'PRIUS PLUG-IN', 'SOUL', 'XC90', 'CAYENNE', 'IONIQ 5', 'XC60',
       'CLARITY', 'MUSTANG MACH-E', 'R1T', 'OUTLANDER', 'TRANSIT',
       'ESCAPE', 'SOUL EV', 'IX', 'FORTWO', 'F-150', '740E', '530E',
       'SOLTERRA', 'BOLT EUV', 'RAV4 PRIME', 'I4', 'KONA ELECTRIC', 'PS2',
       'EQB-CLASS', 'EV6', 'SANTA FE', 'TAYCAN', 'LYRIQ',
       'EQS-CLASS SEDAN', 'I-PACE', 'I8', 'XC40', 'COUNTRYMAN', 'S60',
       'ELR', 'AVIATOR', 'SPARK', 'IONIQ', 'E-TRON GT', 'G80', 'IONIQ 6',
       'ARIYA', 'GLC-CLASS', 'C40', 'EQS-CLASS SUV', 'GV60', 'Q4', 'A7',
       'E-TRON SPORTBACK', 'AIR', 'B-CLASS', 'NX', 'GLE-CLASS', 'C-CLASS',
       'RZ 450E', 'V60', 'EQE-CLASS SUV', 'OPTIMA', 'A8 E', 'BZ4X',
       'PANAMERA', 'FORTWO ELECTRIC DRIVE', 'SONATA', 'KARMA', 'CT6',
       'ROADSTER', 'CX-90', 'PRIUS', 'ACCORD', 'CROSSTREK', 'S90',
       'CORSAIR', 'RANGE ROVER', 'I-MIEV', 'EDV', 'RS E-TRON GT', '745E',
       'CITY', 'EQE-CLASS SEDAN', 'Q8', 'RANGE ROVER SPORT',
       'TRANSIT CONNECT ELECTRIC', 'S-CLASS', 'RANGER', 'GV70', 'TONALE',
       'WHEEGO', '918', 'FLYING SPUR', '745LE', 'S-10 PICKUP', 'BENTAYGA'],
      dtype=object)
```

In [24]:
```
df1['Make'].unique()
```

Out[24]:
```
array(['HYUNDAI', 'JEEP', 'TESLA', 'BMW', 'CHRYSLER', 'FORD', 'TOYOTA',
       'AUDI', 'NISSAN', 'KIA', 'CHEVROLET', 'VOLKSWAGEN', 'FIAT', 'MINI',
       'SMART', 'RIVIAN', 'VOLVO', 'PORSCHE', 'HONDA', 'MITSUBISHI',
       'SUBARU', 'POLESTAR', 'MERCEDES-BENZ', 'CADILLAC', 'JAGUAR',
       'LINCOLN', 'GENESIS', 'LUCID', 'LEXUS', 'FISKER', 'MAZDA',
       'LAND ROVER', 'TH!NK', 'AZURE DYNAMICS', 'ALFA ROMEO',
       'WHEEGO ELECTRIC CARS', 'BENTLEY'], dtype=object)
```

In [25]:
```
df1['Electric Vehicle Type'].unique()
```

Out[25]:
```
array(['Battery Electric Vehicle (BEV)',
       'Plug-in Hybrid Electric Vehicle (PHEV)'], dtype=object)
```

In [26]:
```
df1[ 'Clean Alternative Fuel Vehicle (CAFV) Eligibility'].unique()
```

Out[26]:
```
array(['Clean Alternative Fuel Vehicle Eligible',
       'Not eligible due to low battery range',
       'Eligibility unknown as battery range has not been researched'],
      dtype=object)
```

In [27]:
```
df1[ 'Electric Range'].unique()
```

Out[27]:
```
array([258,  25, 215,  97, 266,  33, 291,   0,  19, 220, 204,  84,  26,
        13, 249,  38, 103, 322, 125, 330, 222, 239,  73, 259,  75,  20,
       150,  16,  32,  76,  21, 151, 200,  72, 149, 238, 289, 208,  34,
        53,  30,  81, 110, 210,  22,  58,  17,  18,  87,   6,  93, 308,
        14, 107, 293,  35,  47,  23, 153,  68, 270,  42, 233,  31, 234,
        12,  41, 265,  83,  37,  82,  29, 203,  10,  40,  24, 114,  39,
         8, 126, 218,  15, 124, 337, 111,  27, 192, 245,  28, 170,  62,
       100,  48,  56,  95,   9,  36,  11,  51,  57,  59,  74], dtype=int64)
```

In [28]:
```
df1['Base MSRP'].unique()
```

Out[28]:
```
array([     0,  69900,  31950,  52900,  32250,  90700,  54950,  36900,
        59900,  45600,  52650,  39995,  53400,  64950,  33950, 102000,
        75095, 110950,  44100,  81100, 184400,  36800,  43700,  55700,
        34995,  91250, 109000,  98950,  89100,  32995, 845000],
      dtype=int64)
```

In [29]:
```
df1['Legislative District'].unique()
```

Out[29]:
```
array([43.,  1., 14., 45., 22., 38., 47., 10., 48., 26., 21., 20., 46.,
       23., 39., 35., 11., 44., 31., 41., 32., 15., 33.,  2., 36., 37.,
        5., 34., 13.,  9., 40., 16.,  3., 12.,  7., 18., 17., 28., 24.,
       30., 19., 49., 27., 29.,  6.,  8.,  4., 25., 42.])
```

In [30]:
```
df1['DOL Vehicle ID'].unique()
```

Out[30]:
```
array([249675142, 233928502, 229675939, ..., 180705626, 240473950,
       207667589], dtype=int64)
```

In [31]:
```
df1['Vehicle Location'].unique()
```

Out[31]:
```
array(['POINT (-122.34301 47.659185)', 'POINT (-122.20578 47.762405)',
       'POINT (-120.6027202 46.5965625)', 'POINT (-122.209285 47.71124)',
       'POINT (-122.89692 47.043535)', 'POINT (-122.1713847 48.10433)',
       'POINT (-122.111625 47.36078)', 'POINT (-122.151665 47.75855)',
       'POINT (-122.6880708 48.2179983)', 'POINT (-122.201905 47.61385)',
       'POINT (-122.20264 47.6785)', 'POINT (-122.6851642 47.506453)',
       'POINT (-122.639265 47.5373)', 'POINT (-122.299965 47.94171)',
       'POINT (-122.12302 47.67668)', 'POINT (-123.09573 46.82158)',
       'POINT (-122.306935 47.62441)', 'POINT (-122.296385 47.71558)',
       'POINT (-122.6468815 47.6344364)', 'POINT (-122.64177 47.737525)',
       'POINT (-122.65223 47.57192)', 'POINT (-122.326965 47.55015)',
       'POINT (-122.112265 48.0047)', 'POINT (-122.668076 47.665978)',
       'POINT (-122.50156 47.8019)', 'POINT (-122.8874781 47.0519573)',
       'POINT (-121.98953 47.20347)', 'POINT (-122.34848 47.632405)',
       'POINT (-122.16937 47.571015)', 'POINT (-122.34584 47.76726)',
       'POINT (-122.222855 47.305065)', 'POINT (-120.3795117 46.55609)',
       'POINT (-120.75692 46.70383)', 'POINT (-122.21024 47.4797047)',
       'POINT (-122.11832 47.6245)', 'POINT (-122.29179 47.43473)',
       'POINT (-122.55717 47.733415)', 'POINT (-122.67499 48.287295)',
       'POINT (-122.1876761 47.820517)', 'POINT (-122.382425 47.77279)',
       'POINT (-122.847462 47.63836)', 'POINT (-122.1298876 47.4451257)',
       'POINT (-121.972215 47.85674)', 'POINT (-122.5235781 47.6293323)',
       'POINT (-122.3175 47.7578146)', 'POINT (-122.61023 46.94126)',
       'POINT (-122.2504747 47.7617128)', 'POINT (-122.92145 47.045935)',
       'POINT (-122.286465 47.476)', 'POINT (-122.3684051 48.2414921)',
       'POINT (-122.180505 47.500055)', 'POINT (-122.228025 47.61598)',
       'POINT (-122.30839 47.610365)', 'POINT (-122.179458 47.802589)',
       'POINT (-122.15134 47.8851158)', 'POINT (-120.31287 46.37534)',
       'POINT (-122.30823 47.581975)', 'POINT (-122.240535 47.91139)',
       'POINT (-122.3164188 47.819664)', 'POINT (-121.999595 47.547635)',
       'POINT (-122.611365 47.575195)', 'POINT (-122.378895 47.66905)',
       'POINT (-122.37275 47.68968)', 'POINT (-122.1820969 47.3198995)',
       'POINT (-122.147385 47.599975)', 'POINT (-122.8285 47.03646)',
       'POINT (-120.477805 46.553505)', 'POINT (-120.54188 46.654175)',
       'POINT (-122.394185 47.639195)', 'POINT (-122.355145 47.505655)',
       'POINT (-122.319115 47.66132)', 'POINT (-122.33589 47.61099)',
       'POINT (-120.524012 46.5973939)', 'POINT (-122.7510253 47.08149)',
       'POINT (-122.81709 46.988575)', 'POINT (-122.9131017 47.0135926)',
       'POINT (-122.3185 47.67949)', 'POINT (-122.37507 47.80807)',
       'POINT (-122.57781 47.903975)', 'POINT (-122.363815 47.63046)',
       'POINT (-122.234385 47.494545)', 'POINT (-122.335685 47.80372)',
       'POINT (-122.853421 46.8553318)',
       'POINT (-122.0313266 47.6285782)',
       'POINT (-122.2012521 47.3931814)', 'POINT (-122.17673 48.05542)',
       'POINT (-117.17912 46.730885)', 'POINT (-122.6898776 46.8908492)',
       'POINT (-122.338975 48.41333)', 'POINT (-120.938305 47.195355)',
       'POINT (-122.15734 47.487175)', 'POINT (-122.33079 48.474765)',
       'POINT (-122.2247757 47.9156409)', 'POINT (-122.34102 47.46593)',
       'POINT (-122.615305 48.501275)', 'POINT (-122.3223156 47.6402108)',
       'POINT (-122.344125 47.61546)', 'POINT (-122.5135345 48.3829111)',
       'POINT (-118.34332 46.063985)', 'POINT (-117.42694 47.67946)',
       'POINT (-122.2551991 47.8650827)', 'POINT (-122.16085 47.624515)',
       'POINT (-122.32284 48.41653)', 'POINT (-122.213105 47.95479)',
       'POINT (-121.759005 48.5374)', 'POINT (-120.500225 46.6043)',
       'POINT (-122.297265 47.84182)', 'POINT (-122.544745 48.03024)',
       'POINT (-120.6619153 47.5970083)', 'POINT (-122.32863 47.46233)',
       'POINT (-120.32009 47.42255)', 'POINT (-119.2599876 47.1240154)',
       'POINT (-117.39787 47.67936)', 'POINT (-119.28953 47.61327)',
