



# ANALYSIS OF ELECTRIC CARS





# ELECTRIC CARS DATA ANALYSIS

## Brands

Aiways

Audi

BMW

Byton

Citroen

CUPRA

DS

Fiat

Ford

Honda

Total Cars

3

Min Price

€ 53.50K

Models

All

Max Price

€ 64.00K

Body Style

1

Brand	Model	Sum of AccelSec	Sum of Efficiency Km/KwH	Sum of Range_ Km	Sum of TopSpeed_ KmH	Sum of Price in E
Byton	M-Byte 95 kWh 4WD	5.50	244	390	190	€ 64,000
Byton	M-Byte 95 kWh 2WD	7.50	238	400	190	€ 62,000
Total		20.50	704	1115	570	1,79,500

Body Style

SUV

Fast Charge Km/H

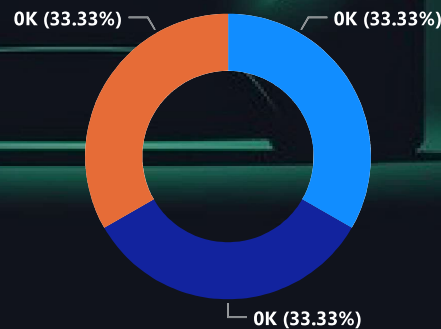
1360

Seats

5

## Plug Types by Model

● M-Byte 72 kW... ● M-Byte 95 k... ● M-Byte 95 k...

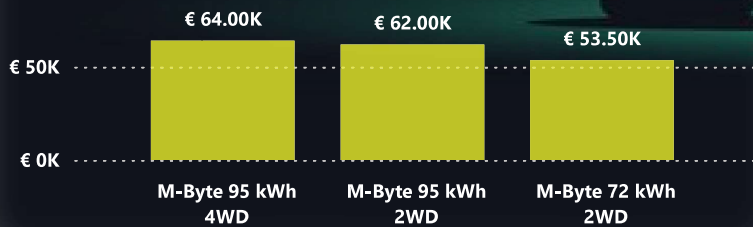


## Total Brand by Rapid Charge

● Yes



## Price Varies by Model





# ELECTRIC CARS DATA ANALYSIS

Brands

Aiways

Audi

BMW

Byton

Citroen

CUPRA

DS

Fiat

Ford

Honda

Total Cars

103

Min Price

€ 20.13K

Models

All

Max Price

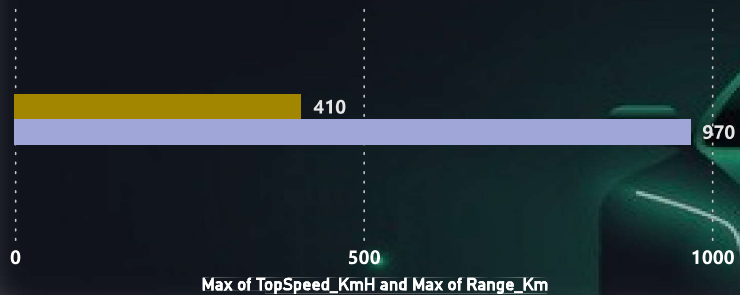
€ 215.0...

Body Style

9

Max of TopSpeed\_KmH and Max of Range\_Km

● Max of TopSpeed\_KmH ● Max of Range\_Km



Body Style

Cabrio

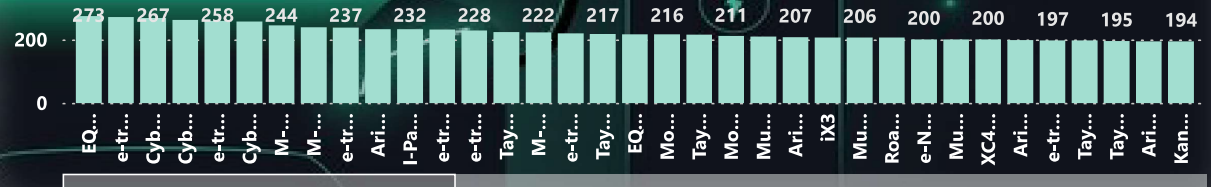
Fast Charge Km/H

45K

Seats

7

Max of Efficiency Km/KwH by Model



Price Varies by Model



Max of AccelSec by Model



# ELECTRIC CAR DATA ANALYSIS

## CONTEXT

The data describes various electric car models with features like acceleration, range, and price. We want to analyze this data to see what features are most important to potential buyers (e.g., long range, fast charging, affordability). This will help both consumers make informed choices and manufacturers understand what electric car features to prioritize.

## OBJECTIVE

The main objective of analyzing this electric car data is to understand what features are most important to potential buyers when choosing an electric car. This will be achieved by:

- Identifying key factors influencing purchase decisions (e.g., range, price, performance).
- Comparing various electric car models based on these factors.

## PROBLEM STATEMENT

**Problem Statement:** Analyze factors influencing electric car selection for potential buyers.

## IMPORT LIBRARIES

```
In [1]: import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline
import warnings
warnings.filterwarnings("ignore")
```

## LOAD DATASET

```
In [2]: df=pd.read_csv("ElectricCarData.csv")
df
```

```
Out[2]:
```

	Brand	Model	AccelSec	TopSpeed_KmH	Range_Km	Efficiency_WhKm	FastChar
0	Tesla	Model 3 Long Range Dual Motor	4.6	233	450	161	
1	Volkswagen	ID.3 Pure	10.0	160	270	167	
2	Polestar	2	4.7	210	400	181	
3	BMW	iX3	6.8	180	360	206	
4	Honda	e	9.5	145	170	168	
...	...	...	...	...	...	...	...
98	Nissan	Ariya 63kWh	7.5	160	330	191	
99	Audi	e-tron S Sportback 55 quattro	4.5	210	335	258	
100	Nissan	Ariya e-4ORCE 63kWh	5.9	200	325	194	
101	Nissan	Ariya e-4ORCE 87kWh Performance	5.1	200	375	232	
102	Byton	M-Byte 95 kWh 2WD	7.5	190	400	238	

103 rows × 14 columns



## INFORMATION ABOUT DATASET

The dataset provided contains information about different electric vehicles. Here's a breakdown of the columns:

- **Brand:** The brand or manufacturer of the electric vehicle.
- **Model:** The model name of the electric vehicle.
- **AccelSec:** Acceleration time from 0 to 100 km/h in seconds.
- **TopSpeed\_KmH:** Top speed of the vehicle in kilometers per hour.
- **Range\_Km:** The range the vehicle can travel on a single charge in kilometers.
- **Efficiency\_WhKm:** Energy efficiency of the vehicle measured in watt-hours per kilometer.
- **FastCharge\_KmH:** Fast charging speed in kilometers per hour.

- **RapidCharge:** Indicates if the vehicle supports rapid charging or not.
- **PowerTrain:** Type of powertrain used in the vehicle.( transmits power from the engine to the wheels)
- **PlugType:** The type of plug used for charging.
- **BodyStyle:** The body style of the vehicle (e.g., sedan, hatchback).
- **Segment:** Segment of the vehicle in terms of size or market positioning.
- **Seats:** Number of seats in the vehicle.
- **PriceEuro:** Price of the vehicle in Euros.

