Analysis of Electric Vehicle Adoption in Washington State

June 1, 2024

[1]:
```
# Load the dataset

import pandas as pd

# Load the datasets
file_path = '/Users/tvvr/Downloads/Electric_Vehicle_Population_Data-3.csv'
data = pd.read_csv(file_path)

# Display the first few rows of the dataset to understand its structure
data.head()
```

[1]:

	VIN (1-10)	County	City	State	Postal Code	Model Year	Make	\
0	WBY8P6C58K	King	Seattle	WA	98115.0	2019	BMW	
1	5YJSA1DN4D	Kitsap	Bremerton	WA	98312.0	2013	TESLA	
2	5YJSA1E26J	King	Kent	WA	98042.0	2018	TESLA	
3	WBY2Z2C54E	King	Bellevue	WA	98004.0	2014	BMW	
4	5YJXCDE23J	King	Bellevue	WA	98004.0	2018	TESLA	

	Model	Electric Vehicle Type	\
0	I3	Battery Electric Vehicle (BEV)	
1	MODEL S	Battery Electric Vehicle (BEV)	
2	MODEL S	Battery Electric Vehicle (BEV)	
3	I8	Plug-in Hybrid Electric Vehicle (PHEV)	
4	MODEL X	Battery Electric Vehicle (BEV)	

	Clean Alternative Fuel Vehicle (CAFV) Eligibility	Electric Range	\
0	Clean Alternative Fuel Vehicle Eligible	153	
1	Clean Alternative Fuel Vehicle Eligible	208	
2	Clean Alternative Fuel Vehicle Eligible	249	
3	Not eligible due to low battery range	14	
4	Clean Alternative Fuel Vehicle Eligible	238	

	Base MSRP	Legislative District	DOL Vehicle ID	\
0	0	43.0	259254397	
1	69900	35.0	127420940	
2	0	47.0	170287183	
3	0	41.0	205545868	

```
4          0                  41.0       237977386

                  Vehicle Location  \
0  POINT (-122.3008235 47.6862671)
1  POINT (-122.6961203 47.5759584)
2  POINT (-122.1145138 47.3581107)
3    POINT (-122.202397 47.619252)
4    POINT (-122.202397 47.619252)

                                Electric Utility  2020 Census Tract
0   CITY OF SEATTLE - (WA)|CITY OF TACOMA - (WA)       5.303300e+10
1                         PUGET SOUND ENERGY INC       5.303508e+10
2  PUGET SOUND ENERGY INC||CITY OF TACOMA - (WA)       5.303303e+10
3  PUGET SOUND ENERGY INC||CITY OF TACOMA - (WA)       5.303302e+10
4  PUGET SOUND ENERGY INC||CITY OF TACOMA - (WA)       5.303302e+10
```

[2]:
```
# Summary statistics for numerical columns
summary_stats = data.describe()

# Check for missing values
missing_values = data.isnull().sum()

# Display the summary statistics and missing values
summary_stats, missing_values
```

[2]:
```
(         Postal Code     Model Year  Electric Range      Base MSRP  \
 count  186876.000000  186879.000000   186879.000000  186879.000000
 mean    98177.500144    2020.659614       56.707790    1011.918487
 std      2374.643395       2.991398       90.788807    8115.767740
 min      1731.000000    1997.000000        0.000000       0.000000
 25%     98052.000000    2019.000000        0.000000       0.000000
 50%     98122.000000    2022.000000        0.000000       0.000000
 75%     98371.000000    2023.000000       73.000000       0.000000
 max     99577.000000    2024.000000      337.000000  845000.000000

        Legislative District  DOL Vehicle ID  2020 Census Tract
 count         186476.000000    1.868790e+05       1.868760e+05
 mean              29.069012    2.225627e+08       5.297773e+10
 std               14.900971    7.463672e+07       1.570887e+09
 min                1.000000    4.385000e+03       1.001020e+09
 25%               18.000000    1.850864e+08       5.303301e+10
 50%               33.000000    2.302081e+08       5.303303e+10
 75%               42.000000    2.577993e+08       5.305307e+10
 max               49.000000    4.792548e+08       5.603300e+10  ,
 VIN (1-10)                                             0
 County                                                 3
 City                                                   3
```

```
State                                                 0
Postal Code                                           3
Model Year                                            0
Make                                                  0
Model                                                 0
Electric Vehicle Type                                 0
Clean Alternative Fuel Vehicle (CAFV) Eligibility     0
Electric Range                                        0
Base MSRP                                             0
Legislative District                                403
DOL Vehicle ID                                        0
Vehicle Location                                      8
Electric Utility                                      3
2020 Census Tract                                     3
dtype: int64)
```

[3]:
```
# Drop rows with missing values for simplicity
cleaned_data = data.dropna()

# Display the number of rows and columns before and after cleaning
original_shape = data.shape
cleaned_shape = cleaned_data.shape

original_shape, cleaned_shape
```

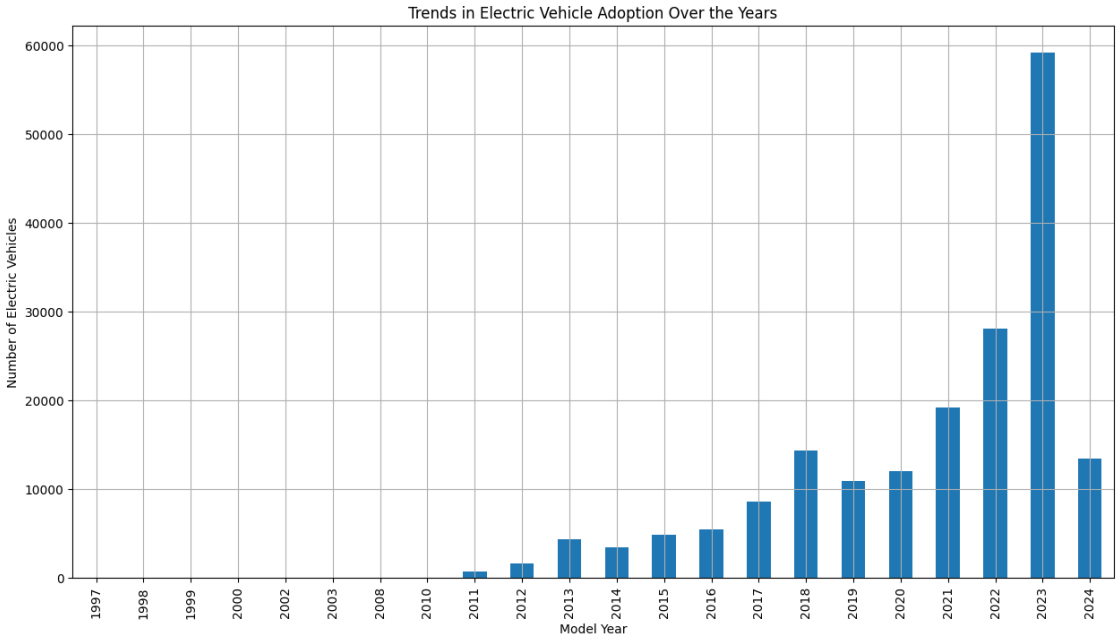
[3]:
```
((186879, 17), (186471, 17))
```

[5]:
```
# Distribution of Electric Vehicle Types
ev_type_distribution = cleaned_data['Electric Vehicle Type'].value_counts()

# Display the distribution
ev_type_distribution
```

[5]:
```
Electric Vehicle Type
Battery Electric Vehicle (BEV)            146021
Plug-in Hybrid Electric Vehicle (PHEV)     40450
Name: count, dtype: int64
```

[13]:
```
# Trends Over Time: Count the number of electric vehicles by model year
trends_over_time = cleaned_data['Model Year'].value_counts().sort_index()

# Plot the trends over time
plt.figure(figsize=(15, 8))
trends_over_time.plot(kind='bar')
plt.title('Trends in Electric Vehicle Adoption Over the Years')
plt.xlabel('Model Year')
plt.ylabel('Number of Electric Vehicles')
```

```
plt.grid(True)
plt.show()
```



[8]:
```
# Re-extract latitude and longitude from Vehicle Location
cleaned_data['Latitude'] = cleaned_data['Vehicle Location'].apply(lambda x:␣
 ↪float(x.strip('POINT ()').split()[1]))
cleaned_data['Longitude'] = cleaned_data['Vehicle Location'].apply(lambda x:␣
 ↪float(x.strip('POINT ()').split()[0]))

# Verify the columns are correctly created
cleaned_data[['Latitude', 'Longitude']].head()
```

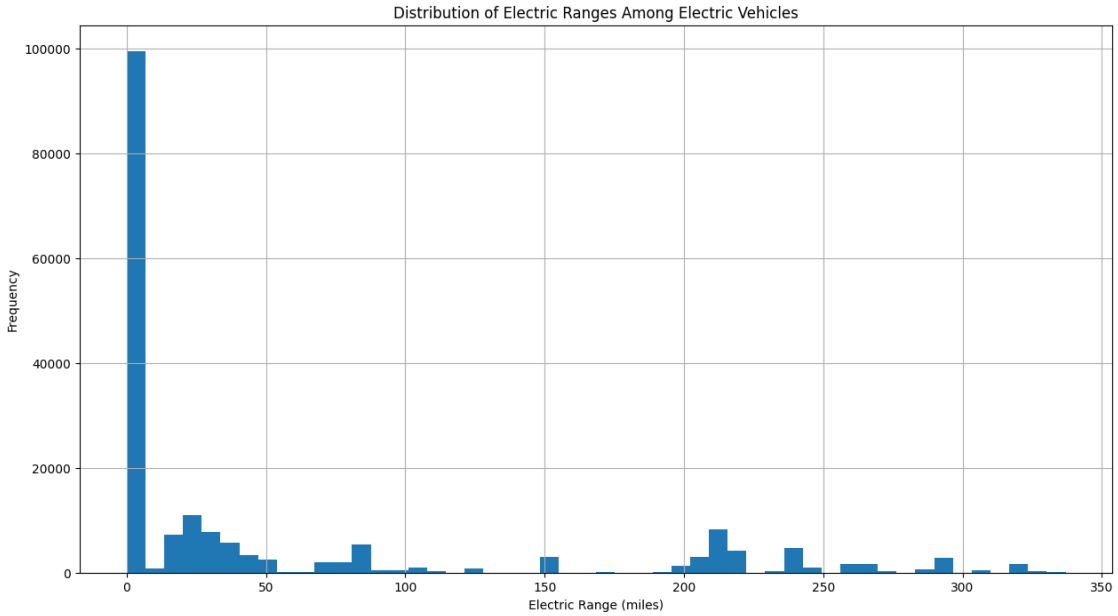
```
/var/folders/nl/520j0msd7vvdfj784n0twgxm0000gn/T/ipykernel_58259/293121837.py:2:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  cleaned_data['Latitude'] = cleaned_data['Vehicle Location'].apply(lambda x:
float(x.strip('POINT ()').split()[1]))
/var/folders/nl/520j0msd7vvdfj784n0twgxm0000gn/T/ipykernel_58259/293121837.py:3:
SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead
```

```
See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  cleaned_data['Longitude'] = cleaned_data['Vehicle Location'].apply(lambda x:
float(x.strip('POINT ()').split()[0]))
```