       'POINT (-122.235035 48.503325)', 'POINT (-122.38679 47.56484)',
       'POINT (-122.4410546 48.5620007)', 'POINT (-120.544475 46.994)',
       'POINT (-122.7061052 47.68848)', 'POINT (-122.197 47.43876)',
       'POINT (-122.3219166 47.4013897)',
       'POINT (-118.1659989 48.428418)', 'POINT (-117.722145 48.27719)',
       'POINT (-122.30842 47.78416)', 'POINT (-122.359364 47.9796552)',
       'POINT (-122.46049 47.44873)', 'POINT (-122.20722 47.979935)',
       'POINT (-120.953125 47.18617)', 'POINT (-122.408015 48.03557)',
       'POINT (-120.4687 47.5173233)', 'POINT (-120.209155 47.68998)',
       'POINT (-122.4853873 45.6083347)', 'POINT (-122.199755 47.37483)',
       'POINT (-121.96994 48.08334)', 'POINT (-117.90431 48.547075)',
       'POINT (-122.5146473 45.67862)', 'POINT (-120.071735 47.64645)',
       'POINT (-122.6706246 45.8662548)', 'POINT (-122.547645 47.176685)',
       'POINT (-122.5485715 45.7336587)',
       'POINT (-122.7644197 48.1195874)',
       'POINT (-122.5918493 45.6617058)',
       'POINT (-121.7814012 47.4935316)',
       'POINT (-122.7033368 45.7042923)',
       'POINT (-122.6483953 45.7010427)', 'POINT (-122.73125 45.923065)',
       'POINT (-122.54483 47.41801)', 'POINT (-122.356145 47.52104)',
       'POINT (-123.105015 48.08104)', 'POINT (-122.250215 47.25014)',
       'POINT (-122.358279 45.9897564)',
       'POINT (-122.2849393 47.3384055)',
       'POINT (-122.9379953 46.1372997)',
       'POINT (-117.5967686 47.8204962)', 'POINT (-122.329075 47.6018)',
       'POINT (-122.03309 47.58153)', 'POINT (-122.03646 47.534065)',
       'POINT (-122.9415583 47.6484648)', 'POINT (-122.839465 46.009825)',
       'POINT (-122.3538285 45.5838634)', 'POINT (-122.666325 45.641205)',
       'POINT (-122.405565 45.59009)', 'POINT (-122.363215 47.3084)',
       'POINT (-121.9810747 47.7377962)', 'POINT (-122.53218 45.77945)',
       'POINT (-122.51692 45.6228)', 'POINT (-122.675975 45.630465)',
       'POINT (-122.6872285 47.9281524)', 'POINT (-122.12324 48.19485)',
       'POINT (-122.589388 45.6228731)',
       'POINT (-122.5404512 47.2074166)', 'POINT (-122.635905 47.57097)',
       'POINT (-122.2377542 47.582905)', 'POINT (-122.641835 45.638545)',
       'POINT (-122.76217 46.30967)', 'POINT (-122.31327 47.32329)',
       'POINT (-122.90724 46.14458)', 'POINT (-122.74291 45.818445)',
       'POINT (-122.091505 47.915555)', 'POINT (-122.5113356 47.2923828)',
       'POINT (-122.0370611 47.6887161)', 'POINT (-122.47913 47.2198)',
       'POINT (-122.5970685 47.1686632)', 'POINT (-122.529685 47.127165)',
       'POINT (-122.66592 45.678565)', 'POINT (-117.57579 47.492775)',
       'POINT (-119.28753 46.29747)', 'POINT (-117.369705 47.62637)',
       'POINT (-122.7658848 47.2739449)', 'POINT (-121.8947186 47.56345)',
       'POINT (-120.011515 46.32189)', 'POINT (-122.329815 47.57981)',
       'POINT (-122.537565 47.231645)', 'POINT (-122.90778 46.2744)',
       'POINT (-122.96692 46.66113)', 'POINT (-120.69918 46.7305)',
       'POINT (-117.288055 47.68043)', 'POINT (-121.48347 45.72977)',
       'POINT (-119.91556 46.249215)', 'POINT (-124.1599804 47.0075271)',
       'POINT (-122.4066726 45.8651816)',
       'POINT (-122.4467982 45.9105287)',
       'POINT (-122.8756509 47.823013)', 'POINT (-122.299155 47.19178)',
       'POINT (-123.8206 46.97789)', 'POINT (-122.23825 47.201625)',
       'POINT (-122.962555 46.716875)', 'POINT (-121.604165 48.252975)',
       'POINT (-119.113535 46.204945)', 'POINT (-122.691675 48.05815)',
       'POINT (-121.648859 45.7491397)', 'POINT (-122.275748 47.1395924)',
       'POINT (-122.5835454 47.3234488)', 'POINT (-117.045685 46.4163)',
       'POINT (-122.31774 47.246525)', 'POINT (-121.98104 47.358625)',
       'POINT (-122.5310901 48.2192797)', 'POINT (-122.183805 47.18062)',
       'POINT (-123.0219704 48.5316467)', 'POINT (-123.424865 48.110695)',
       'POINT (-122.5271049 47.7517942)',
       'POINT (-122.6657985 47.383359)', 'POINT (-122.486115 48.761615)',
       'POINT (-120.1934 46.345135)', 'POINT (-122.05191 47.357985)',
       'POINT (-119.0982 46.232395)', 'POINT (-122.45493 48.76809)',
       'POINT (-121.814825 47.868195)', 'POINT (-122.6066806 47.2254086)',
       'POINT (-122.7759694 48.041403)',
       'POINT (-122.4096963 47.2174975)', 'POINT (-123.46101 48.118925)',
       'POINT (-122.6250117 47.7021263)', 'POINT (-119.55513 47.31936)',
       'POINT (-117.10722 47.698585)', 'POINT (-122.589645 47.342345)',
       'POINT (-122.45153 47.251135)', 'POINT (-117.47501 47.956285)',
       'POINT (-117.19651 47.65942)', 'POINT (-122.235475 47.3809)',
       'POINT (-119.529965 48.41147)', 'POINT (-123.105305 47.211085)',
       'POINT (-122.596275 47.097005)', 'POINT (-121.9105947 47.6483005)',
       'POINT (-121.2837218 45.6947484)', 'POINT (-119.996825 46.21362)',
       'POINT (-120.421245 46.44554)', 'POINT (-120.26034 46.404935)',
       'POINT (-117.431895 47.667155)', 'POINT (-122.490985 47.26365)',
       'POINT (-123.9252642 47.4710694)', 'POINT (-117.40725 47.718625)',
       'POINT (-122.6888403 48.7199947)',
       'POINT (-122.5181098 47.1712579)', 'POINT (-122.388675 47.5415)',
       'POINT (-124.2553016 47.80892)', 'POINT (-122.2147543 48.8233229)',
       'POINT (-122.6011039 48.85324)', 'POINT (-121.8740496 47.5345546)',
       'POINT (-122.2953401 47.0763961)',
       'POINT (-122.872497218 46.794825)', 'POINT (-122.00498 47.31187)',
       'POINT (-122.37439 47.54468)', 'POINT (-121.69427 47.858175)',
       'POINT (-122.7811346 48.0162652)',
       'POINT (-122.7331471 46.3243192)', 'POINT (-120.093285 46.331365)',
       'POINT (-122.28339 47.549285)', 'POINT (-120.713895 46.67108)',
       'POINT (-120.82557 45.821475)', 'POINT (-122.07326 47.75425)',
       'POINT (-121.357795 47.708165)', 'POINT (-122.6084879 48.0755577)',
       'POINT (-122.8551647 47.4495785)', 'POINT (-117.383045 47.661935)',
       'POINT (-122.4569227 48.7470973)',
       'POINT (-119.2952071 46.272495)', 'POINT (-119.901675 46.7377996)',
       'POINT (-117.21264 47.6587754)', 'POINT (-117.0923638 47.6643385)',
       'POINT (-122.522985 46.9876)', 'POINT (-120.405315 48.59203)',
       'POINT (-119.1973001 46.1911488)',
       'POINT (-122.4584536 48.9461196)', 'POINT (-122.40872 47.165675)',
       'POINT (-120.12477 48.365635)', 'POINT (-120.1774093 48.4741766)',
       'POINT (-120.015875 47.839895)', 'POINT (-119.044355 46.965685)',
       'POINT (-123.8035716 46.3722551)', 'POINT (-124.05412 46.3466)',
       'POINT (-120.28619 47.417755)', 'POINT (-122.43827 47.153995)',
       'POINT (-117.411805 47.745795)', 'POINT (-117.1298544 47.4529378)',
       'POINT (-122.94 46.491135)', 'POINT (-122.2987976 47.13795)',
       'POINT (-117.06237 47.704775)', 'POINT (-122.2032237 47.0942956)',
       'POINT (-119.168765 46.826175)', 'POINT (-122.9109264 48.5262809)',
       'POINT (-122.28718 47.190465)', 'POINT (-117.2459114 47.6525293)',
       'POINT (-124.04853 46.4972)',
       'POINT (-122.234180327 48.461412932)',
       'POINT (-122.029685 47.161465)', 'POINT (-119.43496 48.935165)',
       'POINT (-117.460225 47.64927)', 'POINT (-124.625235 48.36824)',
       'POINT (-119.3535873 46.2778489)', 'POINT (-117.049135 48.181095)',
       'POINT (-122.68607 47.158295)', 'POINT (-117.1407 47.673675)',
       'POINT (-120.1531 47.885505)', 'POINT (-117.977095 46.320425)',
       'POINT (-123.380635 46.20034)', 'POINT (-117.425265 47.635365)',
       'POINT (-122.907229 48.7016716)', 'POINT (-122.460375 47.26455)',
       'POINT (-122.8386266 47.3819798)', 'POINT (-117.43167 47.64724)',
       'POINT (-117.212705 47.550885)', 'POINT (-120.4197951 46.9842785)',
       'POINT (-124.05586 46.331995)', 'POINT (-122.377475 47.096645)',
       'POINT (-122.1844551 46.7169232)', 'POINT (-122.270565 46.86822)',