In [70]: df.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 103 entries, 0 to 102
Data columns (total 14 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Brand                 103 non-null   object
1   Model                 103 non-null   object
2   AccelSec              103 non-null   float64
3   TopSpeed_KmH          103 non-null   int64
4   Range_Km              103 non-null   int64
5   Efficiency_WhKm       103 non-null   int64
6   FastCharge_KmH        103 non-null   object
7   RapidCharge           103 non-null   object
8   PowerTrain            103 non-null   object
9   PlugType              103 non-null   object
10  BodyStyle             103 non-null   object
11  Segment               103 non-null   object
12  Seats                 103 non-null   int64
13  PriceEuro             103 non-null   int64
dtypes: float64(1), int64(5), object(8)
memory usage: 11.4+ KB
```

## CHECK NULL VALUES

In [71]: df.isnull().sum()

```
Out[71]: Brand                0
Model                0
AccelSec             0
TopSpeed_KmH         0
Range_Km             0
Efficiency_WhKm      0
FastCharge_KmH       0
RapidCharge          0
PowerTrain           0
PlugType             0
BodyStyle            0
Segment             0
Seats               0
PriceEuro            0
dtype: int64
```

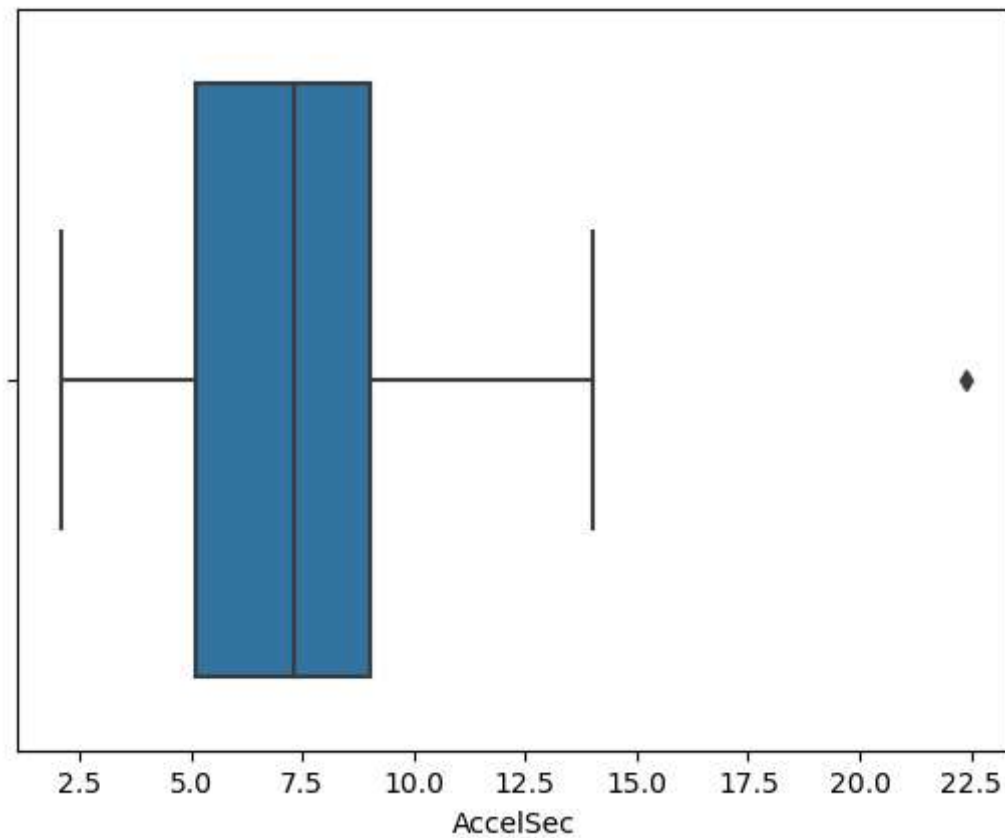
## CHECK DUPLICATED VALUES