[8]:

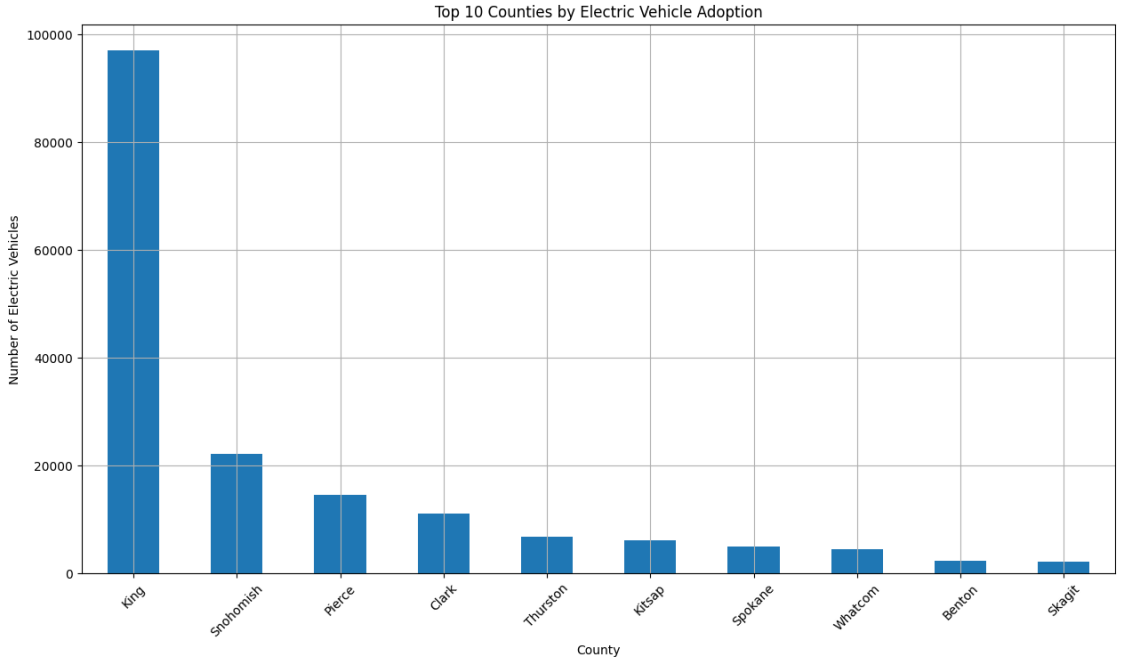
	Latitude	Longitude
0	47.686267	-122.300824
1	47.575958	-122.696120
2	47.358111	-122.114514
3	47.619252	-122.202397
4	47.619252	-122.202397

[9]:
```
# Analyze the distribution of electric ranges
plt.figure(figsize=(15, 8))
cleaned_data['Electric Range'].hist(bins=50)
plt.title('Distribution of Electric Ranges Among Electric Vehicles')
plt.xlabel('Electric Range (miles)')
plt.ylabel('Frequency')
plt.grid(True)
plt.show()
```



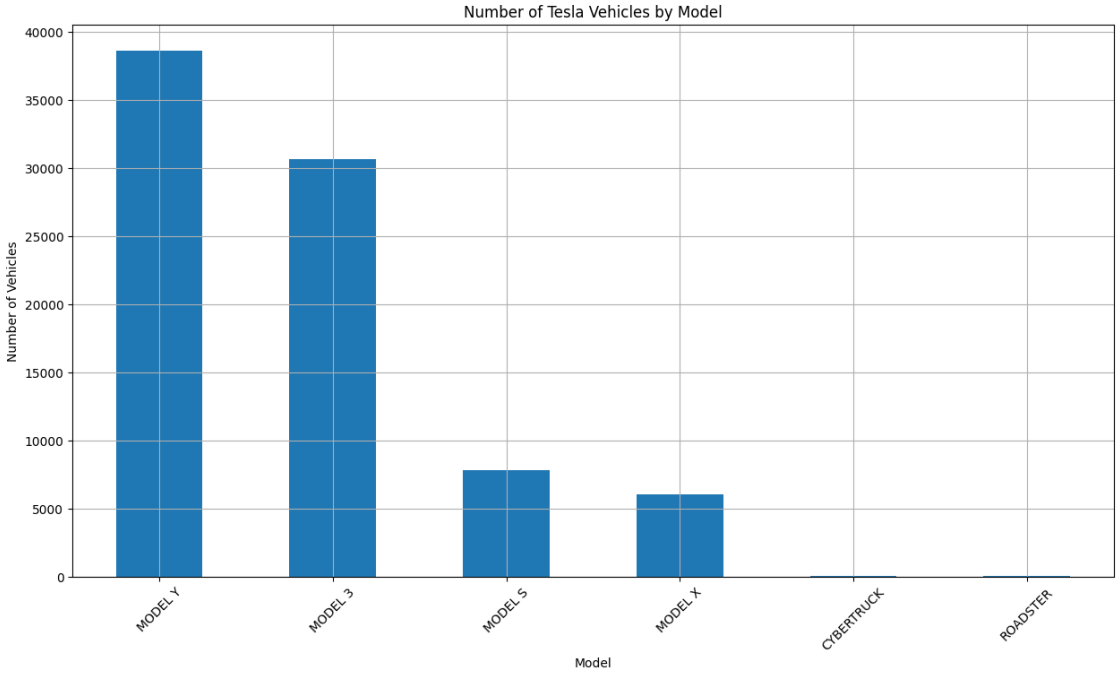
[10]:
```
# Adoption rates of electric vehicles by county
county_adoption = cleaned_data['County'].value_counts().head(10)

# Plot the adoption rates by county
plt.figure(figsize=(15, 8))
county_adoption.plot(kind='bar')
```

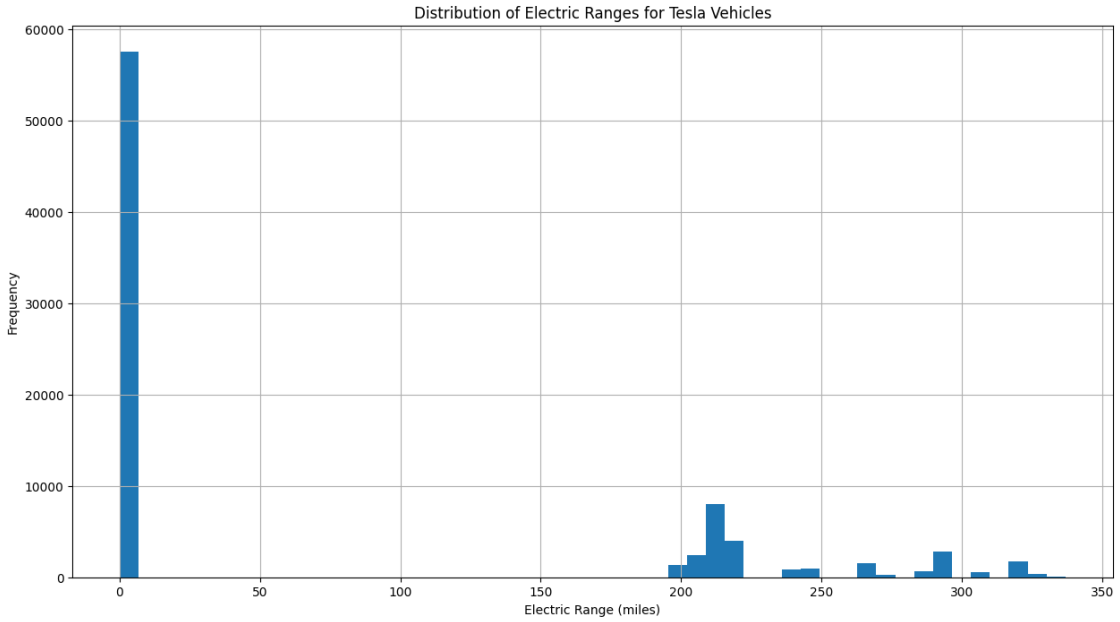
```
plt.title('Top 10 Counties by Electric Vehicle Adoption')
plt.xlabel('County')
plt.ylabel('Number of Electric Vehicles')
plt.xticks(rotation=45)
plt.grid(True)
plt.show()
```