       'POINT (-120.19487 47.704215)', 'POINT (-117.35609 47.68401)',
       'POINT (-119.0127002 46.1971863)', 'POINT (-123.60535 46.982215)',
       'POINT (-122.44131 47.25207)', 'POINT (-122.438105 47.20322)',
       'POINT (-117.6773 47.573335)', 'POINT (-124.03897 46.30954)',
       'POINT (-117.14329 47.00897)', 'POINT (-121.5312858 45.7348285)',
       'POINT (-117.35761 47.76885)', 'POINT (-122.74499 48.99505)',
       'POINT (-119.14533 46.187395)', 'POINT (-117.68591 46.929075)',
       'POINT (-120.21683 45.74239)', 'POINT (-121.8239085 45.7428006)',
       'POINT (-122.846675 46.44142)', 'POINT (-122.643815 47.097455)',
       'POINT (-119.8508676 47.2335835)', 'POINT (-122.435115 47.1045)',
       'POINT (-117.4015511 47.8159901)', 'POINT (-122.769655 47.320505)',
       'POINT (-118.38409 46.043895)', 'POINT (-121.3988798 47.392842)',
       'POINT (-122.01843 46.756945)', 'POINT (-119.9899 46.94468)',
       'POINT (-123.06842 47.33529)', 'POINT (-118.860645 46.660455)',
       'POINT (-122.791154 48.6391921)', 'POINT (-120.60008 47.568765)',
       'POINT (-122.53569 47.249445)', 'POINT (-121.1798228 45.6209509)',
       'POINT (-119.769315 46.204995)', 'POINT (-123.8868787 46.9805308)',
       'POINT (-122.954905 46.400235)', 'POINT (-117.07428 46.91147)',
       'POINT (-122.437085 47.25303)', 'POINT (-122.096795 47.137975)',
       'POINT (-122.332645 47.605785)', 'POINT (-118.616395 46.97191)',
       'POINT (-121.5563475 47.8202753)', 'POINT (-118.376715 47.12569)',
       'POINT (-122.38578 47.28971)', 'POINT (-123.72855 46.686115)',
       'POINT (-121.9755204 45.6395805)', 'POINT (-122.48444 46.530395)',
       'POINT (-117.24097 47.532105)', 'POINT (-119.438935 48.70556)',
       'POINT (-121.66915 46.60968)', 'POINT (-119.630615 46.899585)',
       'POINT (-122.1486861 48.9492931)', 'POINT (-119.509725 48.50223)',
       'POINT (-121.1761632 47.24106)', 'POINT (-122.36151 47.241885)',
       'POINT (-123.7758 46.952755)', 'POINT (-117.04437 46.338085)',
       'POINT (-122.9529423 47.3248964)', 'POINT (-120.199045 47.36607)',
       'POINT (-117.877905 47.667915)', 'POINT (-122.27838 46.556285)',
       'POINT (-122.40594 47.261725)', 'POINT (-123.410105 47.00679)',
       'POINT (-123.0651 48.98823)', 'POINT (-124.084295 46.746475)',
       'POINT (-122.5000729 46.5596671)', 'POINT (-119.783035 48.097155)',
       'POINT (-117.357645 47.89189)', 'POINT (-122.71614 46.5308781)',
       'POINT (-117.587785 47.64677)', 'POINT (-122.3085456 47.1042426)',
       'POINT (-117.369475 47.2369)', 'POINT (-124.107315 46.886855)',
       'POINT (-122.357585 48.90942)', 'POINT (-121.8849392 45.6980854)',
       'POINT (-121.5317803 45.9914877)',
       'POINT (-120.0041169 48.1305433)', 'POINT (-120.144655 47.37695)',
       'POINT (-124.2601571 48.2537459)', 'POINT (-118.056575 48.604475)',
       'POINT (-117.34985 48.017715)', 'POINT (-123.26743 47.052405)',
       'POINT (-122.439165 47.22468)', 'POINT (-118.2539532 47.4808)',
       'POINT (-119.1511997 46.592392)',
       'POINT (-119.0352296 47.9315504)', 'POINT (-122.98528 47.438745)',
       'POINT (-121.0551802 47.2555714)',
       'POINT (-117.9557577 48.6879951)', 'POINT (-119.4897 46.26394)',
       'POINT (-121.5988788 48.486229)', 'POINT (-117.95109 47.5052517)',
       'POINT (-123.2021319 47.4168231)', 'POINT (-120.994415 47.222225)',
       'POINT (-123.0686855 47.4998342)',
       'POINT (-117.6673524 47.6263925)', 'POINT (-122.49046 47.12834)',
       'POINT (-122.71921 46.577)', 'POINT (-119.0189476 46.4536241)',
       'POINT (-123.23165 46.8410324)', 'POINT (-122.6498459 47.561889)',
       'POINT (-122.321835 48.927085)', 'POINT (-119.5875704 48.3648226)',
       'POINT (-117.12633 46.566005)', 'POINT (-124.1960702 47.1969484)',
       'POINT (-117.3631029 46.8783751)',
       'POINT (-124.1921647 47.253853)', 'POINT (-122.23469 46.98436)',
       'POINT (-123.0257673 48.7106423)', 'POINT (-120.964478 45.661388)',
       'POINT (-123.1267988 46.567008)',
       'POINT (-117.3446295 48.5624298)',
       'POINT (-122.5278704 46.5863832)',
       'POINT (-122.9631224 48.5845356)', 'POINT (-122.631295 46.52821)',
       'POINT (-122.26024 48.99634)', 'POINT (-118.15324 46.268335)',
       'POINT (-118.70598 47.755985)', 'POINT (-117.42018 48.74178)',
       'POINT (-120.1158629 48.2496045)', 'POINT (-117.90409 47.201595)',
       'POINT (-117.37668 47.42844)', 'POINT (-122.052435 47.07902)',
       'POINT (-118.15232 47.651205)', 'POINT (-121.420466209 47.754705)',
       'POINT (-123.81213 46.66546)', 'POINT (-118.732885 48.64804)',
       'POINT (-122.6410958 48.919121)', 'POINT (-122.364635 47.19449)',
       'POINT (-117.2823094 48.312794)', 'POINT (-118.6712246 46.040628)',
       'POINT (-122.3016997 47.4559666)', 'POINT (-119.485115 47.38686)',
       'POINT (-122.319018929 48.01306)', 'POINT (-117.60089 46.473565)',
       'POINT (-117.63454 48.05954)', 'POINT (-117.74368 48.05388)',
       'POINT (-119.903265 48.05164)', 'POINT (-118.976655 48.166365)',
       'POINT (-121.294825 46.01829)', 'POINT (-121.920025 46.53731)',
       'POINT (-124.028715 46.54778)', 'POINT (-124.17386 47.11043)',
       'POINT (-118.5109334 48.9895316)', 'POINT (-118.689355 47.331045)',
       'POINT (-117.72382 48.174085)', 'POINT (-123.9965452 46.711147)',
       'POINT (-123.2983149 46.570015)',
       'POINT (-122.1228547 46.5337193)',
       'POINT (-118.4270914 47.8271662)', 'POINT (-117.15512 47.28057)',
       'POINT (-118.2012541 48.2960833)',
       'POINT (-118.1989261 48.1169085)', 'POINT (-119.640515 47.81343)',
       'POINT (-117.8112946 47.8321008)',
       'POINT (-124.3719568 48.2843699)',
       'POINT (-118.6027554 48.8784431)', 'POINT (-122.20984 48.65409)',
       'POINT (-119.880648 47.4480973)', 'POINT (-118.95252 47.998715)',
       'POINT (-117.071775 47.22265)', 'POINT (-123.9490486 46.2769106)',
       'POINT (-117.56666 47.99692)', 'POINT (-121.421895 48.55832)',
       'POINT (-119.0018959 47.9419302)',
       'POINT (-122.9947196 48.6236846)',
       'POINT (-120.6732673 47.067819)', 'POINT (-121.9364687 47.52601)',
       'POINT (-122.059765 48.52525)', 'POINT (-118.985845 47.40771)',
       'POINT (-117.227815 47.34298)', 'POINT (-120.90481 45.75199)',
       'POINT (-119.108275 47.68987)', 'POINT (-117.83807 48.35699)',
       'POINT (-122.303465 46.050445)', 'POINT (-121.1276987 45.8669769)',
       'POINT (-122.0527005 47.109146868)',
       'POINT (-124.347425 48.05613)', 'POINT (-123.483025 47.002225)',
       'POINT (-123.6450789 46.3346014)',
       'POINT (-122.9056795 46.57584759)',
       'POINT (-122.7736637 47.1835998)',
       'POINT (-124.1853792 47.1587097)',
       'POINT (-123.4553939 46.2724165)',
       'POINT (-123.5689506 46.3598681)', 'POINT (-117.085225 46.53802)',
       'POINT (-118.2672241 48.0307304)',
       'POINT (-124.064733792 47.337628)', 'POINT (-118.521405 47.75863)',
       'POINT (-123.95362 46.629685)', 'POINT (-120.72956 46.377845)',
       'POINT (-117.97788 47.29733)', 'POINT (-122.89166 47.03956)',
       'POINT (-122.1438091 46.7596404)', 'POINT (-122.89602 47.04058)',
       'POINT (-122.329075 47.581125)',
       'POINT (-119.273660836 47.417432)', 'POINT (-117.582395 47.08821)',
       'POINT (-122.47818 48.751095)', 'POINT (-119.67319 48.007315)',
       'POINT (-119.35775 45.93503)', 'POINT (-117.879945 46.813895)',
       'POINT (-118.6846777 48.0787283)',
       'POINT (-122.9428013 48.5981289)', 'POINT (-118.14369 48.3068152)',
       'POINT (-123.8113708 47.4931571)', 'POINT (-122.473635 47.219875)',
       'POINT (-120.56105 47.541595)', 'POINT (-117.7910437 47.864569)',
       'POINT (-118.3131408 46.7536204)',
       'POINT (-121.9874081 48.5178627)'], dtype=object)
```

In [32]:

```
df1['Electric Utility'].unique()
```

Out[32]:
```
array(['CITY OF SEATTLE - (WA)|CITY OF TACOMA - (WA)',
       'PUGET SOUND ENERGY INC||CITY OF TACOMA - (WA)', 'PACIFICORP',
       'PUGET SOUND ENERGY INC', 'AVISTA CORP',
       'NO KNOWN ELECTRIC UTILITY SERVICE',
       'MODERN ELECTRIC WATER COMPANY', 'PUD NO 1 OF CHELAN COUNTY',
       'PUD NO 2 OF GRANT COUNTY',
       'BONNEVILLE POWER ADMINISTRATION||PUD NO 1 OF CLARK COUNTY - (WA)',
       'PUD NO 1 OF DOUGLAS COUNTY',
       'BONNEVILLE POWER ADMINISTRATION||CITY OF TACOMA - (WA)||PENINSULA LIGHT COMPANY',
       'BONNEVILLE POWER ADMINISTRATION||PUGET SOUND ENERGY INC||PUD NO 1 OF JEFFERSON COUNTY',
       'BONNEVILLE POWER ADMINISTRATION||PUD NO 1 OF COWLITZ COUNTY',
       'BONNEVILLE POWER ADMINISTRATION||PUD NO 1 OF MASON COUNTY|PUD NO 1 OF JEFFERSON COUNTY',
       'BONNEVILLE POWER ADMINISTRATION||PUD NO 1 OF CLALLAM COUNTY',
       'BONNEVILLE POWER ADMINISTRATION||PACIFICORP||PUD NO 1 OF CLARK COUNTY - (WA)',
       'BONNEVILLE POWER ADMINISTRATION||TOWN OF STEILACOOM|CITY OF TACOMA - (WA)||PENINSULA LIGHT COMPANY',
       'BONNEVILLE POWER ADMINISTRATION||AVISTA CORP||INLAND POWER & LIGHT COMPANY',
       'BONNEVILLE POWER ADMINISTRATION||CITY OF RICHLAND - (WA)',
       'CITY OF TACOMA - (WA)||TANNER ELECTRIC COOP',
       'BONNEVILLE POWER ADMINISTRATION||PACIFICORP||BENTON RURAL ELECTRIC ASSN',
       'BONNEVILLE POWER ADMINISTRATION||PUD 1 OF SNOHOMISH COUNTY',
       'BONNEVILLE POWER ADMINISTRATION||CITY OF TACOMA - (WA)||PUD NO 1 OF LEWIS COUNTY',
       'BONNEVILLE POWER ADMINISTRATION||VERA IRRIGATION DISTRICT #15',
       'BONNEVILLE POWER ADMINISTRATION||PUD NO 1 OF KLICKITAT COUNTY',
       'BONNEVILLE POWER ADMINISTRATION||PUD NO 1 OF GRAYS HARBOR COUNTY',
       'BONNEVILLE POWER ADMINISTRATION||PUD NO 1 OF BENTON COUNTY',
       'BONNEVILLE POWER ADMINISTRATION||AVISTA CORP||PUD NO 1 OF ASOTIN COUNTY',
       'BONNEVILLE POWER ADMINISTRATION||CITY OF MILTON - (WA)|CITY OF TACOMA - (WA)',
       'BONNEVILLE POWER ADMINISTRATION||ORCAS POWER & LIGHT COOP',
       'PUGET SOUND ENERGY INC||PUD NO 1 OF WHATCOM COUNTY',
       'BONNEVILLE POWER ADMINISTRATION||BENTON RURAL ELECTRIC ASSN',
       'BONNEVILLE POWER ADMINISTRATION||PUD NO 1 OF FRANKLIN COUNTY',
       'BONNEVILLE POWER ADMINISTRATION||INLAND POWER & LIGHT COMPANY',
       'PUD NO 1 OF OKANOGAN COUNTY',
       'BONNEVILLE POWER ADMINISTRATION||CITY OF TACOMA - (WA)||PUD NO 3 OF MASON COUNTY',
       'PUD NO 1 OF WHATCOM COUNTY',
       'BONNEVILLE POWER ADMINISTRATION||CITY OF TACOMA - (WA)||LAKEVIEW LIGHT & POWER|PENINSULA LIGHT COMPANY'
,
       'BONNEVILLE POWER ADMINISTRATION||CITY OF PORT ANGELES - (WA)',
       'CITY OF TACOMA - (WA)',
       'BONNEVILLE POWER ADMINISTRATION||PUD NO 1 OF SKAMANIA CO',
       'CITY OF CHEWELAH', 'CITY OF SEATTLE - (WA)',
       'BONNEVILLE POWER ADMINISTRATION||CITY OF ELLENSBURG - (WA)',
       'OKANOGAN COUNTY ELEC COOP, INC',
       'BONNEVILLE POWER ADMINISTRATION||CITY OF TACOMA - (WA)||PARKLAND LIGHT & WATER COMPANY|PENINSULA LIGHT
COMPANY',
       'BONNEVILLE POWER ADMINISTRATION||PUD NO 2 OF PACIFIC COUNTY',
       'BONNEVILLE POWER ADMINISTRATION||AVISTA CORP||BIG BEND ELECTRIC COOP, INC',
       'PUD NO 1 OF PEND OREILLE COUNTY',
       'BONNEVILLE POWER ADMINISTRATION||PUD NO 1 OF WAHKIAKUM COUNTY',
       'BONNEVILLE POWER ADMINISTRATION||CITY OF TACOMA - (WA)||ELMHURST MUTUAL POWER & LIGHT CO|PENINSULA LIGH
T COMPANY',
       'BONNEVILLE POWER ADMINISTRATION||CITY OF CENTRALIA - (WA)|CITY OF TACOMA - (WA)',
       'BONNEVILLE POWER ADMINISTRATION||TOWN OF EATONVILLE - (WA)|CITY OF TACOMA - (WA)',
       'BONNEVILLE POWER ADMINISTRATION||CITY OF TACOMA - (WA)||ALDER MUTUAL LIGHT CO, INC|PENINSULA LIGHT COMP
ANY',
       'BONNEVILLE POWER ADMINISTRATION||COLUMBIA RURAL ELEC ASSN, INC',
       'CITY OF BLAINE - (WA)||PUD NO 1 OF WHATCOM COUNTY',
       'CITY OF CHENEY - (WA)',
       'BONNEVILLE POWER ADMINISTRATION||PUD NO 1 OF KITTITAS COUNTY',
       'BONNEVILLE POWER ADMINISTRATION||PACIFICORP||COLUMBIA RURAL ELEC ASSN, INC',
       'BONNEVILLE POWER ADMINISTRATION||CITY OF TACOMA - (WA)||PUD NO 1 OF MASON COUNTY',
       'BONNEVILLE POWER ADMINISTRATION||BIG BEND ELECTRIC COOP, INC',
       'PORTLAND GENERAL ELECTRIC CO',
       'BONNEVILLE POWER ADMINISTRATION||CITY OF TACOMA - (WA)||OHOP MUTUAL LIGHT COMPANY, INC|PENINSULA LIGHT
COMPANY',
       'BONNEVILLE POWER ADMINISTRATION||CITY OF MCCLEARY - (WA)',
       'BONNEVILLE POWER ADMINISTRATION||CITY OF TACOMA - (WA)||BENTON RURAL ELECTRIC ASSN|PENINSULA LIGHT COMP
ANY',
       'CITY OF SUMAS - (WA)||PUD NO 1 OF WHATCOM COUNTY',
       'BONNEVILLE POWER ADMINISTRATION||PENINSULA LIGHT COMPANY',
       'BONNEVILLE POWER ADMINISTRATION||TOWN OF RUSTON - (WA)|CITY OF TACOMA - (WA)||PENINSULA LIGHT COMPANY',
       'BONNEVILLE POWER ADMINISTRATION||PUD NO 1 OF ASOTIN COUNTY',
       'BONNEVILLE POWER ADMINISTRATION||PUD NO 1 OF FERRY COUNTY',
       'BONNEVILLE POWER ADMINISTRATION||NESPELEM VALLEY ELEC COOP, INC',
       'BONNEVILLE POWER ADMINISTRATION||CITY OF COULEE DAM - (WA)',
       'BONNEVILLE POWER ADMINISTRATION||PUD NO 1 OF JEFFERSON COUNTY',
       'BONNEVILLE POWER ADMINISTRATION||PUD NO 1 OF ASOTIN COUNTY||INLAND POWER & LIGHT COMPANY'],
      dtype=object)
```

In [33]:
```
df1['2020 Census Tract'].unique()
```

Out[33]:
```
array([5.30330048e+10, 5.30330218e+10, 5.30770029e+10, ...,
       5.30770015e+10, 5.30219801e+10, 5.30019503e+10])
```

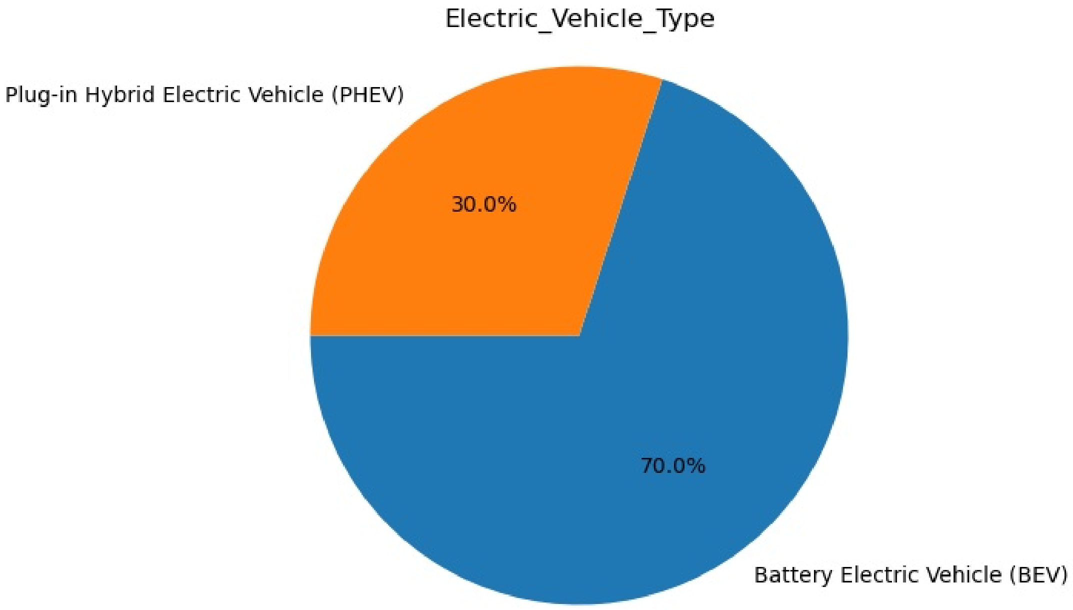
In [34]:
```
# total values count for taking ratio
df["Electric Vehicle Type"].value_counts()
```

Out[34]:

```
Electric Vehicle Type
Battery Electric Vehicle (BEV)            116807
Plug-in Hybrid Electric Vehicle (PHEV)     33675
Name: count, dtype: int64
```

In [35]:

```
# pie charts
Electric_Vehicle_Type = ['Battery Electric Vehicle (BEV)',
       'Plug-in Hybrid Electric Vehicle (PHEV)']
percentages = [70, 30]  # These should add up to 100%

# Create a pie chart
plt.figure(figsize=(10, 5))
plt.pie(percentages, labels=Electric_Vehicle_Type, autopct='%1.1f%%', startangle=180)

# Add a title
plt.title("Electric_Vehicle_Type")

# Display the pie chart
plt.axis('equal')  # Equal aspect ratio ensures that the pie is drawn as a circle.
plt.show()
```



In [36]:

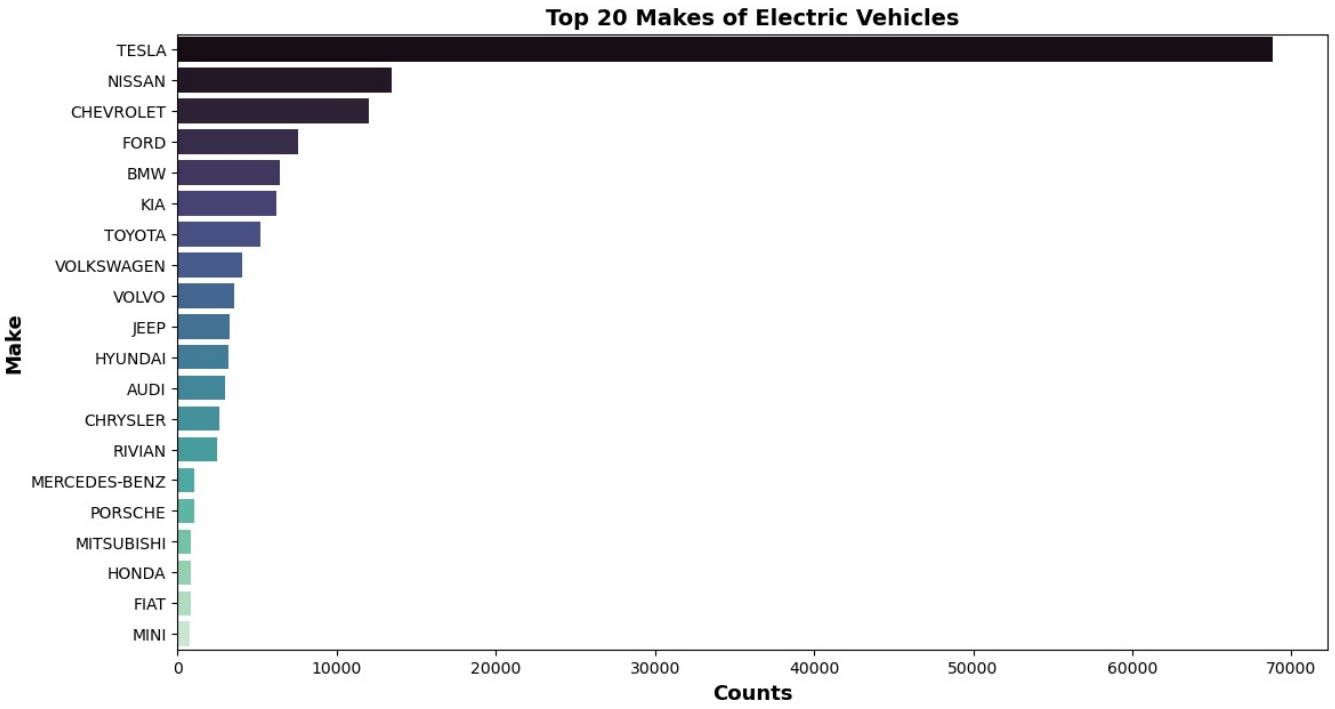
```
df["Make"].value_counts()
```

Out[36]:

```
Make
TESLA                   68983
NISSAN                  13497
CHEVROLET               12026
FORD                     7614
BMW                      6439
KIA                      6198
TOYOTA                   5223
VOLKSWAGEN               4074
VOLVO                    3536
JEEP                     3292
HYUNDAI                  3171
AUDI                     3006
CHRYSLER                 2642
RIVIAN                   2483
MERCEDES-BENZ            1054
PORSCHE                  1028
MITSUBISHI                849
HONDA                     810
FIAT                      806
MINI                      791
POLESTAR                  764
SUBARU                    623
SMART                     274
LINCOLN                   229
JAGUAR                    228
LUCID                     190
CADILLAC                  176
LEXUS                     138
MAZDA                     136
GENESIS                   107
LAND ROVER                 47
FISKER                     17
ALFA ROMEO                 12
AZURE DYNAMICS              9
TH!NK                       5
WHEEGO ELECTRIC CARS        3
BENTLEY                     2
Name: count, dtype: int64
```

In [37]:

```python
# Bar Plot
make_counts = df1['Make'].value_counts().nlargest(20)
plt.figure(figsize=(13, 7))
sns.barplot(x=make_counts.values, y=make_counts.index, palette="mako")
plt.title('Top 20 Makes of Electric Vehicles', fontweight='bold', fontsize=14)
plt.xlabel('Counts', fontweight='bold', fontsize=13)
plt.ylabel('Make', fontweight='bold', fontsize=13)
plt.show()
```



In [38]:

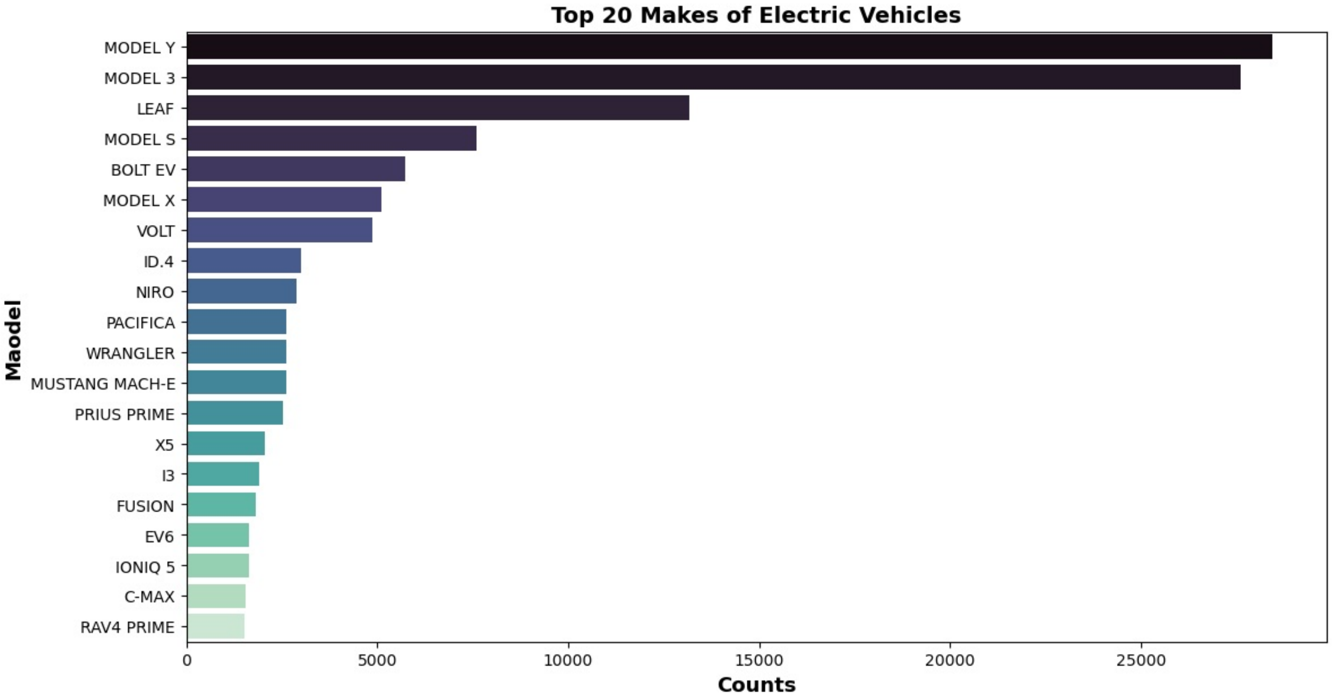
```python
# top Ev Maker
top_makes = ["TESLA", "NISSAN", "CHEVROLET", "FORD", "BMW", "KIA", "TOYOTA", "VOLKSWAGEN", "VOLVO", "JEEP"]

make_evtype = df1[df1["Make"].isin(top_makes)]

top_make_evtype = make_evtype.groupby(["Electric Vehicle Type", "Make"]).size().reset_index(name="Count")

top_make_evtype
```

Out[38]:

	Electric Vehicle Type	Make	Count
0	Battery Electric Vehicle (BEV)	BMW	1793
1	Battery Electric Vehicle (BEV)	CHEVROLET	7127
2	Battery Electric Vehicle (BEV)	FORD	3902
3	Battery Electric Vehicle (BEV)	KIA	4143
4	Battery Electric Vehicle (BEV)	NISSAN	13481
5	Battery Electric Vehicle (BEV)	TESLA	68821
6	Battery Electric Vehicle (BEV)	TOYOTA	188
7	Battery Electric Vehicle (BEV)	VOLKSWAGEN	4064
8	Battery Electric Vehicle (BEV)	VOLVO	1181
9	Plug-in Hybrid Electric Vehicle (PHEV)	BMW	4633
10	Plug-in Hybrid Electric Vehicle (PHEV)	CHEVROLET	4876
11	Plug-in Hybrid Electric Vehicle (PHEV)	FORD	3690
12	Plug-in Hybrid Electric Vehicle (PHEV)	JEEP	3283
13	Plug-in Hybrid Electric Vehicle (PHEV)	KIA	2043
14	Plug-in Hybrid Electric Vehicle (PHEV)	TOYOTA	5007
15	Plug-in Hybrid Electric Vehicle (PHEV)	VOLVO	2348

In [39]:

```
model_counts = df1['Model'].value_counts().nlargest(20)
plt.figure(figsize=(13, 7))
sns.barplot(x=model_counts.values, y=model_counts.index, palette="mako")
plt.title('Top 20 Makes of Electric Vehicles', fontweight='bold', fontsize=14)
plt.xlabel('Counts', fontweight='bold', fontsize=13)
plt.ylabel('Maodel', fontweight='bold', fontsize=13)
plt.show()
```



In [40]:

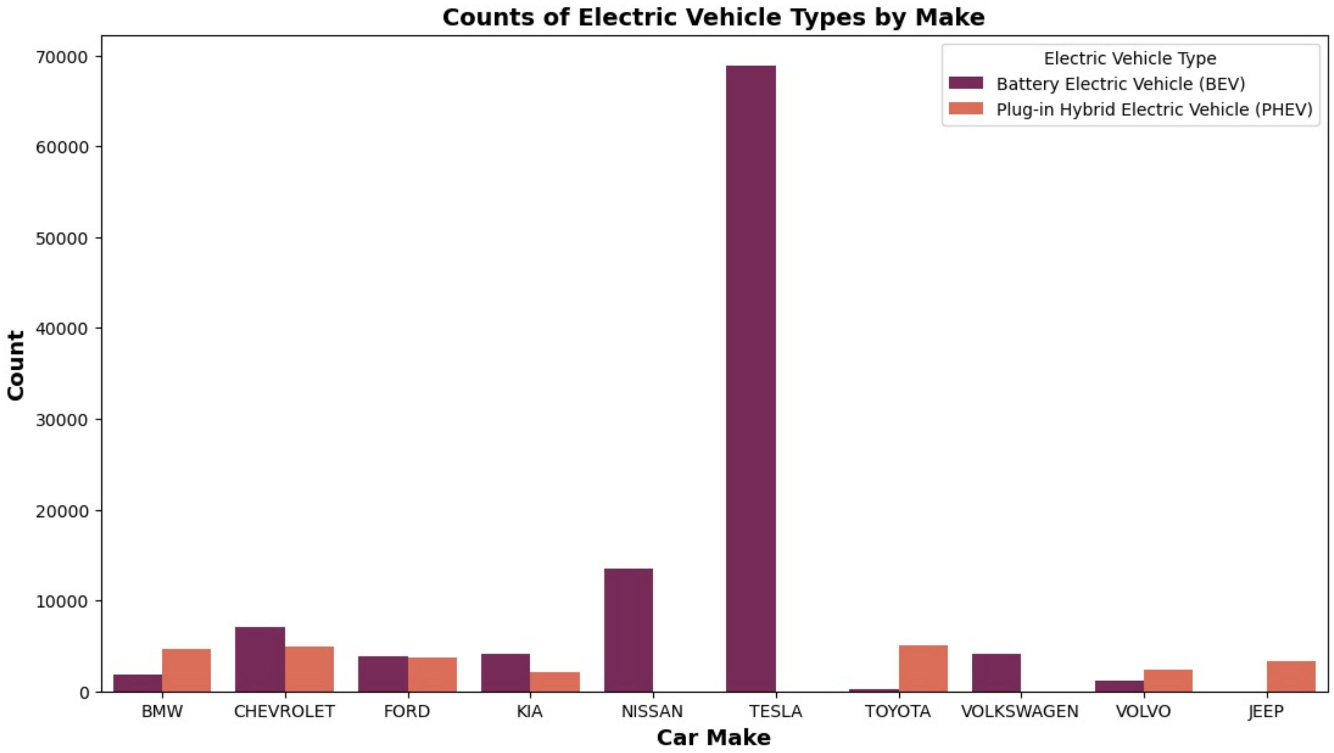
```
top_model = ['MODEL Y','MODEL 3','LEAF','MODEL S','BOLT EV', 'MODEL X','VOLT','ID.4','NIRO','PACIDICA']

model_evtype = df1[df1["Model"].isin(top_model)]

top_model_evtype = model_evtype.groupby(["Electric Vehicle Type", "Model"]).size().reset_index(name="Count")

top_model_evtype
```

Out[40]:

	Electric Vehicle Type	Model	Count
0	Battery Electric Vehicle (BEV)	BOLT EV	5726
1	Battery Electric Vehicle (BEV)	ID.4	2991
2	Battery Electric Vehicle (BEV)	LEAF	13171
3	Battery Electric Vehicle (BEV)	MODEL 3	27626
4	Battery Electric Vehicle (BEV)	MODEL S	7592
5	Battery Electric Vehicle (BEV)	MODEL X	5101
6	Battery Electric Vehicle (BEV)	MODEL Y	28456
7	Battery Electric Vehicle (BEV)	NIRO	1851
8	Plug-in Hybrid Electric Vehicle (PHEV)	NIRO	1018
9	Plug-in Hybrid Electric Vehicle (PHEV)	VOLT	4876

In [41]:

```python
plt.figure(figsize = (13,7))
sns.barplot(data=top_make_evtype, x="Make", y="Count", hue="Electric Vehicle Type", palette="rocket")
plt.xlabel("Car Make", fontsize=13, fontweight="bold")
plt.ylabel("Count", fontsize=13, fontweight="bold")
plt.title("Counts of Electric Vehicle Types by Make", fontsize=14, fontweight="bold")
plt.legend(title='Electric Vehicle Type', loc="upper right")
plt.show()
```



In [42]:

```python
df1["Model"].value_counts()
```

Out[42]:

```
Model
MODEL Y         28456
MODEL 3         27626
LEAF            13171
MODEL S          7592
BOLT EV          5726
                ...  
745LE               2
918                 1
FLYING SPUR         1
S-10 PICKUP         1
BENTAYGA            1
Name: count, Length: 127, dtype: int64
```

In [43]:

```python
df1["County"].value_counts()
```

Out[43]:

```
County
King            79075
Snohomish       17307
Pierce          11542
Clark            8849
Thurston         5403
Kitsap           4923
Spokane          3690
Whatcom          3668
Benton           1800
Skagit           1658
Island           1640
Clallam           920
Jefferson         875
Chelan            863
San Juan          844
Yakima            835
Cowlitz           762
Mason             706
Lewis             598
Grays Harbor      533
Franklin          511
Kittitas          503
Grant             452
Walla Walla       360
Douglas           294
Whitman           262
Klickitat         239
Okanogan          211
Stevens           193
Pacific           173
Skamania          164
Asotin             63
Wahkiakum          49
Pend Oreille       47
Adams              41
Lincoln            39
Ferry              24
Columbia           18
Garfield            3
Name: count, dtype: int64
```

In [44]:

```
df1["Model Year"].value_counts()
```

Out[44]:

```
Model Year
2023    37052
2022    27733
2021    18610
2018    14407
2020    11263
2019    10693
2017     8549
2016     5632
2015     4919
2013     4555
2014     3598
2012     1630
2011      795
2024      642
2010       24
2008       18
2000        8
1999        4
2002        2
1998        1
2003        1
1997        1
Name: count, dtype: int64
```

In [45]:

```
df["State"].value_counts()
```

Out[45]:

```
State
WA    150141
CA        92
VA        35
MD        33
TX        20
NC        13
IL        12
AZ        11
CO        11
FL         9
NJ         9
HI         8
OR         8
NY         7
CT         7
SC         6
NV         6
GA         5
DC         5
MO         4
MA         3
NE         3
LA         3
KY         3
AL         3
WY         2
KS         2
OH         2
IN         2
BC         2
AR         2
ID         2
UT         2
PA         2
AP         1
AK         1
MS         1
NH         1
DE         1
MT         1
MN         1
Name: count, dtype: int64
```

In [46]:

```
df["Base MSRP"].value_counts()
```

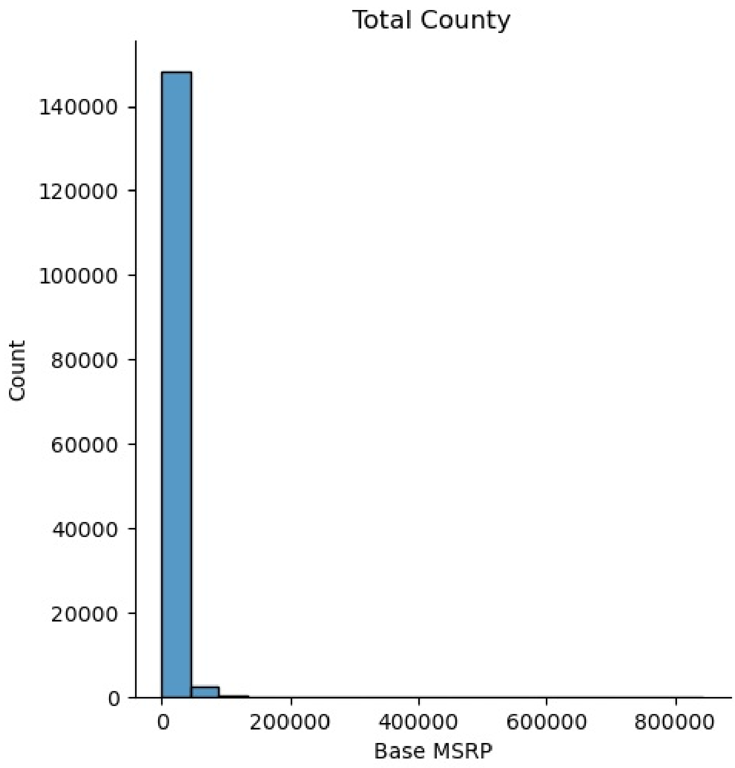
Out[46]:

```
Base MSRP
0         147027
69900       1441
31950        399
52900        218
32250        152
59900        134
54950        133
39995        117
36900        101
44100         97
45600         84
64950         83
33950         74
52650         68
34995         64
36800         53
55700         50
53400         34
110950        21
90700         19
98950         19
81100         19
102000        17
75095         15
184400        11
43700         10
109000         7
89100          6
91250          5
32995          3
845000         1
Name: count, dtype: int64
```

In [47]:

```
plt.figure(figsize=(13,5))
sns.displot(df['Base MSRP'])
plt.title("Total County")
plt.show()
```

```
C:\Users\Admin\anaconda3\Lib\site-packages\seaborn\axisgrid.py:118: UserWarning: The figure layout has changed
to tight
  self._figure.tight_layout(*args, **kwargs)
<Figure size 1300x500 with 0 Axes>
```



In [49]:

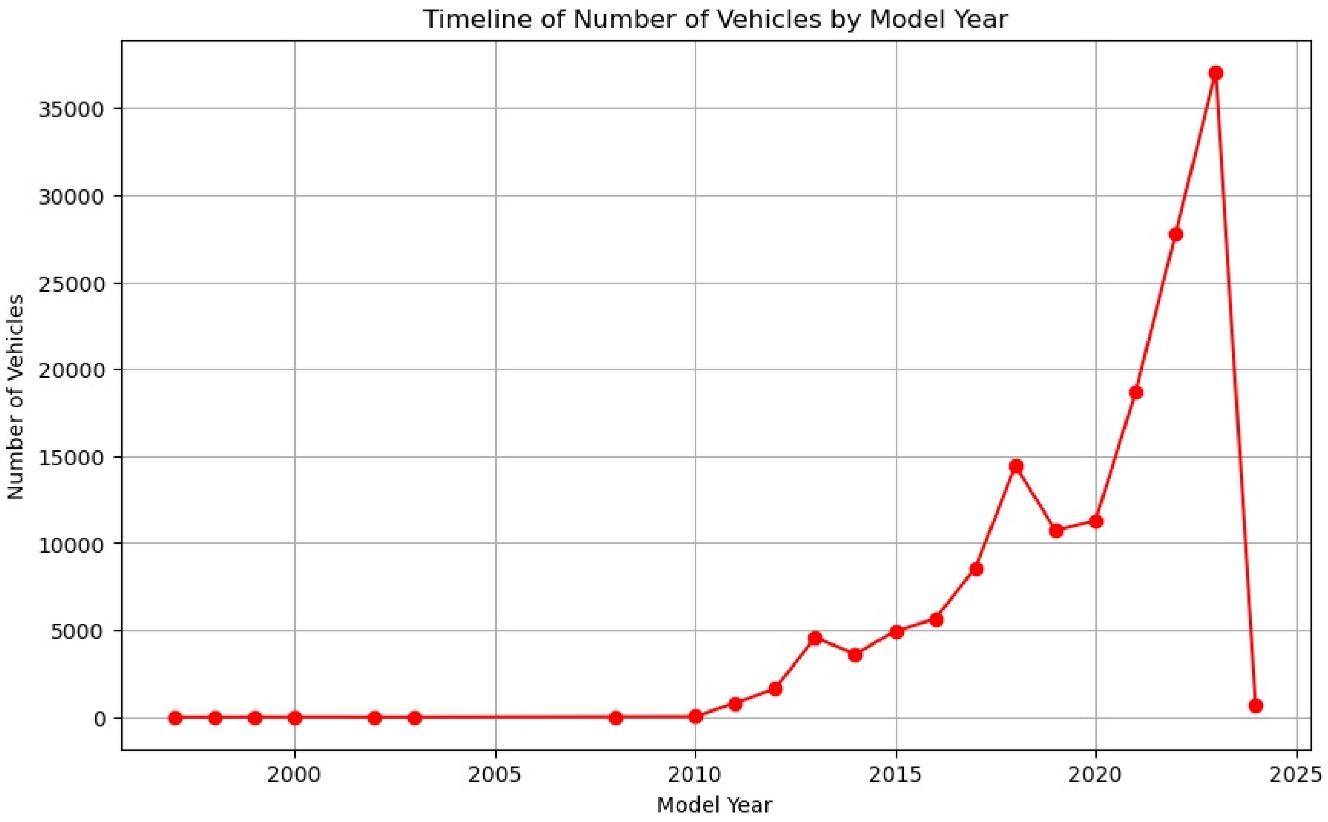
```
counties = ["King", "Snohomish", "Pierce", "Clark", "Thurston"]
makes1 = ["TESLA", "NISSAN", "CHEVROLET", "FORD", "BMW"]

county_makes = df1[(df1["County"].isin(counties) & df1["Make"].isin(makes1))]
top5_county_makes = county_makes.groupby(["County", "Make"]).size().reset_index(name="Count")

top5_county_makes
```

Out[49]:

	County	Make	Count
0	Clark	BMW	347
1	Clark	CHEVROLET	781
2	Clark	FORD	530
3	Clark	NISSAN	803
4	Clark	TESLA	3605
5	King	BMW	3805
6	King	CHEVROLET	4888
7	King	FORD	2897
8	King	NISSAN	6590
9	King	TESLA	39519
10	Pierce	BMW	487
11	Pierce	CHEVROLET	991
12	Pierce	FORD	838
13	Pierce	NISSAN	945
14	Pierce	TESLA	4902
15	Snohomish	BMW	630
16	Snohomish	CHEVROLET	1120
17	Snohomish	FORD	927
18	Snohomish	NISSAN	1552
19	Snohomish	TESLA	9137
20	Thurston	BMW	147
21	Thurston	CHEVROLET	859
22	Thurston	FORD	455
23	Thurston	NISSAN	560
24	Thurston	TESLA	1716

In [50]:

```
plt.figure(figsize=(10, 6))
df['Model Year'].value_counts().sort_index().plot(kind='line', marker='o', color='r')
plt.xlabel('Model Year')
plt.ylabel('Number of Vehicles')
plt.title('Timeline of Number of Vehicles by Model Year')
plt.grid(True)
plt.show()
```



In [55]:

```
df.sample(20).T
```

Out[55]:

	88341	41380	113710	72377	211	103865	61402	113656	
VIN (1-10)	1G1RB6E41C	7SAYGAEE4N	5YJ3E1EA0L	WBY8P2C56K	WBY1Z2C53E	5YJSA1E54N	5YJSA1H15E	5YJ3E1EB5M	
County	King	King	Snohomish	King	Thurston	Chelan	King	Snohomish	
City	Renton	Bellevue	Edmonds	Seattle	Rainier	Leavenworth	Sammamish	Bothell	
State	WA	WA	WA	WA	WA	WA	WA	WA	
Postal Code	98056.0	98006.0	98020.0	98105.0	98576.0	98826.0	98074.0	98012.0	
Model Year	2012	2022	2020	2019	2014	2022	2014	2021	
Make	CHEVROLET	TESLA	TESLA	BMW	BMW	TESLA	TESLA	TESLA	
Model	VOLT	MODEL Y	MODEL 3	I3	I3	MODEL S	MODEL S	MODEL 3	
Electric Vehicle Type	Plug-in Hybrid Electric Vehicle (PHEV)	Battery Electric Vehicle (BEV)	Battery Electric Vehicle (BEV)	Battery Electric Vehicle (BEV)	Battery Electric Vehicle (BEV)	Battery Electric Vehicle (BEV)	Battery Electric Vehicle (BEV)	Battery Electric Vehicle (BEV)	Batt
Clean Alternative Fuel Vehicle (CAFV) Eligibility	Clean Alternative Fuel Vehicle Eligible	Eligibility unknown as battery range has not b...	Clean Alternative Fuel Vehicle Eligible	Clean Alternative Fuel Vehicle Eligible	Clean Alternative Fuel Vehicle Eligible	Eligibility unknown as battery range has not b...	Clean Alternative Fuel Vehicle Eligible	Eligibility unknown as battery range has not b...	Eli batter
Electric Range	35	0	266	153	81	0	208	0	
Base MSRP	0	0	0	0	0	0	69900	0	
Legislative District	41.0	41.0	32.0	43.0	20.0	12.0	45.0	1.0	
DOL Vehicle ID	194393356	192509138	102865528	170443941	121918471	186727992	124151989	148324242	
Vehicle Location	POINT (-122.180505 47.500055)	POINT (-122.16937 47.571015)	POINT (-122.37507 47.80807)	POINT (-122.319115 47.66132)	POINT (-122.6898776 46.8908492)	POINT (-120.6619153 47.5970083)	POINT (-122.0313266 47.6285782)	POINT (-122.1876761 47.820517)	Po
Electric Utility	PUGET SOUND ENERGY INC||CITY OF TACOMA - (WA)	PUGET SOUND ENERGY INC||CITY OF TACOMA - (WA)	PUGET SOUND ENERGY INC	CITY OF SEATTLE - (WA)|CITY OF TACOMA - (WA)	PUGET SOUND ENERGY INC	PUD NO 1 OF CHELAN COUNTY	PUGET SOUND ENERGY INC||CITY OF TACOMA - (WA)	PUGET SOUND ENERGY INC	BON ADMI
2020 Census Tract	53033025304.0	53033024901.0	53061050800.0	53033004401.0	53067012530.0	53007960203.0	53033032316.0	53061052107.0	

In [60]:

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

Out[60]:

	121864	41512	144538	120124	55691
VIN (1-10)	1N4AZ0CP9F	JN1AZ0CP2B	7PDSGABA4P	5YJ3E1EA4J	5YJ3E1EA6P
County	San Juan	King	King	Snohomish	Snohomish
City	Friday Harbor	Renton	North Bend	Lynnwood	Mukilteo
State	WA	WA	WA	WA	WA
Postal Code	98250.0	98057.0	98045.0	98037.0	98275.0
Model Year	2015	2011	2023	2018	2023
Make	NISSAN	NISSAN	RIVIAN	TESLA	TESLA
Model	LEAF	LEAF	R1S	MODEL 3	MODEL 3
Electric Vehicle Type	Battery Electric Vehicle (BEV)	Battery Electric Vehicle (BEV)	Battery Electric Vehicle (BEV)	Battery Electric Vehicle (BEV)	Battery Electric Vehicle (BEV)
Clean Alternative Fuel Vehicle (CAFV) Eligibility	Clean Alternative Fuel Vehicle Eligible	Clean Alternative Fuel Vehicle Eligible	Eligibility unknown as battery range has not b...	Clean Alternative Fuel Vehicle Eligible	Eligibility unknown as battery range has not b...
Electric Range	84	73	0	215	0
Base MSRP	0	0	0	0	0
Legislative District	40.0	37.0	5.0	21.0	21.0
DOL Vehicle ID	227109627	157932477	244188339	308961656	249700839
Vehicle Location	POINT (-123.0219704 48.5316467)	POINT (-122.21024 47.4797047)	POINT (-121.7814012 47.4935316)	POINT (-122.297265 47.84182)	POINT (-122.299965 47.94171)
Electric Utility	BONNEVILLE POWER ADMINISTRATION||ORCAS POWER &...	PUGET SOUND ENERGY INC||CITY OF TACOMA - (WA)	CITY OF TACOMA - (WA)||TANNER ELECTRIC COOP	PUGET SOUND ENERGY INC	PUGET SOUND ENERGY INC
2020 Census Tract	53055960301.0	53033025302.0	53033032800.0	53061050200.0	53061042005.0

```
In [ ]:
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js