```
In [72]: df[df.duplicated()].count()
```

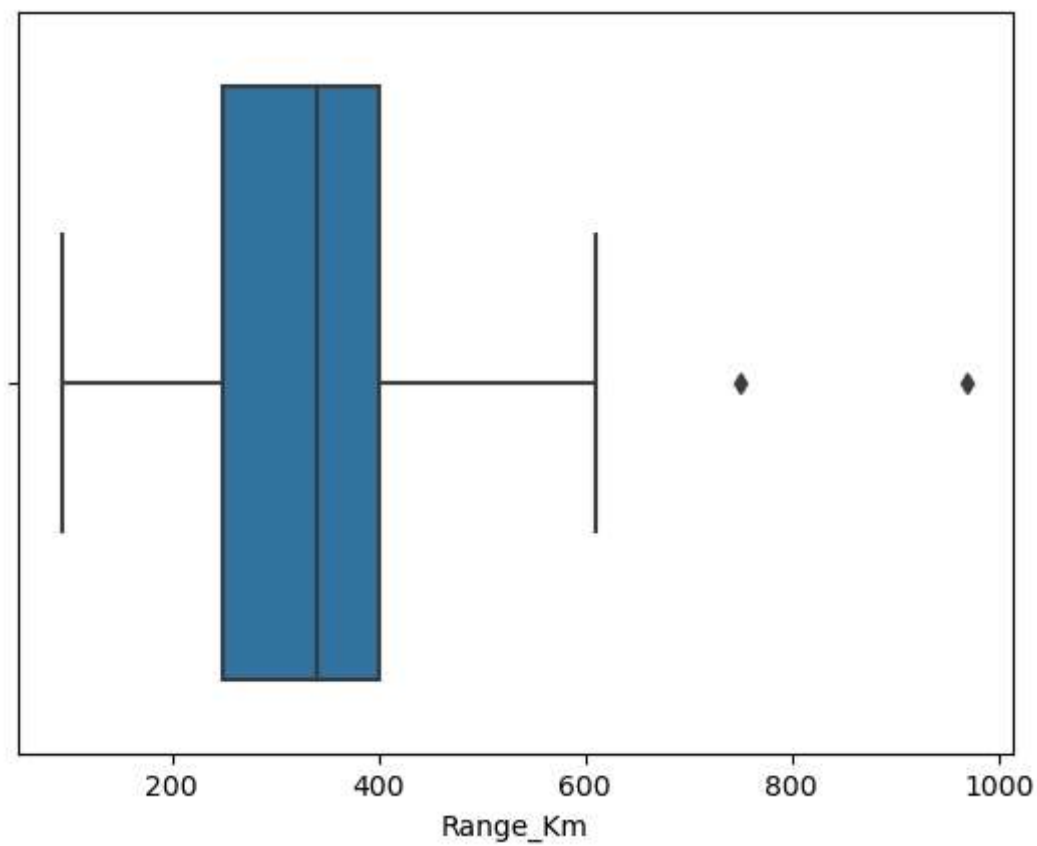
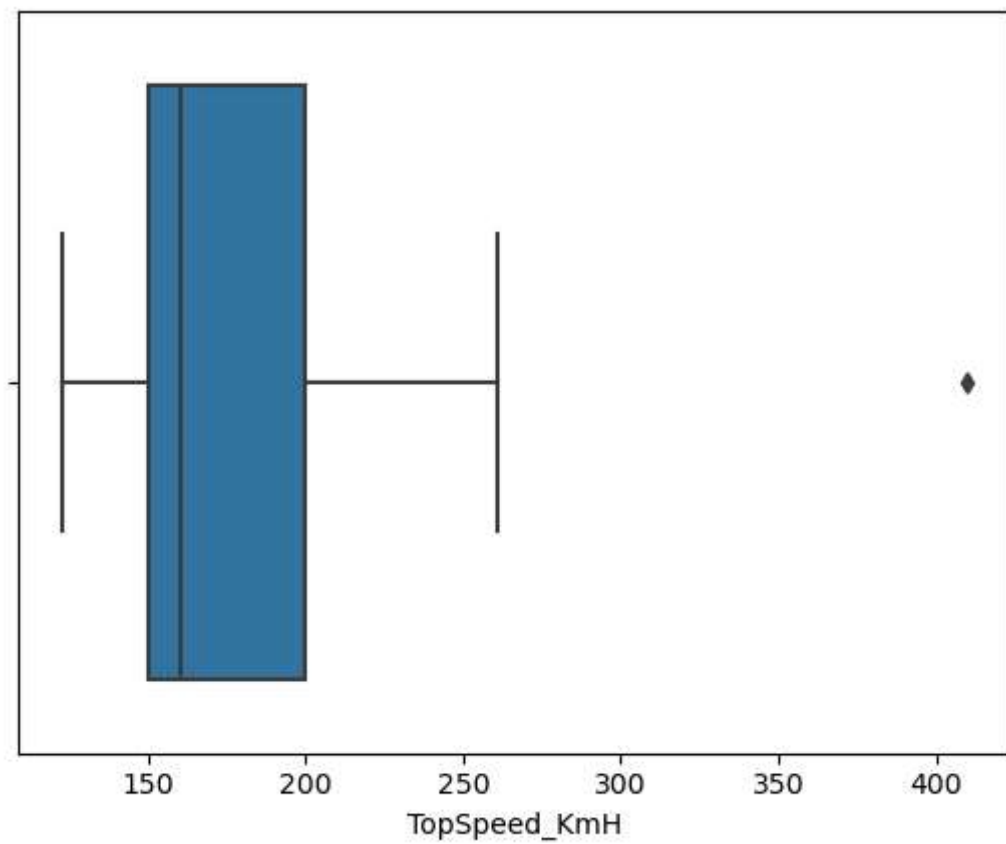
```
Out[72]: Brand          0  
Model          0  
AccelSec       0  
TopSpeed_KmH   0  
Range_Km       0  
Efficiency_WhKm 0  
FastCharge_KmH 0  
RapidCharge    0  
PowerTrain     0  
PlugType       0  
BodyStyle      0  
Segment        0  
Seats          0  
PriceEuro      0  
dtype: int64
```

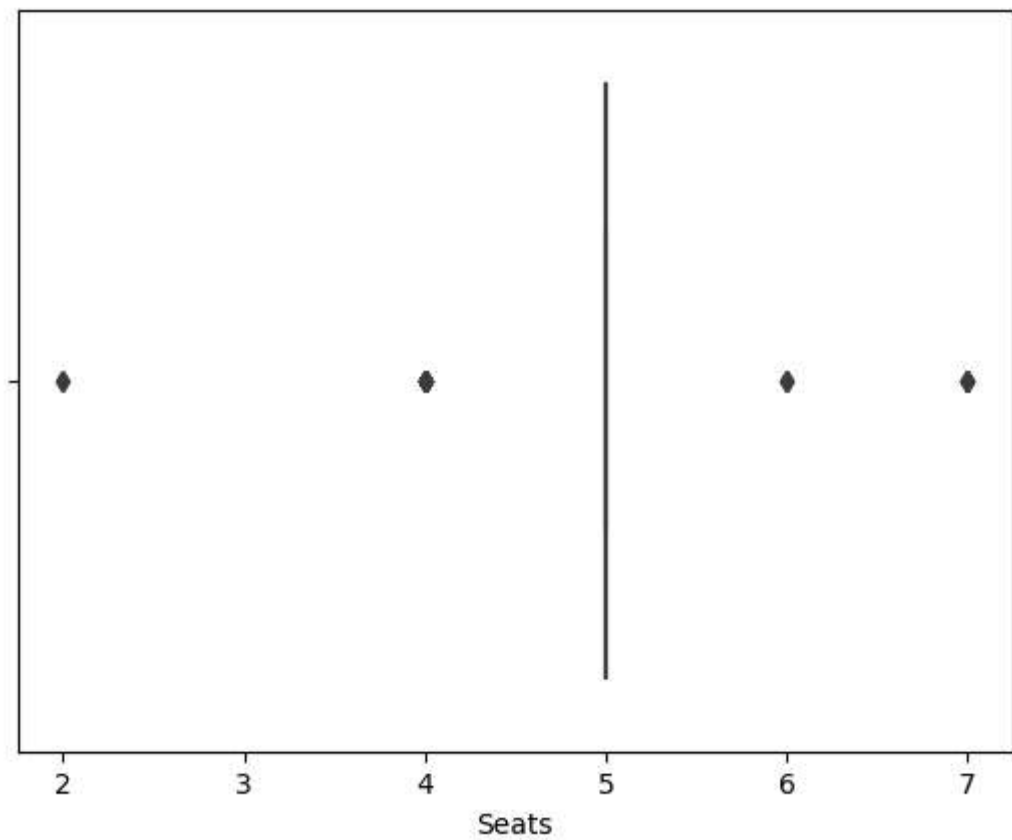
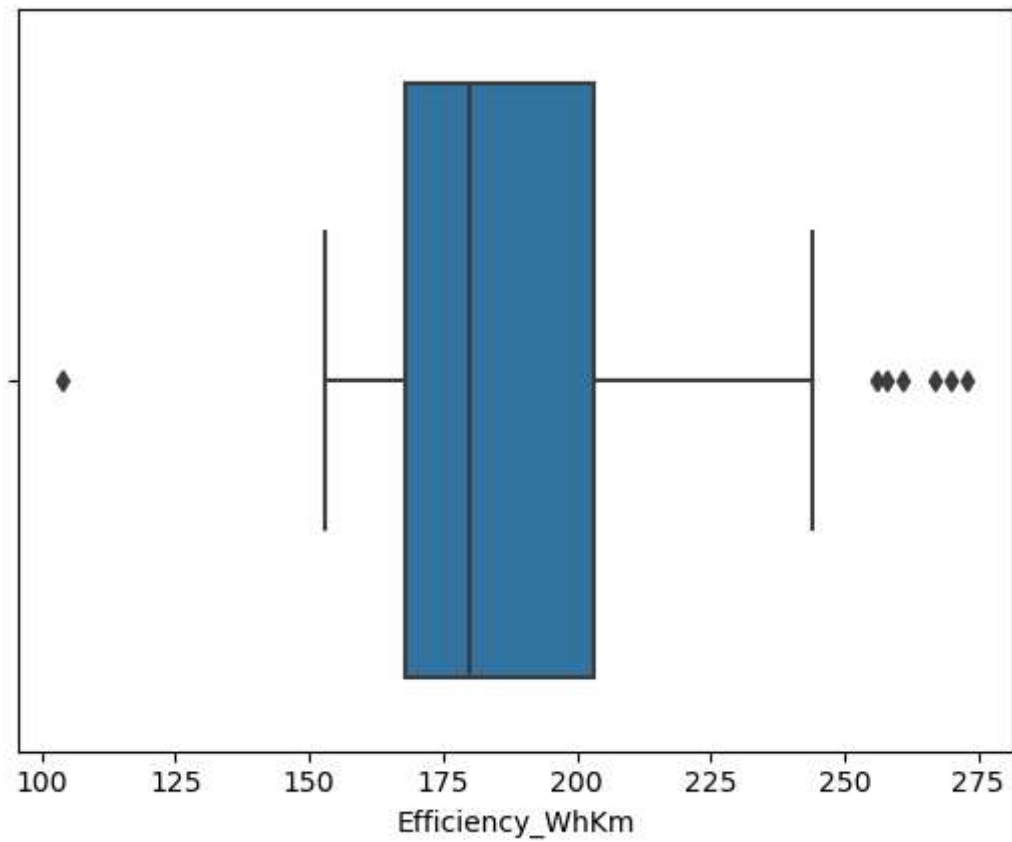
## CHECK OUTLIERS

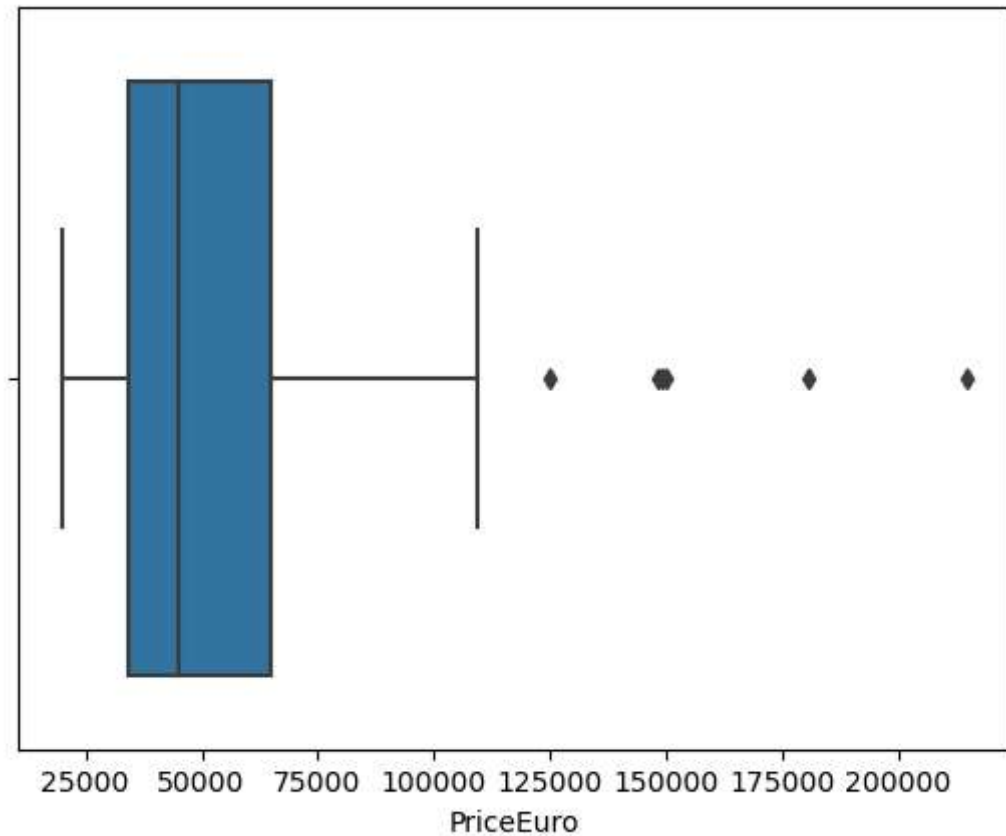
```
In [73]: for i in df.select_dtypes(['int', 'float']):  
         sns.boxplot(data=df, x=i)  
         plt.show()
```









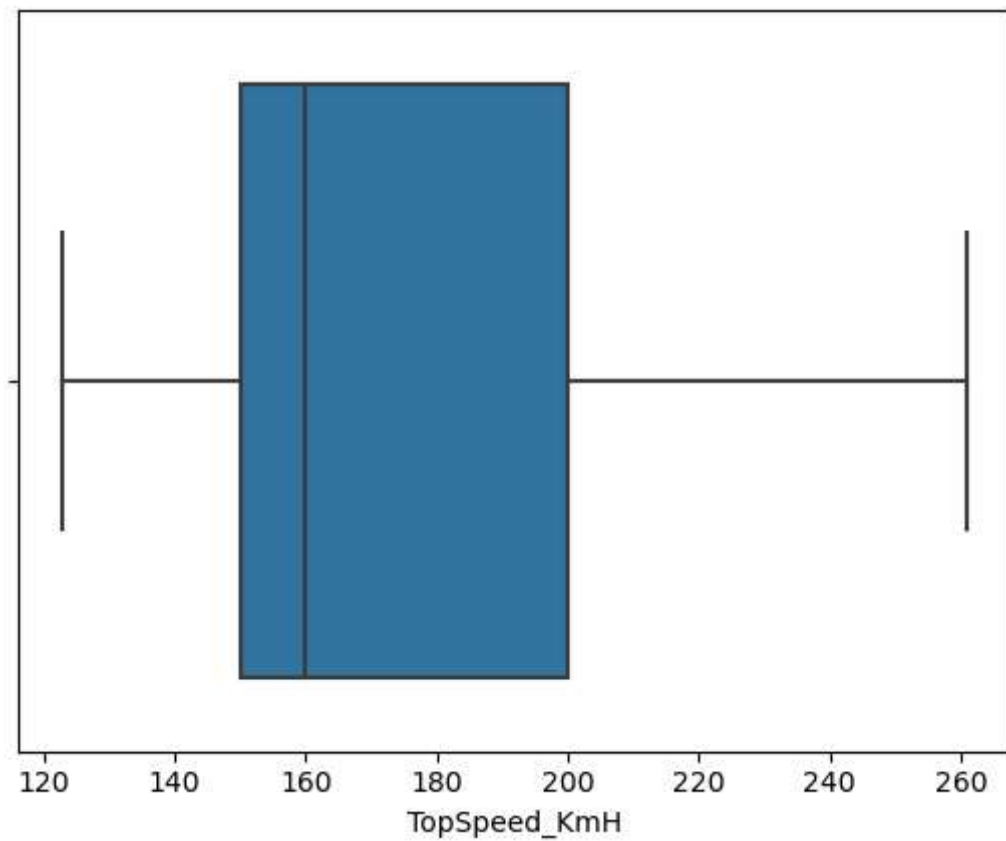
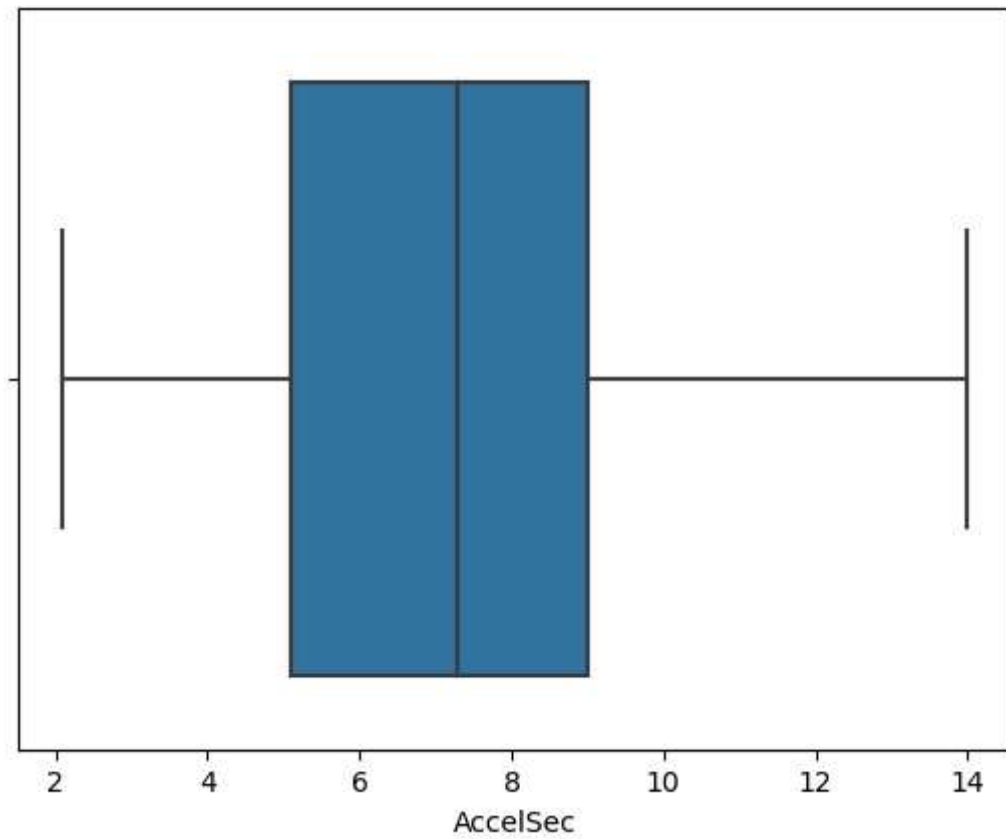


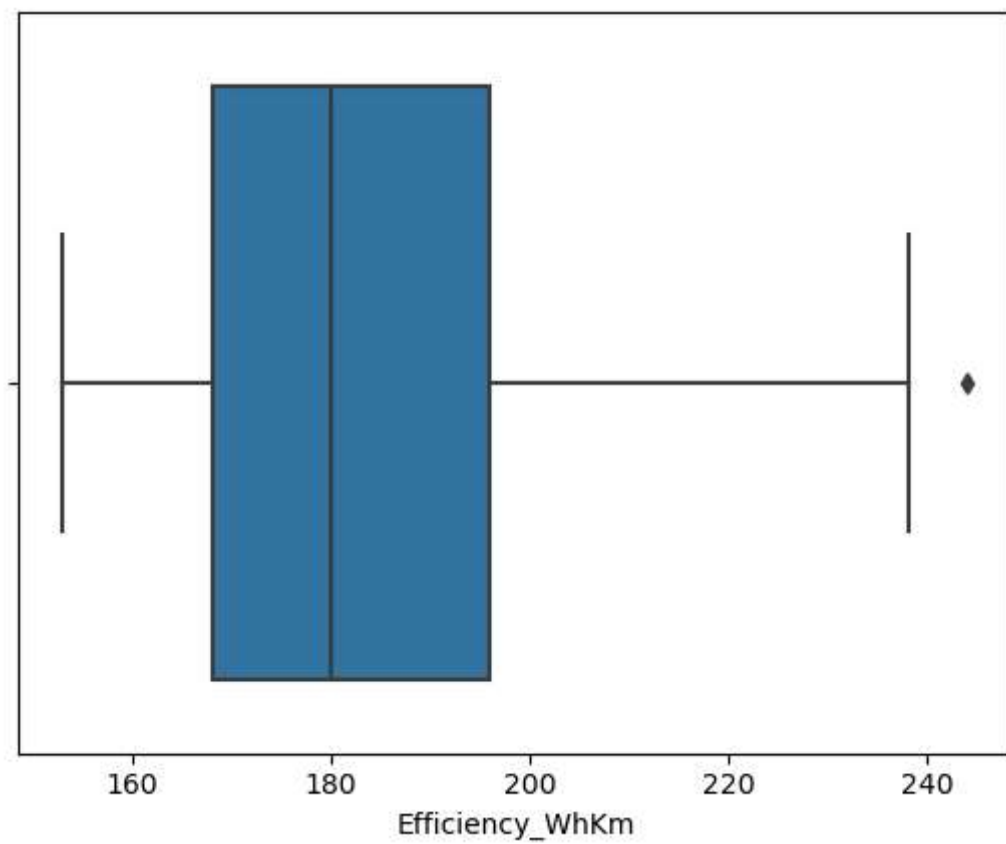
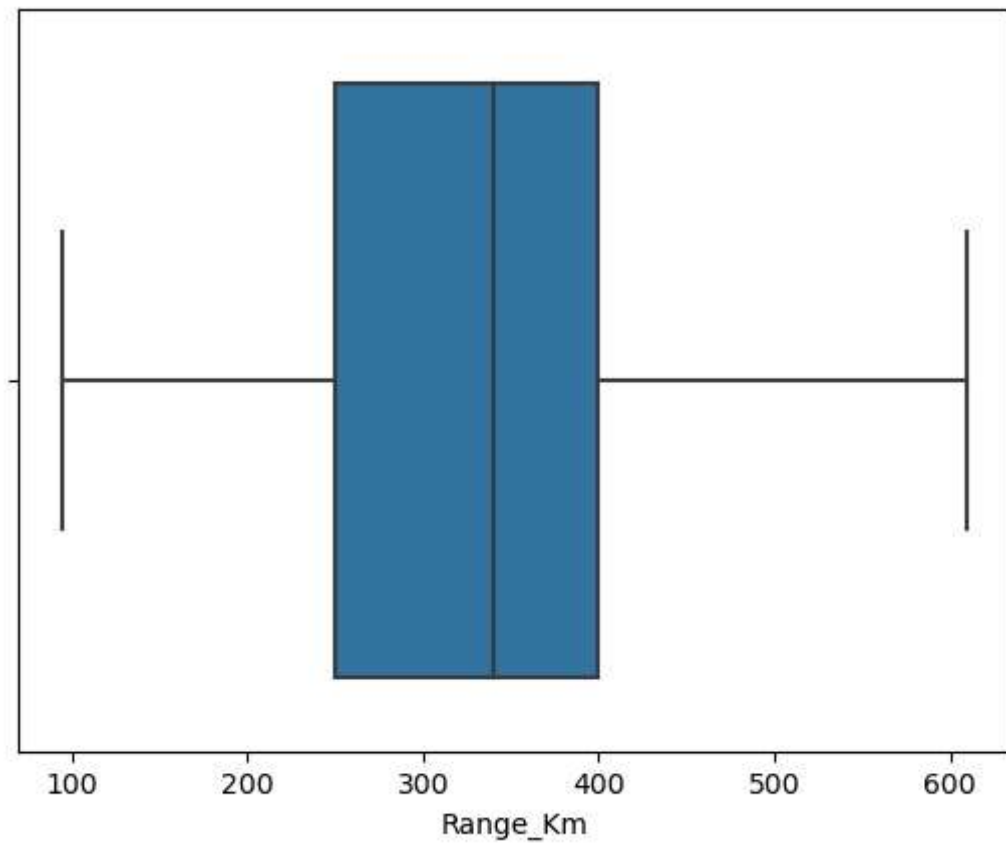
## OUTLIERS TREATMENT

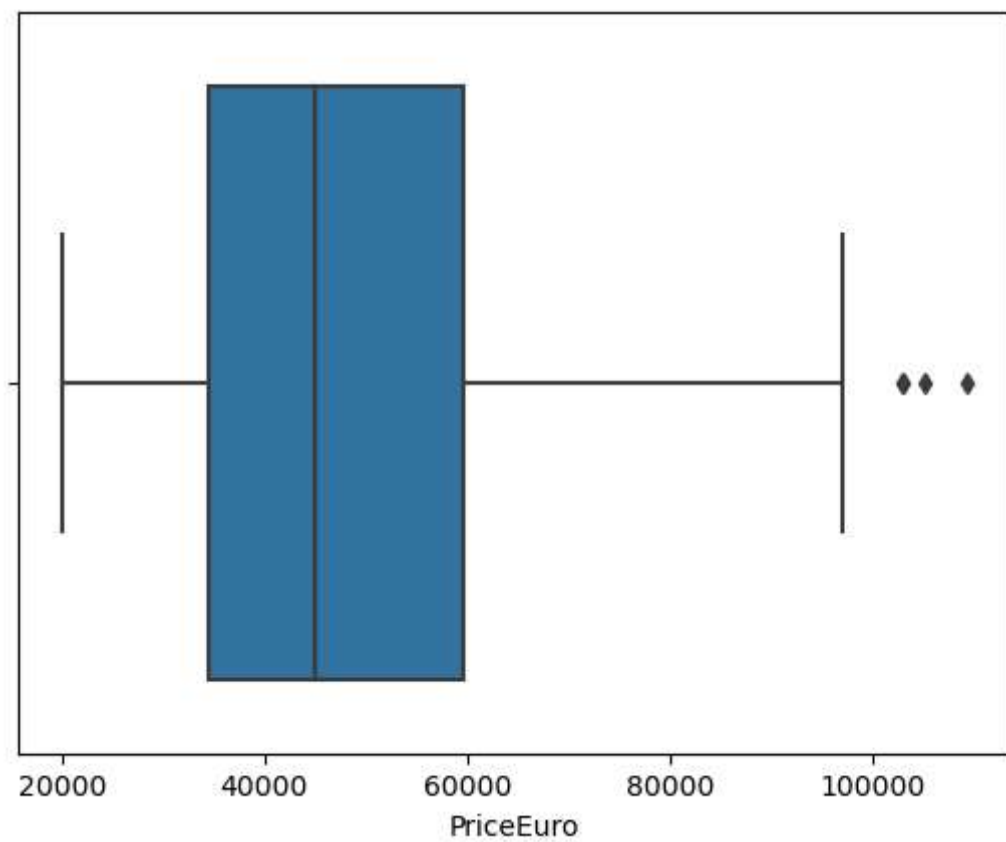
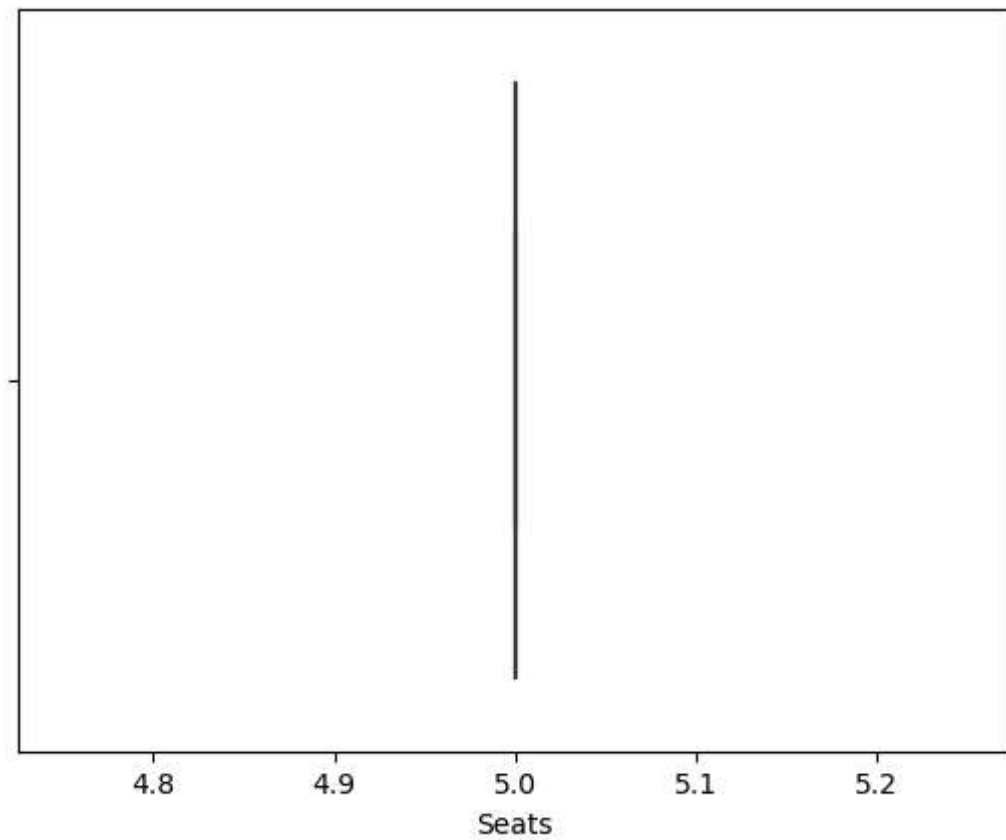
```
In [3]: def outliers_treatment(col):  
        Q1=df[col].quantile(0.25)  
        Q3=df[col].quantile(0.75)  
        IQR= Q3 - Q1  
        UB=Q3+1.5*IQR  
        LB=Q1-1.5*IQR  
        Upper_Outlier=df[col]>UB  
        Lower_Outlier=df[col]<LB  
        df.loc[Upper_Outlier,col]=df[col].median()  
        df.loc[Lower_Outlier,col]=df[col].median()
```

```
In [4]: for i in df.select_dtypes(['int','float']):  
        outliers_treatment(i)
```

```
In [5]: for i in df.select_dtypes(['int', 'float']):  
        sns.boxplot(data=df, x=i)  
        plt.show()
```



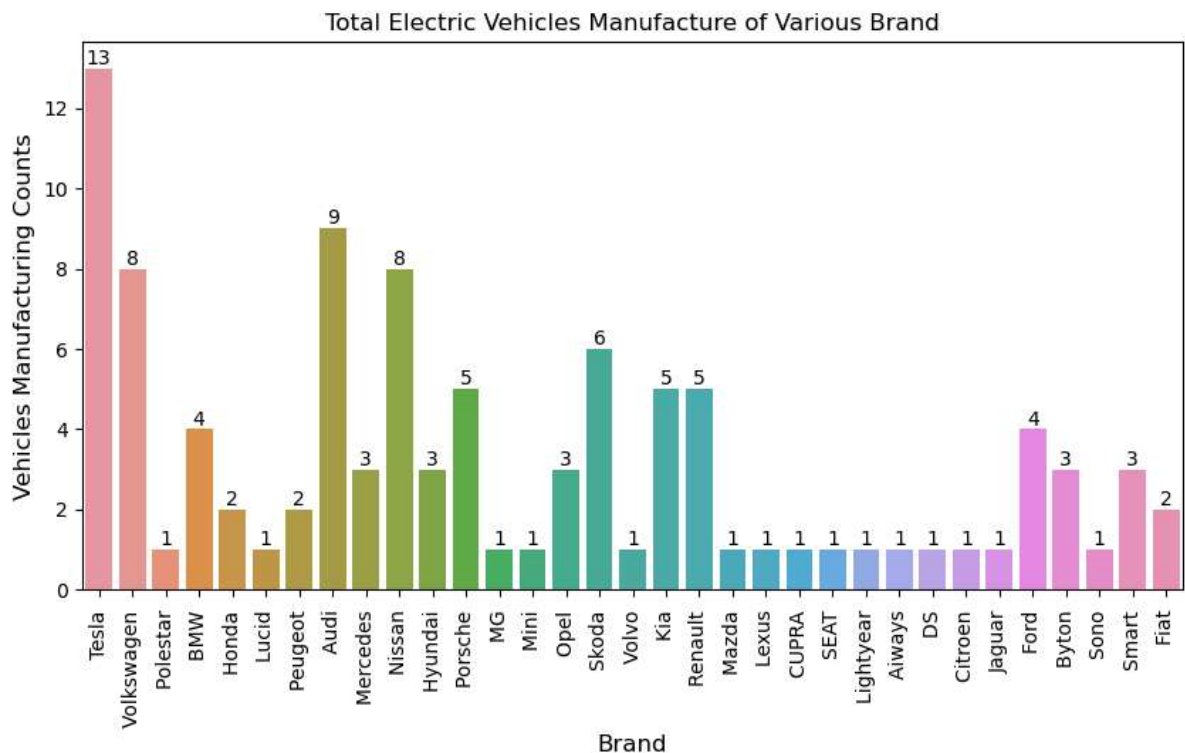




## Visualisation

### The most number of manufacturing vehicles

```
In [7]: plt.figure(figsize=(10,5))
S=sns.countplot(data=df,x="Brand")
for v in S.containers:
    S.bar_label(v)
plt.title("Total Electric Vehicles Manufacture of Various Brand", size=12)
plt.xlabel("Brand", size=12)
plt.ylabel("Vehicles Manufacturing Counts", size=12)
plt.xticks(rotation=90)
plt.show()
```



### Observation:

The chart shows the number of cars sold of different brands according to the dataset. It seems to be a bar chart with brand names on the x-axis and the number of cars model manufacturing on the y-axis.

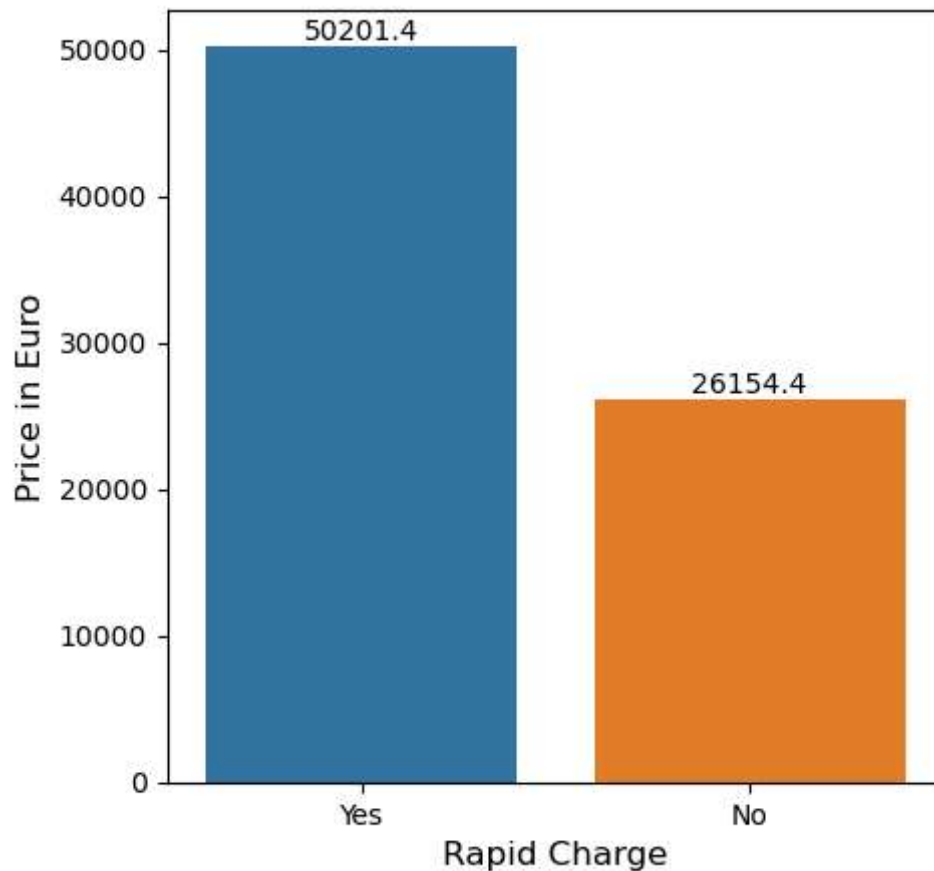
- Tesla has 13 models which is the highest quantity of vehicles.
- Audi is in the second position, with around 9 cars model manufacture.
- Volkswagem, Nissan and Skoda follow closely behind with around 8, 8 and 6 cars manufacture respectively.
- The number of cars model manufacture by other brands including Lucid, Peugeot, Ford, etc. are all fewer than 5.

## Rapid charge

```
In [115]: df["RapidCharge"].value_counts()
```

```
Out[115]: RapidCharge
Yes      98
No       5
Name: count, dtype: int64
```

```
In [8]: plt.figure(figsize=(5,5))
p=sns.barplot(data=df, x= "RapidCharge", y="PriceEuro", ci=False)
for v in p.containers:
    p.bar_label(v)
plt.xlabel("Rapid Charge", size=12)
plt.ylabel("Price in Euro", size=12)
plt.show()
```



### Observation:

- graph shows the average price in Euros in Europe according to whether it has rapid charge or not. The y-axis shows the price in Euros and the x-axis shows rapid charge capability. There are two data points represented by bars. The blue bar on the left is labeled "No" for rapid charge and shows an average price of 26,154.4 Euros. The orange bar on the right is labeled "Yes" for rapid charge and shows an average price of 50,201.4 Euros.



## Model Vehicles Acceleration

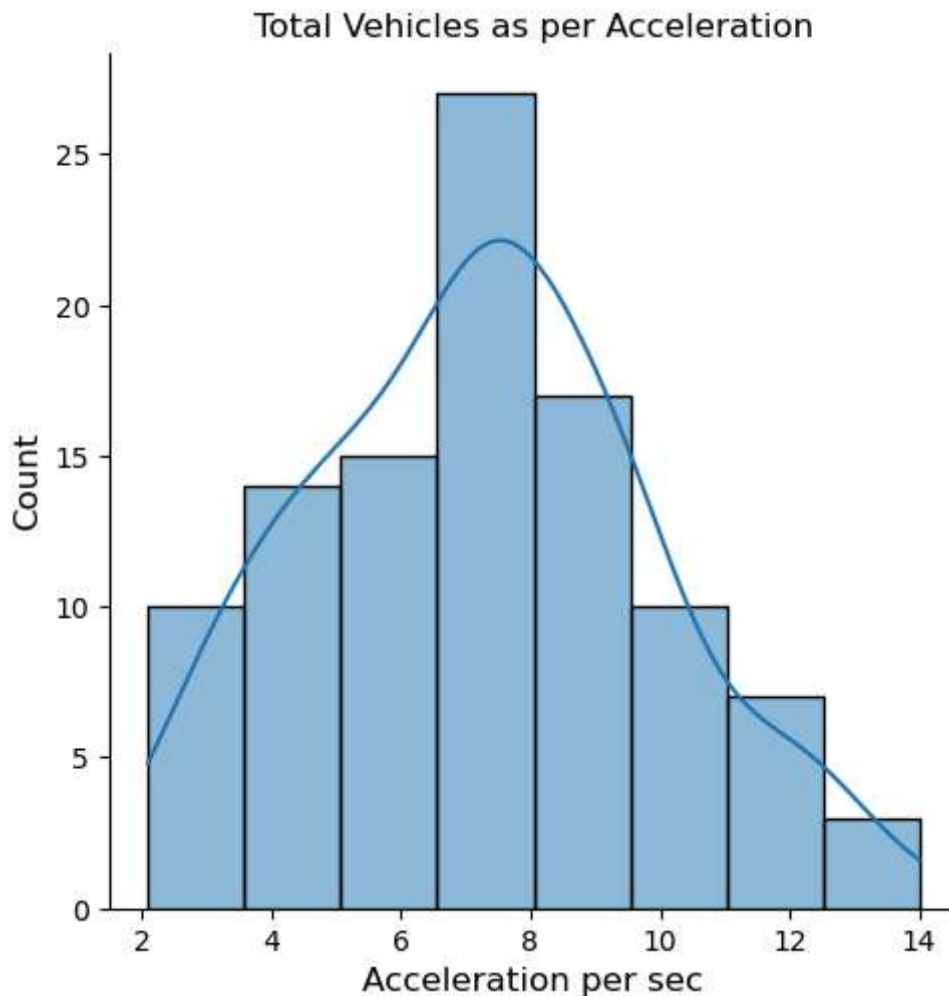
```
In [10]: A=df.groupby(["Brand", "Model"], as_index=False)["AccelSec"].max().sort_values  
A
```

```
Out[10]:
```

	Brand	Model	AccelSec
52	Nissan	e-NV200 Evalia	14.0
76	Smart	EQ forfour	12.7
65	Renault	Twingo ZE	12.6
70	Skoda	CITIGOe iV	12.3
69	SEAT	Mii Electric	12.3

```
In [9]: plt.figure(figsize=(10,5))  
sns.displot(data=df, x="AccelSec", kde=True)  
plt.title("Total Vehicles as per Acceleration", size=12)  
plt.xlabel("Acceleration per sec", size=12)  
plt.ylabel("Count", size=12)  
plt.show()
```

<Figure size 1000x500 with 0 Axes>



**OBJECTIVES:**

- Graph shows, The maximum range of cars as per acceleration is 6 to 8 per second.
- The count of manufacturing cars as per Acceleration per second increasing range from 2 to 8, after that decline the manufacturing cars range as highest acceleration per second.

**Model vehicles efficiency**

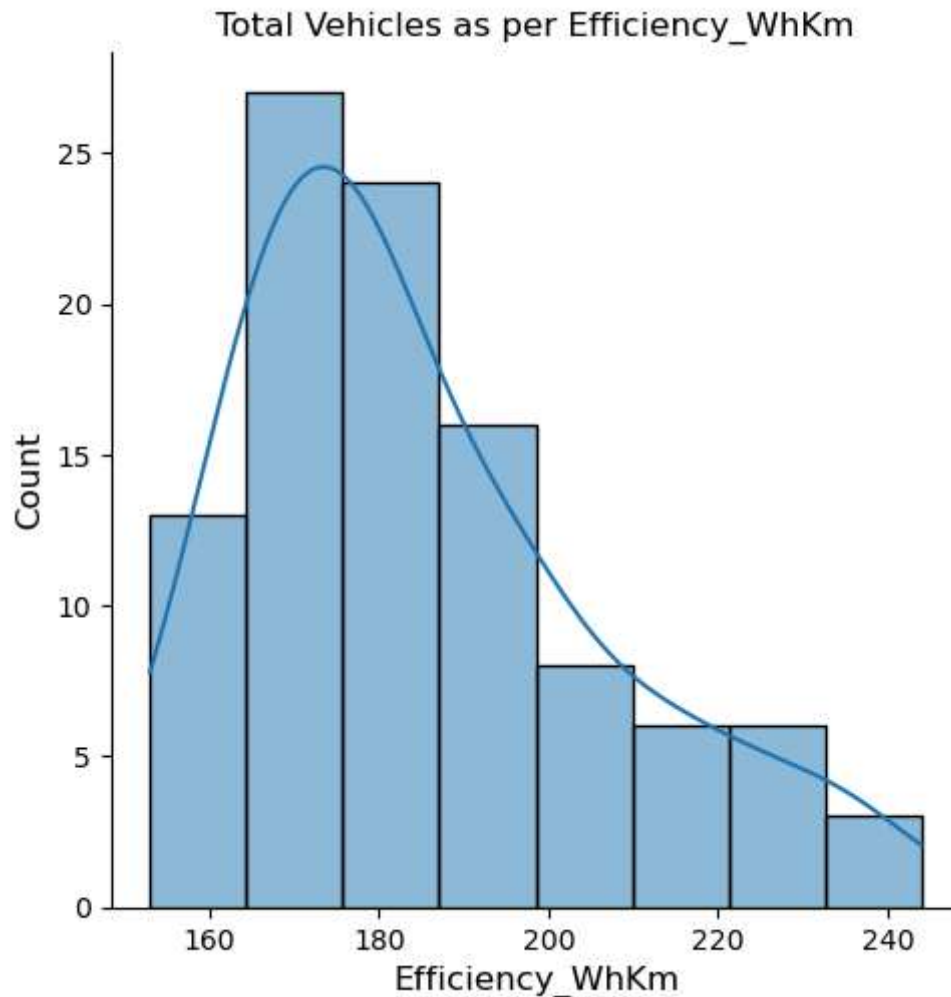
```
In [88]: B=df.groupby(["Brand", "Model"], as_index=False)["Efficiency_WhKm"].max().sort_index()
```

Out[88]:

	Brand	Model	Efficiency_WhKm
16	Byton	M-Byte 95 kWh 4WD	244
15	Byton	M-Byte 95 kWh 2WD	238
4	Audi	e-tron 55 quattro	237
31	Jaguar	I-Pace	232
49	Nissan	Ariya e-4ORCE 87kWh Performance	232

```
In [12]: plt.figure(figsize=(10,5))
sns.displot(data=df, x="Efficiency_WhKm",kde=True)
plt.title("Total Vehicles as per Efficiency_WhKm", size=12)
plt.xlabel("Efficiency_WhKm", size=12)
plt.ylabel("Count", size=12)
plt.show()
```

<Figure size 1000x500 with 0 Axes>



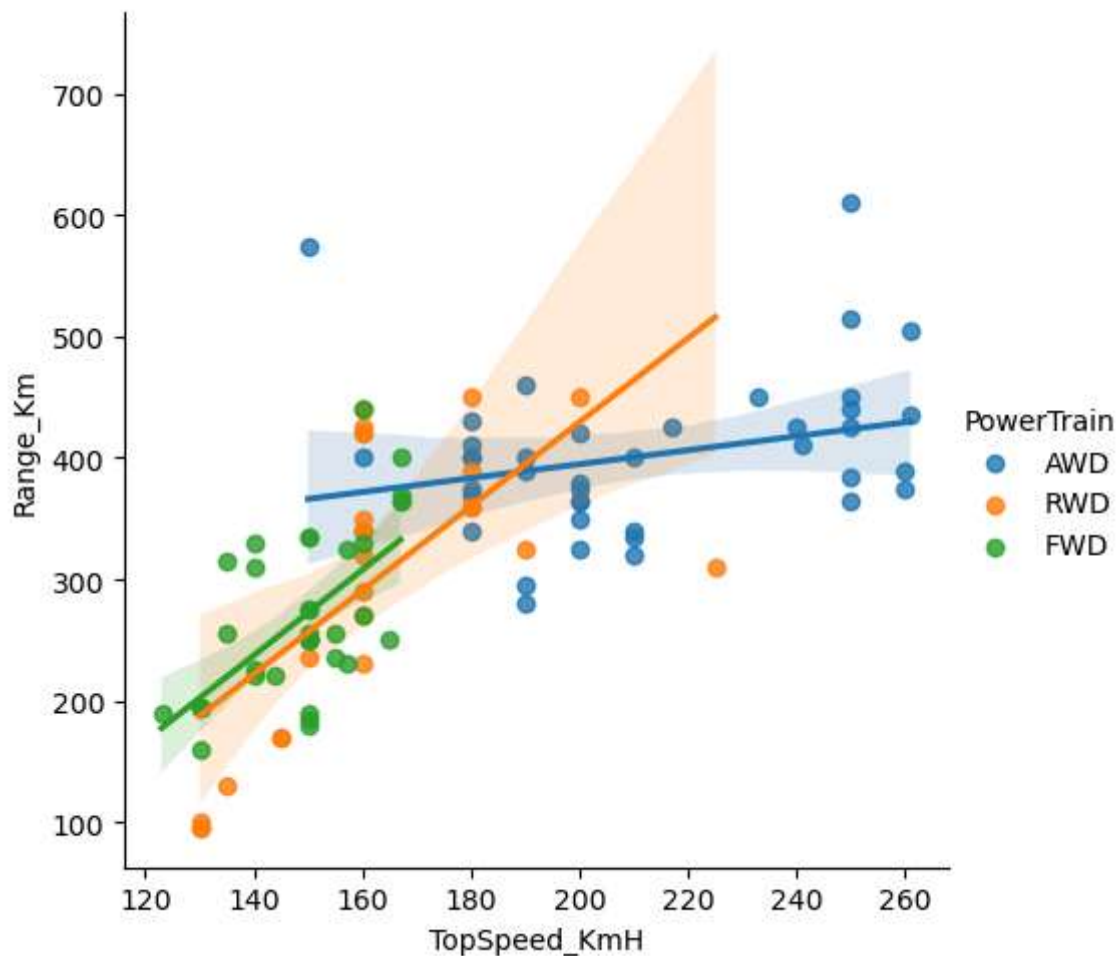
### OBJECTIVES:

- Graph shows, car has the highest efficiency range from 150whkm to 175 whkm. afterthat, decline the graph of counts of cars as per highest efficiency range 180 onwards.

## Relationship Range, Power Train and Top Speed

```
In [97]: plt.figure(figsize=(5,5))
sns.lmplot(data=df,x="TopSpeed_KmH", y="Range_Km",hue="PowerTrain")
plt.title("Total Vehicles as per Efficiency_WhKm", size=12)
plt.xlabel("TopSpeed_KmH", size=12)
plt.ylabel("Range_Km", size=12)
plt.show()
```

<Figure size 500x500 with 0 Axes>

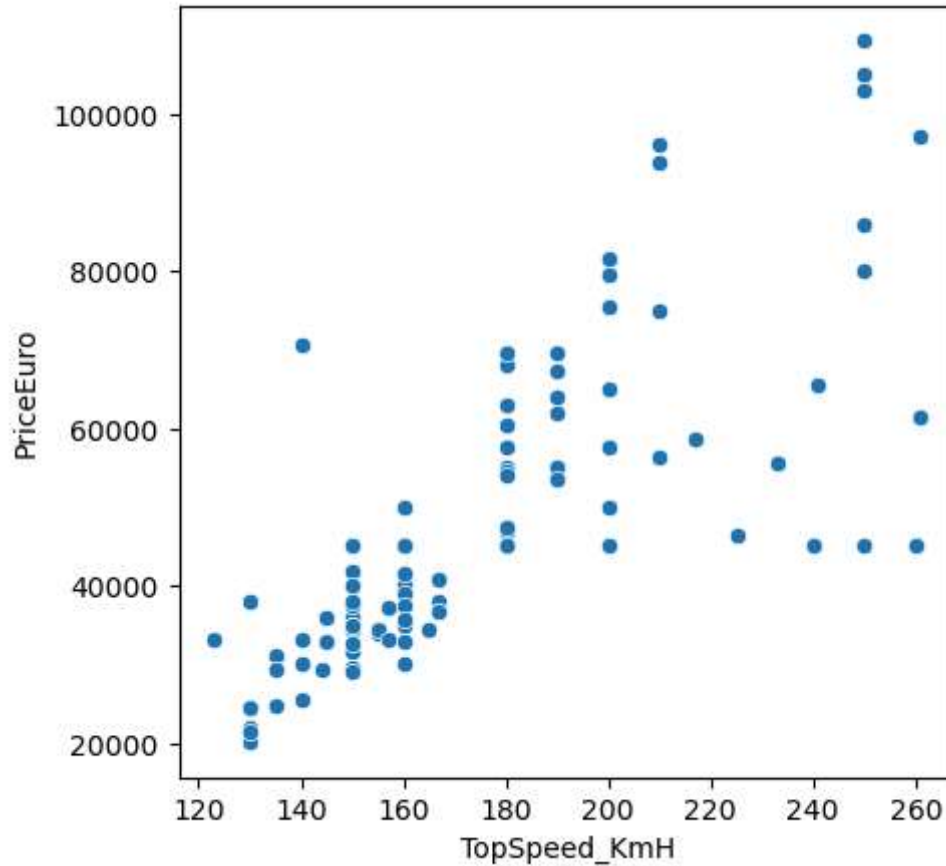


### Observation:

- it appears to show a negative correlation between the top speed (TopSpeed\_KmH) and the range (Range\_Km) of a powertrain. This means that as the top speed of a powertrain increases, the range tends to decrease.

## Car Price Depend on Top Speed

```
In [120]: plt.figure(figsize=(5,5))  
sns.scatterplot(data=df,x="TopSpeed_KmH", y="PriceEuro")  
plt.show()
```



### Observation:

- As per the graph shows, The top speed range from 120kmhr to 170kmhr maximum cars is available.
- As per top speed increase there car price in Euro also increase, but car model is specific.

### Conclusion:

In conclusion, factors such as performance, range, charging infrastructure, and price play significant roles in influencing electric car selection for potential buyers. The decision ultimately depends on individual preferences, priorities, and budget constraints.

```
In [ ]:
```

