Annie Phan

Exploratory Data Analysis (EDA) of Electric Vehicle Adoption in Washington State

Step One: Understand Problem and Data

The dataset used in this analysis was sourced from the U.S. Government's open data portal and contains detailed records of Battery Electric Vehicles (BEVs) and Plug-in Hybrid Electric Vehicles (PHEVs) registered with the Washington State Department of Licensing (DOL). Each record represents a unique registered vehicle, providing comprehensive information on vehicle specifications, registration details, and regional distribution.

Research questions

This analysis investigates patterns of electric vehicle (EV) adoption in Washington State and is guided by the following research questions:

1. What is the distribution of electric vehicle types (e.g., BEV vs. PHEV) across the state?
2. Which manufacturers and models dominate the EV market, and how do these patterns vary geographically?
3. How has EV adoption evolved over time based on model year, vehicle type, and electric range capabilities?
4. Which areas show the fastest growth in EV registrations over time?
5. What relationship, if any, exists between vehicle price and electric range performance? How do electric range and base MSRP vary across manufacturers and models?

Step Two: Import and Inspect Data

This dataset comprises 264,628 records of electric vehicle registrations in Washington State, encompassing 17 variables that capture detailed information on vehicle specifications, ownership, and geographic distribution.

```
# Check data size
df.shape
```

```
(264628, 17)
```

Figure 1. Dataset dimensions (rows and columns)

```
# View top 5 records
df.head()
```

	VIN (1-10)	County	City	State	Postal Code	Model Year	Make	Model	Electric Vehicle Type	Clean Alternative Fuel Vehicle (CAFV) Eligibility	Electric Range	Base MSRP	Legislative District	DOL Vehicle ID	Vehicle Location	Electric Utility	2020 Census Tract
0	WA1E2AFY8R	Thurston	Olympia	WA	98512.00	2024	AUDI	Q5 E	Plug-in Hybrid Electric Vehicle (PHEV)	Not eligible due to low battery range	23.00	0.00	22.00	263239938	POINT (-122.90787 46.9461)	PUGET SOUND ENERGY INC	53067010910.00
1	WAUUPBFF4J	Yakima	Wapato	WA	98951.00	2018	AUDI	A3	Plug-in Hybrid Electric Vehicle (PHEV)	Not eligible due to low battery range	16.00	0.00	15.00	318160860	POINT (-120.42083 46.44779)	PACIFICORP	53077940008.00
2	1N4AZ0CP0F	King	Seattle	WA	98125.00	2015	NISSAN	LEAF	Battery Electric Vehicle (BEV)	Clean Alternative Fuel Vehicle Eligible	84.00	0.00	46.00	184963586	POINT (-122.30253 47.72656)	CITY OF SEATTLE - (WA)|CITY OF TACOMA - (WA)	53033000700.00
3	WA1VAAGE5K	King	Kent	WA	98031.00	2019	AUDI	E-TRON	Battery Electric Vehicle (BEV)	Clean Alternative Fuel Vehicle Eligible	204.00	0.00	11.00	259426821	POINT (-122.17743 47.41185)	PUGET SOUND ENERGY INC|CITY OF TACOMA - (WA)	53033029306.00
4	7SAXCAE57N	Snohomish	Bothell	WA	98021.00	2022	TESLA	MODEL X	Battery Electric Vehicle (BEV)	Eligibility unknown as battery range has not b...	0.00	0.00	1.00	208182236	POINT (-122.18384 47.8031)	PUGET SOUND ENERGY INC	53061051922.00

Figure 2. Sample of data records (first 5 rows)

```
# Electric Vehicle Info
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 264628 entries, 0 to 264627
Data columns (total 17 columns):
 #   Column                                             Non-Null Count   Dtype
---  ------                                             --------------   -----
 0   VIN (1-10)                                         264628 non-null  object
 1   County                                             264619 non-null  object
 2   City                                               264619 non-null  object
 3   State                                              264628 non-null  object
 4   Postal Code                                        264619 non-null  float64
 5   Model Year                                         264628 non-null  int64
 6   Make                                               264628 non-null  object
 7   Model                                              264628 non-null  object
 8   Electric Vehicle Type                              264628 non-null  object
 9   Clean Alternative Fuel Vehicle (CAFV) Eligibility  264628 non-null  object
 10  Electric Range                                     264624 non-null  float64
 11  Base MSRP                                          264624 non-null  float64
 12  Legislative District                               263969 non-null  float64
 13  DOL Vehicle ID                                     264628 non-null  int64
 14  Vehicle Location                                   264611 non-null  object
 15  Electric Utility                                   264619 non-null  object
 16  2020 Census Tract                                  264619 non-null  float64
```

Figure 3. Overview of dataset structure and variable types

The dataset contains seventeen variables: fifteen categorical and two numerical.

Variable	Type	Description
VIN (1-10)	Categorical	First ten characters of the Vehicle Identification Number used for partial vehicle identification
County, City, State, Postal Code	Categorical	Geographic indicators denoting the vehicle's registration location within Washington State
Model Year	Categorical	Year in which the vehicle was manufactured
Make, Model	Categorical	Manufacturer and specific model designation

Electric Vehicle Type	Categorical	Indicates BEV (Battery Electric Vehicle) or PHEV (Plug-in Hybrid Electric Vehicle)
Clean Alternative Fuel Vehicle (CAFV) Eligibility	Categorical	Whether the vehicle qualifies under the Clean Alternative Fuel Vehicle program
Legislative District	Categorical	Washington State legislative district associated with the registration location
DOL Vehicle ID	Categorical	Unique identifier assigned by the Department of Licensing
Vehicle Location	Categorical	Latitude and longitude of the registration address
Electric Utility	Categorical	Electric service provider for the registered location
2020 Census Tract	Categorical	Census tract identifier used for demographic or spatial analysis
Electric Range	Numerical	Maximum distance (miles) the vehicle can travel using only electric power
Base MSRP	Numerical	Manufacturer's suggested retail price for the base model

Table 1. Summary of variables in the dataset

The dataset is largely complete, with missing values concentrated in only a few variables. Most core identifiers and vehicle attributes contain no missing entries. The distribution of missing data is summarized below:

- **No missing values:** VIN, Model Year, Make, Model, Electric Vehicle Type, CAFV Eligibility, DOL Vehicle ID, County
- **Minimal missingness (9 entries each):** City, State, Postal Code, Electric Utility, 2020 Census Tract
- **Very few missing values (4 entries each):** Electric Range, Base MSRP
- **Notable missingness:** Legislative District: 659 missing, Vehicle Location: 17 missing

Overall, the dataset demonstrates a high level of completeness. Most variables contain minimal or no missing values, and only two fields display notable gaps that stand out relative to the rest of the dataset.

```
df.isna().sum()
```

	0
VIN (1-10)	0
County	9
City	9
State	0
Postal Code	9
Model Year	0
Make	0
Model	0
Electric Vehicle Type	0
Clean Alternative Fuel Vehicle (CAFV) Eligibility	0
Electric Range	4
Base MSRP	4
Legislative District	659
DOL Vehicle ID	0
Vehicle Location	17
Electric Utility	9
2020 Census Tract	9

Figure 4. Null value counts

```
# Electric Range
print("Mean = ", df['Electric Range'].mean())
print("Standard Deviation = ", df['Electric Range'].std())
print("Median = ", df['Electric Range'].median())
print("Mode = ", df['Electric Range'].mode())
print("Minimum = ", df['Electric Range'].min())
print("Maximum = ", df['Electric Range'].max())
print("Count = ", df['Electric Range'].count())
print("Length =", len(df['Electric Range']))
```

```
Mean =  41.713159048310054
Standard Deviation =  80.37797717197819
Median =  0.0
Mode =  0   0.00
Name: Electric Range, dtype: float64
Minimum =  0.0
Maximum =  337.0
Count =  264624
Length = 264628
```

Figure 5. Descriptive statistics of the electric range variable

The Electric Range variable exhibits substantial variation and a strongly right-skewed distribution. The mean range is 41.71 miles, while both the median and mode are 0, reflecting the large proportion of vehicles recorded with no electric range. Values span from 0 to 337 miles, and the high standard deviation (80.38 miles) indicates considerable variability among vehicles with nonzero ranges.

A closer breakdown by vehicle type reveals that the majority of zero-range entries originate from Battery Electric Vehicles (BEVs). In total, 163,797 BEVs, approximately 61% of all records and 78% of the BEV subset, are listed with an electric range of 0 miles. Because BEVs operate exclusively on battery power and should always possess a positive electric range, these zero values almost certainly reflect missing or unreported data rather than true vehicle characteristics. This pattern highlights a notable data quality issue within the electric range variable.

```
(df['Electric Range'] == 0).groupby(df['Electric Vehicle Type']).sum()
```

Electric Vehicle Type	Electric Range
Battery Electric Vehicle (BEV)	163797
Plug-in Hybrid Electric Vehicle (PHEV)	0

dtype: int64

```
# Calculate electric range mean by vehicle type
df['Electric Range'].groupby(df['Electric Vehicle Type']).describe()
```

Electric Vehicle Type	count	mean	std	min	25%	50%	75%	max
Battery Electric Vehicle (BEV)	210575.00	44.32	89.63	0.00	0.00	0.00	0.00	337.00
Plug-in Hybrid Electric Vehicle (PHEV)	54049.00	31.56	14.14	1.00	21.00	32.00	38.00	153.00

Figure 6. Distribution of zero electric range across electric vehicle type

```
# Base MSRP
print("Mean = ", df['Base MSRP'].mean())
print("Standard Deviation = ", df['Base MSRP'].std())
print("Median = ", df['Base MSRP'].median())
print("Mode = ", df['Base MSRP'].mode())
print("Minimum = ", df['Base MSRP'].min())
print("Maximum = ", df['Base MSRP'].max())
print("Count = ", df['Base MSRP'].count())
print("Length =", len(df['Base MSRP']))
```

```
Mean =  678.90219707963
Standard Deviation =  6868.919926418542
Median =  0.0
Mode =  0    0.00
Name: Base MSRP, dtype: float64
Minimum =  0.0
Maximum =  845000.0
Count =  264624
Length = 264628
```

Figure 7. Descriptive statistics of the base MSRP variable

The Base MSRP variable displays extreme right-skew and substantial data quality limitations. While the maximum reported MSRP is $845,000, the mean is only $678.90, and both the median and mode are $0, indicating that most records lack valid pricing information. The standard deviation ($6,868.92) is substantially larger than the mean combined with a median of zero indicates that the distribution is dominated by

noninformative or incomplete entries rather than meaningful variation in vehicle pricing.

A closer inspection further confirms the extent of missing data. As shown in Figure 8, 261,476 records, about 98.8% of the dataset, list a Base MSRP of $0. Because a true MSRP of zero is not realistic, these values almost certainly represent unrecorded or unavailable pricing information rather than actual vehicle prices.

```
# Count zero values in Base MSRP
print((df['Base MSRP'] == 0).sum())
print((df['Base MSRP'] == 0).sum()/len(df['Base MSRP'])*100)
```

```
261476
98.80889399458863
```

Figure 8. Counts of zero values in base MSRP

Step Three: Handle Missing Data

Several variables are excluded from further analysis: Legislative District (659 missing values), Vehicle Location (17 missing), 2020 Census Tract (9 missing), and Postal Code (9 missing). These fields are primarily relevant for political, spatial, or detailed geolocation analyses, domains outside the scope of this study. As they do not directly contribute to the analytical objectives, they are set aside at this stage.

```
# View 4 rows that have missing electric range
df_clean[df_clean['Electric Range'].isna()]
```

	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	Electric Utility
11897	ZHWUC1ZM3S	King	Mercer Island	WA	98040.00	2025	LAMBORGHINI	REVUELTO	Plug-in Hybrid Electric Vehicle (PHEV)	Clean Alternative Fuel Vehicle Eligible	NaN	NaN	PUGET SOUND ENERGY INC||CITY OF TACOMA - (WA)
28337	ZHWUC1ZM5S	King	Seattle	WA	98125.00	2025	LAMBORGHINI	REVUELTO	Plug-in Hybrid Electric Vehicle (PHEV)	Clean Alternative Fuel Vehicle Eligible	NaN	NaN	CITY OF SEATTLE - (WA)||CITY OF TACOMA - (WA)
51695	ZHWUC1ZM5S	Snohomish	Snohomish	WA	98296.00	2025	LAMBORGHINI	REVUELTO	Plug-in Hybrid Electric Vehicle (PHEV)	Clean Alternative Fuel Vehicle Eligible	NaN	NaN	PUGET SOUND ENERGY INC
182701	ZHWUC1ZM9S	Spokane	Spokane	WA	99005.00	2025	LAMBORGHINI	REVUELTO	Plug-in Hybrid Electric Vehicle (PHEV)	Clean Alternative Fuel Vehicle Eligible	NaN	NaN	BONNEVILLE POWER ADMINISTRATION||INLAND POWER ...

Figure 9. Rows with missing electric range and base MSRP values

Four records with missing values in both Electric Range and Base MSRP were imputed with 0 to align with the dataset's existing coding convention, where zero commonly represents missing or unreported information. This approach allows the records to be retained without introducing unsupported estimates.

For the nine records containing missing values in categorical variables such as County, City, Electric Utility, and State, the entries were imputed using an "Unknown" category.

This approach preserves all observations and maintains the integrity of the dataset while avoiding unverifiable assumptions about the true values.

After applying these imputation procedures and exclusions, all remaining variables in the dataset contain zero missing values, resulting in a complete dataset suitable for subsequent analysis.

```
# Check missing values after imputation
df_clean.isna().sum()
```

	0
VIN (1-10)	0
County	0
City	0
State	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
DOL Vehicle ID	0
Electric Utility	0

Figure 10. Null value counts after imputation

Step Four: Explore Data Patterns

Numerical Electric range

The Electric Range variable exhibits a pronounced right-skew, largely driven by a high frequency of zero values. Across the full dataset (264,628 entries), the mean is 41.71 miles, the median is 0, the standard deviation is 80.38 miles, and values range from 0 to 337 miles (Figure 11).

When zero values are excluded (100,827 entries), the distribution changes substantially: the mean rises to 109.48 miles, the median to 53 miles, the standard deviation to 97.66 miles, and the range spans 1–337 miles (Figure 11).

```
df_clean[['Electric Range']].describe()
```

	Electric Range
count	264628.00
mean	41.71
std	80.38
min	0.00
25%	0.00
50%	0.00
75%	34.00
max	337.00

```
# Filter non-zero electric range
df_non_zero_ER = df_clean[df_clean['Electric Range'] != 0]
```

```
df_non_zero_ER[['Electric Range']].describe()
```

	Electric Range
count	100827.00
mean	109.48
std	97.66
min	1.00
25%	30.00
50%	53.00
75%	215.00
max	337.00

Figure 11. Descriptive statistics of electric range (full dataset vs. excluding zero values)

The histogram after removing zeros (Figure 12) shows two approximately bell-shaped subpopulations, indicating heterogeneity in electric-range performance.

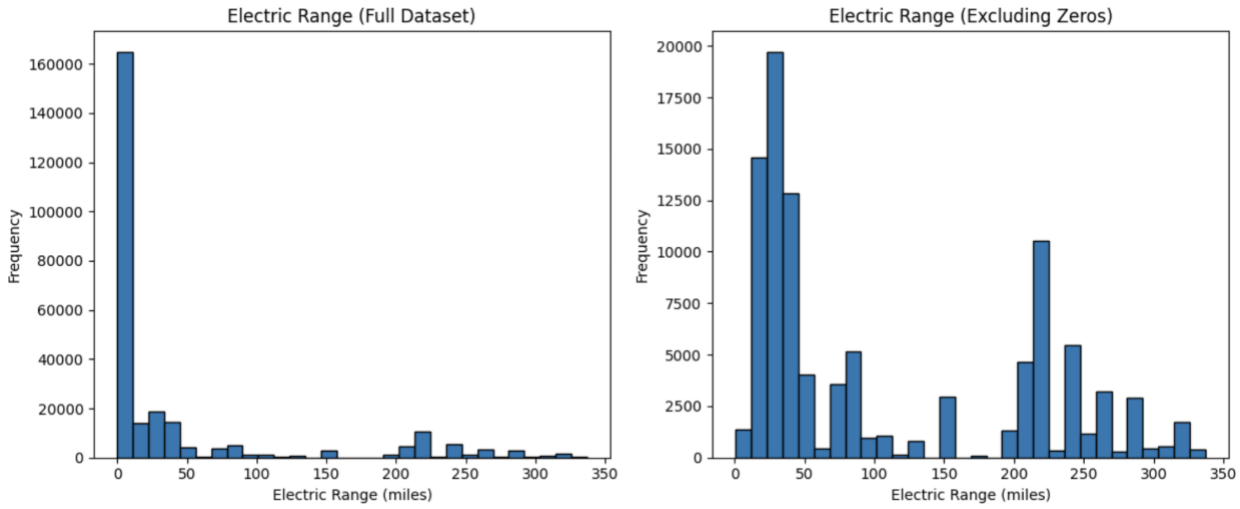


Figure 12. Histograms of electric range (full dataset vs. excluding zero values)

```
# Describe ER by Electric Vehicle Type
df_non_zero_ER['Electric Range'].groupby(df_non_zero_ER['Electric Vehicle Type']).describe()
```

Electric Vehicle Type	count	mean	std	min	25%	50%	75%	max
Battery Electric Vehicle (BEV)	46778.00	199.50	72.16	29.00	150.00	215.00	238.00	337.00
Plug-in Hybrid Electric Vehicle (PHEV)	54049.00	31.56	14.14	1.00	21.00	32.00	38.00	153.00

Figure 13. Descriptive statistics of electric range by electric vehicle (EV) types

Further analysis by EV type confirms this variation. Plug-in Hybrid Electric Vehicles (PHEVs) have ranges of 1–153 miles (mean = 31.56, SD = 14.14), while Battery Electric Vehicles (BEVs) range from 29–337 miles (mean = 199.50, SD = 72.16) (Figure 13). These differences highlight distinct subpopulations within the dataset, which are clearly visualized in Figure 14.

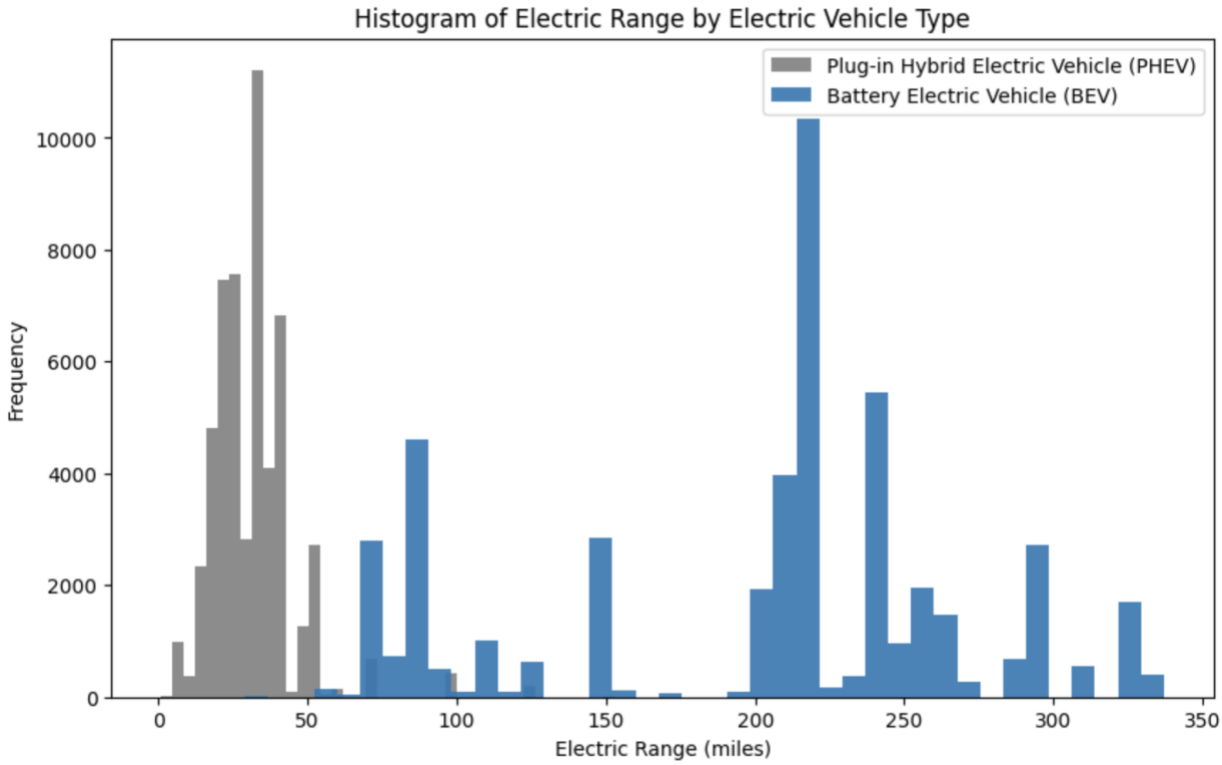


Figure 14. Histogram of electric range by electric vehicle (EV) types

Numerical variable: base MSRP

The full dataset contains 264,628 vehicle records, but the Base MSRP variable is dominated by zero values, with more than 260,000 entries recorded as $0. As shown in Figure 15, these zero values substantially distort the summary statistics: the mean is suppressed to approximately $678, and the median and interquartile range are also zero. These values clearly do not represent actual vehicle prices and are likely placeholders for missing data.

After removing zero-MSRP entries, the dataset decreases to 3,148 valid records. The descriptive statistics become more meaningful, with a mean of approximately $57,069, a median of $55,700, and a standard deviation of $27,354. This cleaned distribution, illustrated in Figure 16, is easier to interpret; however, the presence of extremely high-priced vehicles (primarily >$150,000) results in a strongly right-skewed pattern.

```
df_clean[['Base MSRP']].describe()
```

	Base MSRP
count	264628.00
mean	678.89
std	6868.87
min	0.00
25%	0.00
50%	0.00
75%	0.00
max	845000.00

```
# Describe only Base MSRP
df_non_zero_MSRP['Base MSRP'].describe()
```

	Base MSRP
count	3148.00
mean	57069.19
std	27354.07
min	31950.00
25%	39221.25
50%	55700.00
75%	69900.00
max	845000.00

Figure 15. Descriptive statistics of base MSRP (full dataset vs. excluding zero values)

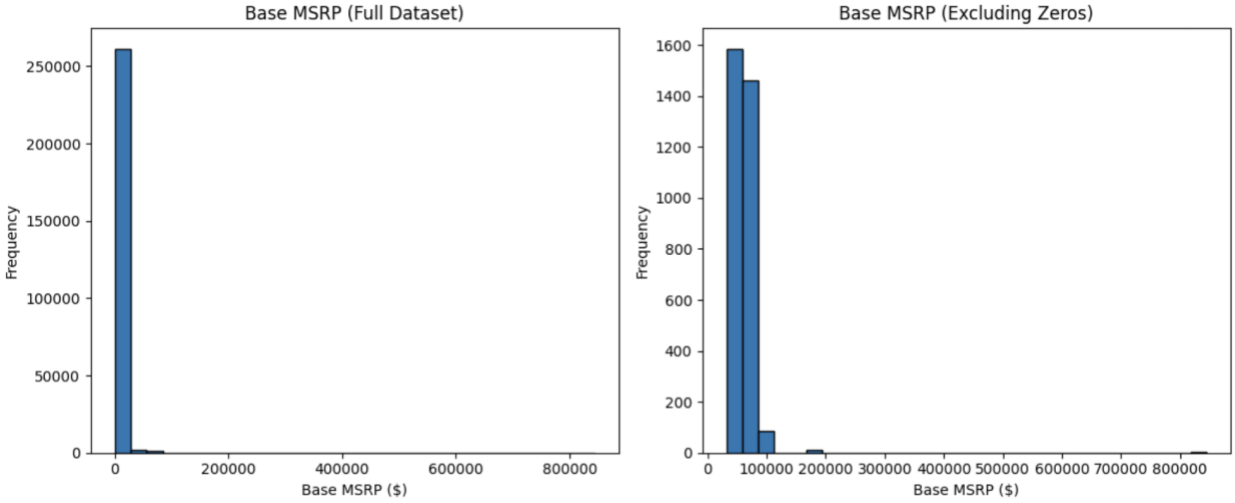


Figure 16. Histograms of base MSRP (full dataset vs. excluding zero values)

To better capture typical EV pricing, outliers above $150,000 were removed. The resulting descriptive statistics (Figure 17) show a lower mean ($56,037), a smaller standard deviation ($16,941), and a narrower range ($31,950 to $110,950), indicating a more stable and representative dataset.

Once outliers are excluded, the distribution (Figure 18) exhibits a clearer right-skew without the influence of ultra-luxury vehicles. A large portion of MSRPs cluster around $70,000, consistent with mid- to high-range EV models. Below $70,000, the density increases further, reflecting mainstream market pricing. Above this threshold, frequencies decline sharply, indicating that high-end models represent a small share of EV registrations.

```
# Describe Base MSRP after excluding zeros and outliers
df_MSRP_no_outliers['Base MSRP'].describe()
```

	Base MSRP
count	3133.00
mean	56037.86
std	16941.74
min	31950.00
25%	36900.00
50%	55700.00
75%	69900.00
max	110950.00

Figure 17. Descriptive statistics of base MSRP (excluding zero values and outliers)

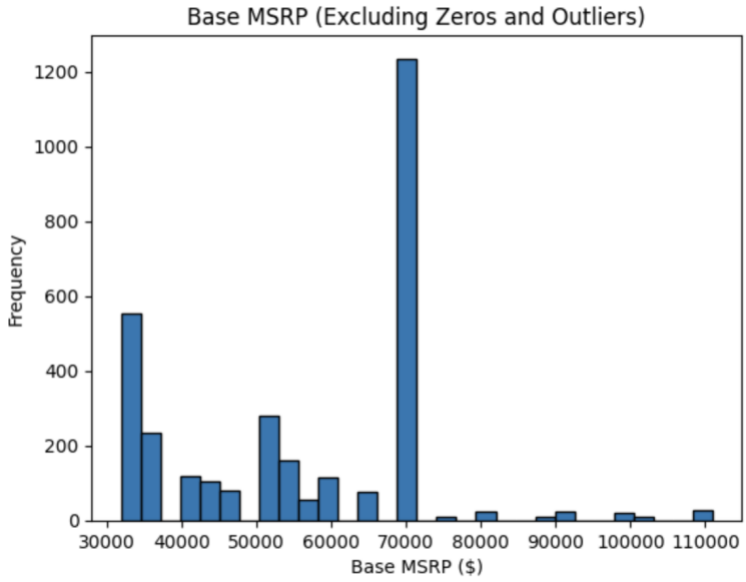


Figure 18. Histogram of base MSRP (excluding zero values and outliers)

Categorical variables

Figure 19 summarizes the key categorical attributes in the dataset. Most EV registrations are concentrated in Seattle, King County, Washington State. The 2023 model year shows the highest adoption, and Tesla, particularly the Model Y, remains the dominant manufacturer, reflecting strong market penetration.

Figure 20 further highlights that Battery Electric Vehicles (BEVs) account for the majority of registrations, while Plug-in Hybrid Electric Vehicles (PHEVs) represent a smaller share. The distribution of Clean Alternative Fuel Vehicle (CAFV) eligibility shows

that most vehicles fall into the unknown category due to unresearched or missing electric-range values, limiting the interpretability of this field.

Finally, Puget Sound Energy emerges as the leading electric utility provider, underscoring its central role in supporting EV adoption across the region.

```
# describe all
df_clean[['County', 'City', 'State', 'Model Year', 'Make', 'Model',
          'Electric Vehicle Type',
          'Clean Alternative Fuel Vehicle (CAFV) Eligibility',
          'Electric Utility']].describe(include='all').transpose()
```

	count	unique	top	freq
County	264628	240	King	131179
City	264628	859	Seattle	41533
State	264628	51	WA	263969
Model Year	264628	22	2023	60157
Make	264628	46	TESLA	108633
Model	264628	182	MODEL Y	55187
Electric Vehicle Type	264628	2	Battery Electric Vehicle (BEV)	210575
Clean Alternative Fuel Vehicle (CAFV) Eligibility	264628	3	Eligibility unknown as battery range has not b...	163797
Electric Utility	264628	77	PUGET SOUND ENERGY INCIICITY OF TACOMA - (WA)	94223

Figure 19. Overview of categorical variables

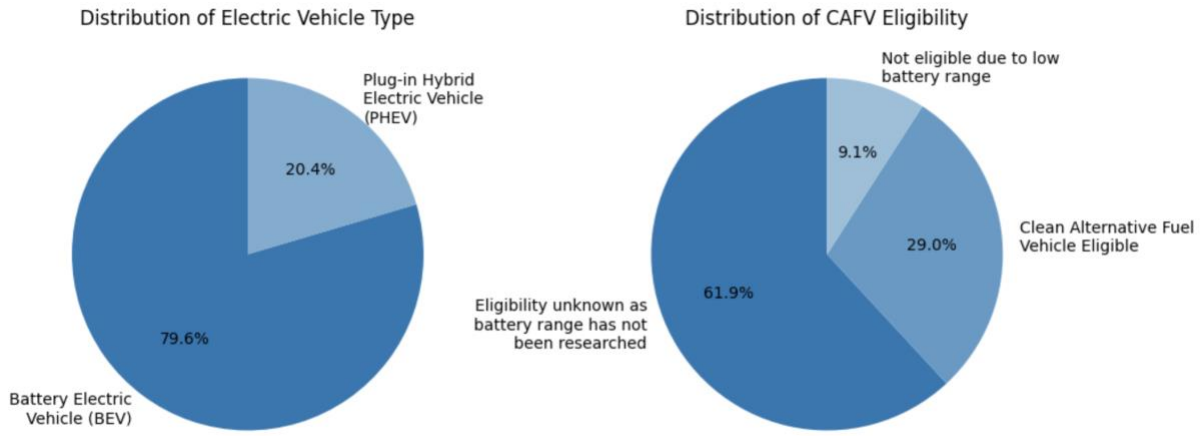


Figure 20. Distribution of electric vehicle type and CAFV eligibility

Step Five: Transform Data

Several data transformations were applied to prepare the dataset for analysis.

- **Model year** was converted from a numerical to a categorical variable to better reflect its discrete nature and facilitate categorical comparisons.

- A combined **"City_County"** variable was created to eliminate ambiguity where cities occur in multiple counties.
- A **"Model_Make"** variable was added to provide clearer model identification.
- Key categorical fields including **EV type**, **CAFV eligibility**, **make**, **model**, **county**, and **electric utility** were numerically encoded to enable computation of the correlation matrix.

These transformations ensure that the dataset is properly structured for statistical analysis and modeling.

Step Six: Visualize Correlations

Correlation Matrix

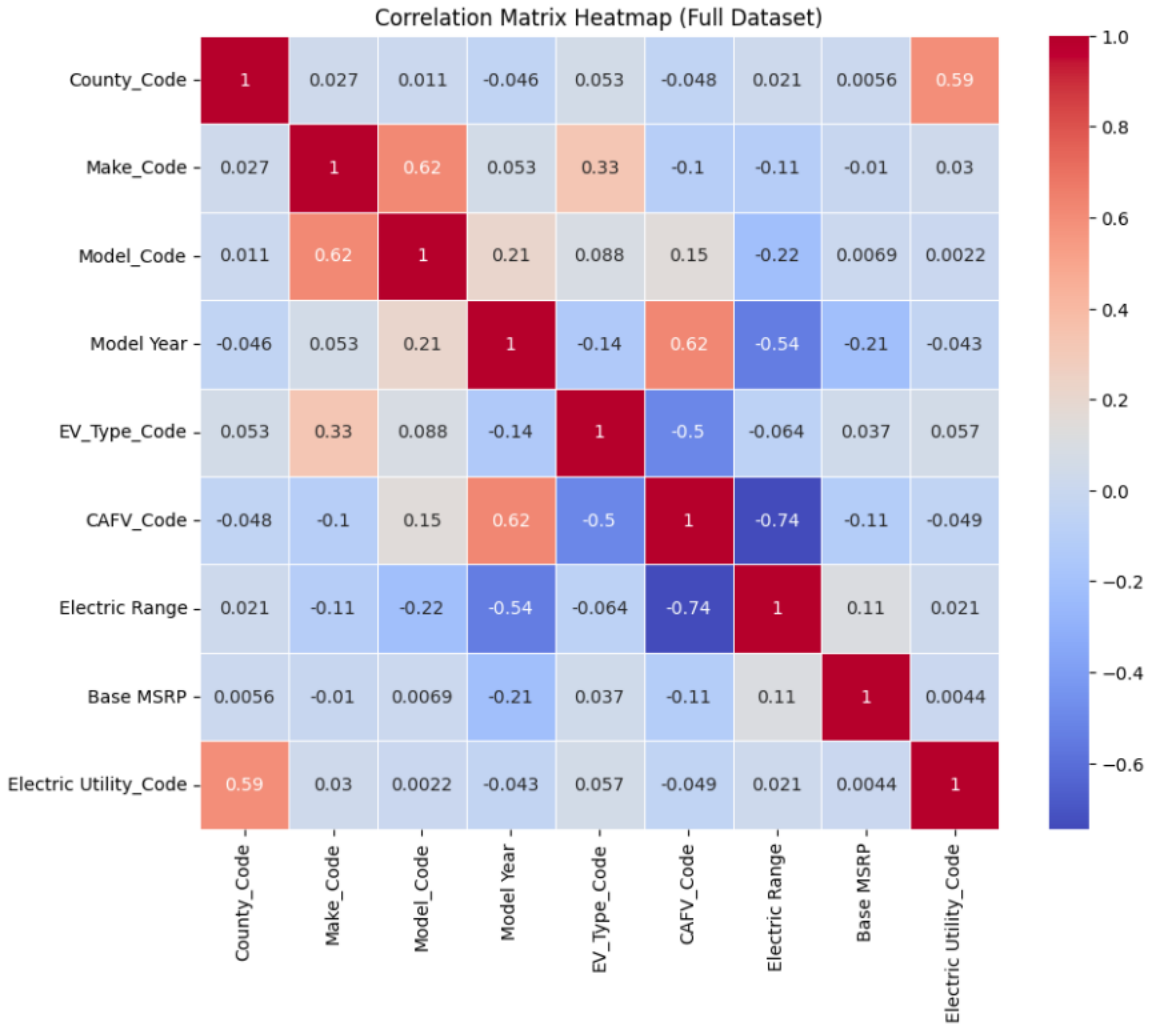


Figure 21. Correlation Matrix Heatmap (Full Dataset)

The correlation matrix generated from the full dataset shows predominantly weak associations, with most coefficients clustering near zero. This is largely driven by the high volume of zero MSRP values, zero electric-range entries, and unknown CAFV

eligibility records, all of which introduce noise and mask meaningful relationships. Consequently, expected patterns such as the relationship between Base MSRP and Electric Range do not emerge because both variables contain large proportions of invalid zeros.

A few logical associations are still visible, including relationships between county and electric utility, make and model, CAFV eligibility and model year, and electric range and CAFV eligibility. However, these are limited, and the overall matrix reflects the impact of missing or uninformative data rather than true underlying trends.

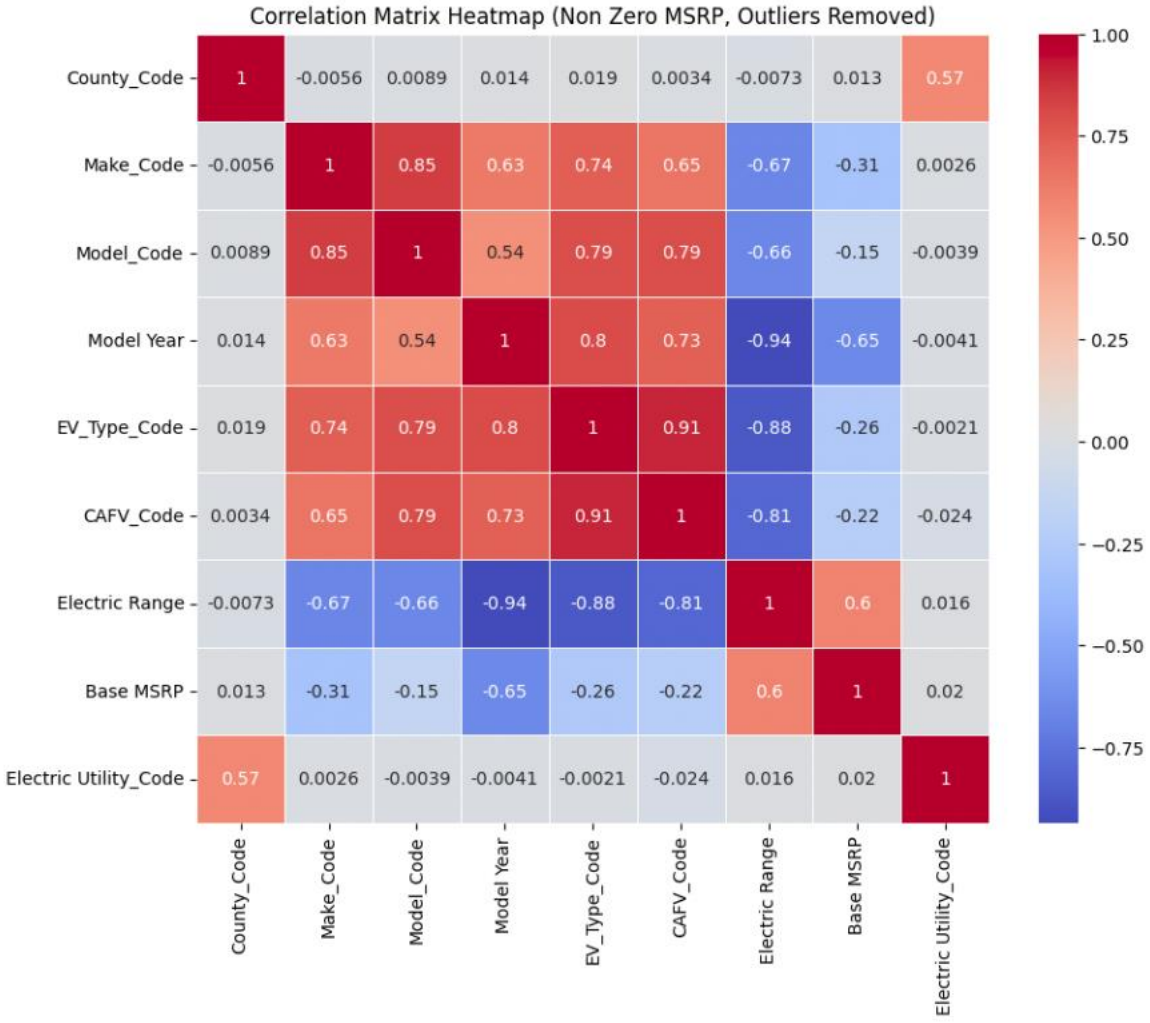


Figure 22. Correlation Matrix Heatmap Based on Cleaned Base MSRP Data (Excluding Zero Values and Outliers)

After removing zero MSRPs and outliers, the cleaned correlation matrix reveals clearer and more meaningful patterns. Strong positive relationships appear between **make and model, model year and EV type, EV type and CAFV eligibility, and CAFV eligibility and electric range**, reflecting consistent links between vehicle technology, manufacturing structure, and policy qualification. The relationship between **EV type**

and electric range also highlights the expected range differences between BEVs and PHEVs.

Moderate correlations are observed between **county and electric utility**, due to region-specific service areas, and between **Base MSRP and model year**, indicating that newer vehicles tend to carry higher prices. All remaining variable pairs show weak or negligible correlations, suggesting limited direct associations.

Exploring Relationships Between Variables

Research question 1: What is the distribution of electric vehicle types (BEV vs. PHEV) across the state?

The distribution of electric vehicle types in Washington State is heavily skewed toward Battery Electric Vehicles (BEVs), which account for approximately 80% of all registered EVs. This pattern holds consistently across both counties and cities. In every top location, BEVs make up the clear majority of the EV fleet, with PHEVs representing only about 20%. High-adoption areas such as King County and Seattle show the same trend, reinforcing that BEVs dominate statewide regardless of geography.

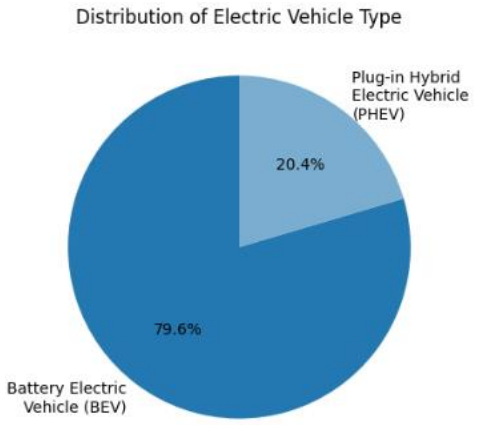


Figure 23. Distribution of Electric Vehicle Type

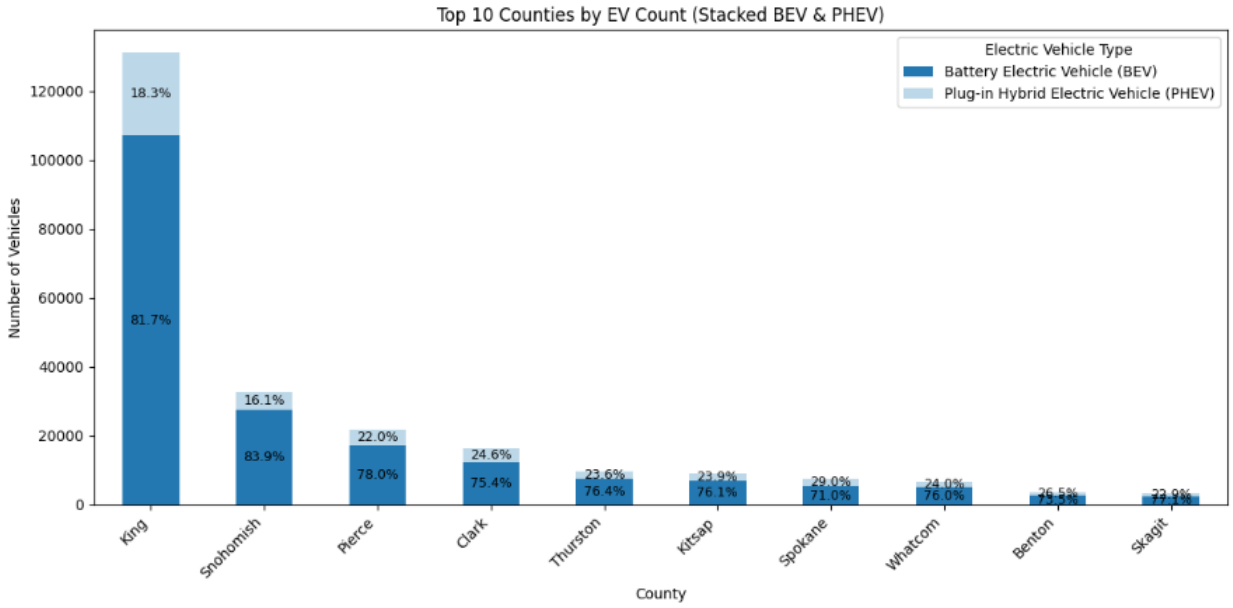


Figure 24. Top 10 Counties by EV Registrations and EV Type Mix

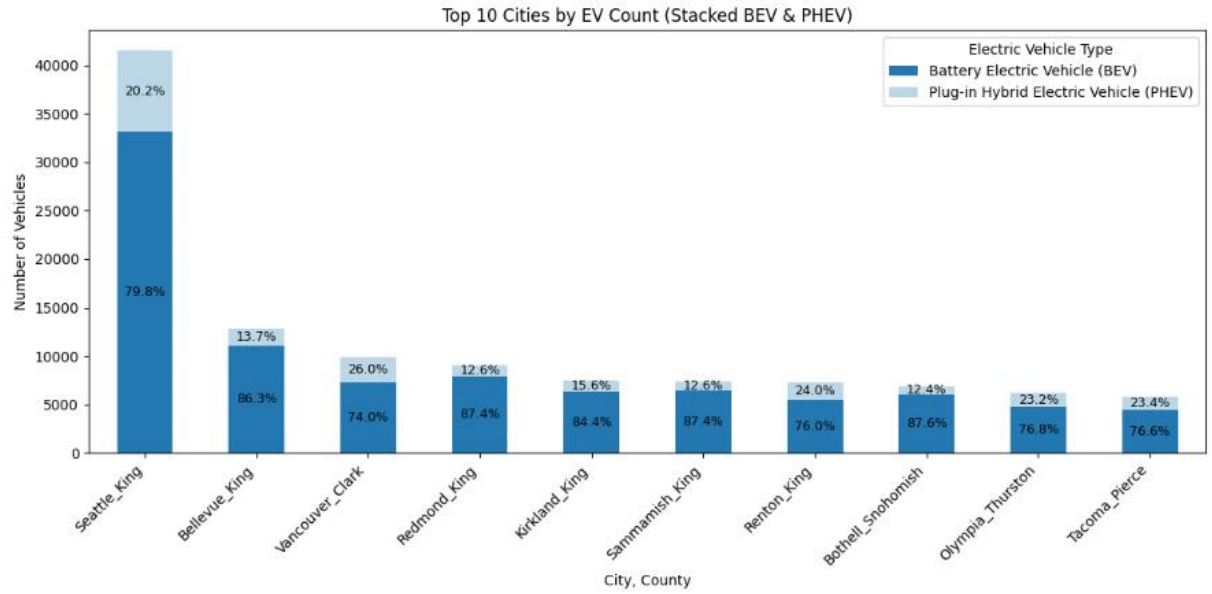


Figure 25. Top 10 Cities by EV Registrations and EV Type Mix

Research question 2: Which manufacturers and models dominate the EV market, and how do these patterns vary geographically?

The EV market in Washington State is overwhelmingly dominated by Tesla, both in terms of manufacturers and specific models. Tesla is the leading brand by a wide margin, and the Model Y and Model 3 are the two most registered EVs in the state. This dominance is consistent across major cities and counties, where Tesla accounts for the majority of EV registrations in nearly every top location. Other manufacturers such as

Chevrolet, Ford, Kia, and Nissan hold much smaller and more evenly distributed shares, indicating that Tesla's leadership is both statewide and geographically consistent.

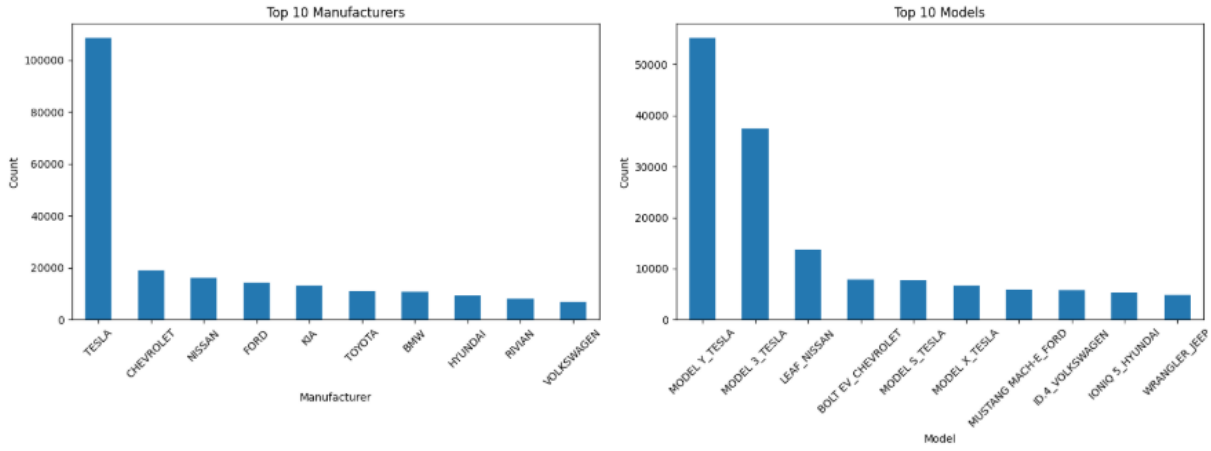


Figure 26. Top 10 Manufacturers and Models in Washington's EV Market

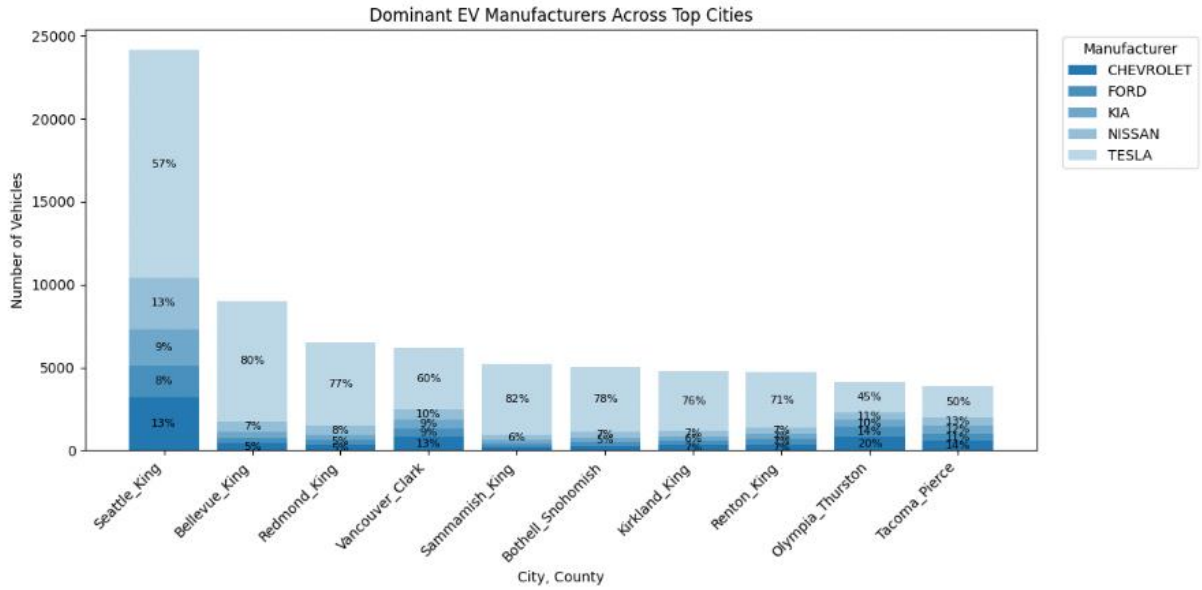


Figure 27. Dominant EV Manufacturers Across Top 10 Cities

Research question 3: How has electric vehicle adoption evolved over time by model year, vehicle type and electric range capabilities?

EV adoption in Washington State has increased sharply over time, with the most pronounced growth occurring from 2018 onward, peaking in 2023. When disaggregated by vehicle type, BEVs drive most of this growth, expanding rapidly in recent years, while PHEVs grow more slowly and represent a smaller share of new registrations.

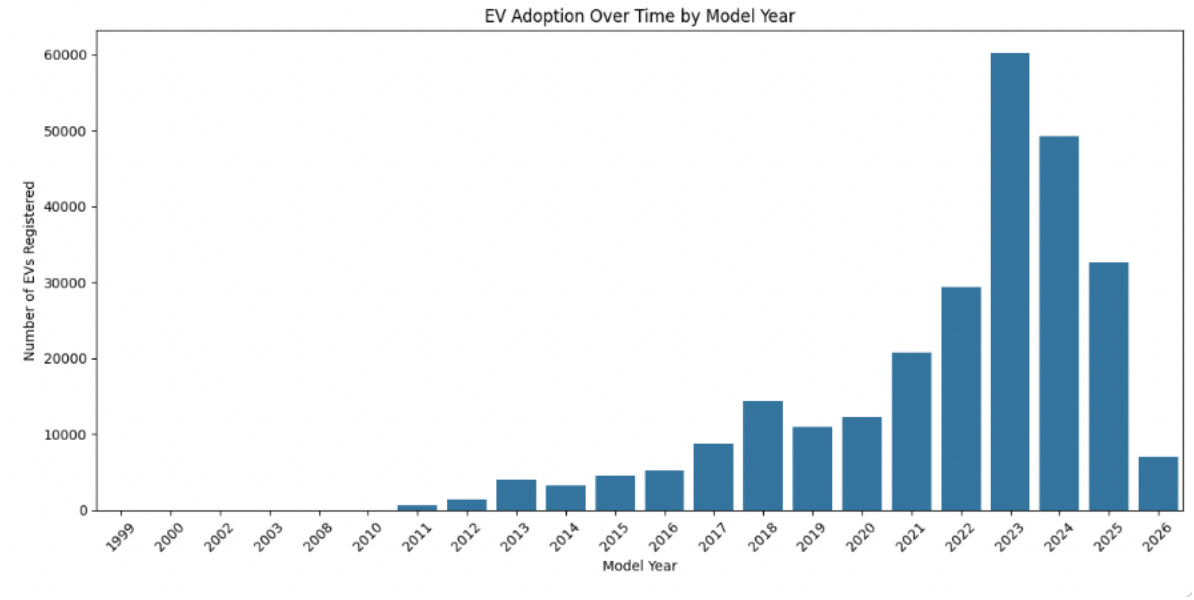


Figure 28. EV adoption over time by model year

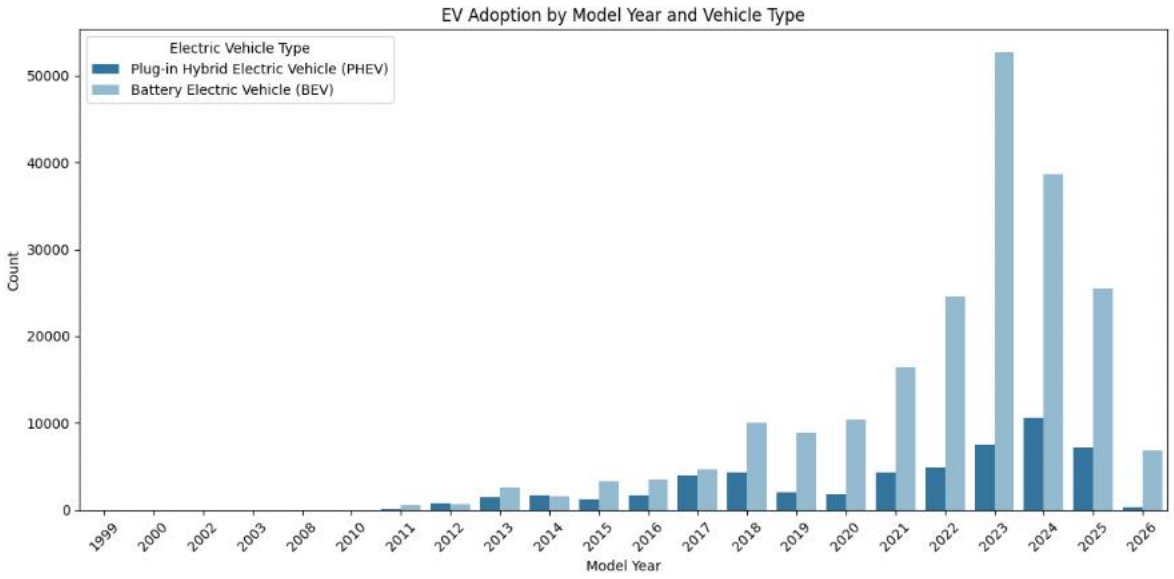


Figure 29. EV adoption by model year and vehicle type

Electric-range trends mirror these adoption patterns. BEVs show a substantial increase in median range over time, reflecting advances in battery technology, whereas PHEV ranges remain relatively low and stable. This divergence reinforces the technological shift toward fully electric platforms with longer driving capabilities.

Overall, EV adoption has accelerated across newer model years, led by BEVs with improving range, underscoring a broader market transition toward higher-range, fully electric vehicles.

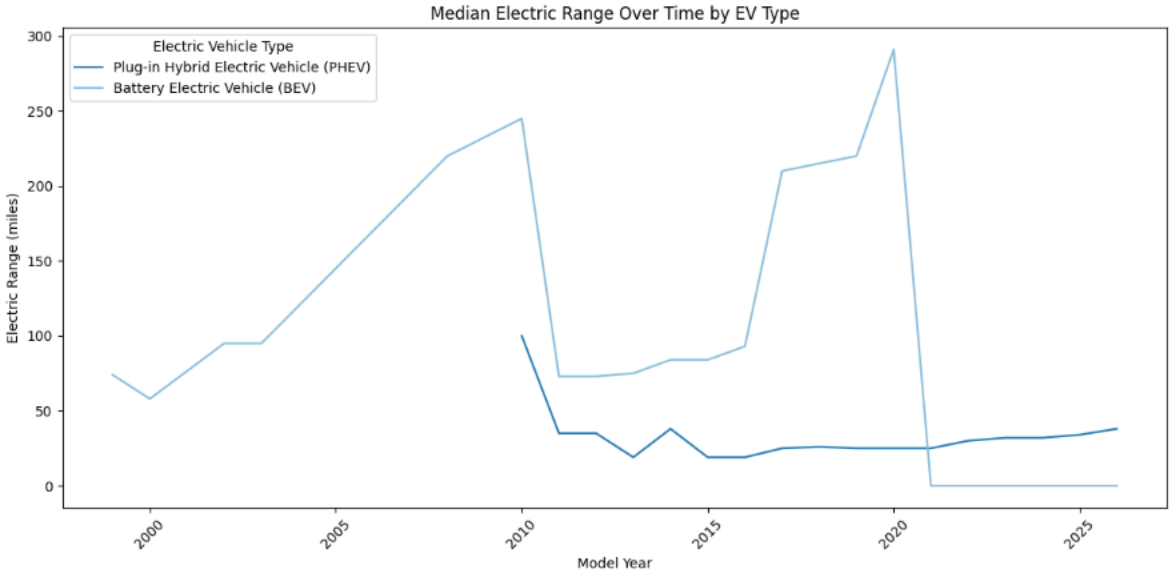


Figure 30. Median Electric Range Over Time by EV Type

Research question 4: Which areas show the fastest growth in EV registrations over time?

EV registration growth varies substantially across locations, with Seattle, Bellevue, and Redmond showing the fastest and most sustained increases over time. These areas experience sharp growth beginning around 2018, peaking in 2022–2023, reflecting strong adoption in major urban and tech-oriented communities. Other cities such as Sammamish, Kirkland, and Renton also show steady growth but at lower magnitudes. Overall, the most rapid expansion occurs in high-population, high-income areas within King County, indicating that EV adoption is concentrated in economically and infrastructure-advantaged regions.

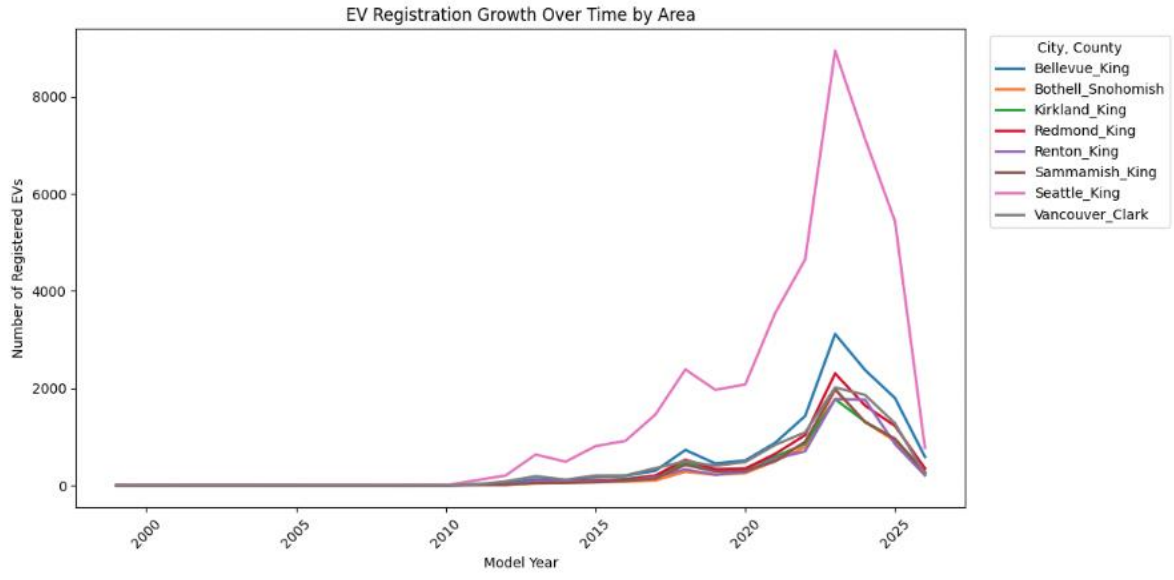


Figure 31. EV Registration Growth Over Time by Cities

Research question 5: What relationship, if any, exists between base MSRP and electric range performance? How do electric range and base MSRP vary across manufacturers and models?

The full-dataset scatterplot shows a dense cluster of observations at zero MSRP, making it difficult to observe any meaningful relationship between vehicle price and electric range.

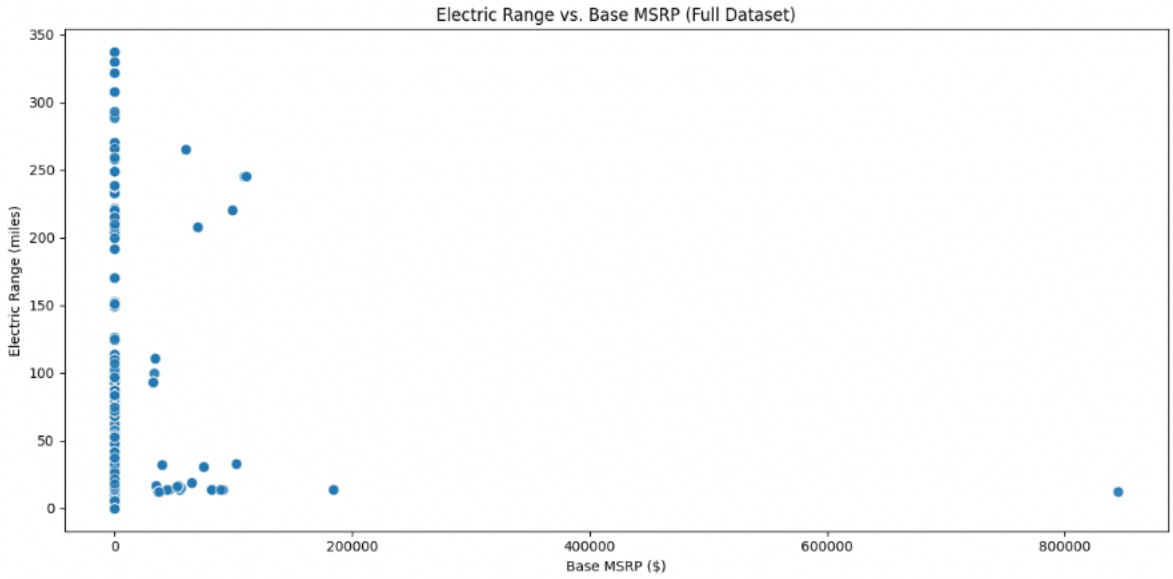


Figure 32. Scatterplot of Electric Range vs. Base MSRP (Full Database)

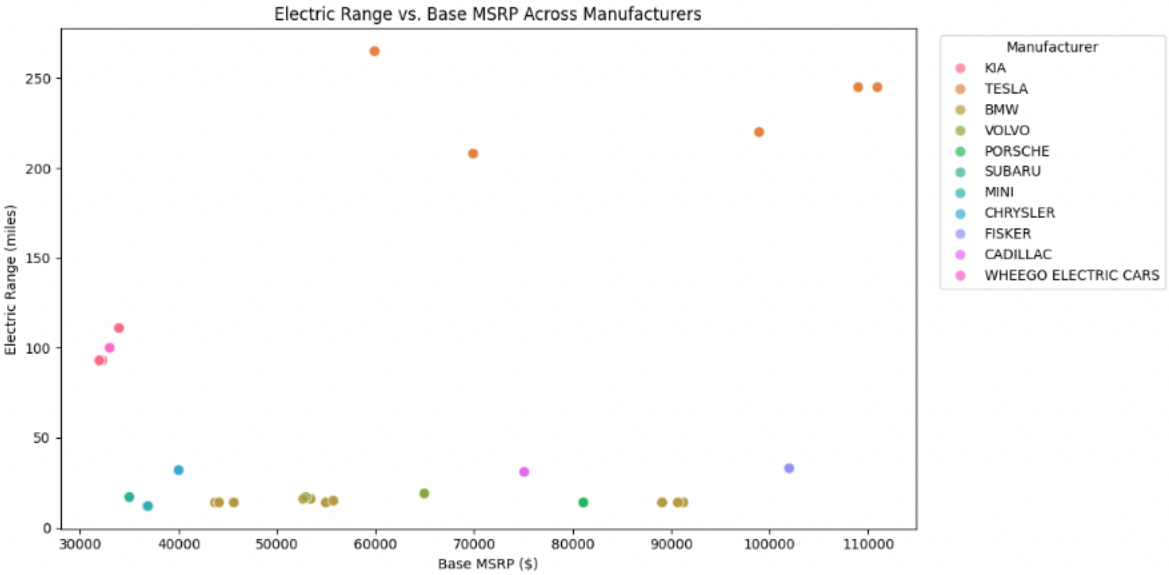


Figure 33. Scatterplot of Electric Range vs. Base MSRP (Non-zero MSRP, outlier excluded)

After excluding zero-MSRP entries and outliers, the cleaned scatterplot indicates that the relationship between base MSRP and electric range remains weak—higher prices do not consistently correspond to longer ranges. However, manufacturer-level patterns become clearer: brands such as Tesla, Volvo, and Porsche tend to offer higher-range vehicles at higher price points, whereas manufacturers like Kia, MINI, and Chrysler cluster at lower ranges and lower prices.

The boxplots confirm substantial variation in both range and MSRP across manufacturers and models, suggesting that electric-range performance is more strongly shaped by manufacturer design choices than by price alone.

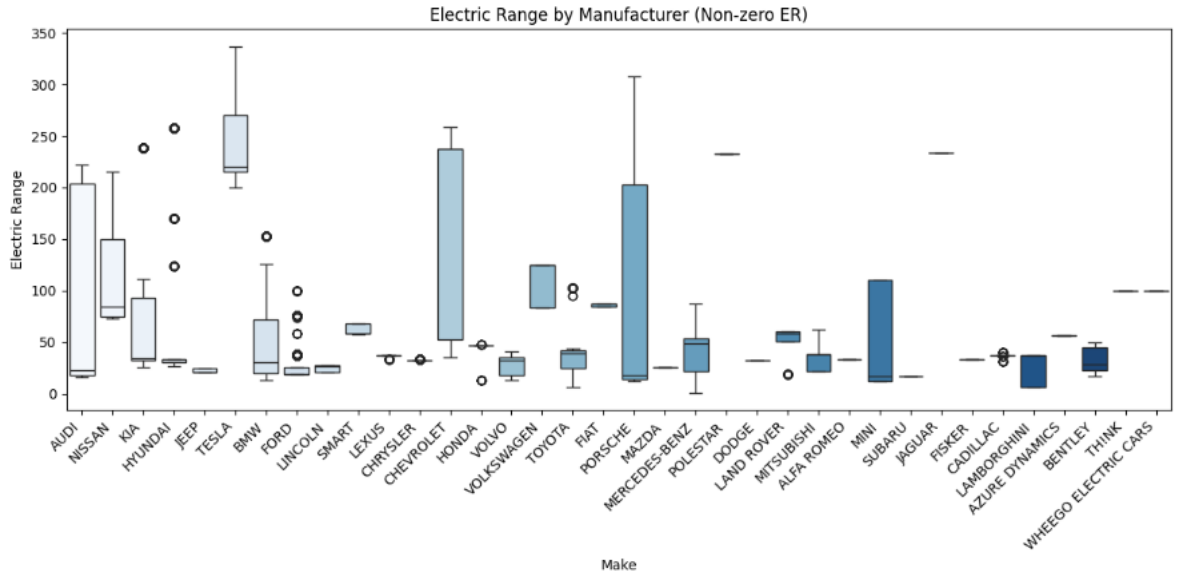


Figure 34. Boxplots of Electric Range by Manufacturer (Non-zero ER)

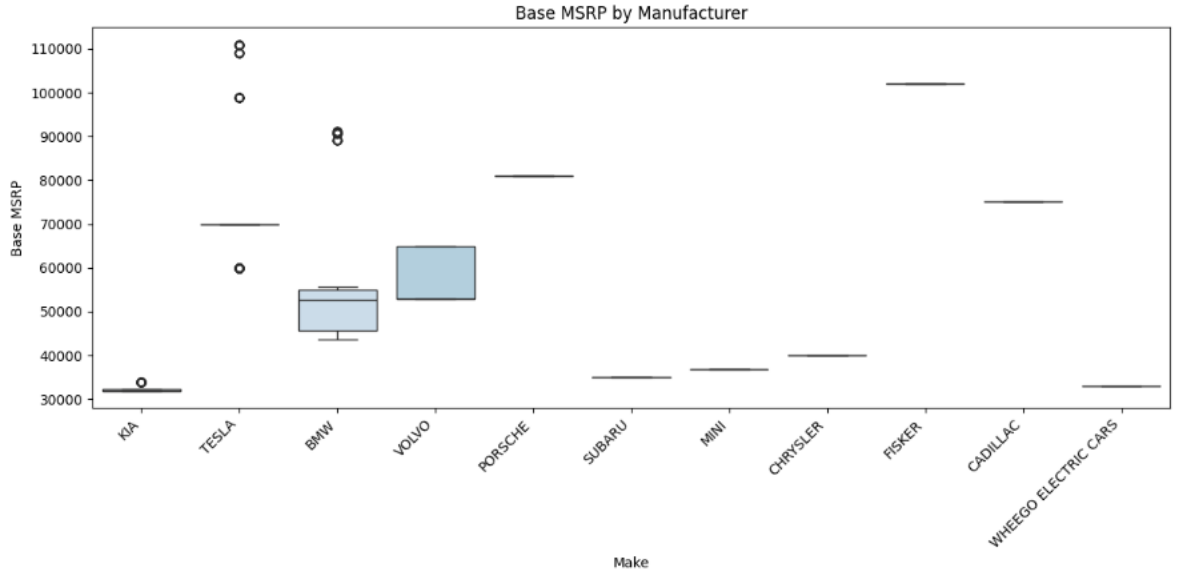


Figure 35. Boxplots of Base MSRP by Manufacturer (Non-zero MSRP, outliers excluded)

Step Seven: Handle Outliers

Outliers were assessed for the numerical variables using boxplots (Figures 36 & 37). The approach differs for Base MSRP and Electric Range based on the behavior of each variable.

Base MSRP

The boxplot revealed a small number of extremely high-priced vehicles far outside the typical range. These values represent ultra-luxury EV models and would greatly distort any statistical analysis involving price. In addition, the dataset contained a large number of zero MSRP values, which do not reflect actual vehicle pricing.

To ensure that the analysis focuses on typical EV pricing patterns in Washington State, both zero-value MSRPs and extreme outliers were removed. As a result, all analyses involving Base MSRP, including descriptive summaries and any relationships with other variables, use the cleaned dataset that excludes these values.

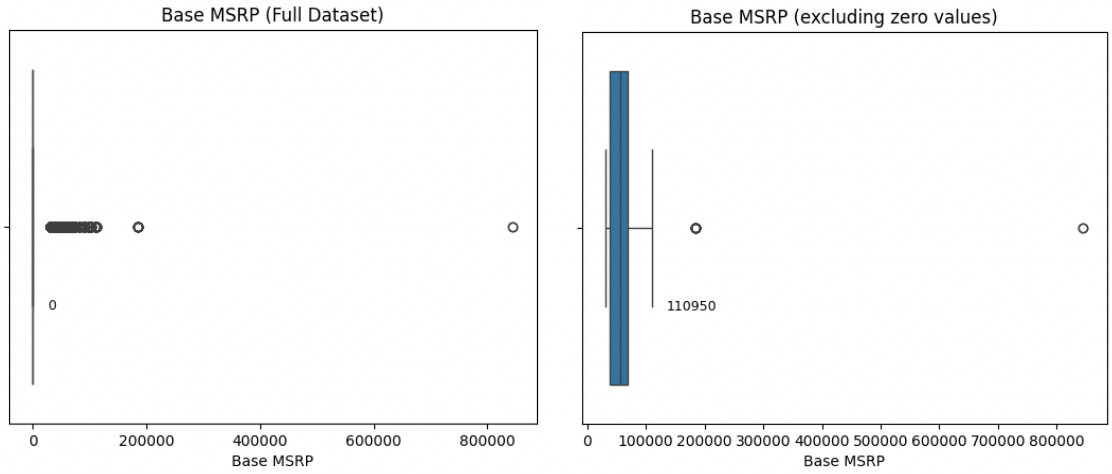


Figure 36. Boxplot of Base MSRP (full dataset vs. excluding zero values)

Electric Range

For electric range, the full dataset includes many zero values, likely representing missing or unreported manufacturer range data. However, once these zero values are excluded, the boxplot shows no meaningful outliers in the remaining observations. The non-zero electric-range values fall within a consistent and interpretable range.

Therefore, all analyses involving electric range use the dataset that excludes only zero values, with no additional outlier removal required.

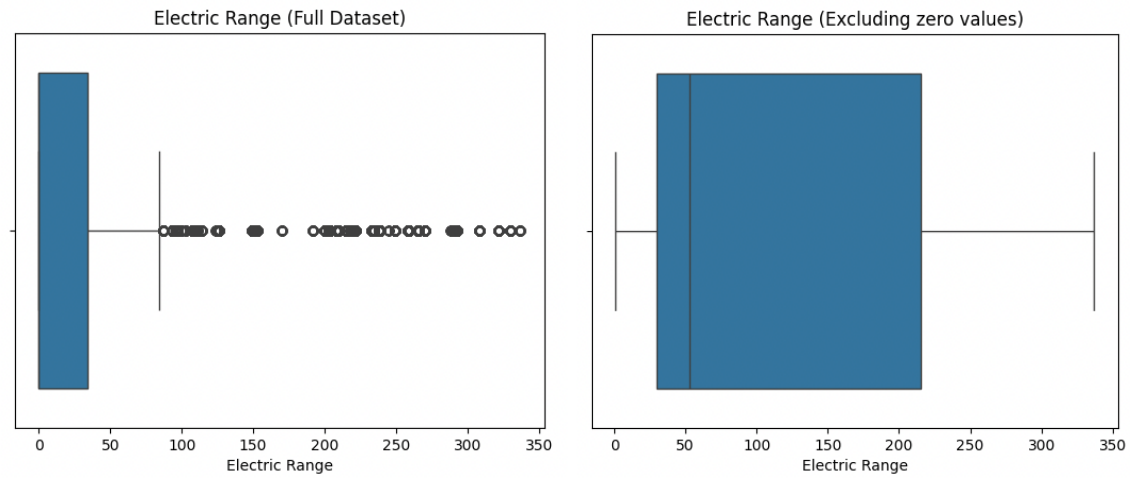


Figure 37. Boxplot of Electric Range (full dataset vs. excluding zero values)

Step Eight: Communicate Insights

The analyses conducted in this project reveal several meaningful insights into electric vehicle adoption patterns in Washington State. First, Battery Electric Vehicles (BEVs) overwhelmingly dominate the market, consistently representing about 80% of registrations across counties and cities. This conclusion is supported by both statewide proportions and stacked bar charts showing BEVs as the majority in all high-adoption areas.

Second, Tesla and its leading models—especially the Model Y and Model 3—clearly dominate the EV landscape, a finding reinforced by manufacturer rankings and stacked manufacturer distributions across major cities. The concentration of EV adoption in high-income, urban areas within King County further highlights the role of socioeconomic and infrastructure factors in driving EV uptake.

Temporal analyses show that EV adoption has accelerated sharply since 2018, with BEVs driving most of the growth. Improvements in BEV electric range over time support this trend, as newer models offer substantially higher range performance than earlier generations. Conversely, PHEVs show limited improvement and remain relatively low in range, which aligns with their smaller battery capacities.

Despite expectations, the relationship between base MSRP and electric range is weak. This conclusion follows from scatterplots that show large variation in electric range at similar price points and substantial differences across manufacturers. Boxplots confirm that range performance is manufacturer-specific rather than strongly price-dependent, suggesting that engineering choices and battery technologies drive range more than MSRP alone.

These insights collectively reveal how technology, geography, market dynamics, and policy factors shape EV adoption in the state.

Emerging Questions

The findings also raise new questions that extend beyond the original research scope:

- What demographic or socioeconomic factors explain why King County leads EV adoption so strongly?
- How does charging infrastructure availability influence adoption patterns across cities?
- Do policy incentives (e.g., CAFV eligibility) meaningfully shift consumer behavior, or do range and brand reputation dominate choices?
- How will the introduction of new long-range models or lower-cost EVs affect future adoption trends?
- What role do fleet vehicles or corporate purchases play in shaping local EV patterns?

These questions suggest opportunities for deeper analysis involving demographic data, charging network density, policy impacts, and future forecasting.

References

Data.gov. Electric Vehicle Population Data. Retrieved from https://catalog.data.gov/dataset/electric-vehicle-population-data

Appendix

Appendix A. Step Two Python Code – Import & Inspect Data

```
import pandas as pd
from sklearn.preprocessing import LabelEncoder
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

import textwrap # to clean the labels for piecharts
import matplotlib.colors as mcolors # color for charts

from google.colab import drive
drive.mount('/content/drive')

from datetime import datetime
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive
```

```
# Universal formatting
# Turn off scientific notation
pd.set_option('display.float_format', '{:.2f}'.format)

# Base color
base_color = "#1f77b4"

# Function to lighten the base color
def lighten(color, amount=0.5):
    """
    amount: 0 = original color, 1 = white
    """
    c = np.array(mcolors.to_rgb(color))
    white = np.array([1, 1, 1])
    return mcolors.to_hex(c + (white - c) * amount)
```

```
# Import
# File name
path = "/content/drive/MyDrive/Colab/2_ISE 201/Data/Project/"
filename = "Electric_Vehicle_Population_Data.csv"

# Load data
df = pd.read_csv(path + filename, encoding='latin1')

# Check data size
df.shape
```

```
(264628, 17)
```

```
# Preview dataset
df.head()
```

	VIN (1-10)	County	City	State	Postal Code	Model Year	Make	Model	Electric Vehicle Type	Clean Alternative Fuel Vehicle (CAFV) Eligibility	Electric Range	Ba MS
0	WA1E2AFY8R	Thurston	Olympia	WA	98512.00	2024	AUDI	Q5 E	Plug-in Hybrid Electric Vehicle (PHEV)	Not eligible due to low battery range	23.00	0
1	WAUUPBFF4J	Yakima	Wapato	WA	98951.00	2018	AUDI	A3	Plug-in Hybrid Electric Vehicle (PHEV)	Not eligible due to low battery range	16.00	0

Check null values

```
# Electric Vehicle Info
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 264628 entries, 0 to 264627
Data columns (total 17 columns):
 #   Column                                             Non-Null Count   Dtype
---  ------                                             --------------   -----
 0   VIN (1-10)                                         264628 non-null  object
 1   County                                             264619 non-null  object
 2   City                                               264619 non-null  object
 3   State                                              264628 non-null  object
 4   Postal Code                                        264619 non-null  float64
 5   Model Year                                         264628 non-null  int64
 6   Make                                               264628 non-null  object
 7   Model                                              264628 non-null  object
 8   Electric Vehicle Type                              264628 non-null  object
 9   Clean Alternative Fuel Vehicle (CAFV) Eligibility  264628 non-null  object
 10  Electric Range                                     264624 non-null  float64
 11  Base MSRP                                          264624 non-null  float64
 12  Legislative District                               263969 non-null  float64
 13  DOL Vehicle ID                                     264628 non-null  int64
 14  Vehicle Location                                   264611 non-null  object
 15  Electric Utility                                   264619 non-null  object
 16  2020 Census Tract                                  264619 non-null  float64
dtypes: float64(5), int64(2), object(10)
memory usage: 34.3+ MB
```

```
df.isna().sum()
```

	0
VIN (1-10)	0
County	9
City	9
State	0
Postal Code	9
Model Year	0
Make	0
Model	0
Electric Vehicle Type	0
Clean Alternative Fuel Vehicle (CAFV) Eligibility	0
Electric Range	4
Base MSRP	4
Legislative District	659
DOL Vehicle ID	0
Vehicle Location	17
Electric Utility	9
2020 Census Tract	9

dtype: int64

Descriptive Statistics

```
# Electric Range
print("Mean = ", df['Electric Range'].mean())
print("Standard Deviation = ", df['Electric Range'].std())
print("Median = ", df['Electric Range'].median())
print("Mode = ", df['Electric Range'].mode())
print("Minimum = ", df['Electric Range'].min())
print("Maximum = ", df['Electric Range'].max())
print("Count = ", df['Electric Range'].count())
print("Length =", len(df['Electric Range']))
```

```
Mean =  41.713159048310054
Standard Deviation =  80.37797717197819
Median =  0.0
Mode =  0   0.00
Name: Electric Range, dtype: float64
Minimum =  0.0
Maximum =  337.0
Count =  264624
Length = 264628
```

```
# Count zero values in electric range
print((df['Electric Range'] == 0).sum())
print((df['Electric Range'] == 0).sum()/len(df['Electric Range'])*100)
```

```
163797
61.89707816255271
```

```
(df['Electric Range'] == 0).groupby(df['Electric Vehicle Type']).sum()
```

Electric Vehicle Type	Electric Range
Battery Electric Vehicle (BEV)	163797
Plug-in Hybrid Electric Vehicle (PHEV)	0

dtype: int64

```
# Calculate electric range mean by vehicle type
df['Electric Range'].groupby(df['Electric Vehicle Type']).describe()
```

Electric Vehicle Type	count	mean	std	min	25%	50%	75%	max
Battery Electric Vehicle (BEV)	210575.00	44.32	89.63	0.00	0.00	0.00	0.00	337.00
Plug-in Hybrid Electric Vehicle (PHEV)	54049.00	31.56	14.14	1.00	21.00	32.00	38.00	153.00

```
# Base MSRP
print("Mean = ", df['Base MSRP'].mean())
print("Standard Deviation = ", df['Base MSRP'].std())
print("Median = ", df['Base MSRP'].median())
print("Mode = ", df['Base MSRP'].mode())
print("Minimum = ", df['Base MSRP'].min())
print("Maximum = ", df['Base MSRP'].max())
print("Count = ", df['Base MSRP'].count())
print("Length =", len(df['Base MSRP']))
```

```
Mean =  678.90219707963
Standard Deviation =  6868.919926418542
Median =  0.0
Mode =  0   0.00
Name: Base MSRP, dtype: float64
Minimum =  0.0
Maximum =  845000.0
Count =  264624
Length = 264628
```

Appendix B. Step Three Python Code – Handle Missing Data

```
# Drop columns that not being used
df_clean = df.drop(columns=['Legislative District',
                            'Vehicle Location',
                            '2020 Census Tract',
                            'Postal Code'])
```

```
# View 4 rows that have missing electric range
df_clean[df_clean['Electric Range'].isna()]
```

	VIN (1-10)	County	City	State	Model Year	Make	Model	Electric Vehicle Type	Clean Alternative Fuel Vehicle (CAFV) Eligibility	Elec R
11897	ZHWUC1ZM3S	King	Mercer Island	WA	2025	LAMBORGHINI	REVUELTO	Plug-in Hybrid Electric Vehicle (PHEV)	Clean Alternative Fuel Vehicle Eligible	
28337	ZHWUC1ZM5S	King	Seattle	WA	2025	LAMBORGHINI	REVUELTO	Plug-in Hybrid Electric Vehicle (PHEV)	Clean Alternative Fuel Vehicle Eligible	
51695	ZHWUC1ZM5S	Snohomish	Snohomish	WA	2025	LAMBORGHINI	REVUELTO	Plug-in Hybrid Electric Vehicle (PHEV)	Clean Alternative Fuel Vehicle Eligible	
182701	ZHWUC1ZM9S	Spokane	Spokane	WA	2025	LAMBORGHINI	REVUELTO	Plug-in Hybrid Electric Vehicle (PHEV)	Clean Alternative Fuel Vehicle Eligible	

```
# Impute missing electric range and Base MSRP with 0
num_to_impute = ['Electric Range', 'Base MSRP']
df_clean[num_to_impute] = df_clean[num_to_impute].fillna(0)
```

```
# View the 9 rows that have missing county
df_clean[df_clean['County'].isna()]
```

	VIN (1-10)	County	City	State	Model Year	Make	Model	Electric Vehicle Type	Clean Alternative Fuel Vehicle (CAFV) Eligibility	Electric Range	Base MSRP
211	3FA6P0SU7E	NaN	NaN	AE	2014	FORD	FUSION	Plug-in Hybrid Electric Vehicle (PHEV)	Not eligible due to low battery range	19.00	0.0
133651	7SAYGDEE0P	NaN	NaN	AP	2023	TESLA	MODEL Y	Battery Electric Vehicle (BEV)	Eligibility unknown as battery range has not b...	0.00	0.0
139165	5YJXCAE24H	NaN	NaN	BC	2017	TESLA	MODEL X	Battery Electric Vehicle (BEV)	Clean Alternative Fuel Vehicle Eligible	200.00	0.0
146691	YV4ED3UR8N	NaN	NaN	BC	2022	VOLVO	XC40	Battery Electric Vehicle (BEV)	Eligibility unknown as battery range has not b...	0.00	0.0

```
# Impute missing value in County, City, Postal Code, Electric Utility with 'Unknown'
cat_to_impute = ['County', 'City', 'Electric Utility']
df_clean[cat_to_impute] = df_clean[cat_to_impute].fillna('Unknown')
```

```
# Check missing values after imputation
df_clean.isna().sum()
```

	0
VIN (1-10)	0
County	0
City	0
State	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
DOL Vehicle ID	0
Electric Utility	0

dtype: int64

Appendix C. Step Four Python Code – Explore Data Patterns

Numerical Variables - Electric Range

```
df_clean[['Electric Range']].describe()
```

	Electric Range
count	264628.00
mean	41.71
std	80.38
min	0.00
25%	0.00
50%	0.00
75%	34.00
max	337.00

```
# Filter non-zero electric range
df_non_zero_ER = df_clean[df_clean['Electric Range'] != 0]
```

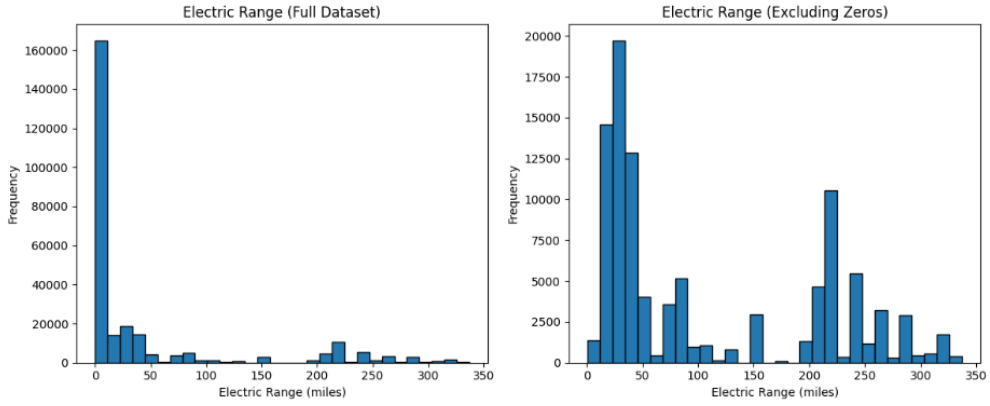
```
df_non_zero_ER[['Electric Range']].describe()
```

	Electric Range
count	100827.00
mean	109.48
std	97.66
min	1.00
25%	30.00
50%	53.00
75%	215.00
max	337.00

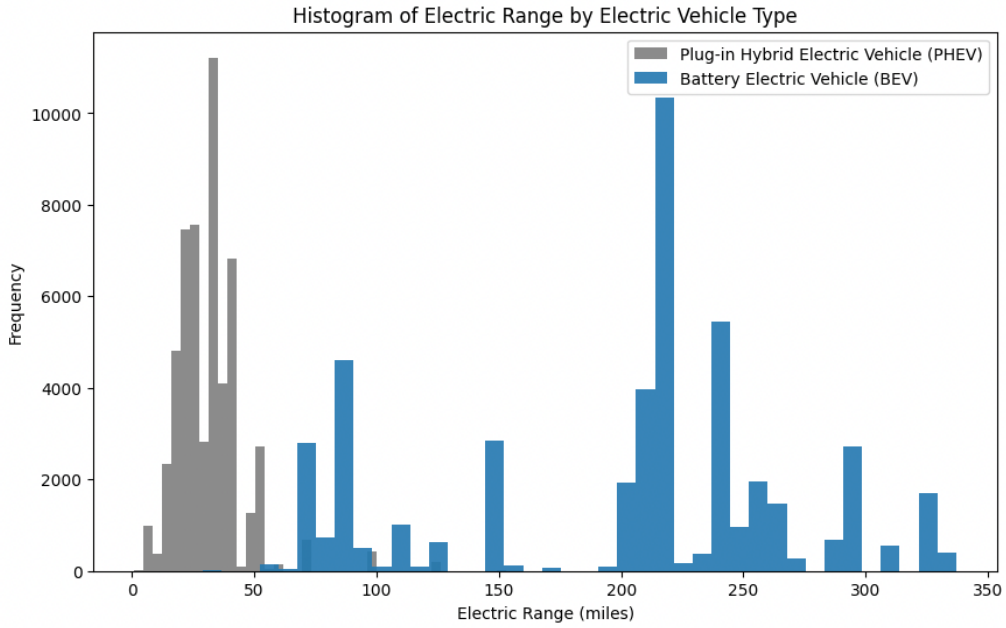
```
# Describe ER by Electric Vehicle Type
df_non_zero_ER['Electric Range'].groupby(df_non_zero_ER['Electric Vehicle Type']).describe()
```

Electric Vehicle Type	count	mean	std	min	25%	50%	75%	max
Battery Electric Vehicle (BEV)	46778.00	199.50	72.16	29.00	150.00	215.00	238.00	337.00
Plug-in Hybrid Electric Vehicle (PHEV)	54049.00	31.56	14.14	1.00	21.00	32.00	38.00	153.00

```
# Histogram
fig, axes = plt.subplots(1, 2, figsize=(12, 5))  # make it wider for readability

# Histogram for Electric Range with 0 data
axes[0].hist(df_clean['Electric Range'], bins=30, edgecolor='black')
axes[0].set_title('Electric Range (Full Dataset)')
axes[0].ticklabel_format(style='plain')  # avoid scientific notation
axes[0].set_xlabel('Electric Range (miles)')
axes[0].set_ylabel('Frequency')

# Histogram for Electric Range
axes[1].hist(df_non_zero_ER['Electric Range'], bins=30, edgecolor='black')
axes[1].set_title('Electric Range (Excluding Zeros)')
axes[1].ticklabel_format(style='plain')  # avoid scientific notation
axes[1].set_xlabel('Electric Range (miles)')
axes[1].set_ylabel('Frequency')

plt.tight_layout()  # adjust spacing
plt.show()
```



```python
# Create histogram by EV type
ev_types = df_non_zero_ER["Electric Vehicle Type"].unique()

# Set the size for the graph
plt.figure(figsize=(10, 6))

# Set the color
colors = ["grey", base_color]

# Draw the graph for each vehicle type
for ev, color in zip(ev_types, colors):
    subset = df_non_zero_ER[df_non_zero_ER["Electric Vehicle Type"] == ev]
    plt.hist(subset["Electric Range"].dropna(),
             alpha=0.9,
             bins=40,
             label=ev,
             color=color)

# Plot lables
plt.xlabel("Electric Range (miles)")
plt.ylabel("Frequency")
plt.title("Histogram of Electric Range by Electric Vehicle Type")
plt.legend()
plt.show()
```



Numerical Variables - Base MSRP

```python
df_clean[['Base MSRP']].describe()
```

	Base MSRP
count	264628.00
mean	678.89
std	6868.87
min	0.00
25%	0.00
50%	0.00
75%	0.00
max	845000.00

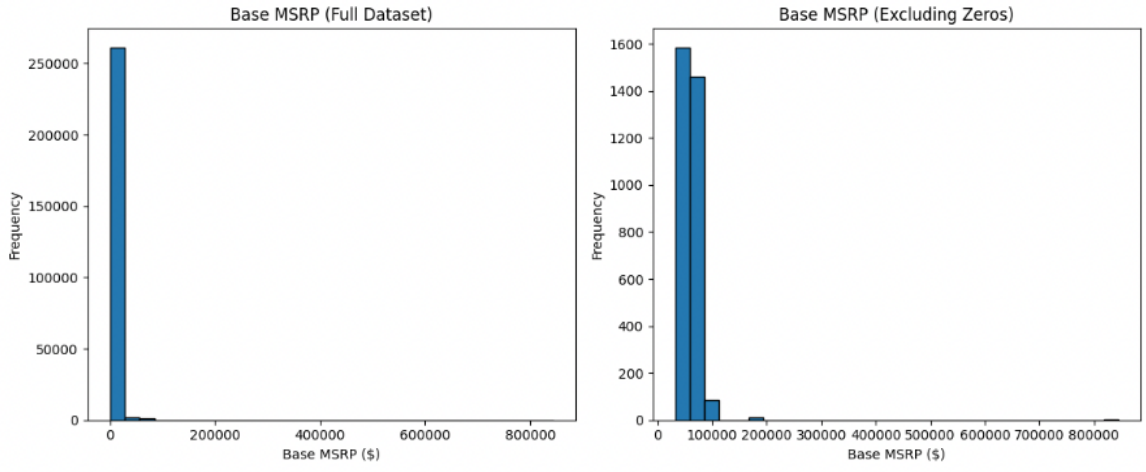
```
# Filter non-zero Base MSRP
df_non_zero_MSRP = df_clean[df_clean['Base MSRP'] != 0]
```

```
# Describe only Base MSRP
df_non_zero_MSRP['Base MSRP'].describe()
```

	Base MSRP
count	3148.00
mean	57069.19
std	27354.07
min	31950.00
25%	39221.25
50%	55700.00
75%	69900.00
max	845000.00

dtype: float64

```
# Histogram
fig, axes = plt.subplots(1, 2, figsize=(12, 5))  # make it wider for readability

# Histogram for Base MSRP
axes[0].hist(df_clean['Base MSRP'], bins=30, edgecolor='black')
axes[0].set_title('Base MSRP (Full Dataset)')
axes[0].ticklabel_format(style='plain')  # avoid scientific notation
axes[0].set_xlabel('Base MSRP ($)')
axes[0].set_ylabel('Frequency')

# Histogram for Base MSRP
axes[1].hist(df_non_zero_MSRP['Base MSRP'], bins=30, edgecolor='black')
axes[1].set_title('Base MSRP (Excluding Zeros)')
axes[1].ticklabel_format(style='plain')  # avoid scientific notation
axes[1].set_xlabel('Base MSRP ($)')
axes[1].set_ylabel('Frequency')

plt.tight_layout()  # adjust spacing
plt.show()
```



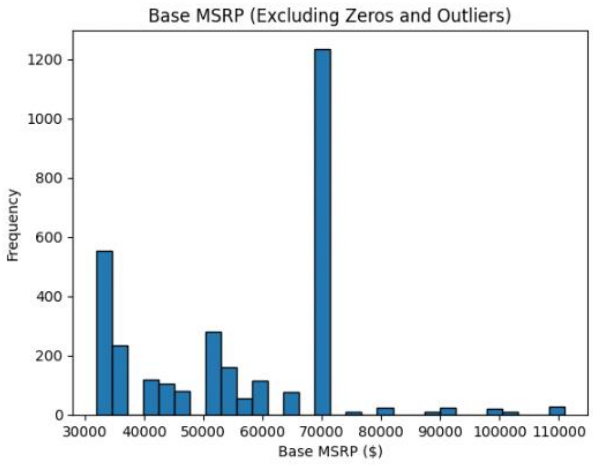

```
# Remove Base MSRP > 150k
df_MSRP_no_outliers = df_non_zero_MSRP[df_non_zero_MSRP['Base MSRP'] <= 150000]
```

```
# Describe Base MSRP after excluding zeros and outliers
df_MSRP_no_outliers['Base MSRP'].describe()
```

	Base MSRP
count	3133.00
mean	56037.86
std	16941.74
min	31950.00
25%	36900.00
50%	55700.00
75%	69900.00
max	110950.00

dtype: float64

```
# Histogram of MSRP wo outliers and zeros
plt.hist(df_MSRP_no_outliers['Base MSRP'], bins=30, edgecolor='black')

plt.title('Base MSRP (Excluding Zeros and Outliers)')
plt.xlabel('Base MSRP ($)')
plt.ylabel('Frequency')
plt.show()
```



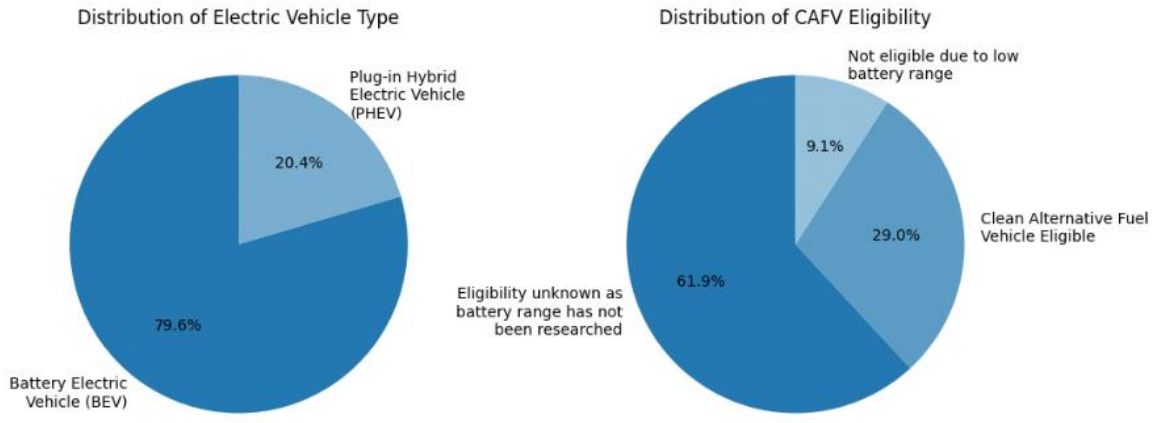
Categorical variables

```
# Transform model year from numerical to categorical
df_clean['Model Year'] = pd.Categorical(df_clean['Model Year'])
```

```
# describe all
df_clean[['County', 'City', 'State', 'Make', 'Model',
          'Electric Vehicle Type',
          'Clean Alternative Fuel Vehicle (CAFV) Eligibility',
          'Electric Utility',
          'Model Year']].describe(include='all').transpose()
```

	count	unique	top	freq
County	264628	240	King	131179
City	264628	859	Seattle	41533
State	264628	51	WA	263969
Make	264628	46	TESLA	108633
Model	264628	182	MODEL Y	55187
Electric Vehicle Type	264628	2	Battery Electric Vehicle (BEV)	210575
Clean Alternative Fuel Vehicle (CAFV) Eligibility	264628	3	Eligibility unknown as battery range has not b...	163797
Electric Utility	264628	77	PUGET SOUND ENERGY INCIICITY OF TACOMA - (WA)	94223
Model Year	264628	22	2023	60157

```
# Draw pie charts for Electric Vehicle Type and CAFV eligibility
fig, axes = plt.subplots(1, 2, figsize=(12, 5))

# Electric Vehicle Type Counts
ev_counts = df['Electric Vehicle Type'].value_counts()

# Set a width for EV pie chart
ev_labels = [textwrap.fill(label, width=18) for label in ev_counts.index]

# Create a list of shades (slightly lighter each time)
n_ev = len(ev_counts)
ev_colors = [lighten(base_color, amount=i*(0.8/n_ev)) for i in range(n_ev)]

# EV pie chart
axes[0].pie(ev_counts, labels=ev_labels, autopct='%1.1f%%', colors=ev_colors, startangle=90)
axes[0].set_title("Distribution of Electric Vehicle Type")

# CAFV eligibility counts
cafv_counts = df['Clean Alternative Fuel Vehicle (CAFV) Eligibility'].value_counts()

# Set a width for EV pie chart
cafv_labels = [textwrap.fill(label, width=24) for label in cafv_counts.index]

# CAFV colors
n_cafv = len(cafv_counts)
cafv_colors = [lighten(base_color, amount=i*(0.8/n_cafv)) for i in range(n_cafv)]

# CAFV pie chart
axes[1].pie(cafv_counts, labels=cafv_labels, autopct='%1.1f%%', colors=cafv_colors, startangle=90)
axes[1].set_title("Distribution of CAFV Eligibility")

plt.show()
```




```
# Describe ER by Electric Vehicle Type
df_clean['Electric Range'].groupby(df_clean['Clean Alternative Fuel Vehicle (CAFV) Eligibility']).descri
```

Clean Alternative Fuel Vehicle (CAFV) Eligibility	count	mean	std	min	25%	50%	75%	max
Clean Alternative Fuel Vehicle Eligible	76677.00	137.44	96.27	0.00	40.00	97.00	220.00	337.00
Eligibility unknown as battery range has not been researched	163797.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Not eligible due to low battery range	24154.00	20.69	4.91	1.00	18.00	21.00	25.00	29.00

Appendix D. Step Five Python Code – Transform Data

- Convert Model Year from numerical to categorical in previous step
- Create new variables

```
# Create a column City + County
df_clean['City_County'] = df_clean['City'] + '_' + df_clean['County']
```

```
# Create a column Model + Make
df_clean['Model_Make'] = df_clean['Model'] + '_' + df_clean['Make']
```

- Encode categorical variables for correlation matrix

```
# Mapping Electric Vehicle Type
mapping_EV = {'Battery Electric Vehicle (BEV)': 1, 'Plug-in Hybrid Electric Vehicle (PHEV)': 2}

df_clean['EV_Type_Code'] = df_clean['Electric Vehicle Type'].map(mapping_EV)
df_non_zero_MSRP.loc[:,'EV_Type_Code'] = df_non_zero_MSRP['Electric Vehicle Type'].map(mapping_EV)
df_MSRP_no_outliers.loc[:,'EV_Type_Code'] = df_MSRP_no_outliers['Electric Vehicle Type'].map(mapping_EV)
```

```
# Mapping CAFV eligibility
mapping_CAFV = {'Clean Alternative Fuel Vehicle Eligible': 1,
                'Not eligible due to low battery range': 2,
                'Eligibility unknown as battery range has not been researched': 3}

df_clean['CAFV_Code'] = df_clean['Clean Alternative Fuel Vehicle (CAFV) Eligibility'].map(mapping_CAFV)
df_non_zero_MSRP.loc[:,'CAFV_Code'] = df_non_zero_MSRP['Clean Alternative Fuel Vehicle (CAFV) Eligibilit
df_MSRP_no_outliers.loc[:,'CAFV_Code'] = df_MSRP_no_outliers['Clean Alternative Fuel Vehicle (CAFV) Elig
```

```
# Mapping Make
mapping_make = {'AUDI':1, 'NISSAN':2, 'TESLA':3, 'KIA':4, 'HYUNDAI':5, 'JEEP':6, 'BMW':7, 'FORD':8,
       'LINCOLN':9, 'SMART':10, 'LEXUS':11, 'RIVIAN':12, 'CHRYSLER':13, 'CHEVROLET':14,
       'HONDA':15, 'VOLVO':16, 'VOLKSWAGEN':17, 'TOYOTA':18, 'FIAT':19, 'POLESTAR':20,
       'PORSCHE':21, 'SUBARU':22, 'ACURA':23, 'MAZDA':24, 'MERCEDES-BENZ':25, 'GENESIS':26,
       'DODGE':27, 'GMC':28, 'MINI':29, 'CADILLAC':30, 'LUCID':31, 'FISKER':32,
       'LAND ROVER':33, 'MITSUBISHI':34, 'ALFA ROMEO':35, 'RAM':36, 'ROLLS-ROYCE':37,
       'JAGUAR':38, 'VINFAST':39, 'LAMBORGHINI':40, 'BRIGHTDROP':41, 'AZURE DYNAMICS':42,
       'MULLEN AUTOMOTIVE INC.':43, 'BENTLEY':44, 'TH!NK':45,
       'WHEEGO ELECTRIC CARS':46}

df_clean['Make_Code'] = df_clean['Make'].map(mapping_make)
df_non_zero_MSRP.loc[:,'Make_Code'] = df_non_zero_MSRP['Make'].map(mapping_make)
df_MSRP_no_outliers.loc[:,'Make_Code'] = df_MSRP_no_outliers['Make'].map(mapping_make)
```

```
# Mapping Model
mapping_model = {'Q5 E':1, 'A3':2, 'LEAF':3, 'E-TRON':4, 'MODEL X':5, 'SOUL':6, 'NIRO':7,
       'TUCSON':8, 'WRANGLER':9, 'MODEL Y':10, 'MODEL S':11, 'MODEL 3':12, 'X5':13,
       'C-MAX':14, 'AVIATOR':15, 'FORTWO':16, 'NX':17, 'SPORTAGE':18, 'I3':19, 'R1T':20,
       'PACIFICA':21, 'VOLT':22, 'BOLT EV':23, 'FUSION':24, 'CLARITY':25, 'ESCAPE':26,
       'V60':27, 'EV6':28, 'EQ FORTWO':29, 'R1S':30, 'E-GOLF':31, 'EX90':32,
       'RAV4 PRIME (PHEV)':33, 'ID.4':34, 'KONA':35, 'PROLOGUE':36, 'GRAND CHEROKEE':37,
       '500':38, 'PS2':39, 'SORENTO':40, 'Q5':41, 'EV9':42, 'PRIUS PRIME (PHEV)':43, 'Q4':44,
       'XC90':45, 'I4':46, 'IX':47, 'PANAMERA':48, '740E':49, 'F-150':50, 'SOLTERRA':51,
       'IONIQ 5':52, 'XM':53, 'XC60':54, 'BZ4X':55, 'ZDX':56, 'FOCUS':57, '330E':58, 'SPARK':59,
       'CX-90':60, '530E':61, 'C40':62, 'CYBERTRUCK':63, 'B-CLASS':64, 'XC40':65, 'X3':66,
       'I8':67, 'GLE-CLASS':68, 'CAYENNE':69, '500E':70, 'BOLT EUV':71, 'EX30':72, 'GV70':73,
       'IONIQ 6':74, 'IONIQ 9':75, 'MUSTANG MACH-E':76, 'ROADSTER':77, 'PRIUS':78,
       'CHARGER':79, 'TAYCAN':80, 'HUMMER EV PICKUP':81, 'TRANSIT':82, 'RS E-TRON GT':83,
       'HARDTOP':84, 'ID. BUZZ':85, 'GLC-CLASS':86, 'RZ':87, 'LYRIQ':88, 'BLAZER':89,
       'E-TRON GT':90, 'S60':91, 'AIR':92, 'OCEAN':93, 'EQE-CLASS SUV':94, 'S-CLASS':95,
       'I5':96, 'CORSAIR':97, 'HORNET':98, 'ARIYA HATCHBACK':99, 'RAV4':100,
       'RANGE ROVER':101, 'OUTLANDER':102, 'EQUINOX':103, 'Q6':104, 'TONALE':105, 'OPTIQ':106,
       'MACAN':107, 'ARIYA MPV':108, 'OPTIMA':109, 'EQB-CLASS':110, 'EQS-CLASS SUV':111,
       'PROMASTER 3500':112, 'WAGONEER S':113, 'COUNTRYMAN':114, 'EX40':115, 'GV60':116, 'Q8':117,
       'SANTA FE':118, 'CX-70':119, 'IONIQ':120, 'HUMMER EV SUV':121, 'EQE-CLASS SEDAN':122,
       'SIERRA':123, 'EQS-CLASS SEDAN':124, 'SQ6':125, 'SONATA':125,
       'FORTWO ELECTRIC DRIVE':127, 'CROSSTREK':128, '550E':129, 'SILVERADO':130, 'RX':131,
       'VISTIQ':132, 'M5':133, 'E-TRON SPORTBACK':134, 'E-CLASS':135, 'RANGE ROVER SPORT':136,
       'G-CLASS':137, 'POLESTAR 3':138, 'S E-TRON GT':139, 'SPECTRE':140, 'I-PACE':141,
       'ACCORD':142, 'I-MIEV':143, 'MX-30':144, 'TX':145, 'A6':146, 'GRAVITY':147, 'KARMA':148,
       '745E':149, 'SQ8':150, 'VF 8':151, 'BRIGHTDROP':152, 'REVUELTO':153, 'ELR':154, 'AMG GT':155,
       'C-CLASS':156, 'I7':157, 'S6':158, 'EDV':159, 'S90':160, 'CT6':161, 'ZEVO':162, 'IONIQ 5 N':163,
       'RANGER':164, 'URUS':165, 'RCV':166, 'ESPRINTER':167, 'G80':168,
```

```
        'TRANSIT CONNECT ELECTRIC':169, 'SL-CLASS':170, 'ONE':171, '918':172, 'BENTAYGA':173,
        'A7 E':174, '750E':175, '745LE':176, 'CITY':177, 'FLYING SPUR':178, 'WHEEGO':179, 'A8 E':180,
        'CONTINENTAL':181, 'MIRAI':182}

df_clean['Model_Code'] = df_clean['Model'].map(mapping_model)
df_non_zero_MSRP.loc[:, 'Model_Code'] = df_non_zero_MSRP['Model'].map(mapping_model)
df_MSRP_no_outliers.loc[:, 'Model_Code'] = df_MSRP_no_outliers['Model'].map(mapping_model)
```

```
# Mapping county
mapping_county = {'Thurston':1, 'Yakima':2, 'King':3, 'Snohomish':4, 'Kitsap':5, 'Sauk':6,
       'Island':7, 'Fresno':8, 'Whitman':9, 'Spokane':10, 'Skagit':11, 'Walla Walla':12,
       'Douglas':13, 'Chelan':14, 'Kittitas':15, 'Unknown':16, 'Stevens':17, 'Grant':18,
       'Clark':19, 'Bannock':20, 'Lewis':21, 'Pierce':22, 'Benton':23, 'Whatcom':24,
       'Clallam':25, 'San Juan':26, 'Jefferson':27, 'Mason':28, 'Pend Oreille':29,
       'Cowlitz':30, 'Klickitat':31, 'Grays Harbor':32, 'Pacific':33, 'Adams':34,
       'Franklin':35, 'Okanogan':36, 'San Diego':37, 'Garfield':38, 'Ferry':39,
       'Lincoln':40, 'Wahkiakum':41, 'Skamania':42, 'Asotin':43, 'El Paso':44,
       'Los Angeles':45, 'Riverside':46, 'Anne Arundel':47, "Prince George's":48,
       'Churchill':49, 'Gwinnett':50, 'Columbia':51, 'Hillsborough':52, 'Miami-Dade':53,
       'Fairfax':54, 'New York':55, 'Orange':56, 'Forsyth':57, 'Arlington':58, 'Marin':59,
       'Philadelphia':60, 'Guam':61, 'Ventura':62, 'Maricopa':63, 'Multnomah':64,
       'Chesapeake':65, 'St. Charles':66, "St. Mary's":67, 'Cook':68, 'Kings':69,
       'Cuyahoga':70, 'Alexandria':71, 'Leavenworth':72, 'Middlesex':73,
       'District of Columbia':74, 'Collin':75, 'San Bernardino':76, 'Miller':77,
       'Allegheny':78, 'Harris':79, 'Pinal':80, 'Shelby':81, 'Bucks':82, 'Kent':83,
       'Williamson':84, 'Duval':85, 'Berkeley':86, 'Okaloosa':87, 'Hoke':88, 'Suffolk':89,
       'Harnett':90, 'Cumberland':91, 'Saratoga':92, 'Calvert':93, 'Montgomery':94,
       'Yuba':95, 'Denver':96, 'Polk':97, 'Santa Clara':98, 'York':99, 'Lake':100, 'DeKalb':101,
       'Maui':102, 'Niagara':103, 'Contra Costa':104, 'Bexar':105, 'Harford':106, 'Washoe':107,
       'Travis':108, 'Milwaukee':109, 'Sarpy':110, 'DuPage':111, 'Monterey':112, 'Charles':113,
       'Sussex':114, 'Hamilton':115, 'Ada':116, 'Alameda':117, 'Newport':118, 'Palm Beach':119,
       'Galveston':120, 'Elmore':121, 'James City':122, 'Albemarle':123, 'Henrico':124,
       'Broward':125, 'Washington':126, 'Virginia Beach':127, 'Centre':128, 'Talladega':129,
       'Kootenai':130, 'Olmsted':131, 'Mercer':132, 'Prince William':133, 'Weber':134,
       'Ocean':135, 'Platte':136, 'Bell':137, 'Barnstable':138, 'Autauga':139, 'Richmond':140,
       'Placer':141, 'San Mateo':142, 'Pulaski':143, 'Clackamas':144, 'Honolulu':145,
       'Salt Lake':145, 'Cochise':146, 'Stafford':147, 'Norfolk':148, 'Portsmouth':149,
       'Kane':150, 'Geary':151, 'Escambia':152, 'Monmouth':153, 'Beaver':154, 'Nueces':155,
       'Camden':156, 'Currituck':157, 'Anchorage':158, 'Santa Cruz':159, 'Schuylkill':160,
       'Hartford':161, 'Tooele':162, 'Comanche':163, 'Loudoun':164, 'Washtenaw':165, 'Riley':166,
       'Penobscot':167, 'Davidson':168, 'Frederick':169, 'Stanislaus':170, 'Atlantic':171,
       'Manatee':172, 'Manassas':173, 'Meade':174, 'Sacramento':175, 'Johnson':176,
       'Muscogee':177, 'Dallas':178, 'Hudson':179, 'St. Louis':180, 'San Luis Obispo':181,
       'Lexington':182, 'Sonoma':183, 'Marion':184, 'Denton':185, 'Weld':186, 'Greene':187,
       'Cobb':188, 'Laramie':189, 'Oldham':190, 'Otero':191, 'Caddo':192, 'Macomb':193,
       'La Plata':194, 'San Francisco':195, 'Burlington':196, 'Allen':197, 'Larimer':198,
       'Fort Bend':199, 'Jackson':200, 'Wake':201, 'Dale':202, 'New London':203, 'Moore':204,
       'St. Clair':205, 'Pima':206, 'Howard':207, 'Lee':208, 'Arapahoe':209, 'Solano':210, 'Knox':211,
       'Rockingham':212, 'St. Landry':213, 'Brown':214, 'Beaufort':215, 'Brevard':216,
       'Kauai':217, 'Texas':218, 'Saginaw':219, 'Nassau':220, 'Wayne':221, 'Carroll':222,
       'Charlottesville':223, 'Clay':224, 'Sarasota':225, 'Tarrant':226, 'Cleveland':227,
       'Prince George':228, 'Medina':229, 'Essex':230, 'Monroe':231, 'Kern':232, 'Houston':233,
       'Hardin':234, 'Hennepin':235, 'Bristol':236, 'Madison':237, 'Nye':238, 'Osceola':239}

df_clean['County_Code'] = df_clean['County'].map(mapping_county)
df_non_zero_MSRP.loc[:, 'County_Code'] = df_non_zero_MSRP['County'].map(mapping_county)
df_MSRP_no_outliers.loc[:, 'County_Code'] = df_MSRP_no_outliers['County'].map(mapping_county)
```

```
# Mapping electric utility
mapping_utility = {'PUGET SOUND ENERGY INC':1, 'PACIFICORP':2,
       'CITY OF SEATTLE - (WA)|CITY OF TACOMA - (WA)':3,
       'PUGET SOUND ENERGY INC||CITY OF TACOMA - (WA)':4,
       'NON WASHINGTON STATE ELECTRIC UTILITY':5, 'AVISTA CORP':6,
       'MODERN ELECTRIC WATER COMPANY':7, 'PUD NO 1 OF DOUGLAS COUNTY':8,
       'PUD NO 1 OF CHELAN COUNTY':9, 'Unknown':10, 'PUD NO 2 OF GRANT COUNTY':11,
       'PORTLAND GENERAL ELECTRIC CO':12,
       'BONNEVILLE POWER ADMINISTRATION||PUD NO 1 OF CLARK COUNTY - (WA)':13,
       'BONNEVILLE POWER ADMINISTRATION||CITY OF CENTRALIA - (WA)|CITY OF TACOMA - (WA)':14,
       'BONNEVILLE POWER ADMINISTRATION||CITY OF TACOMA - (WA)||PENINSULA LIGHT COMPANY':15,
       'BONNEVILLE POWER ADMINISTRATION||PUD NO 1 OF BENTON COUNTY':16,
       'PUGET SOUND ENERGY INC||PUD NO 1 OF WHATCOM COUNTY':17,
       'BONNEVILLE POWER ADMINISTRATION||PUD NO 1 OF CLALLAM COUNTY':18,
       'BONNEVILLE POWER ADMINISTRATION||ORCAS POWER & LIGHT COOP':19,
       'BONNEVILLE POWER ADMINISTRATION||AVISTA CORP||INLAND POWER & LIGHT COMPANY':20,
       'BONNEVILLE POWER ADMINISTRATION||CITY OF TACOMA - (WA)||OHOP MUTUAL LIGHT COMPANY, INC|PENINSULA
       'BONNEVILLE POWER ADMINISTRATION||PUGET SOUND ENERGY INC||PUD NO 1 OF JEFFERSON COUNTY':22,
       'BONNEVILLE POWER ADMINISTRATION||CITY OF RICHLAND - (WA)':23,
       'BONNEVILLE POWER ADMINISTRATION||CITY OF TACOMA - (WA)||PUD NO 3 OF MASON COUNTY':24,
```

```
        'BONNEVILLE POWER ADMINISTRATION||PUD 1 OF SNOHOMISH COUNTY':25,
        'PUD NO 1 OF PEND OREILLE COUNTY':26,
        'CITY OF TACOMA - (WA)||TANNER ELECTRIC COOP':27,
        'BONNEVILLE POWER ADMINISTRATION||PUD NO 1 OF COWLITZ COUNTY':28,
        'BONNEVILLE POWER ADMINISTRATION||PUD NO 1 OF KLICKITAT COUNTY':29,
        'BONNEVILLE POWER ADMINISTRATION||PUD NO 1 OF GRAYS HARBOR COUNTY':30,
        'BONNEVILLE POWER ADMINISTRATION||PUD NO 2 OF PACIFIC COUNTY':31,
        'BONNEVILLE POWER ADMINISTRATION||TOWN OF STEILACOOM|CITY OF TACOMA - (WA)||PENINSULA LIGHT COMPA
        'BONNEVILLE POWER ADMINISTRATION||PACIFICORP||PUD NO 1 OF CLARK COUNTY - (WA)':33,
        'BONNEVILLE POWER ADMINISTRATION||AVISTA CORP||BIG BEND ELECTRIC COOP, INC':34,
        'BONNEVILLE POWER ADMINISTRATION||CITY OF TACOMA - (WA)||ELMHURST MUTUAL POWER & LIGHT CO|PENINSU
        'BONNEVILLE POWER ADMINISTRATION||PUD NO 1 OF FRANKLIN COUNTY':36,
        'NO KNOWN ELECTRIC UTILITY SERVICE':37,
        'OKANOGAN COUNTY ELEC COOP, INC':38,
        'BONNEVILLE POWER ADMINISTRATION||CITY OF TACOMA - (WA)||LAKEVIEW LIGHT & POWER|PENINSULA LIGHT C
        'BONNEVILLE POWER ADMINISTRATION||VERA IRRIGATION DISTRICT #15':40,
        'BONNEVILLE POWER ADMINISTRATION||CITY OF TACOMA - (WA)||PARKLAND LIGHT & WATER COMPANY|PENINSULA
        'BONNEVILLE POWER ADMINISTRATION||CITY OF ELLENSBURG - (WA)':42,
        'BONNEVILLE POWER ADMINISTRATION||CITY OF TACOMA - (WA)||PUD NO 1 OF LEWIS COUNTY':43,
        'BONNEVILLE POWER ADMINISTRATION||TOWN OF EATONVILLE - (WA)|CITY OF TACOMA - (WA)':44,
        'BONNEVILLE POWER ADMINISTRATION||INLAND POWER & LIGHT COMPANY':45,
        'BONNEVILLE POWER ADMINISTRATION||PUD NO 1 OF FERRY COUNTY':46,
        'BONNEVILLE POWER ADMINISTRATION||CITY OF TACOMA - (WA)||BENTON RURAL ELECTRIC ASSN|PENINSULA LIG
        'CITY OF BLAINE - (WA)||PUD NO 1 OF WHATCOM COUNTY':48,
        'BONNEVILLE POWER ADMINISTRATION||PACIFICORP||BENTON RURAL ELECTRIC ASSN':49,
        'BONNEVILLE POWER ADMINISTRATION||PUD NO 1 OF MASON COUNTY|PUD NO 1 OF JEFFERSON COUNTY':50,
        'BONNEVILLE POWER ADMINISTRATION||CITY OF MILTON - (WA)|CITY OF TACOMA - (WA)':51,
        'BONNEVILLE POWER ADMINISTRATION||CITY OF PORT ANGELES - (WA)':52,
        'BONNEVILLE POWER ADMINISTRATION||BIG BEND ELECTRIC COOP, INC':53,
        'PUD NO 1 OF WHATCOM COUNTY':54, 'PUD NO 1 OF OKANOGAN COUNTY':55,
        'BONNEVILLE POWER ADMINISTRATION||PUD NO 1 OF WAHKIAKUM COUNTY':56,
        'BONNEVILLE POWER ADMINISTRATION||PUD NO 1 OF SKAMANIA CO':57,
        'CITY OF TACOMA - (WA)':58,
        'BONNEVILLE POWER ADMINISTRATION||CITY OF TACOMA - (WA)||PUD NO 1 OF MASON COUNTY':59,
        'BONNEVILLE POWER ADMINISTRATION||AVISTA CORP||PUD NO 1 OF ASOTIN COUNTY':60,
        'BONNEVILLE POWER ADMINISTRATION||TOWN OF RUSTON - (WA)|CITY OF TACOMA - (WA)||PENINSULA LIGHT CO
        'CITY OF SUMAS - (WA)||PUD NO 1 OF WHATCOM COUNTY':62,
        'BONNEVILLE POWER ADMINISTRATION||CITY OF COULEE DAM - (WA)':63,
        'CITY OF CHENEY - (WA)':64,
        'BONNEVILLE POWER ADMINISTRATION||PUD NO 1 OF KITTITAS COUNTY':65,
        'CITY OF SEATTLE - (WA)':66,
        'BONNEVILLE POWER ADMINISTRATION||CITY OF MCCLEARY - (WA)':67,
        'BONNEVILLE POWER ADMINISTRATION||PACIFICORP||COLUMBIA RURAL ELEC ASSN, INC':68,
        'BONNEVILLE POWER ADMINISTRATION||COLUMBIA RURAL ELEC ASSN, INC':69,
        'BONNEVILLE POWER ADMINISTRATION||BENTON RURAL ELECTRIC ASSN':70,
        'CITY OF CHEWELAH':71,
        'BONNEVILLE POWER ADMINISTRATION||NESPELEM VALLEY ELEC COOP, INC':72,
        'BONNEVILLE POWER ADMINISTRATION||PUD NO 1 OF ASOTIN COUNTY':73,
        'BONNEVILLE POWER ADMINISTRATION||CITY OF TACOMA - (WA)||ALDER MUTUAL LIGHT CO, INC|PENINSULA LIG
        'BONNEVILLE POWER ADMINISTRATION||PUD NO 1 OF ASOTIN COUNTY||INLAND POWER & LIGHT COMPANY':75,
        'BONNEVILLE POWER ADMINISTRATION||PUD NO 1 OF JEFFERSON COUNTY':76,
        'BONNEVILLE POWER ADMINISTRATION||PENINSULA LIGHT COMPANY':77}

df_clean['Electric Utility_Code'] = df_clean['Electric Utility'].map(mapping_utility)
df_non_zero_MSRP.loc[:, 'Electric Utility_Code'] = df_non_zero_MSRP['Electric Utility'].map(mapping_util
df_MSRP_no_outliers.loc[:, 'Electric Utility_Code'] = df_MSRP_no_outliers['Electric Utility'].map(mappin
```

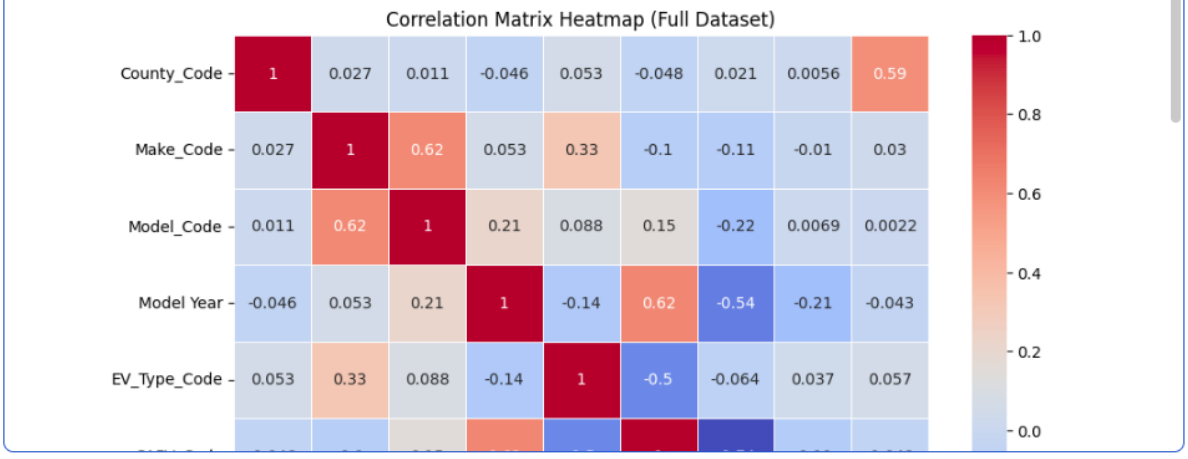
Appendix E. Step Six Python Code – Visualize Correlations

1. Correlation Matrix

- Full Dataset

```
# correlation matrix
correlation_matrix = df_clean[['County_Code',
                               'Make_Code',
                               'Model_Code',
                               'Model Year',
                               'EV_Type_Code',
                               'CAFV_Code',
                               'Electric Range',
                               'Base MSRP',
                               'Electric Utility_Code']].corr()
print("\nCorrelation Matrix (Full Dataset)")
print(correlation_matrix)

# plot correlation matrix
plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', linewidths=0.5)
plt.title('Correlation Matrix Heatmap (Full Dataset)')
plt.show()
```

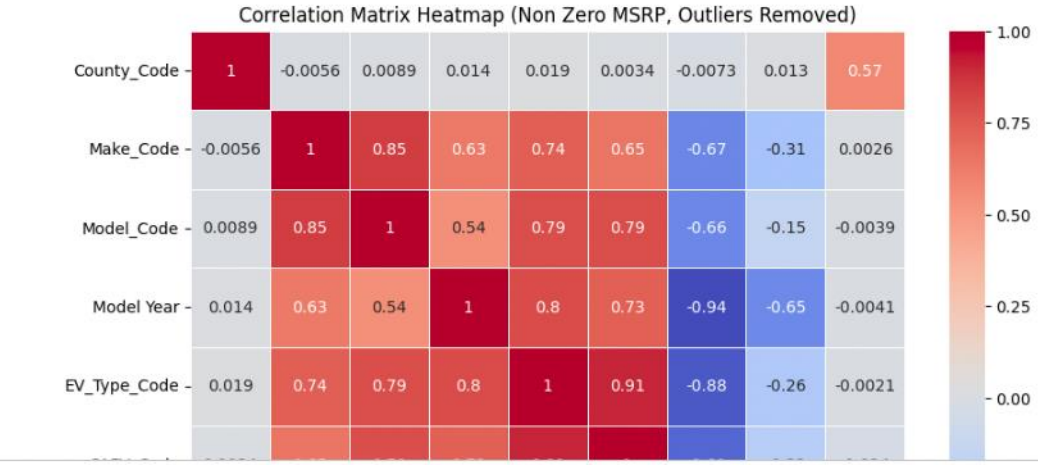
```
Correlation Matrix (Full Dataset)
                       County_Code  Make_Code  Model_Code  Model Year  \
County_Code                   1.00       0.03        0.01       -0.05
Make_Code                     0.03       1.00        0.62        0.05
Model_Code                    0.01       0.62        1.00        0.21
Model Year                   -0.05       0.05        0.21        1.00
EV_Type_Code                  0.05       0.33        0.09       -0.14
CAFV_Code                    -0.05      -0.10        0.15        0.62
Electric Range                0.02      -0.11       -0.22       -0.54
Base MSRP                     0.01      -0.01        0.01       -0.21
Electric Utility_Code         0.59       0.03        0.00       -0.04

                       EV_Type_Code  CAFV_Code  Electric Range  Base MSRP  \
County_Code                    0.05      -0.05            0.02       0.01
Make_Code                      0.33      -0.10           -0.11      -0.01
Model_Code                     0.09       0.15           -0.22       0.01
Model Year                    -0.14       0.62           -0.54      -0.21
EV_Type_Code                   1.00      -0.50           -0.06       0.04
CAFV_Code                     -0.50       1.00           -0.74      -0.11
Electric Range                -0.06      -0.74            1.00       0.11
Base MSRP                      0.04      -0.11            0.11       1.00
Electric Utility_Code          0.06      -0.05            0.02       0.00

                       Electric Utility_Code
County_Code                             0.59
Make_Code                               0.03
Model_Code                              0.00
Model Year                             -0.04
EV_Type_Code                            0.06
CAFV_Code                              -0.05
Electric Range                          0.02
Base MSRP                               0.00
Electric Utility_Code                   1.00
```



- Non zero Base MSRP, outliers removed

```
# correlation matrix
correlation_matrix = df_MSRP_no_outliers[['County_Code',
                                'Make_Code',
                                'Model_Code',
                                'Model Year',
                                'EV_Type_Code',
                                'CAFV_Code',
                                'Electric Range',
                                'Base MSRP',
                                'Electric Utility_Code']].corr()
print("\nCorrelation Matrix (Non Zero MSRP, Outliers Removed)")
print(correlation_matrix)

# plot correlation matrix
plt.figure(figsize=(10, 8))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', linewidths=0.5)
plt.title('Correlation Matrix Heatmap (Non Zero MSRP, Outliers Removed)')
plt.show()
```

```
Correlation Matrix (Non Zero MSRP, Outliers Removed)
                     County_Code  Make_Code  Model_Code  Model Year  \
County_Code                 1.00      -0.01        0.01        0.01
Make_Code                  -0.01       1.00        0.85        0.63
Model_Code                  0.01       0.85        1.00        0.54
Model Year                  0.01       0.63        0.54        1.00
EV_Type_Code                0.02       0.74        0.79        0.80
CAFV_Code                   0.00       0.65        0.79        0.73
Electric Range             -0.01      -0.67       -0.66       -0.94
Base MSRP                   0.01      -0.31       -0.15       -0.65
Electric Utility_Code       0.57       0.00       -0.00       -0.00

                     EV_Type_Code  CAFV_Code  Electric Range  Base MSRP  \
County_Code                  0.02       0.00           -0.01       0.01
Make_Code                    0.74       0.65           -0.67      -0.31
Model_Code                   0.79       0.79           -0.66      -0.15
Model Year                   0.80       0.73           -0.94      -0.65
EV_Type_Code                 1.00       0.91           -0.88      -0.26
CAFV_Code                    0.91       1.00           -0.81      -0.22
Electric Range              -0.88      -0.81            1.00       0.60
Base MSRP                   -0.26      -0.22            0.60       1.00
Electric Utility_Code       -0.00      -0.02            0.02       0.02

                     Electric Utility_Code
County_Code                           0.57
Make_Code                             0.00
Model_Code                           -0.00
Model Year                           -0.00
EV_Type_Code                         -0.00
CAFV_Code                            -0.02
Electric Range                        0.02
Base MSRP                             0.02
Electric Utility_Code                 1.00
```



2. Relationship between variables

1. What is the distribution of electric vehicle types (BEV vs. PHEV) across the state?

```
# Count by Ev type
df_clean['Electric Vehicle Type'].value_counts()
```

Electric Vehicle Type	count
Battery Electric Vehicle (BEV)	210575
Plug-in Hybrid Electric Vehicle (PHEV)	54053

dtype: int64

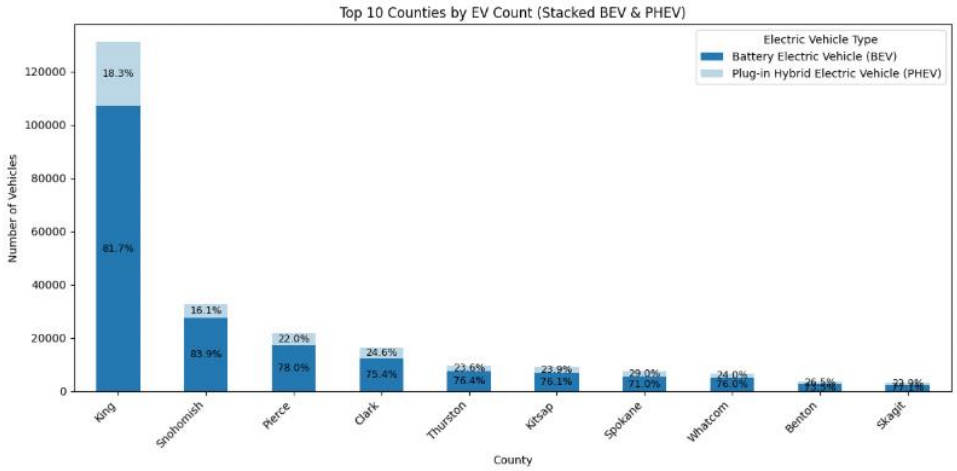
```
# Calculate percentages
df_clean['Electric Vehicle Type'].value_counts(normalize=True) * 100
```

Electric Vehicle Type	proportion
Battery Electric Vehicle (BEV)	79.57
Plug-in Hybrid Electric Vehicle (PHEV)	20.43

dtype: float64

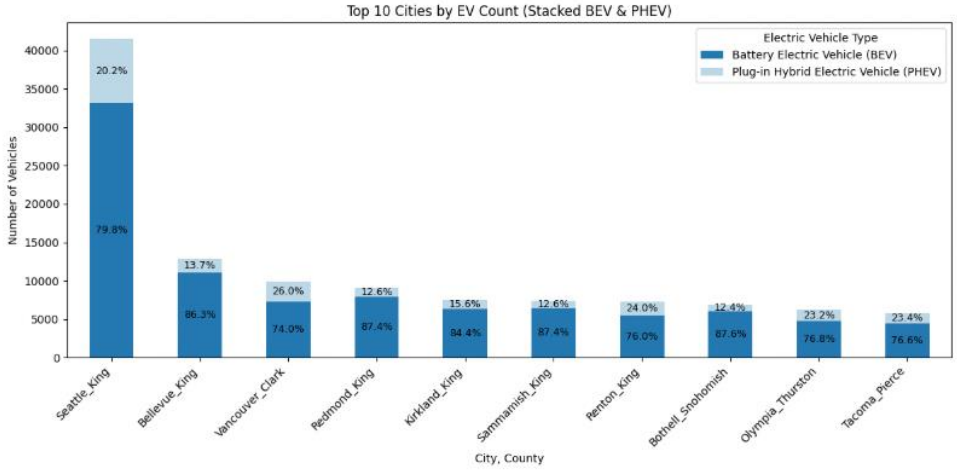
```
# Count EV types per county → rows = counties, columns = BEV/PHEV
counts = df_clean.groupby(['County', 'Electric Vehicle Type']).size().unstack(fill_value=0)

# Get top 10 counties by total EV count
top10 = counts.sum(axis=1).sort_values(ascending=False).head(10)
counts_top10 = counts.loc[top10.index]

plt.figure(figsize=(12, 6))

ax = counts_top10.plot(
    kind='bar',
    stacked=True,
    figsize=(12, 6),
    color=[base_color, lighten(base_color, amount=0.7)]
)

# ----- Add percentage labels -----
for i, county in enumerate(counts_top10.index):
    total = counts_top10.loc[county].sum()
    bottom = 0

    for ev_type in counts_top10.columns:
        value = counts_top10.loc[county, ev_type]
        pct = value / total

        # Only label if segment is > 3% to avoid clutter
        if pct > 0.03:
            ax.text(
                i,                      # x position
                bottom + value/2,       # y position (middle of segment)
                f"{pct*100:.1f}%",      # percentage text
                ha='center', va='center',
                color='black', fontsize=9
            )

        bottom += value

plt.xlabel("County")
plt.ylabel("Number of Vehicles")
plt.title("Top 10 Counties by EV Count (Stacked BEV & PHEV)")
plt.xticks(rotation=45, ha='right')
plt.legend(title="Electric Vehicle Type")
plt.tight_layout()

plt.show()
```

```
<Figure size 1200x600 with 0 Axes>
```




```python
# Count EV types per county → rows = counties, columns = BEV/PHEV
counts = df_clean.groupby(['City_County', 'Electric Vehicle Type']).size().unstack(fill_value=0)

# Get top 10 counties by total EV count
top10 = counts.sum(axis=1).sort_values(ascending=False).head(10)
counts_top10 = counts.loc[top10.index]

plt.figure(figsize=(12, 6))

ax = counts_top10.plot(
    kind='bar',
    stacked=True,
    figsize=(12, 6),
    color=[base_color, lighten(base_color, amount=0.7)]
)

# percentage labels
for i, city in enumerate(counts_top10.index):
    total = counts_top10.loc[city].sum()
    bottom = 0

    for ev_type in counts_top10.columns:
        value = counts_top10.loc[city, ev_type]
        pct = value / total

        # Label only if > 3% (remove this condition if you want ALL labels)
        if pct > 0.03:
            ax.text(
                i,                          # x position
                bottom + value / 2,         # y position (middle of bar segment)
                f"{pct*100:.1f}%",          # percentage text
                ha='center', va='center',
                fontsize=9, color='black'
            )

        bottom += value

plt.xlabel("City, County")
plt.ylabel("Number of Vehicles")
plt.title("Top 10 Cities by EV Count (Stacked BEV & PHEV)")
plt.xticks(rotation=45, ha='right')
plt.legend(title="Electric Vehicle Type")
plt.tight_layout()

plt.show()
```

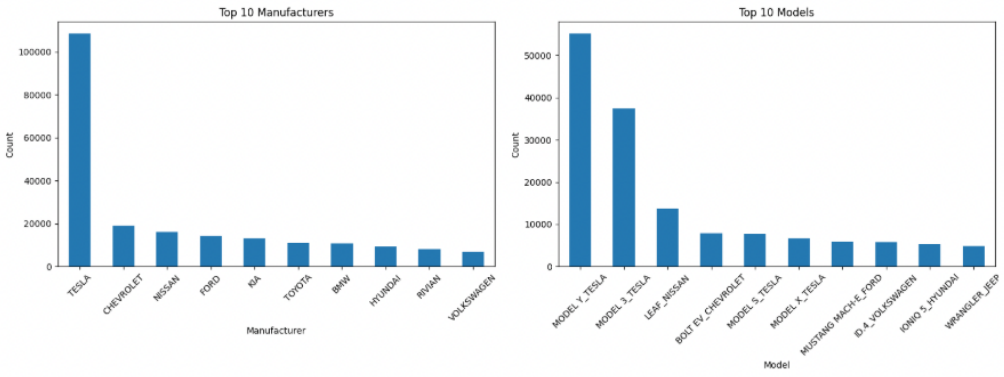
```
<Figure size 1200x600 with 0 Axes>
```



2. Which manufacturers and models dominate the EV market, and how do these patterns vary geographically?

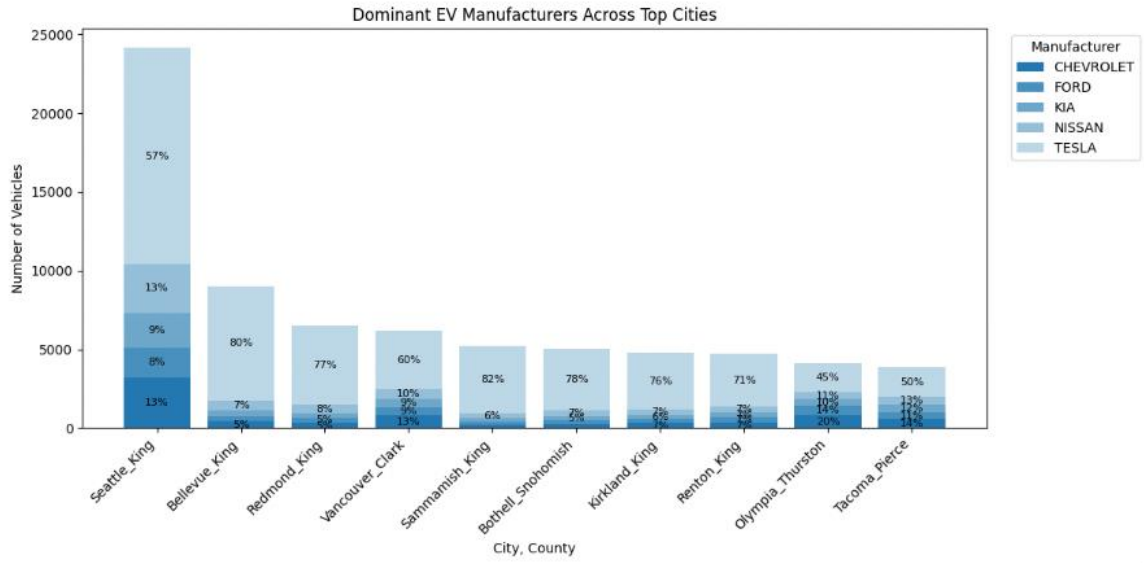
```
# Top 10 manufacturers
top_makes = df_clean['Make'].value_counts().head(10)

# Top 10 EV models
top_models = df_clean['Model_Make'].value_counts().head(10)
```

```
fig, axes = plt.subplots(1, 2, figsize=(16, 6))

# Manufacturers
top_makes.plot(kind='bar', ax=axes[0], color="#1f77b4")
axes[0].set_title("Top 10 Manufacturers")
axes[0].set_xlabel("Manufacturer")
axes[0].set_ylabel("Count")
axes[0].tick_params(axis='x', rotation=45)

# Models
top_models.plot(kind='bar', ax=axes[1], color="#1f77b4")
axes[1].set_title("Top 10 Models")
axes[1].set_xlabel("Model")
axes[1].set_ylabel("Count")
axes[1].tick_params(axis='x', rotation=45)

plt.tight_layout()
plt.show()
```



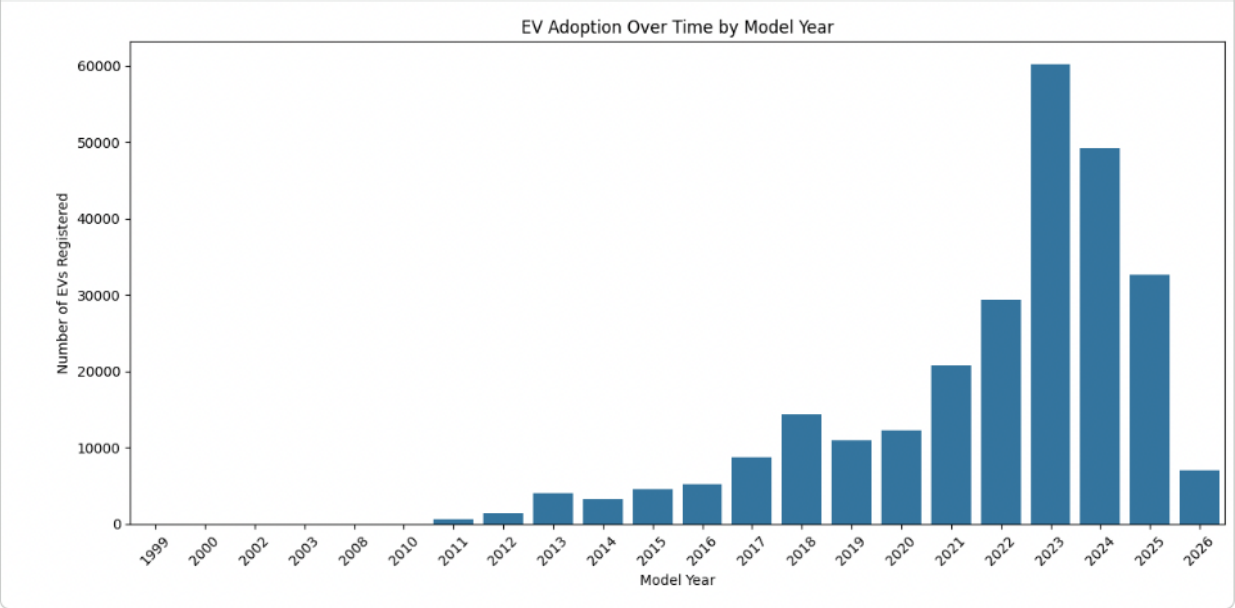

```
# 1. Top manufacturers and cities
top_5_makes = df_clean['Make'].value_counts().head(5).index
top_10_cities = df_clean['City_County'].value_counts().head(10).index

df_sub = df_clean[df_clean['Make'].isin(top_5_makes) & df_clean['City_County'].isin(top_10_cities)]

# 2. Pivot table: rows = cities, columns = manufacturers, values = counts
counts = df_sub.groupby(['City_County', 'Make']).size().unstack(fill_value=0)

# Sort cities by total vehicles
counts = counts.loc[counts.sum(axis=1).sort_values(ascending=False).index]

# 3. Create blue shades
n_makes = len(counts.columns)
colors = [lighten(base_color, amount=i * (0.7 / max(1, n_makes - 1))) for i in range(n_makes)]

# 4. Plot stacked count bar chart
fig, ax = plt.subplots(figsize=(12, 6))

bottom = np.zeros(len(counts))

for make, color in zip(counts.columns, colors):
    values = counts[make].values
    ax.bar(
        counts.index,
        values,
        bottom=bottom,
        label=make,
        color=color
    )
```

```
        )

        # Optional: add % label for segments >5%
        total_per_city = counts.sum(axis=1).values
        for i, v in enumerate(values):
            pct = v / total_per_city[i]
            if pct > 0.05:   # label only if >5% to avoid clutter
                ax.text(
                    i,
                    bottom[i] + v/2,
                    f"{pct*100:.0f}%",       # raw counts inside bars
                    ha='center',
                    va='center',
                    fontsize=8,
                    color='black'
                )

        bottom += values

    # Labels and title
    ax.set_xlabel("City, County")
    ax.set_ylabel("Number of Vehicles")
    ax.set_title("Dominant EV Manufacturers Across Top Cities")
    plt.xticks(rotation=45, ha='right')

    # Legend
    ax.legend(title="Manufacturer", bbox_to_anchor=(1.02, 1), loc="upper left")

    plt.tight_layout()
    plt.show()
```

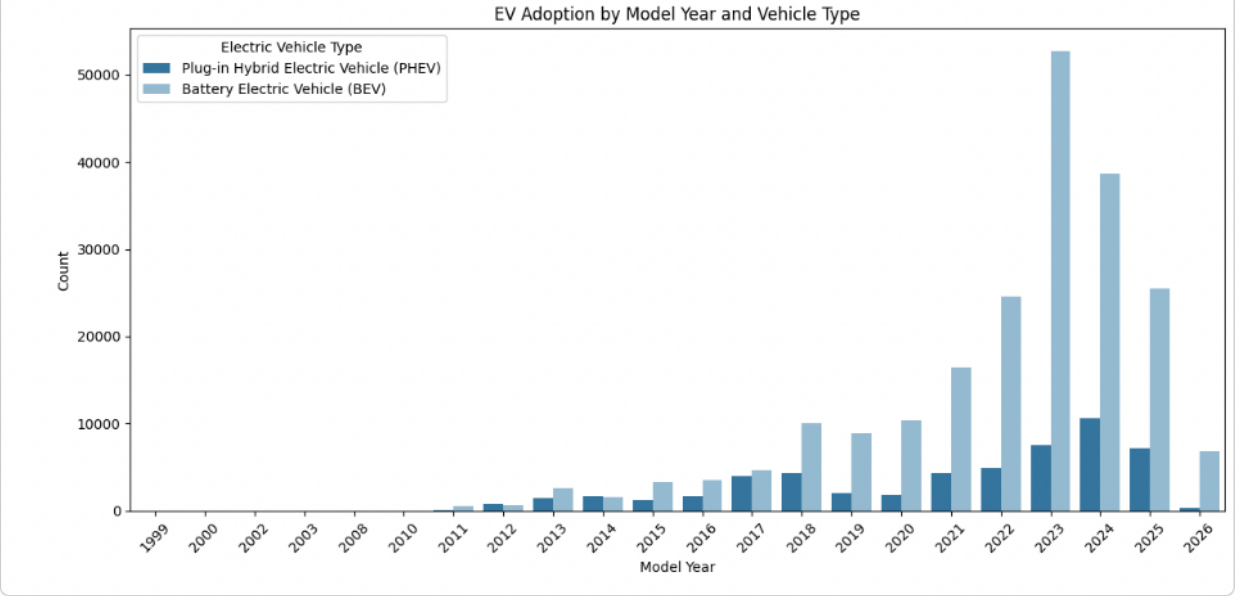


3. How has electric vehicle adoption evolved over time by model year, vehicle type and electric range capabilities?

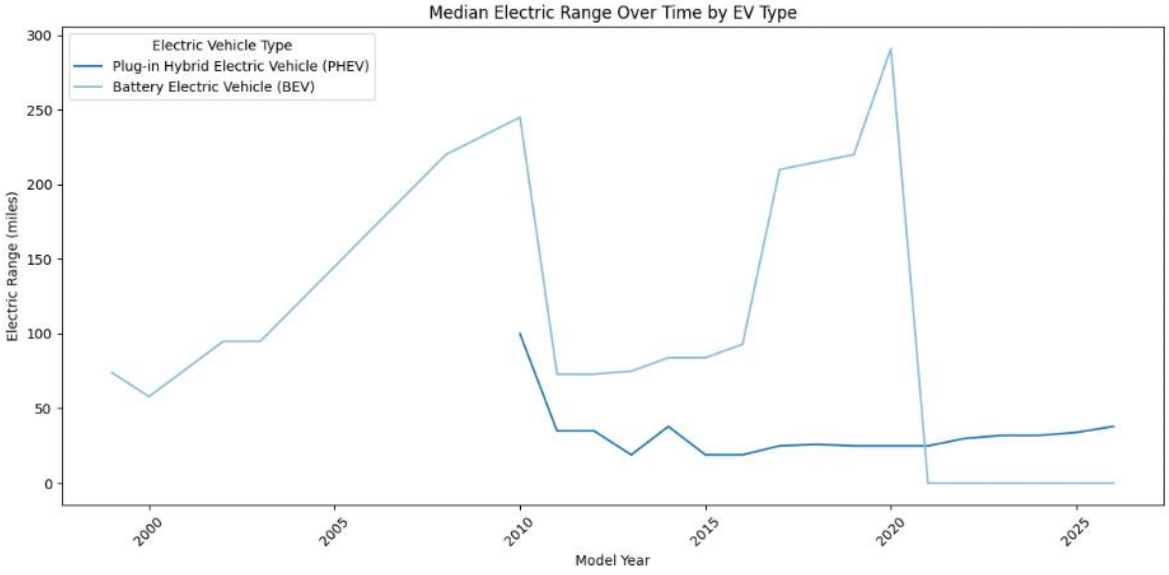
```
    plt.figure(figsize=(12,6))
    sns.countplot(data=df_clean, x='Model Year', color="#1f77b4")

    plt.title("EV Adoption Over Time by Model Year")
    plt.xlabel("Model Year")
    plt.ylabel("Number of EVs Registered")
    plt.xticks(rotation=45)
    plt.tight_layout()
    plt.show()
```




```
plt.figure(figsize=(12,6))
sns.countplot(
    data=df_clean,
    x='Model Year',
    hue='Electric Vehicle Type',
    palette=["#1f77b4", "#8abbdc"]
)

plt.title("EV Adoption by Model Year and Vehicle Type")
plt.xlabel("Model Year")
plt.ylabel("Count")
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```

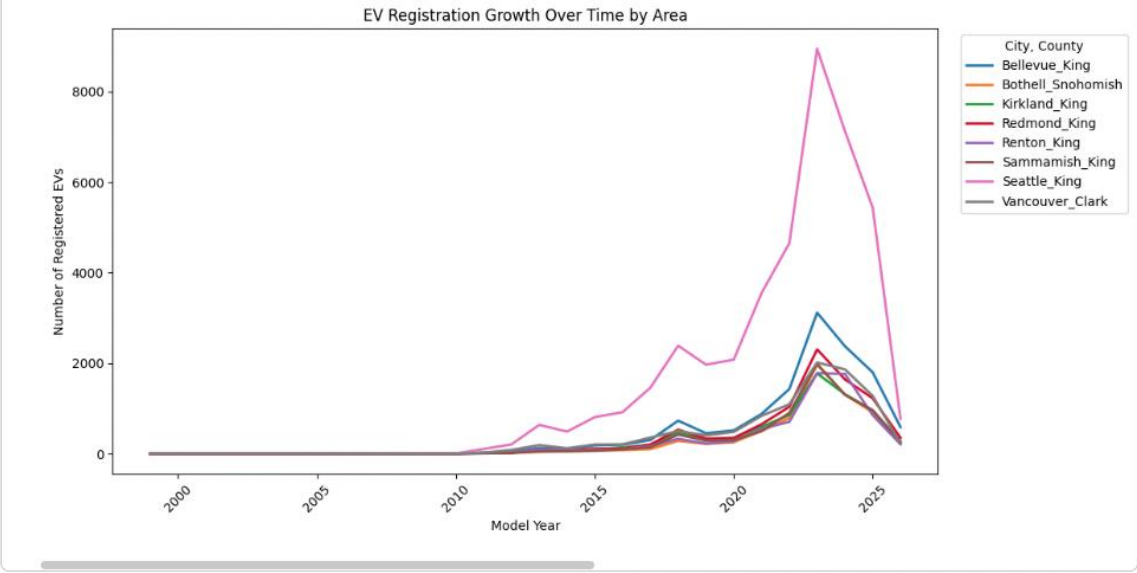


```
plt.figure(figsize=(12,6))

sns.lineplot(
    data=df_clean,
    x='Model Year',
    y='Electric Range',
    hue='Electric Vehicle Type',
    estimator='median',
    errorbar=None,
    palette=["#1f77b4", "#8abbdc"]
)

plt.title("Median Electric Range Over Time by EV Type")
plt.xlabel("Model Year")
plt.ylabel("Electric Range (miles)")
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



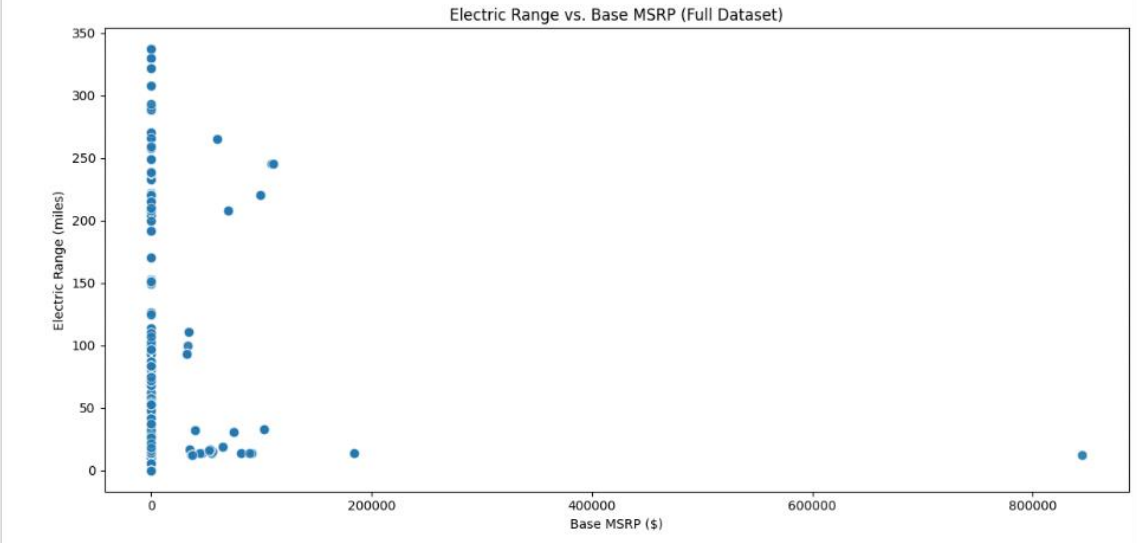
4. Which areas show the fastest growth in EV registrations over time?

```
plt.figure(figsize=(12, 6))

# Group by Model Year and City_County
growth = df_clean.groupby(['Model Year', 'City_County']).size().reset_index(name='Count')

# Select top 8 fastest-growing areas (highest total registrations)
top_areas = df_clean['City_County'].value_counts().head(8).index

growth_top = growth[growth['City_County'].isin(top_areas)]

sns.lineplot(
    data=growth_top,
    x='Model Year',
    y='Count',
    hue='City_County',
    linewidth=2
)

plt.title("EV Registration Growth Over Time by Area")
plt.xlabel("Model Year")
plt.ylabel("Number of Registered EVs")
plt.xticks(rotation=45)
plt.legend(title="City, County", bbox_to_anchor=(1.02, 1), loc="upper left")
plt.tight_layout()
plt.show()
```



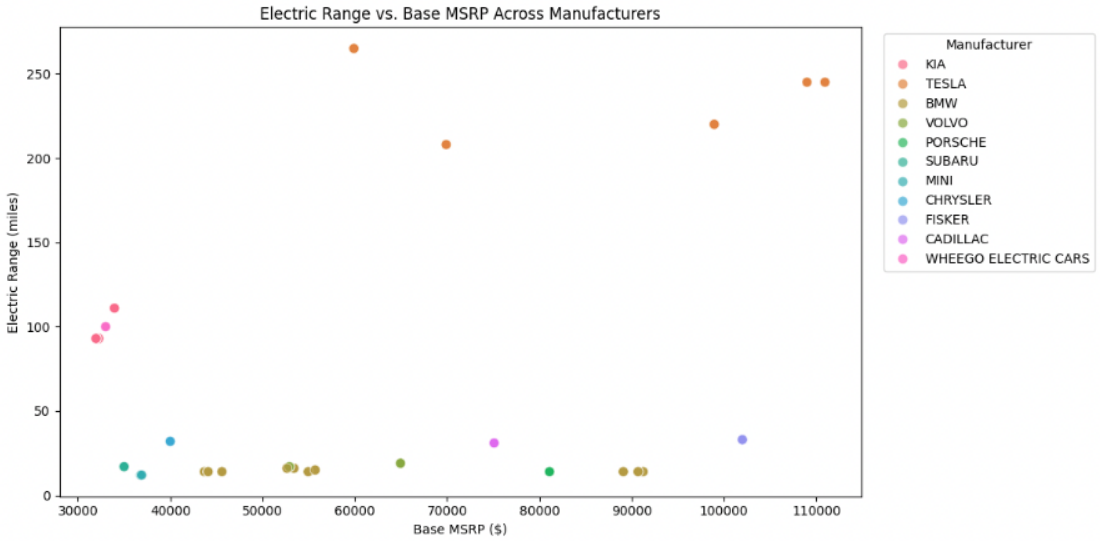
5. What relationship, if any, exists between base MSRP and electric range performance? How do electric range and base MSRP vary across manufacturers and models?

```
plt.figure(figsize=(12, 6))

sns.scatterplot(
    data=df_clean,
    x='Base MSRP',
    y='Electric Range',
    alpha=0.7,
    s=60
)

plt.title("Electric Range vs. Base MSRP (Full Dataset)")
plt.xlabel("Base MSRP ($)")
plt.ylabel("Electric Range (miles)")
plt.tight_layout()
plt.show()
```



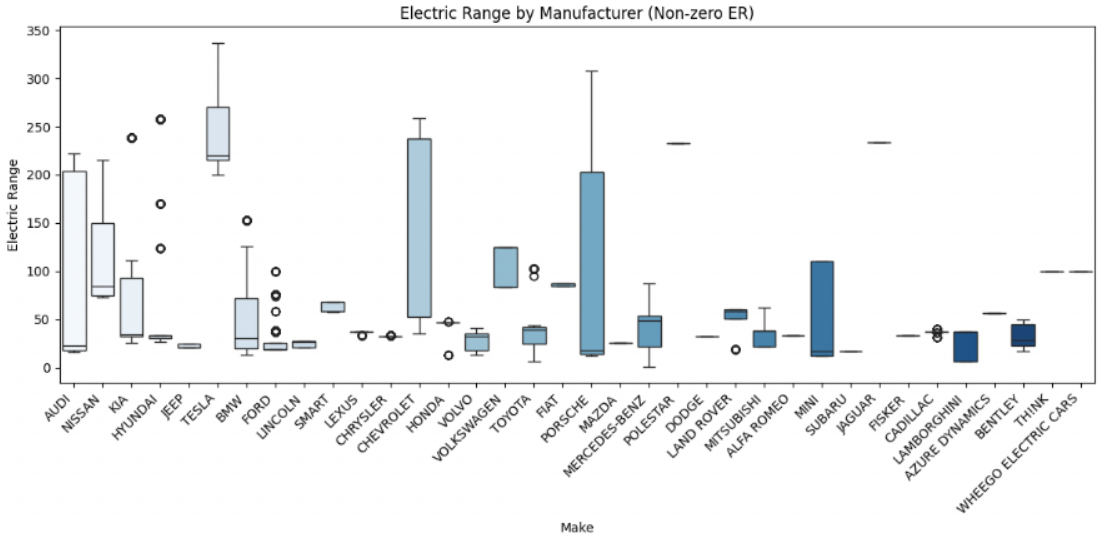

```
plt.figure(figsize=(12, 6))

sns.scatterplot(
    data=df_MSRP_no_outliers,
    x='Base MSRP',
    y='Electric Range',
    hue='Make',
    alpha=0.7,
    s=60
)
```

```
plt.title("Electric Range vs. Base MSRP Across Manufacturers")
plt.xlabel("Base MSRP ($)")
plt.ylabel("Electric Range (miles)")
plt.legend(title="Manufacturer", bbox_to_anchor=(1.02, 1), loc="upper left")
plt.tight_layout()
plt.show()
```



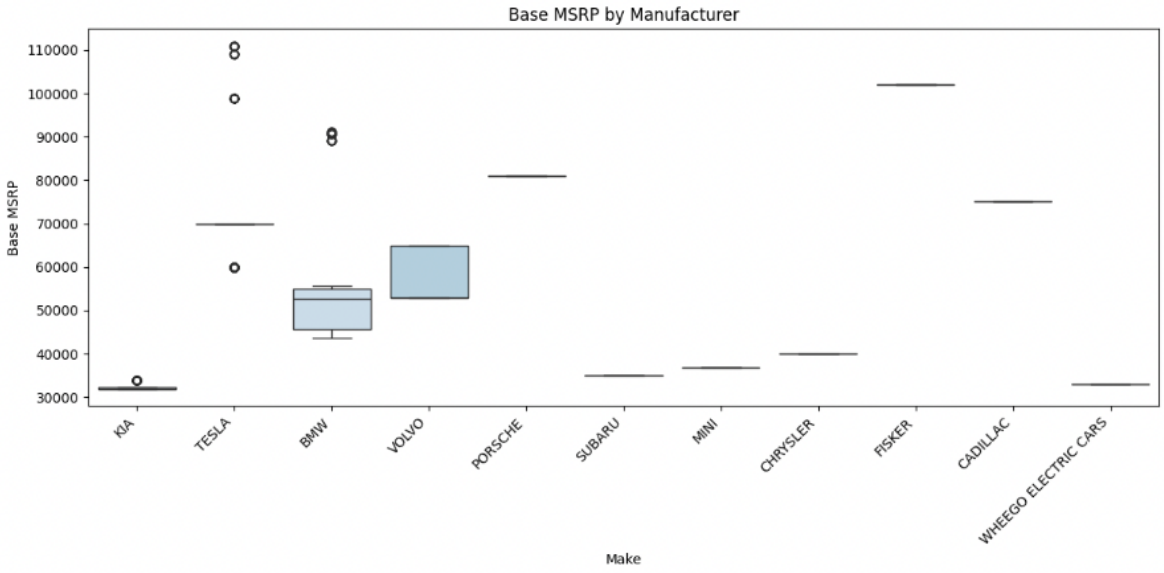

```
plt.figure(figsize=(12,6))
sns.boxplot(
    data=df_non_zero_ER,
    x='Make',
    y='Electric Range',
    palette='Blues'
)
plt.xticks(rotation=45, ha='right')
plt.title("Electric Range by Manufacturer (Non-zero ER)")
plt.tight_layout()
plt.show()
```

```
/tmp/ipython-input-3944493441.py:2: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` va

  sns.boxplot(
```



```
plt.figure(figsize=(12,6))
sns.boxplot(
    data=df_MSRP_no_outliers,
    x='Make',
    y='Base MSRP',
    palette='Blues'
)
plt.xticks(rotation=45, ha='right')
plt.title("Base MSRP by Manufacturer")
plt.tight_layout()
plt.show()
```

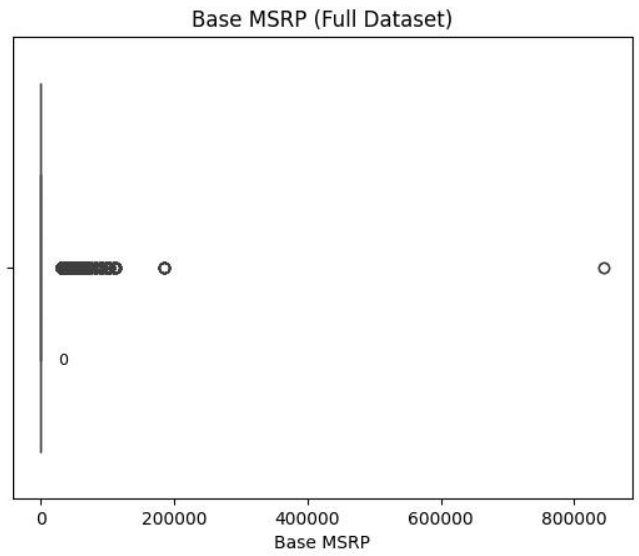
```
/tmp/ipython-input-2226682068.py:2: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` va

  sns.boxplot(
```



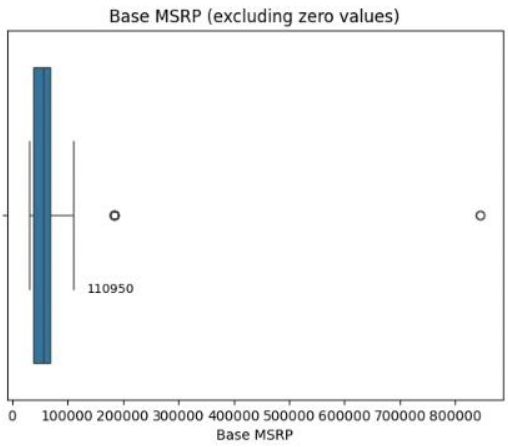
Appendix F. Step Seven Python Code – Handle Outliers

Base MSRP

```
# Box plots for Base MSRP
ax = sns.boxplot(x=df_clean['Base MSRP'])

lines = ax.lines
upper_whisker_line = lines[3]
upper_whisker_y = upper_whisker_line.get_ydata().max()
upper_whisker_x = upper_whisker_line.get_xdata().max()

# Annotate the max whisker value
ax.annotate(
    f"{upper_whisker_x:.0f}",
    xy=(upper_whisker_x, upper_whisker_y),
    xytext=(10, 0),
    textcoords="offset points",
    ha="left", va="center",
    fontsize=9
)


plt.title('Base MSRP (Full Dataset)')
plt.show()
```

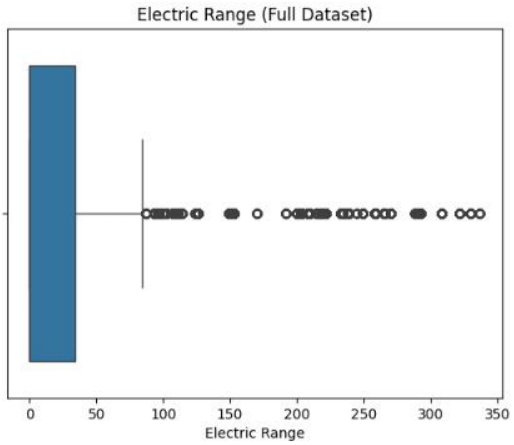



```
# Box plots for Base MSRP
ax = sns.boxplot(x=df_non_zero_MSRP['Base MSRP'])

lines = ax.lines
upper_whisker_line = lines[3]
upper_whisker_y = upper_whisker_line.get_ydata().max()
upper_whisker_x = upper_whisker_line.get_xdata().max()

# Annotate the max whisker value
ax.annotate(
    f"{upper_whisker_x:.0f}",
    xy=(upper_whisker_x, upper_whisker_y),
    xytext=(10, 0),
    textcoords="offset points",
    ha="left", va="center",
    fontsize=9
)


plt.title('Base MSRP (excluding zero values)')
plt.show()
```



Electric range

```
# Box plots for Electric Range
sns.boxplot(x=df_clean['Electric Range'])
plt.title('Electric Range (Full Dataset)')
plt.show()
```




```
# Box plots for Electric Range
sns.boxplot(x=df_non_zero_ER['Electric Range'])
plt.title('Electric Range (Excluding zero values)')
plt.show()
```