Annie Phan

Project Proposal

Project Objective

The objective of this project is to conduct a comprehensive Exploratory Data Analysis (EDA) of a dataset to examine its descriptive statistics and identify trends, patterns and relationships among its features using Python and statistical methods.

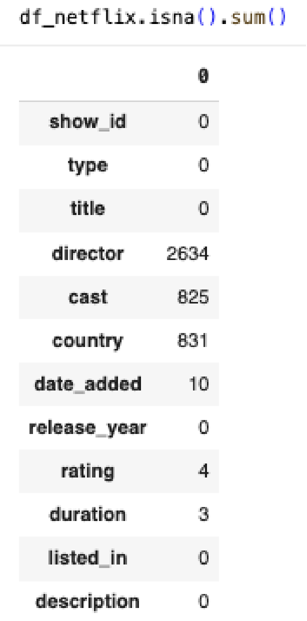
Initial Dataset Exploration

To select a dataset suitable for in-depth EDA, three publicly available datasets were initially evaluated:

1. Netflix Movies and TV Shows:

This dataset contains information on 8,809 movies and TV shows available from 2008 to 2021 on the Netflix streaming platform. It includes 8,809 entries and 12 columns. Most columns are categorical variables, including show ID, type, title, director, cast, country of production, date added to the platform, release year, duration, rating, genre, and description.

No missing values were found in key columns such as show ID, type, title, release year, genre, and description. Columns such as director, cast, and country contain a high proportion of null entries, which may affect analyses involving these attributes. A summary of null value counts for each column is provided below:



```
df_netflix.isna().sum()
```

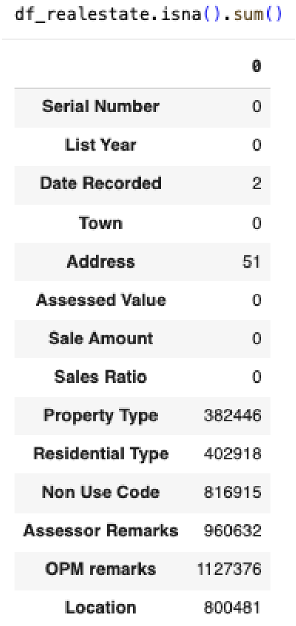
	0
show_id	0
type	0
title	0
director	2634
cast	825
country	831
date_added	10
release_year	0
rating	4
duration	3
listed_in	0
description	0

Figure 1. Null value counts in the Netflix dataset

2. Real Estate Sales (2001–2023):

This dataset comprises 1,141,722 property sales records in Connecticut, spanning the years 2001 to 2023, and includes 14 columns. The categorical variables are serial number, list year, sales date, town, address, property type, residential type, and location. The numerical variables are assessed value, sale price, and sales ratio.

Key columns, such as serial number, list year, sales date, town, address, assessed value, sale amount, sales ratio, have no or minimal missing values, ensuring that analyses related to sale prices and assessment ratios can be conducted with high confidence. Other columns, such as property type, residential type, and location contain a substantial number of missing values, which may limit the reliability of analyses involving these attributes. A summary of the null value counts for each column is presented in the table below.

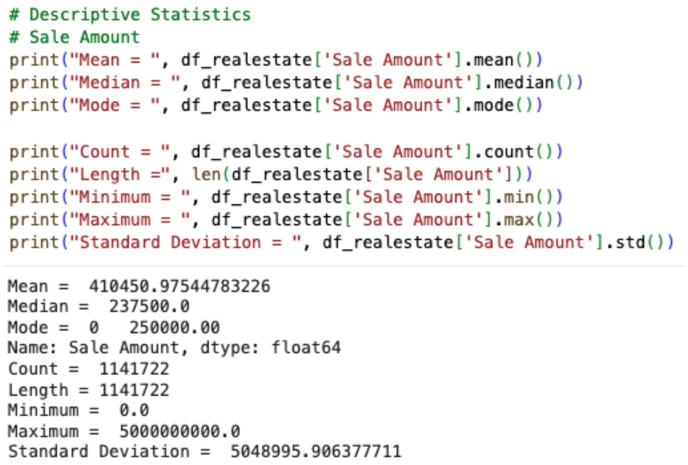


```
df_realestate.isna().sum()
```

	0
Serial Number	0
List Year	0
Date Recorded	2
Town	0
Address	51
Assessed Value	0
Sale Amount	0
Sales Ratio	0
Property Type	382446
Residential Type	402918
Non Use Code	816915
Assessor Remarks	960632
OPM remarks	1127376
Location	800481

Figure 2. Null value counts in the real estate sales dataset

The sale prices in the real estate dataset exhibit significant variability. The mean sale price is $410,451, while the median is considerably lower at $237,500, indicating a right-skewed distribution influenced by extreme high-value transactions. The mode occurs at $0 and $250,000, reflecting the presence of properties with missing sale prices and the clustering of many transactions around $250,000. Sale prices range from a minimum of $0 to a maximum of $5,000,000,000, and the standard deviation of approximately $5,048,996 highlights the substantial dispersion and presence of outliers within the dataset.



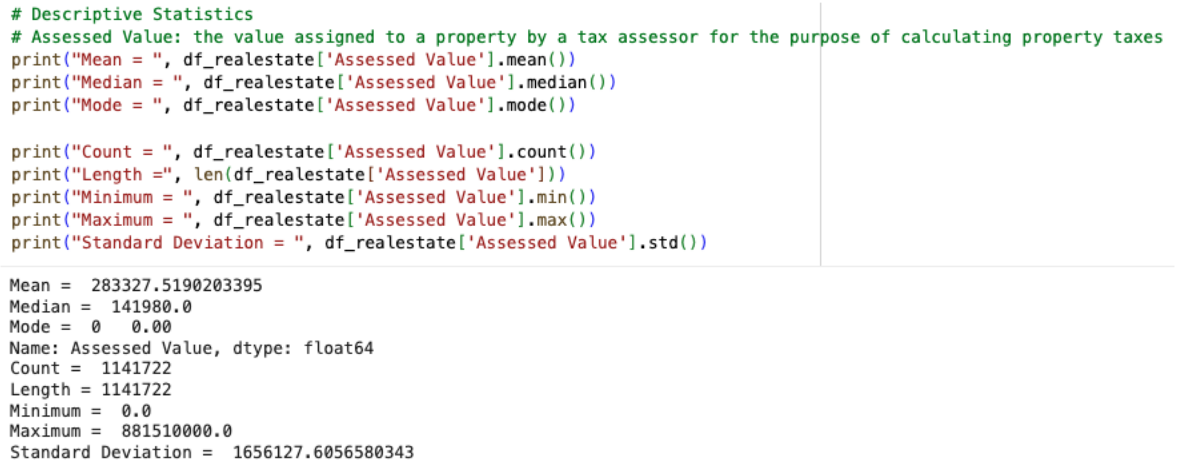
```
# Descriptive Statistics
# Sale Amount
print("Mean = ", df_realestate['Sale Amount'].mean())
print("Median = ", df_realestate['Sale Amount'].median())
print("Mode = ", df_realestate['Sale Amount'].mode())

print("Count = ", df_realestate['Sale Amount'].count())
print("Length =", len(df_realestate['Sale Amount']))
print("Minimum = ", df_realestate['Sale Amount'].min())
print("Maximum = ", df_realestate['Sale Amount'].max())
print("Standard Deviation = ", df_realestate['Sale Amount'].std())
```

```
Mean =  410450.97544783226
Median =  237500.0
Mode =  0   250000.00
Name: Sale Amount, dtype: float64
Count =  1141722
Length = 1141722
Minimum =  0.0
Maximum =  5000000000.0
Standard Deviation =  5048995.906377711
```

Figure 3. Descriptive statistics of the sale price variable

The assessed values in the real estate dataset exhibit a wide range and significant variability. The mean assessed value is $283,328, while the median is considerably lower at $141,980, indicating a right-skewed distribution influenced by extreme high-value properties. The mode is $0, reflecting the presence of properties with missing or unassessed values. Assessed values span from a minimum of $0 to a maximum of $881,510,000, and the standard deviation of approximately $1,656,128 highlights substantial dispersion and the influence of outliers on the overall distribution.



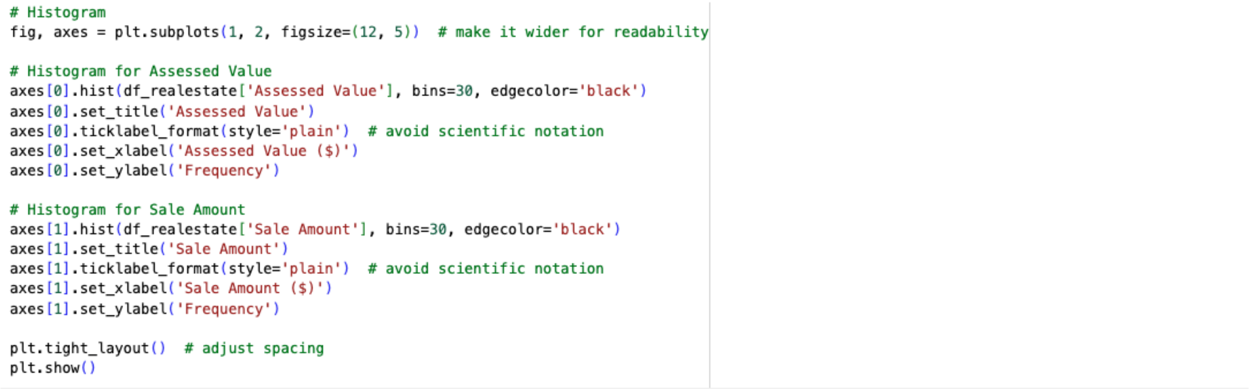
```
# Descriptive Statistics
# Assessed Value: the value assigned to a property by a tax assessor for the purpose of calculating property taxes
print("Mean = ", df_realestate['Assessed Value'].mean())
print("Median = ", df_realestate['Assessed Value'].median())
print("Mode = ", df_realestate['Assessed Value'].mode())

print("Count = ", df_realestate['Assessed Value'].count())
print("Length =", len(df_realestate['Assessed Value']))
print("Minimum = ", df_realestate['Assessed Value'].min())
print("Maximum = ", df_realestate['Assessed Value'].max())
print("Standard Deviation = ", df_realestate['Assessed Value'].std())
```

```
Mean =  283327.5190203395
Median =  141980.0
Mode =  0   0.00
Name: Assessed Value, dtype: float64
Count =  1141722
Length = 1141722
Minimum =  0.0
Maximum =  881510000.0
Standard Deviation =  1656127.6056580343
```

Figure 4. Descriptive statistics of the assessed value variable

The histograms show that the distributions of both assessed value and sale amount are right-skewed, with a concentration of values at 0. Specifically, 64% of properties have an assessed value of 0, while 0.15% of the records have a sale amount of 0.



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

# Histogram for Assessed Value
axes[0].hist(df_realestate['Assessed Value'], bins=30, edgecolor='black')
axes[0].set_title('Assessed Value')
axes[0].ticklabel_format(style='plain')  # avoid scientific notation
axes[0].set_xlabel('Assessed Value ($)')
axes[0].set_ylabel('Frequency')

# Histogram for Sale Amount
axes[1].hist(df_realestate['Sale Amount'], bins=30, edgecolor='black')
axes[1].set_title('Sale Amount')
axes[1].ticklabel_format(style='plain')  # avoid scientific notation
axes[1].set_xlabel('Sale Amount ($)')
axes[1].set_ylabel('Frequency')

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

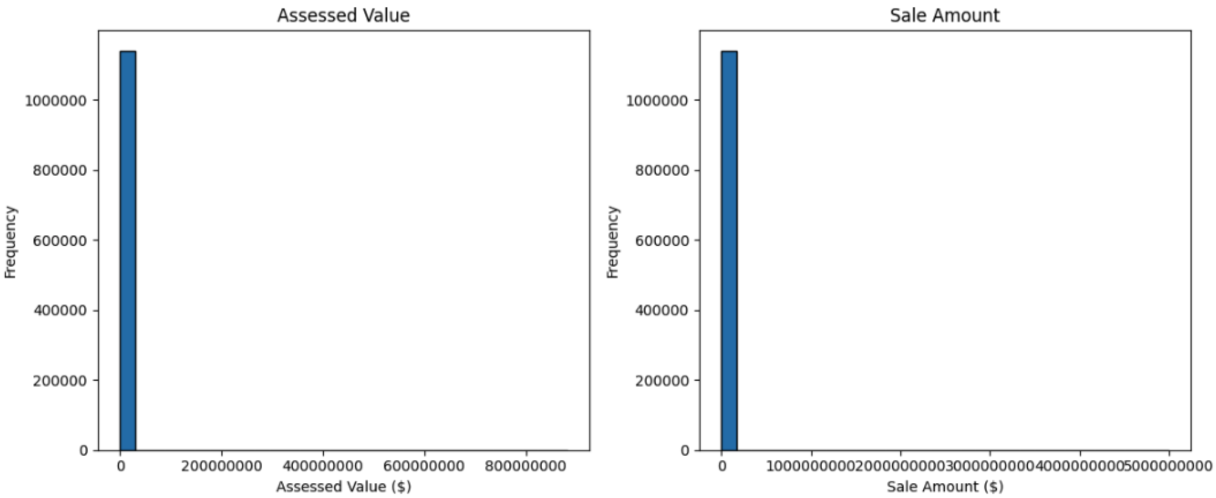
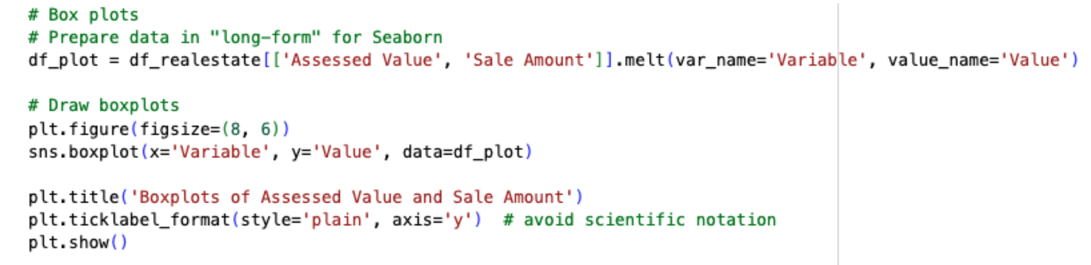


Figure 5. Histograms of Assessed Value and Sale Amount

The box plots confirm the presence of extreme outliers in the dataset, with one property having an assessed value of $881,510,000 and another transaction showing a sale price of $5,000,000,000. These extreme values contribute to the right-skewness observed in both distributions and substantially increase the mean compared to the median. While most properties fall within a much lower range, these outliers highlight the need for careful handling in statistical analyses to avoid distortion of summary measures.



```
# Box plots
# Prepare data in "long-form" for Seaborn
df_plot = df_realestate[['Assessed Value', 'Sale Amount']].melt(var_name='Variable', value_name='Value')

# Draw boxplots
plt.figure(figsize=(8, 6))
sns.boxplot(x='Variable', y='Value', data=df_plot)

plt.title('Boxplots of Assessed Value and Sale Amount')
plt.ticklabel_format(style='plain', axis='y')  # avoid scientific notation
plt.show()
```

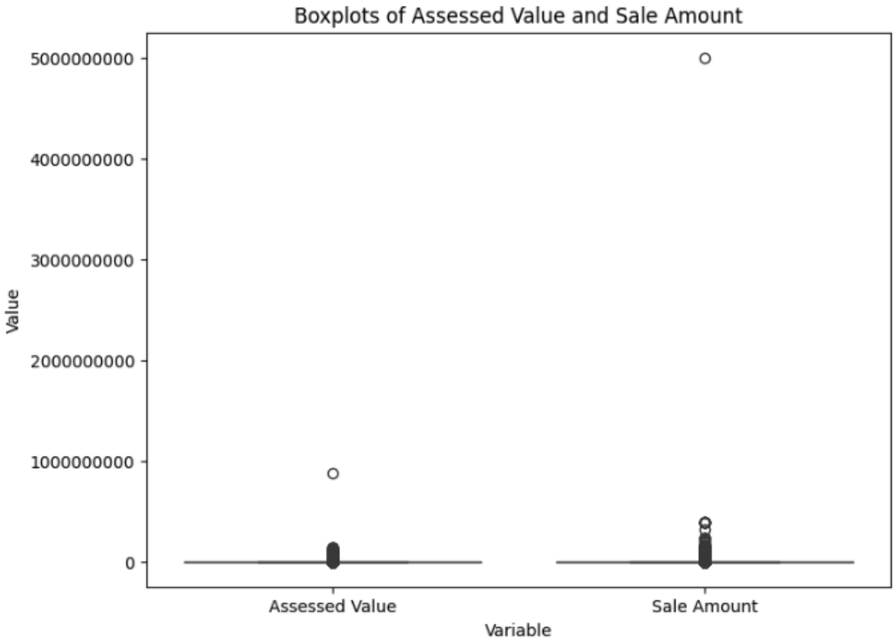
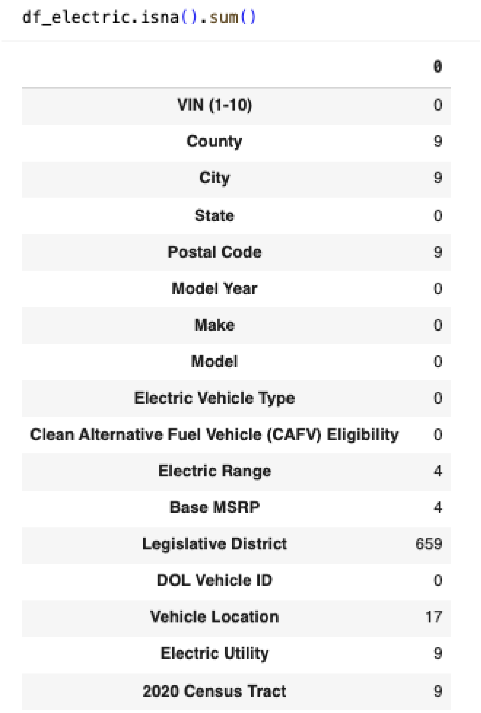


Figure 6. Boxplots of Assessed Value and Sale Amount

3. Electric Vehicle Population Data:

This dataset contains records of electric vehicle registrations in Washington State, comprising 264,628 entries and 17 variables. It includes two numerical variables electric range and base MSRP, and thirteen categorical variables, such as VIN, county, city, state, postal code, model year, manufacturer, model, vehicle type, clean alternative fuel vehicle eligibility, legislative district, vehicle location, and electric utility.

The dataset is largely complete, with most variables containing no or minimal missing values. Only a few columns, such as legislative district (659 nulls), vehicle location (17 nulls), and several others with fewer than 10 missing entries, show limited data gaps that are unlikely to affect overall analysis quality.

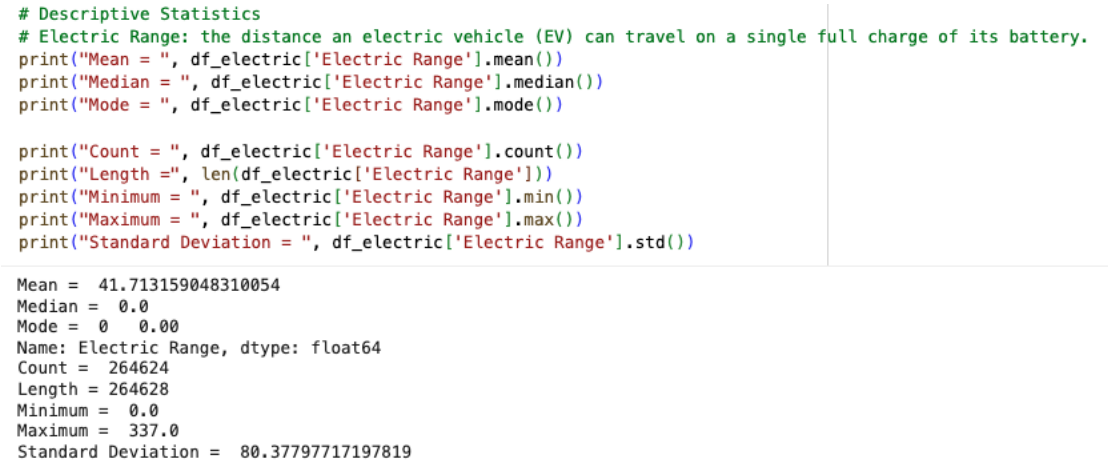


```
df_electric.isna().sum()
```

	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 7. Null value counts in the electric vehicle population dataset

The Electric Range variable shows substantial variation and a highly right-skewed distribution. The mean electric range is approximately 41.71 miles, while the median and mode are both 0, indicating that a large number of vehicles in the dataset have no electric range, likely representing hybrid or non-electric models. The values range from 0 to 337 miles, and the standard deviation of about 80.38 reflects considerable dispersion among vehicles with nonzero ranges. This suggests that while most vehicles offer limited or no electric driving capability, a subset of fully electric models achieves significantly higher ranges.



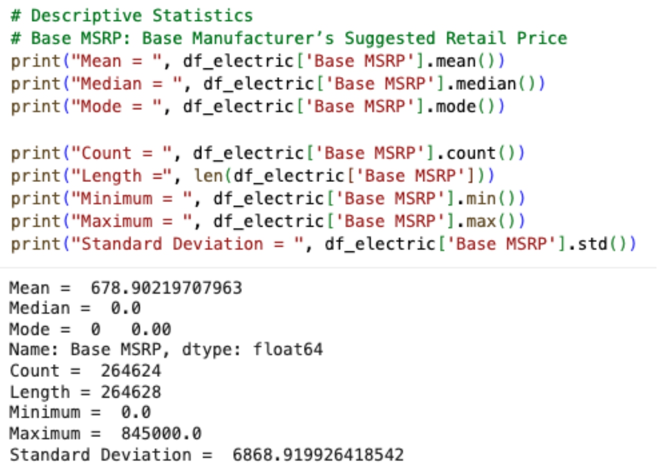
```
# Descriptive Statistics
# Electric Range: the distance an electric vehicle (EV) can travel on a single full charge of its battery.
print("Mean = ", df_electric['Electric Range'].mean())
print("Median = ", df_electric['Electric Range'].median())
print("Mode = ", df_electric['Electric Range'].mode())

print("Count = ", df_electric['Electric Range'].count())
print("Length =", len(df_electric['Electric Range']))
print("Minimum = ", df_electric['Electric Range'].min())
print("Maximum = ", df_electric['Electric Range'].max())
print("Standard Deviation = ", df_electric['Electric Range'].std())
```

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

Figure 8. Descriptive statistics of the electric range variable

The Base MSRP variable displays extreme variability and a highly right-skewed distribution. The mean value is approximately $678.90, while both the median and mode are $0, suggesting that a large portion of entries either lack MSRP data or represent incomplete records. The range spans from $0 to $845,000, and the standard deviation of about $6,868.92 indicates substantial dispersion caused by a few vehicles with very high base prices. Overall, the data suggest that most records do not include MSRP information, while a small subset of luxury or specialized electric vehicles contributes to the upper extreme of the distribution.



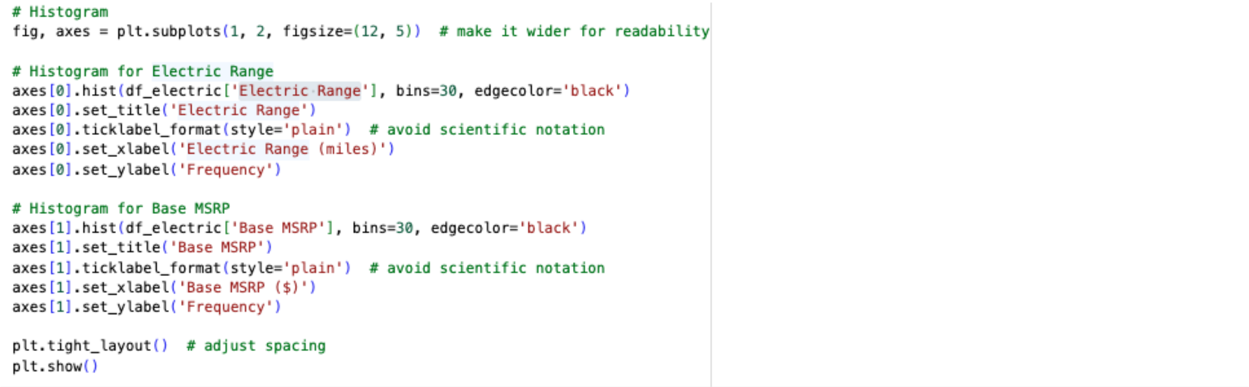
```python
# Descriptive Statistics
# Base MSRP: Base Manufacturer’s Suggested Retail Price
print("Mean = ", df_electric['Base MSRP'].mean())
print("Median = ", df_electric['Base MSRP'].median())
print("Mode = ", df_electric['Base MSRP'].mode())

print("Count = ", df_electric['Base MSRP'].count())
print("Length =", len(df_electric['Base MSRP']))
print("Minimum = ", df_electric['Base MSRP'].min())
print("Maximum = ", df_electric['Base MSRP'].max())
print("Standard Deviation = ", df_electric['Base MSRP'].std())
```

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

Figure 9. Descriptive statistics of the base MSRP variable

The histograms for both Electric Range and Base MSRP show highly right-skewed distributions with a large concentration of values at zero. This suggests that many registered vehicles either have no electric-only driving capability or are missing MSRP information. A small subset of vehicles with substantially higher values creates a long right tail in both distributions, indicating the presence of high-range fully electric models and premium-priced vehicles.



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

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

# Histogram for Base MSRP
axes[1].hist(df_electric['Base MSRP'], bins=30, edgecolor='black')
axes[1].set_title('Base MSRP')
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()
```

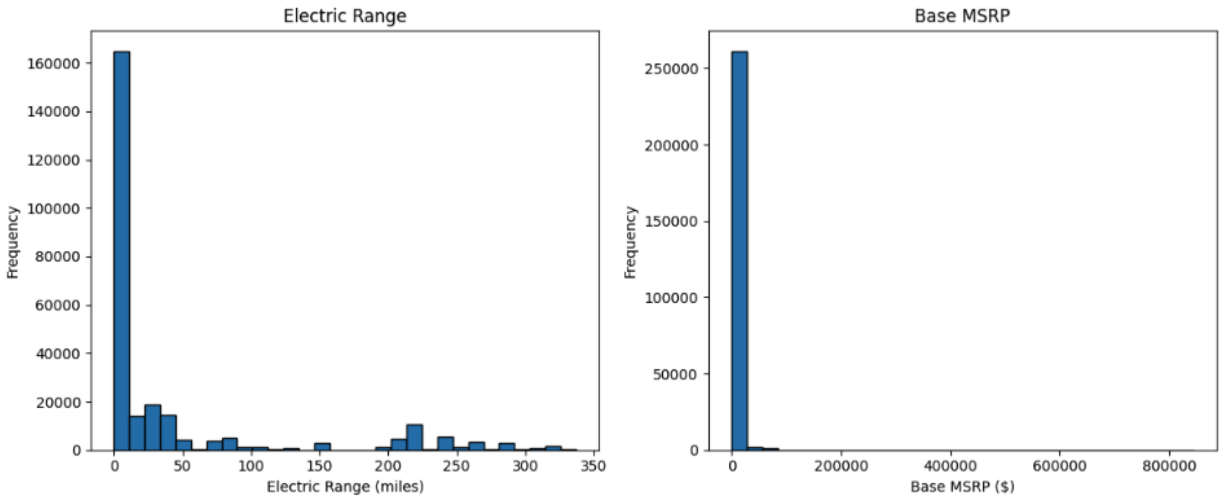
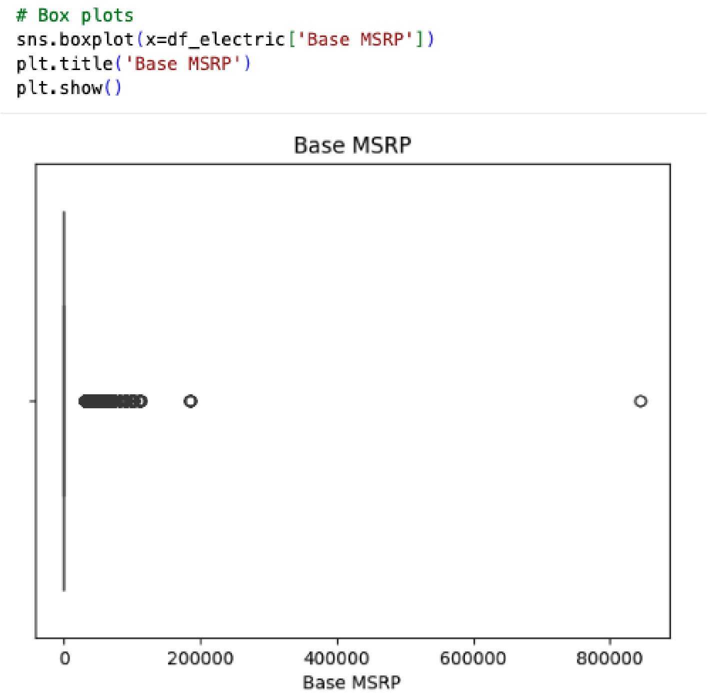


Figure 10. Histograms of electric range and base MSRP

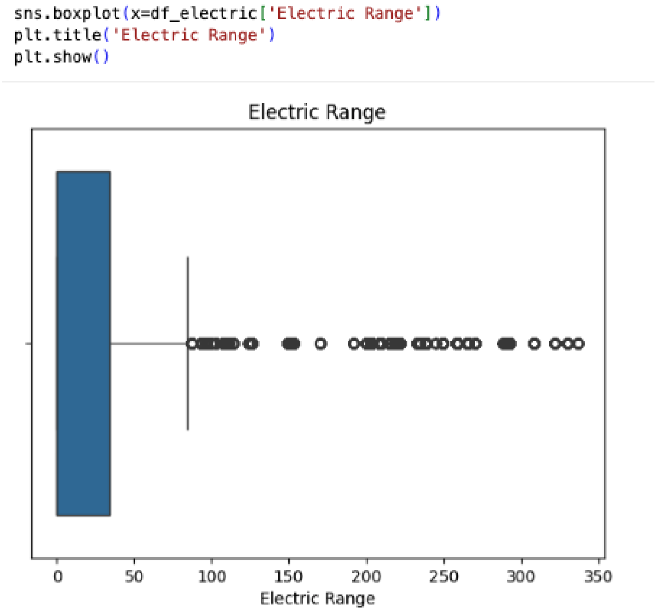
The box plot shows that Base MSRP is heavily right-skewed, with most values clustered near zero and a few extreme outliers extending far to the right. This indicates that while most vehicles have low or missing MSRP values, a small number of high-priced models distort the distribution.



```
# Box plots
sns.boxplot(x=df_electric['Base MSRP'])
plt.title('Base MSRP')
plt.show()
```

Figure 11. Boxplot of base MSRP

The box plot for Electric Range shows a strongly right-skewed distribution, with most observations clustered near zero. Approximately 75% of the vehicles have an electric range between 0 and 34 miles, reflecting the predominance of plug-in hybrids with limited electric-only capability. A smaller subset of fully electric vehicles with much higher ranges appears as outliers to the right.



```
sns.boxplot(x=df_electric['Electric Range'])
plt.title('Electric Range')
plt.show()
```

Figure 12. Boxplot of electric range

Dataset Selection

Among the three datasets initially evaluated, the electric vehicle population dataset was selected for in-depth exploratory data analysis. This dataset contains both categorical and numerical variables that enable meaningful descriptive and inferential statistical analysis. In addition, it has fewer missing values relative to the other datasets, which improves reliability and reduces the need for extensive data cleaning.

Research Questions

The EDA will focus on addressing the following research questions related to electric vehicle adoption in Washington State:

1. What is the distribution of electric vehicle types (e.g., BEV vs. PHEV) across the dataset?
2. How do electric range and base MSRP vary across manufacturers and models?
3. Which counties and cities have the highest concentration of electric vehicle registrations?
4. How does electric vehicle adoption vary by model year over time?
5. Is there a relationship between electric range and eligibility for clean alternative fuel incentives?
6. Are there geographic or legislative district patterns in EV adoption?

References

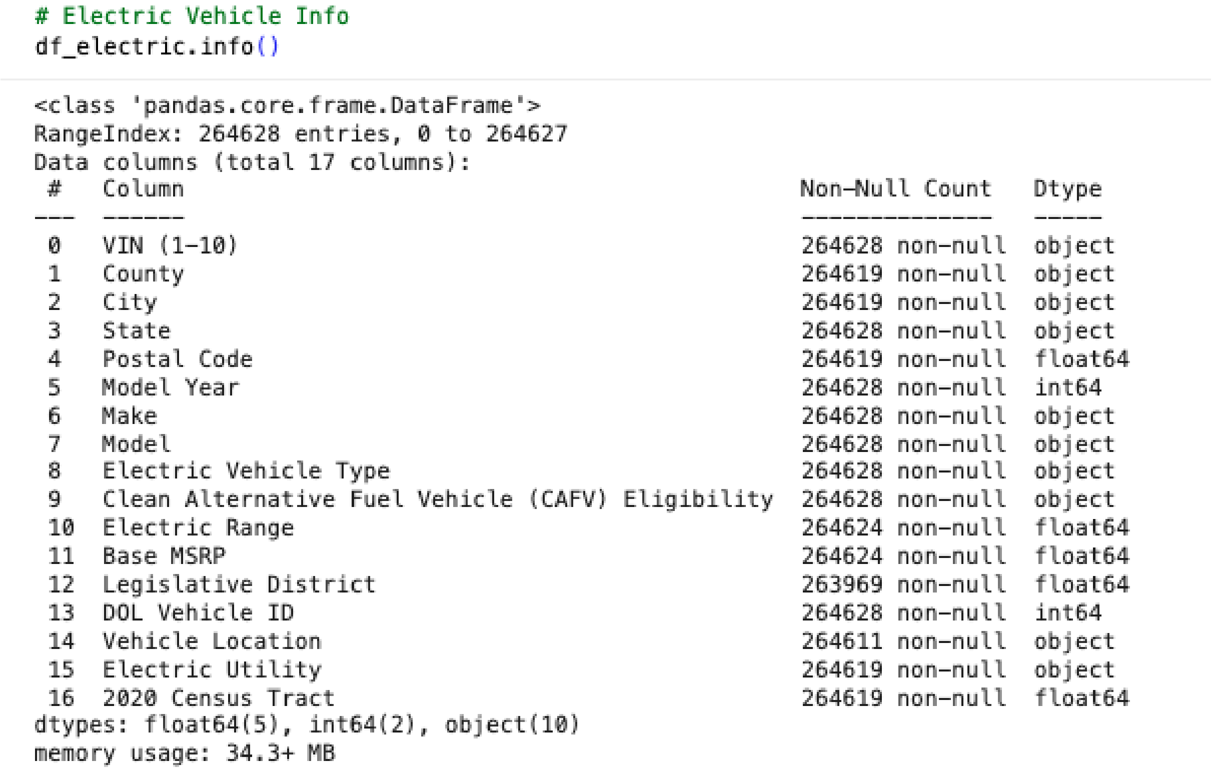
- Kaggle. Netflix Movies and TV Shows Dataset. Retrieved from https://www.kaggle.com/datasets/rahulvyasm/netflix-movies-and-tv-shows

- Data.gov. Real Estate Sales (2001–2018). Retrieved from https://catalog.data.gov/dataset/real-estate-sales-2001-2018

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

Appendix

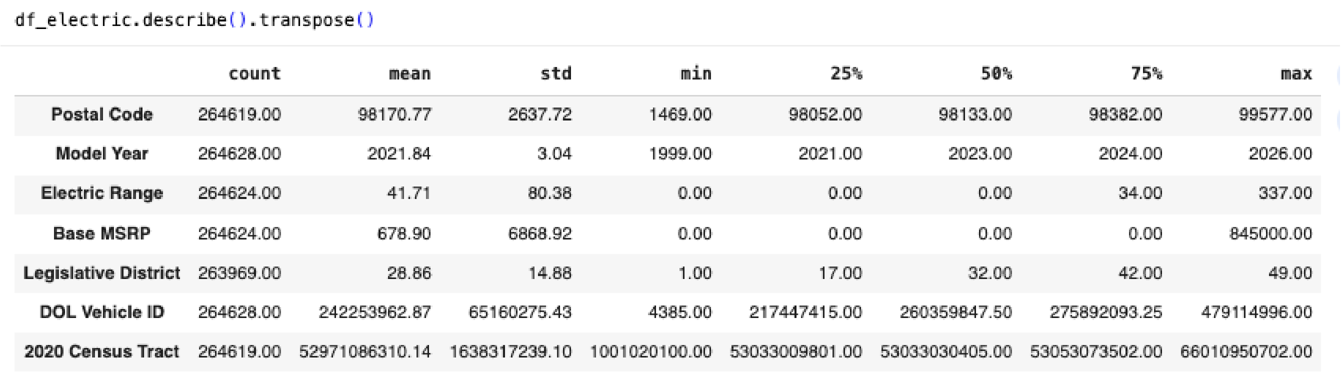
Appendix A.
Data Structure and Variable Information for the Electric Vehicle Dataset



```
# Electric Vehicle Info
df_electric.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
```

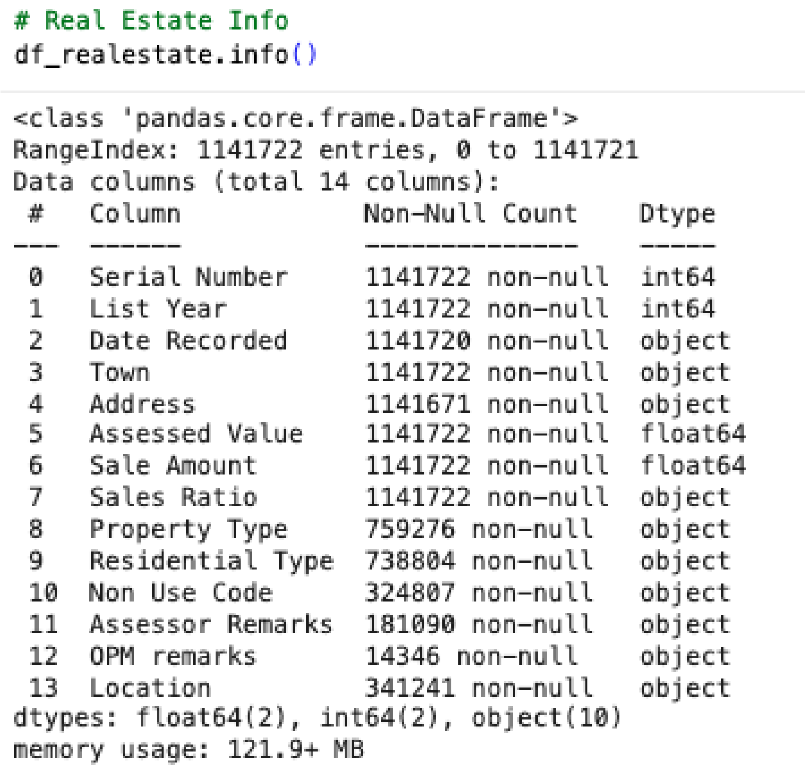
Appendix B.
Summary Descriptive Statistics for the Electric Vehicle Dataset



```
df_electric.describe().transpose()
```

	count	mean	std	min	25%	50%	75%	max
Postal Code	264619.00	98170.77	2637.72	1469.00	98052.00	98133.00	98382.00	99577.00
Model Year	264628.00	2021.84	3.04	1999.00	2021.00	2023.00	2024.00	2026.00
Electric Range	264624.00	41.71	80.38	0.00	0.00	0.00	34.00	337.00
Base MSRP	264624.00	678.90	6868.92	0.00	0.00	0.00	0.00	845000.00
Legislative District	263969.00	28.86	14.88	1.00	17.00	32.00	42.00	49.00
DOL Vehicle ID	264628.00	242253962.87	65160275.43	4385.00	217447415.00	260359847.50	275892093.25	479114996.00
2020 Census Tract	264619.00	52971086310.14	1638317239.10	1001020100.00	53033009801.00	53033030405.00	53053073502.00	66010950702.00

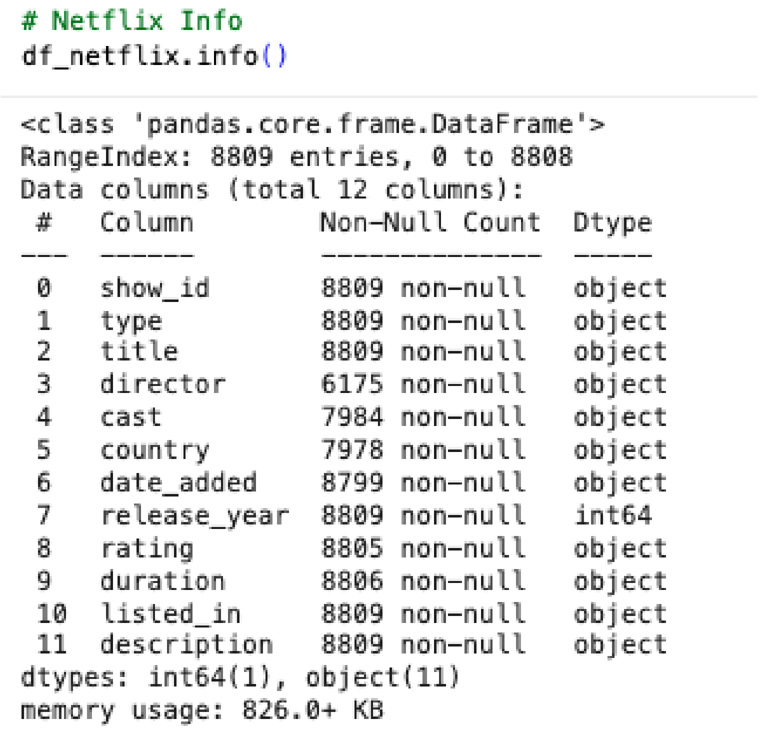
Appendix C.
Data Structure and Variable Information for the Real Estate Sales Dataset



```
# Real Estate Info
df_realestate.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1141722 entries, 0 to 1141721
Data columns (total 14 columns):
 #   Column            Non-Null Count    Dtype  
---  ------            --------------    -----  
 0   Serial Number     1141722 non-null  int64  
 1   List Year         1141722 non-null  int64  
 2   Date Recorded     1141720 non-null  object 
 3   Town              1141722 non-null  object 
 4   Address           1141671 non-null  object 
 5   Assessed Value    1141722 non-null  float64
 6   Sale Amount       1141722 non-null  float64
 7   Sales Ratio       1141722 non-null  object 
 8   Property Type     759276 non-null   object 
 9   Residential Type  738804 non-null   object 
 10  Non Use Code      324807 non-null   object 
 11  Assessor Remarks  181090 non-null   object 
 12  OPM remarks       14346 non-null    object 
 13  Location          341241 non-null   object 
dtypes: float64(2), int64(2), object(10)
memory usage: 121.9+ MB
```

Appendix D.
Data Structure and Variable Information for the Netflix Dataset



```
# Netflix Info
df_netflix.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 8809 entries, 0 to 8808
Data columns (total 12 columns):
 #   Column        Non-Null Count  Dtype 
---  ------        --------------  ----- 
 0   show_id       8809 non-null   object
 1   type          8809 non-null   object
 2   title         8809 non-null   object
 3   director      6175 non-null   object
 4   cast          7984 non-null   object
 5   country       7978 non-null   object
 6   date_added    8799 non-null   object
 7   release_year  8809 non-null   int64 
 8   rating        8805 non-null   object
 9   duration      8806 non-null   object
 10  listed_in     8809 non-null   object
 11  description   8809 non-null   object
dtypes: int64(1), object(11)
memory usage: 826.0+ KB
```