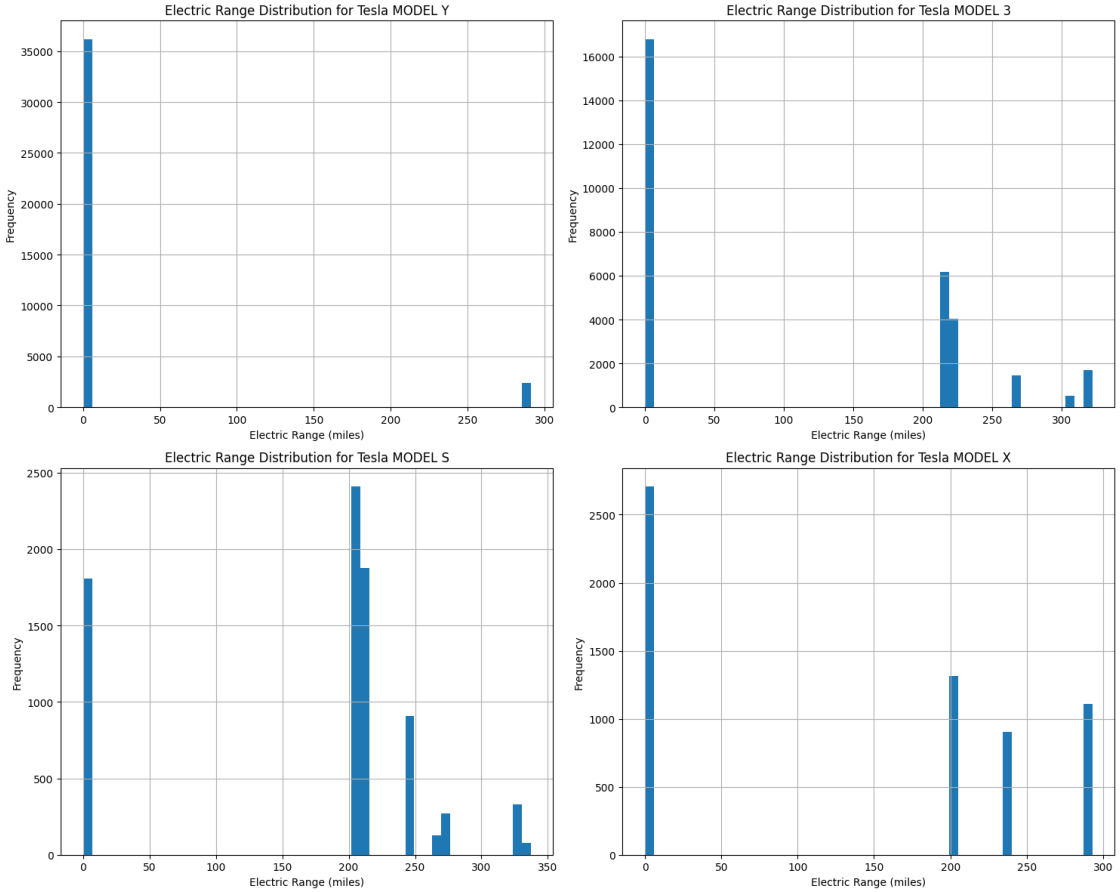
[11]:
```
# Filter the dataset for Tesla vehicles
tesla_data = cleaned_data[cleaned_data['Make'] == 'TESLA']

# Count the number of Tesla vehicles by model
tesla_model_counts = tesla_data['Model'].value_counts()

# Plot the number of Tesla vehicles by model
plt.figure(figsize=(15, 8))
tesla_model_counts.plot(kind='bar')
plt.title('Number of Tesla Vehicles by Model')
plt.xlabel('Model')
plt.ylabel('Number of Vehicles')
plt.xticks(rotation=45)
plt.grid(True)
plt.show()

# Analyze the distribution of electric ranges for Tesla models
plt.figure(figsize=(15, 8))
```

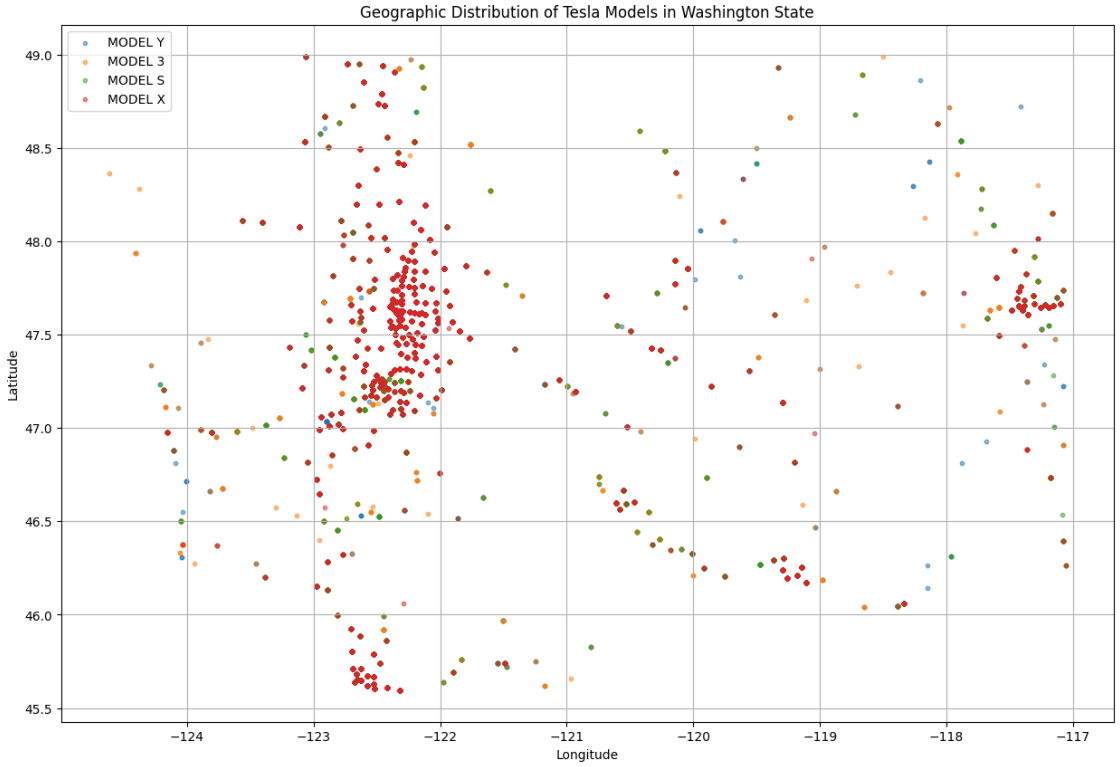
```
tesla_data['Electric Range'].hist(bins=50)
plt.title('Distribution of Electric Ranges for Tesla Vehicles')
plt.xlabel('Electric Range (miles)')
plt.ylabel('Frequency')
plt.grid(True)
plt.show()
```



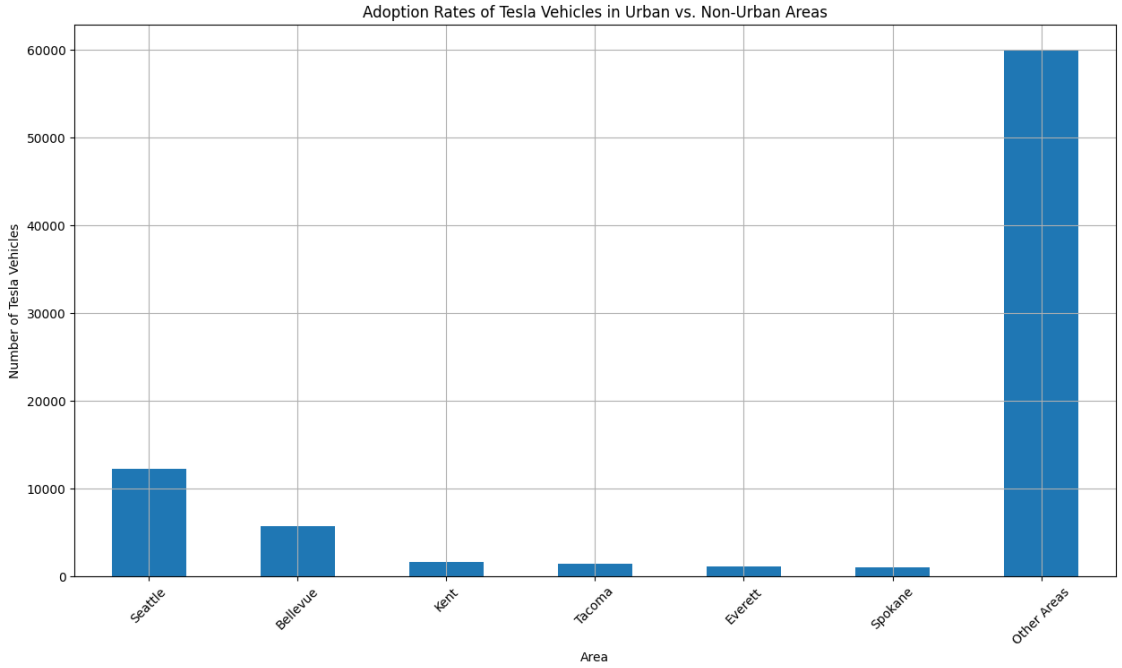


[12]:
```
# Detailed analysis by Tesla model: Electric range distribution for each model
models = ['MODEL Y', 'MODEL 3', 'MODEL S', 'MODEL X']

plt.figure(figsize=(15, 12))

for i, model in enumerate(models, 1):
    plt.subplot(2, 2, i)
    model_data = tesla_data[tesla_data['Model'] == model]
    model_data['Electric Range'].hist(bins=50)
    plt.title(f'Electric Range Distribution for Tesla {model}')
    plt.xlabel('Electric Range (miles)')
    plt.ylabel('Frequency')
    plt.grid(True)

plt.tight_layout()
plt.show()

# Regional analysis: Geographic distribution of Tesla models
plt.figure(figsize=(15, 10))
for model in models:
    model_data = tesla_data[tesla_data['Model'] == model]
    plt.scatter(model_data['Longitude'], model_data['Latitude'], label=model,␣
 ↪alpha=0.5, s=10)

plt.title('Geographic Distribution of Tesla Models in Washington State')
plt.xlabel('Longitude')
plt.ylabel('Latitude')
plt.legend()
plt.grid(True)
plt.show()
```


Electric Range Distribution for Tesla MODEL Y
Frequency
Electric Range (miles)
Electric Range Distribution for Tesla MODEL 3
Frequency
Electric Range (miles)
Electric Range Distribution for Tesla MODEL S
Frequency
Electric Range (miles)
Electric Range Distribution for Tesla MODEL X
Frequency
Electric Range (miles)



[15]:
```
# Prepare data for comparison using pd.concat
adoption_comparison = pd.concat([urban_tesla_adoption, pd.
  ↪Series(non_urban_tesla_adoption, index=['Other Areas'])])

# Plot the adoption rates
plt.figure(figsize=(15, 8))
adoption_comparison.plot(kind='bar')
plt.title('Adoption Rates of Tesla Vehicles in Urban vs. Non-Urban Areas')
plt.xlabel('Area')
plt.ylabel('Number of Tesla Vehicles')
plt.xticks(rotation=45)
plt.grid(True)
plt.show()

# Display the adoption rates for urban and non-urban areas
adoption_comparison
```

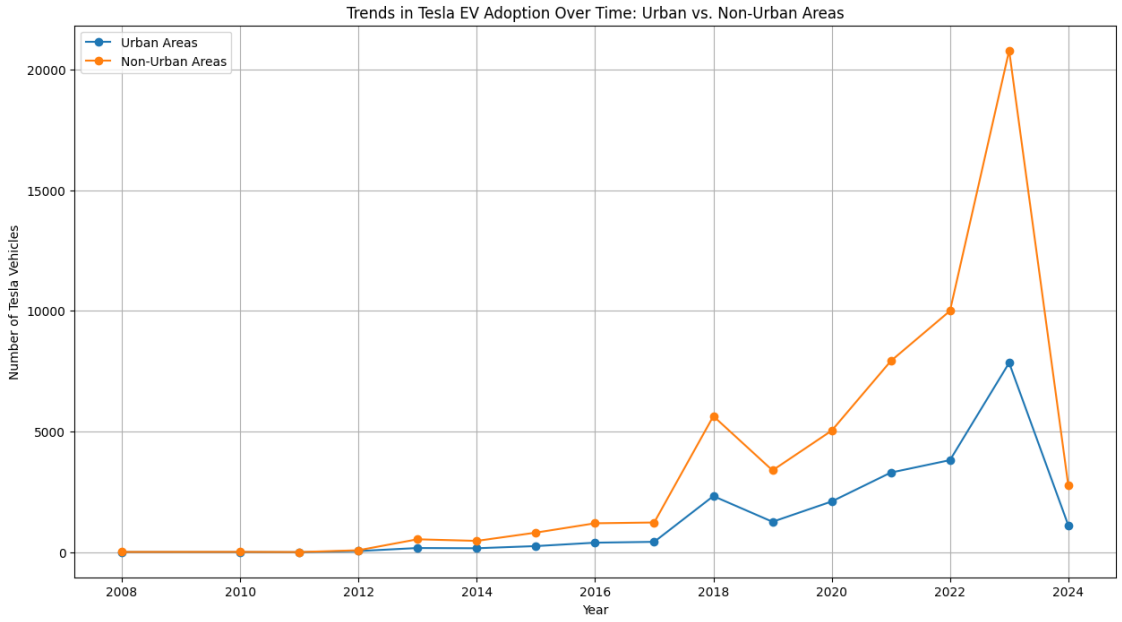


[15]:
```
Seattle        12237
Bellevue        5739
Kent            1628
Tacoma          1413
Everett         1159
Spokane         1072
Other Areas    59901
dtype: int64
```

[16]:
```
# Extract the year from the model year column
tesla_data['Year'] = tesla_data['Model Year']

# Define major urban areas
urban_areas = ['Seattle', 'Bellevue', 'Kent', 'Everett', 'Tacoma', 'Spokane']

# Filter Tesla data for urban and non-urban areas
urban_tesla_data = tesla_data[tesla_data['City'].isin(urban_areas)]
non_urban_tesla_data = tesla_data[~tesla_data['City'].isin(urban_areas)]

# Calculate yearly trends for urban and non-urban areas
urban_trends = urban_tesla_data['Year'].value_counts().sort_index()
non_urban_trends = non_urban_tesla_data['Year'].value_counts().sort_index()

# Plot the trends over time
plt.figure(figsize=(15, 8))
```

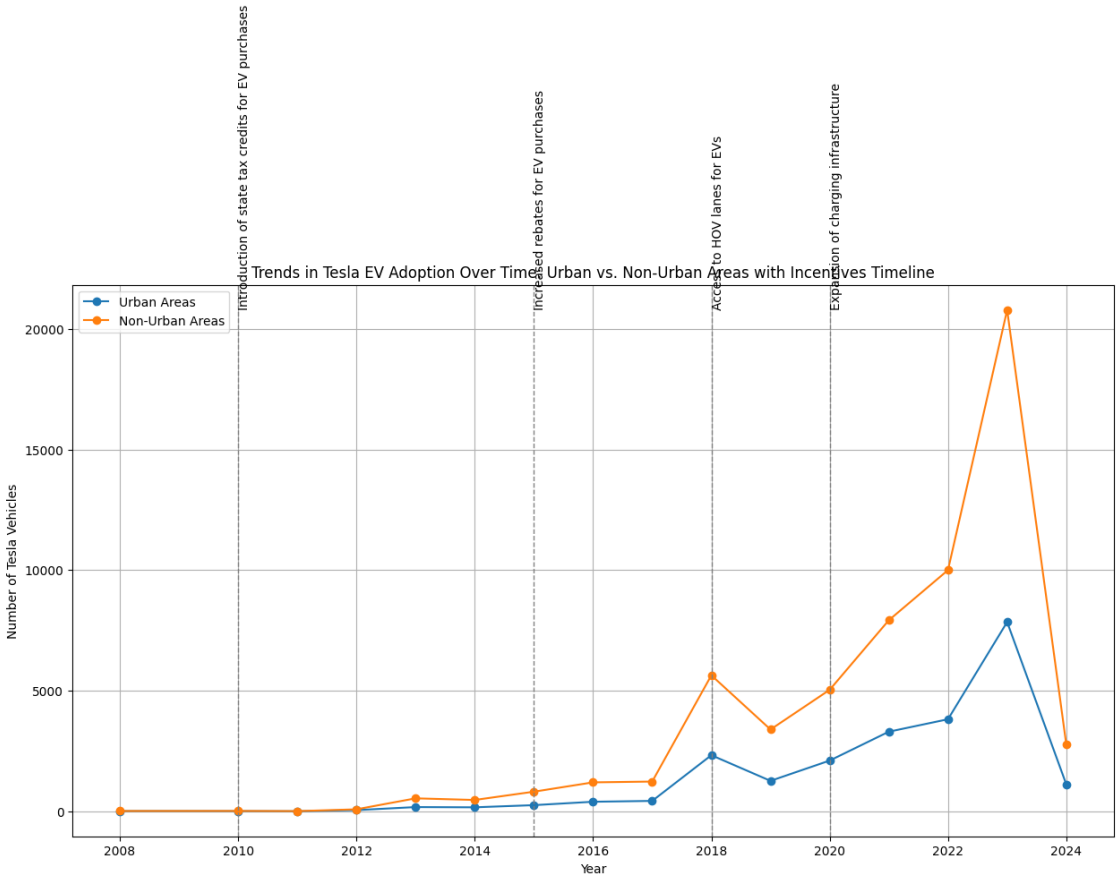
```
plt.plot(urban_trends.index, urban_trends.values, label='Urban Areas',␣
 ↪marker='o')
plt.plot(non_urban_trends.index, non_urban_trends.values, label='Non-Urban␣
 ↪Areas', marker='o')
plt.title('Trends in Tesla EV Adoption Over Time: Urban vs. Non-Urban Areas')
plt.xlabel('Year')
plt.ylabel('Number of Tesla Vehicles')
plt.legend()
plt.grid(True)
plt.show()
```

```
/var/folders/nl/520j0msd7vvdfj784n0twgxm0000gn/T/ipykernel_58259/3069946266.py:2
: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-
docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
  tesla_data['Year'] = tesla_data['Model Year']
```



[17]:
```
# Mock-up data for incentives
incentives_data = {
    'Year': [2010, 2015, 2018, 2020],
    'Incentive': [
        'Introduction of state tax credits for EV purchases',
        'Increased rebates for EV purchases',
        'Access to HOV lanes for EVs',
```

```
        'Expansion of charging infrastructure'
    ]
}

# Create a DataFrame for incentives
incentives_df = pd.DataFrame(incentives_data)

# Plot the EV adoption trends with incentives timeline
plt.figure(figsize=(15, 8))
plt.plot(urban_trends.index, urban_trends.values, label='Urban Areas',␣
 ↪marker='o')
plt.plot(non_urban_trends.index, non_urban_trends.values, label='Non-Urban␣
 ↪Areas', marker='o')

# Overlay incentives timeline
for i in range(len(incentives_df)):
    plt.axvline(x=incentives_df['Year'][i], color='gray', linestyle='--',␣
 ↪linewidth=1)
    plt.text(incentives_df['Year'][i], max(max(urban_trends.values),␣
 ↪max(non_urban_trends.values)),
             incentives_df['Incentive'][i], rotation=90,␣
 ↪verticalalignment='bottom')

plt.title('Trends in Tesla EV Adoption Over Time: Urban vs. Non-Urban Areas␣
 ↪with Incentives Timeline')
plt.xlabel('Year')
plt.ylabel('Number of Tesla Vehicles')
plt.legend()
plt.grid(True)
plt.show()
```


Introduction of state tax credits for EV purchases
Increased rebates for EV purchases
Access to HOV lanes for EVs
Expansion of charging infrastructure
Trends in Tesla EV Adoption Over Time: Urban vs. Non-Urban Areas with Incentives Timeline
Urban Areas
Non-Urban Areas
Number of Tesla Vehicles
0
5000
10000
15000
20000
2008
2010
2012
2014
2016
2018
2020
2022
2024
Year

[ ]: