In [11]:
```
#Import Libraries
import matplotlib.pyplot as plt
import pandas as pd
```

In [12]:
```
#Read in Electric Vehicle Population dataset in from data.gov
data = pd.read_csv('https://data.wa.gov/api/views/f6w7-q2d2/rows.csv?accessType=DOWNLOAD')
data.head()
```

Out[12]:

	VIN (1-10)	County	City	State	Postal Code	Model Year	Make	Model	Electric Vehicle Type	Clean Alternative Fuel Vehicle (CAFV) Eligibility	Electric Range	Base MSRP	Legislative District	Vehi
0	5YJ3E1EB4L	Yakima	Yakima	WA	98908.0	2020	TESLA	MODEL 3	Battery Electric Vehicle (BEV)	Clean Alternative Fuel Vehicle Eligible	322	0	14.0	1271
1	5YJ3E1EA7K	San Diego	San Diego	CA	92101.0	2019	TESLA	MODEL 3	Battery Electric Vehicle (BEV)	Clean Alternative Fuel Vehicle Eligible	220	0	NaN	2666
2	7JRBR0FL9M	Lane	Eugene	OR	97404.0	2021	VOLVO	S60	Plug-in Hybrid Electric Vehicle (PHEV)	Not eligible due to low battery range	22	0	NaN	1445
3	5YJXCBE21K	Yakima	Yakima	WA	98908.0	2019	TESLA	MODEL X	Battery Electric Vehicle (BEV)	Clean Alternative Fuel Vehicle Eligible	289	0	14.0	4770
4	5UXKT0C5XH	Snohomish	Bothell	WA	98021.0	2017	BMW	X5	Plug-in Hybrid Electric Vehicle (PHEV)	Not eligible due to low battery range	14	0	1.0	1063

In [13]:
```
# Create a subset of data that only has records of fully electric vehicles from Washington State
wa_bev_data = data[(data["State"] == "WA") &
                   (data["Electric Vehicle Type"] == "Battery Electric Vehicle (BEV)")]
wa_bev_data.head()
```

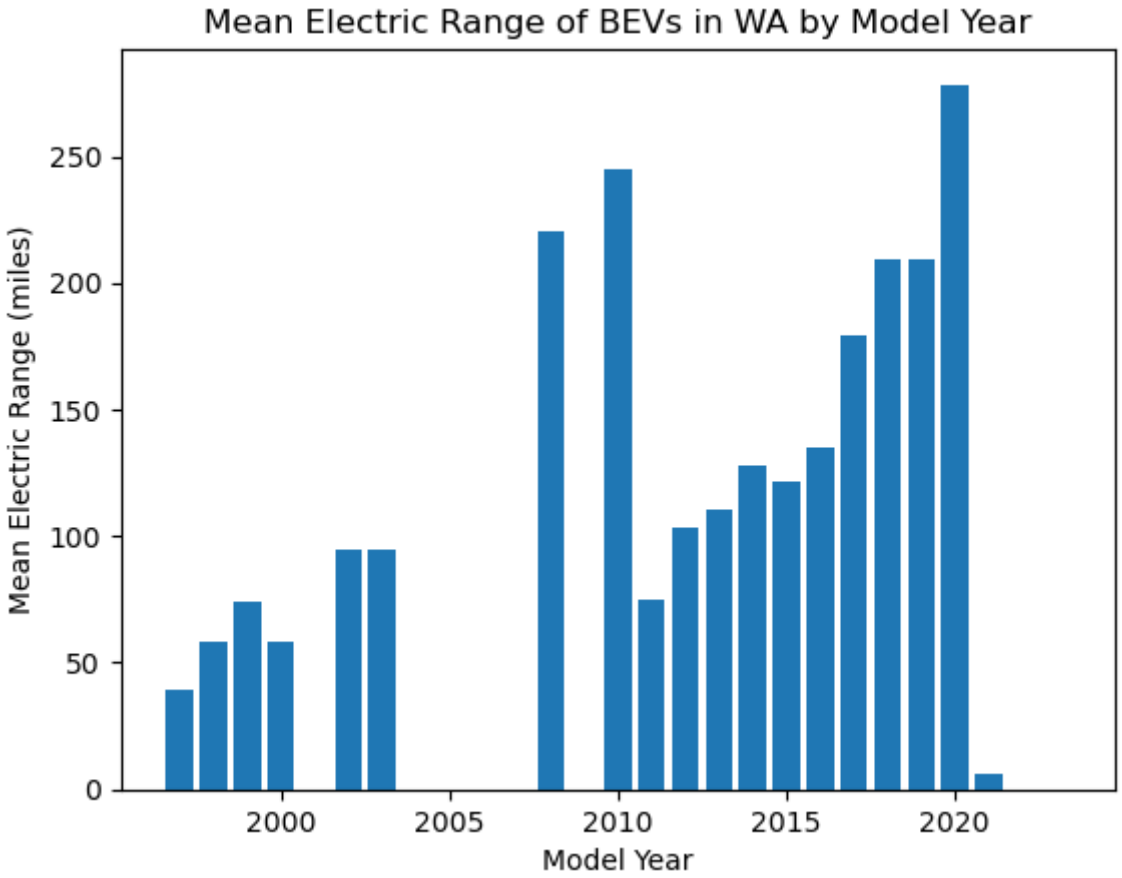
Out[13]:

	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	Veh
0	5YJ3E1EB4L	Yakima	Yakima	WA	98908.0	2020	TESLA	MODEL 3	Battery Electric Vehicle (BEV)	Clean Alternative Fuel Vehicle Eligible	322	0	14.0	127
3	5YJXCBE21K	Yakima	Yakima	WA	98908.0	2019	TESLA	MODEL X	Battery Electric Vehicle (BEV)	Clean Alternative Fuel Vehicle Eligible	289	0	14.0	4770
5	1N4AZ0CP4F	Snohomish	Everett	WA	98201.0	2015	NISSAN	LEAF	Battery Electric Vehicle (BEV)	Clean Alternative Fuel Vehicle Eligible	84	0	38.0	1079
6	5YJ3E1EBXJ	Kitsap	Poulsbo	WA	98370.0	2018	TESLA	MODEL 3	Battery Electric Vehicle (BEV)	Clean Alternative Fuel Vehicle Eligible	215	0	23.0	4750
8	1N4AZ0CP3D	Kitsap	Port Orchard	WA	98366.0	2013	NISSAN	LEAF	Battery Electric Vehicle (BEV)	Clean Alternative Fuel Vehicle Eligible	75	0	26.0	2492

In [14]:
```
#Find the mean electric range per year
mean_range_by_year = wa_bev_data.groupby('Model Year')['Electric Range'].mean()
print(mean_range_by_year)
```

```
Model Year
1997     39.000000
1998     58.000000
1999     74.000000
2000     58.000000
2002     95.000000
2003     95.000000
2008    220.000000
2010    245.000000
2011     74.618351
2012    103.372180
2013    110.780089
2014    127.981677
2015    121.456612
2016    135.345520
2017    179.644719
2018    209.507606
2019    208.974407
2020    278.297417
2021      6.298762
2022      0.000000
2023      0.000000
Name: Electric Range, dtype: float64
```

In [15]:
```
# Create bar plot of electric range vs. model year
plt.bar(mean_range_by_year.index, mean_range_by_year)
plt.xlabel('Model Year')
plt.ylabel('Mean Electric Range (miles)')
plt.title('Mean Electric Range of BEVs in WA by Model Year')
plt.show()
```

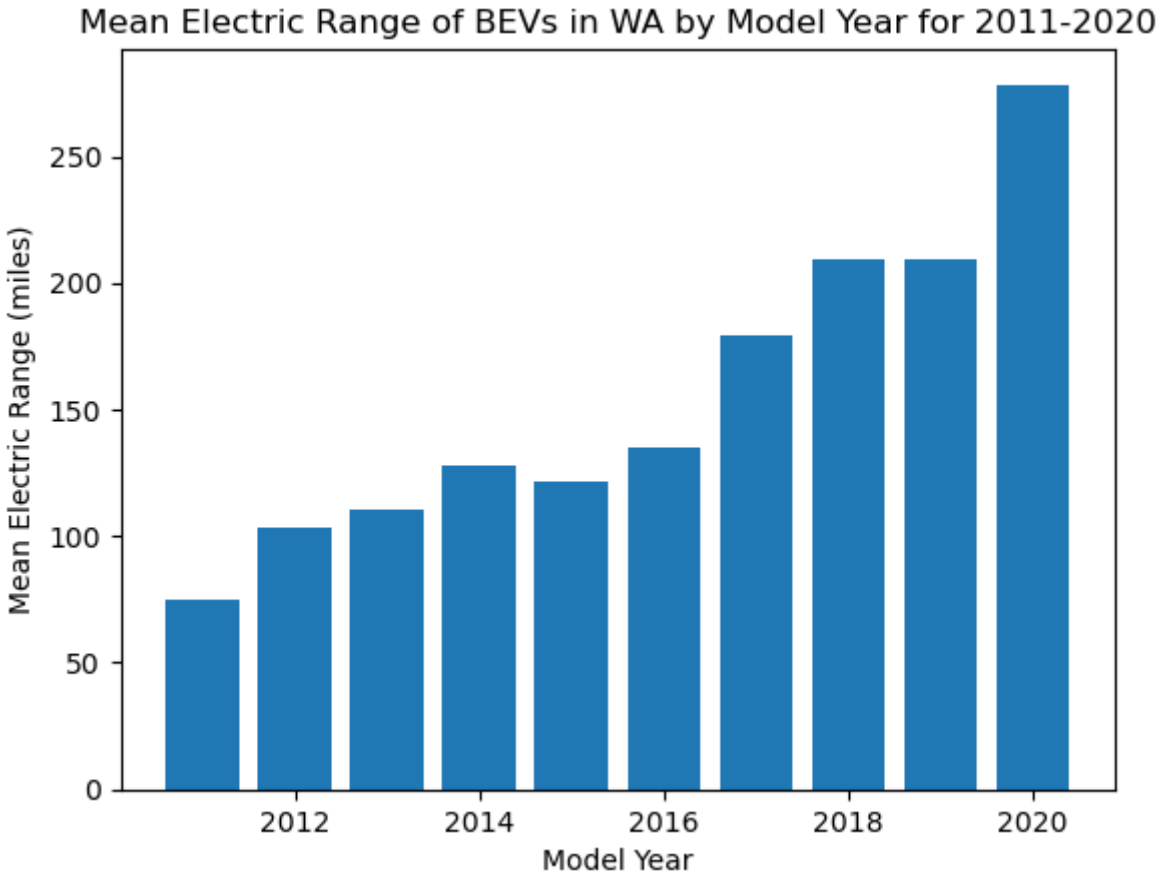


In [17]:
```
#Count the data points per model year
counts = wa_bev_data['Model Year'].value_counts()
print(counts)
```

```
2022    23466
2021    14704
2023    11846
2018     9927
2020     9290
2019     8557
2017     4450
2016     3884
2015     3607
2013     2933
2014     1801
2012      798
2011      752
2008       21
2010       21
2000        9
1999        4
2002        2
1998        1
2003        1
1997        1
Name: Model Year, dtype: int64
```

In [18]:

```
#Based on the first barplot and counts/year output, as well as our desire to see trends in more recent years.
#Create bar plot of electric range vs. model year for years 2011-2020
filtered_wa_bev_data = wa_bev_data[(wa_bev_data['Model Year'] >= 2011) & (wa_bev_data['Model Year'] <= 2020)]
filtered_mean = filtered_wa_bev_data.groupby('Model Year')['Electric Range'].mean()

plt.bar(filtered_mean.index, filtered_mean)
plt.xlabel('Model Year')
plt.ylabel('Mean Electric Range (miles)')
plt.title('Mean Electric Range of BEVs in WA by Model Year for 2011-2020')
plt.savefig('barplot.png')
plt.show()
```



In [25]:
```
# The following analysis was not included in the research paper.
# I still wanted to show you the work I did, since I spent a lot of time figureing it out.
```

In [26]:
```
#Linear regression analysis of electric range and model year.
#Hypothesis: As time passes, the electric range of BEVs increases.
```

In [27]:
```
# Import Libraries
import numpy as np
from sklearn.linear_model import LinearRegression
```

In [22]:
```
# Sort the subset of data by Model Year
subset_sorted = filtered_wa_bev_data .sort_values(by=['Model Year'])
```

In [29]:
```
# Since simple linear regression with time series data does not work, I relabeled model year
# Replace Model Year values 2011-2020 with 1-10
```

```
subset_sorted['Model Year'] = subset_sorted['Model Year'].apply(lambda x: x - min(subset_sorted['Model Year']
subset_sorted.head()
```

Out[29]:

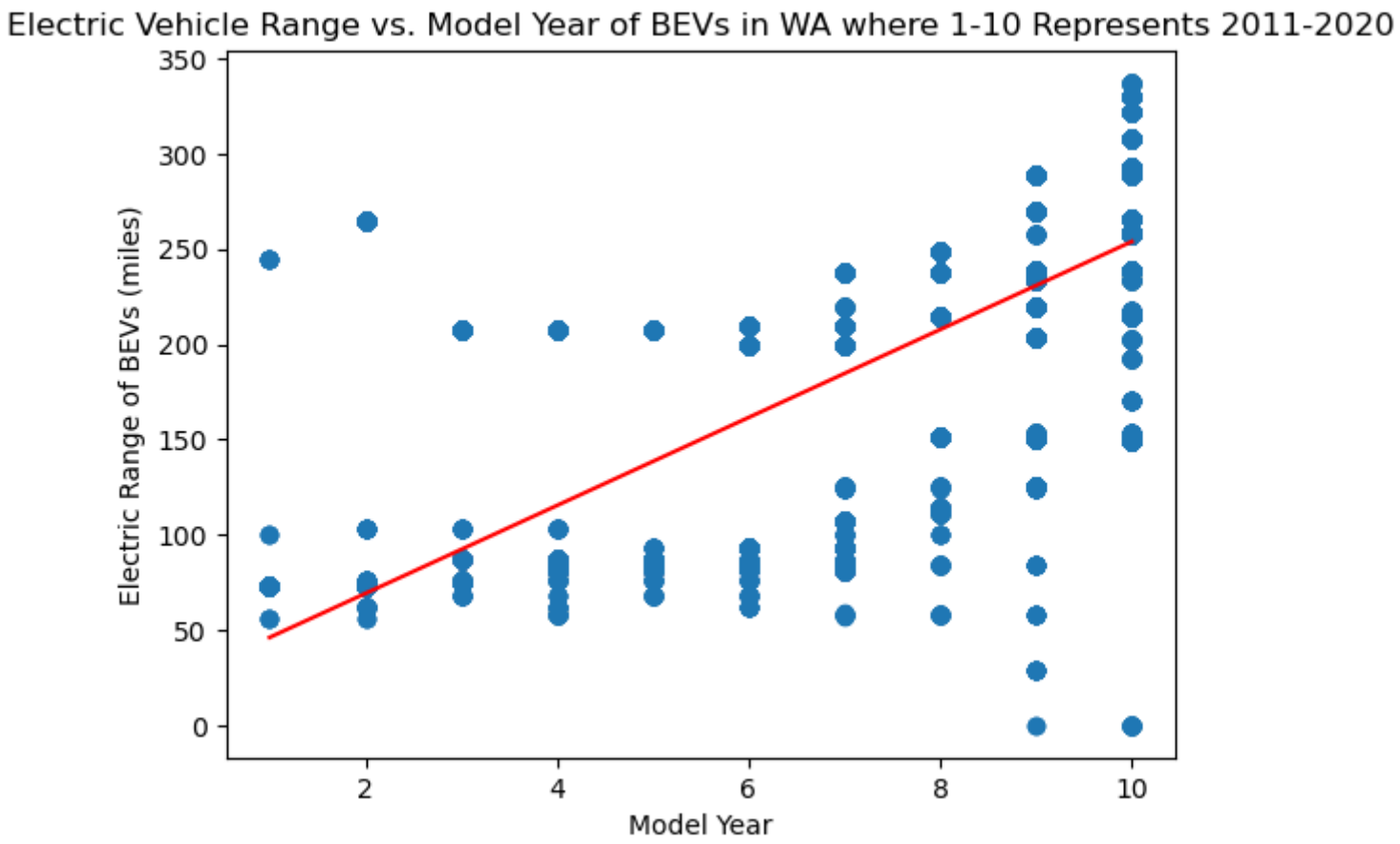
	VIN (1-10)	County	City	State	Postal Code	Model Year	Make	Model	Electric Vehicle Type	Clean Alternative Fuel Vehicle (CAFV) Eligibility	Electric Range	Base MSRP	Legislative District
52733	JN1AZ0CP5B	King	Seattle	WA	98107.0	1	NISSAN	LEAF	Battery Electric Vehicle (BEV)	Clean Alternative Fuel Vehicle Eligible	73	0	36.0
79402	JN1AZ0CP0B	Thurston	Olympia	WA	98502.0	1	NISSAN	LEAF	Battery Electric Vehicle (BEV)	Clean Alternative Fuel Vehicle Eligible	73	0	35.0
119584	JN1AZ0CP7B	Pierce	University Place	WA	98467.0	1	NISSAN	LEAF	Battery Electric Vehicle (BEV)	Clean Alternative Fuel Vehicle Eligible	73	0	28.0
71541	JN1AZ0CP6B	King	Bothell	WA	98011.0	1	NISSAN	LEAF	Battery Electric Vehicle (BEV)	Clean Alternative Fuel Vehicle Eligible	73	0	1.0
38310	JN1AZ0CP5B	King	Covington	WA	98042.0	1	NISSAN	LEAF	Battery Electric Vehicle (BEV)	Clean Alternative Fuel Vehicle Eligible	73	0	47.0

In [33]:

```
# Set X and Y
X = subset_sorted['Model Year'].values.reshape(-1, 1)
Y = subset_sorted['Electric Range'].values.reshape(-1, 1)

# Create a scatter plot of the data
plt.scatter(subset_sorted['Model Year'], subset_sorted['Electric Range'])
plt.xlabel('Model Year')
plt.ylabel('Electric Range of BEVs (miles)')
```

```
plt.title('Electric Vehicle Range vs. Model Year of BEVs in WA where 1-10 Represents 2011-2020')

#Linear Regression
model = LinearRegression()
model.fit(X, Y)
clf = model.fit(X, Y)
predicitons = np.dot(X, clf.coef_)

for index in range (len(predicitons)):
    predicitons[index] = predicitons[index] + clf.intercept_

#Print equation for linear regression line in slope-intercept form
m = model.coef_[0]
b = model.intercept_
print(f"y = {m}x + {b}")

#R squared score --> positive correlation between model year and electric range
r_squared = model.score(X, Y)
print(f"R-squared score: {r_squared:.2f}")

# Display the linear regression line on the scatter plot
plt.plot(X, predicitons, color='red')
```

Out[33]:
```
y = [23.08433411]x + [23.0004163]
R-squared score: 0.52
[<matplotlib.lines.Line2D at 0x7f9dc02a3fd0>]
```


Electric Vehicle Range vs. Model Year of BEVs in WA where 1-10 Represents 2011-2020
Electric Range of BEVs (miles)
Model Year
0
50
100
150
200
250
300
350
2
4
6
8
10

In [ ]: