

```
In [1]: import pandas as pd
import numpy as np
from matplotlib import pyplot as plt
ev = pd.read_csv("D:\ElectricCarData.csv")
ev.head(8)
```

Out [1]:

	Brand	Model	Accel	TopSpeed	Range	Efficiency	FastCharge	PowerTrain	PlugType	BodyStyle	Seats	PriceEuro
0	Tesla	Model 3 Long Range Dual Motor	4.6	233	450	161	940	All Wheel Drive	Type 2 CCS	Sedan	5	55480
1	Volkswagen	ID.3 Pure	10.0	160	270	167	250	Rear Wheel Drive	Type 2 CCS	Hatchback	5	30000
2	Polestar	2	4.7	210	400	181	620	All Wheel Drive	Type 2 CCS	Liftback	5	56440
3	BMW	iX3	6.8	180	360	206	560	Rear Wheel Drive	Type 2 CCS	SUV	5	68040
4	Honda	e	9.5	145	170	168	190	Rear Wheel Drive	Type 2 CCS	Hatchback	4	32997
5	Lucid	Air	2.8	250	610	180	620	All Wheel Drive	Type 2 CCS	Sedan	5	105000
6	Volkswagen	e-Golf	9.6	150	190	168	220	Front Wheel Drive	Type 2 CCS	Hatchback	5	31900
7	Peugeot	e-208	8.1	150	275	164	420	Front Wheel Drive	Type 2 CCS	Hatchback	5	29682

```
In [2]: ev = pd.read_csv("D:\ElectricCarData.csv")
ev_tesla= ev[ev['Brand']=='Tesla ']
print(ev_tesla.reset_index())
```

index	Brand	Model	Accel	TopSpeed	Range	Efficiency	FastCharge	PowerTrain	PlugType	BodyStyle	Seats	PriceEuro
0	0 Tesla	Model 3 Long Range Dual Motor	4.6	233	450	161	940	All Wheel Drive	Type 2 CCS	Sedan	5	55480
1	8 Tesla	Model 3 Standard Range Plus	5.6	225	310	153	650	Rear Wheel Drive	Type 2 CCS	Sedan	5	46380
2	21 Tesla	Model Y Long Range Dual Motor	5.1	217	425	171	930	All Wheel Drive	Type 2 CCS	SUV	7	58620
3	24 Tesla	Model 3 Long Range Performance	3.4	261	435	167	910	All Wheel Drive	Type 2 CCS	Sedan	5	61480
4	33 Tesla	Cybertruck Tri Motor	3.0	210	750	267	710	All Wheel Drive	Type 2 CCS	Pickup	6	75000
5	40 Tesla	Model S Long Range	3.8	250	515	184	560	All Wheel Drive	Type 2 CCS	Liftback	5	79990
6	51 Tesla	Roadster	2.1	410	970	206	920	All Wheel Drive	Type 2 CCS	Cabrio	4	215000
7	54 Tesla	Model X Long Range	4.6	250	450	211	490	All Wheel Drive	Type 2 CCS	SUV	7	85990
8	59 Tesla	Model S Performance	2.5	261	505	188	550	All Wheel Drive	Type 2 CCS	Liftback	5	96990
9	61 Tesla	Model Y Long Range Performance	3.7	241	410	177	900	All Wheel Drive	Type 2 CCS	SUV	7	65620
10	67 Tesla	Cybertruck Dual Motor	5.0	190	460	261	710	All Wheel Drive	Type 2 CCS	Pickup	6	55000
11	81 Tesla	Model X Performance	2.8	250	440	216	480	All Wheel Drive	Type 2 CCS	SUV	7	102990
12	86 Tesla	Cybertruck Single Motor	7.0	180	390	256	740	Rear Wheel Drive	Type 2 CCS	Pickup	6	45000

```
In [3]: print(ev.info())
print(" -----")
print(ev.isna().sum())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 103 entries, 0 to 102
Data columns (total 12 columns):
#   Column      Non-Null Count  Dtype
---  ---
0   Brand       103 non-null   object
1   Model       103 non-null   object
2   Accel       103 non-null   float64
```

```

3 TopSpeed    103 non-null    int64
4 Range       103 non-null    int64
5 Efficiency   103 non-null    int64
6 FastCharge   103 non-null    int64
7 PowerTrain   103 non-null    object
8 PlugType     103 non-null    object
9 BodyStyle    103 non-null    object
10 Seats       103 non-null    int64
11 PriceEuro   103 non-null    int64
dtypes: float64(1), int64(6), object(5)
memory usage: 9.8+ KB
None
-----
Brand        0
Model        0
Accel        0
TopSpeed     0
Range        0
Efficiency   0
FastCharge   0
PowerTrain   0
PlugType     0
BodyStyle    0
Seats        0
PriceEuro    0
dtype: int64

```

```

In [4]: print(ev.describe())
print("_____")

```

```

count      Accel    TopSpeed    Range  Efficiency  FastCharge    Seats \
mean      7.396117  179.194175  338.786408  189.165049  455.339806    4.883495
std       3.017430   43.573030  126.014444   29.566839  196.861729    0.795834
min       2.100000  123.000000   95.000000  104.000000  170.000000    2.000000
25%       5.100000  150.000000  250.000000  168.000000  305.000000    5.000000
50%       7.300000  160.000000  340.000000  180.000000  440.000000    5.000000
75%       9.000000  200.000000  400.000000  203.000000  555.000000    5.000000
max      22.400000  410.000000  970.000000  273.000000  940.000000    7.000000

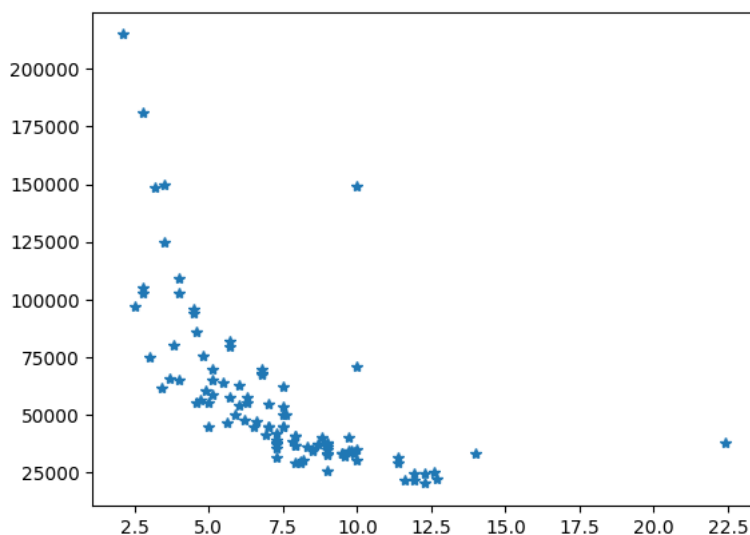
      PriceEuro
count      103.000000
mean     55811.563107
std      34134.665280
min      20129.000000
25%      34429.500000
50%      45000.000000
75%      65000.000000
max      215000.000000

```

```

In [5]: plot_Accel= ev["PriceEuro"]
plot_price= ev["Accel"]
plt.plot(plot_price,plot_Accel,linestyle=" ",marker="*")
plt.show()

```



```

In [6]: ev["Top"] = ev["FastCharge"]/ev["TopSpeed"] # البطارية يتخلص في مدة ( ساعات )

```

```

In [7]: ev.sort_values(["Top"],ascending=False)

```

```

Out [7]:
   Brand  Model  Accel  TopSpeed  Range  Efficiency  FastCharge  PowerTrain  PlugType  BodyStyle  Seats  PriceEuro  4.2
21  Tesla  Model Y Long Range Dual Motor    5.1    217    425    171    930    All Wheel Drive  Type 2 CCS    SUV    7    58620    4.2

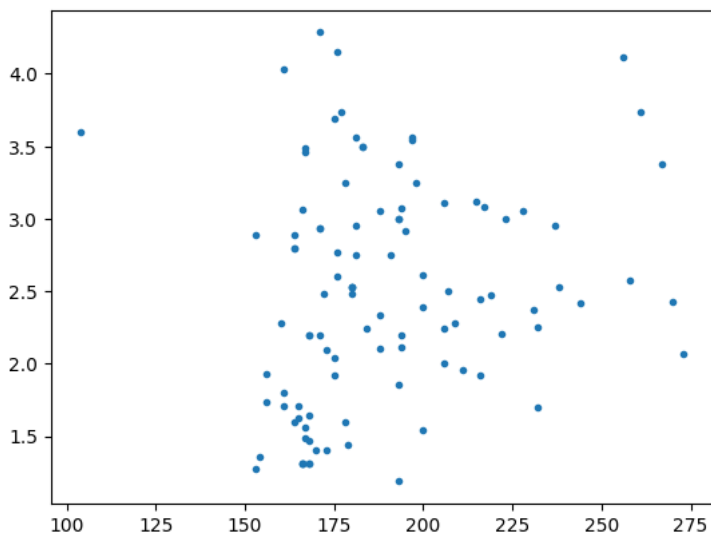
```

	Brand	Model	Accel	TopSpeed	Range	Efficiency	FastCharge	PowerTrain	PlugType	BodyStyle	Seats	PriceEuro	
91	Smart	EQ fortwo cabrio	11.9	130	95	176	540	Rear Wheel Drive	Type 2	Cabrio	2	24565	4.1
86	Tesla	Cybertruck Single Motor	7.0	180	390	256	740	Rear Wheel Drive	Type 2 CCS	Pickup	6	45000	4.1
0	Tesla	Model 3 Long Range Dual Motor	4.6	233	450	161	940	All Wheel Drive	Type 2 CCS	Sedan	5	55480	4.0
67	Tesla	Cybertruck Dual Motor	5.0	190	460	261	710	All Wheel Drive	Type 2 CCS	Pickup	6	55000	3.7
...
17	Volkswagen	e-Up!	11.9	130	195	166	170	Front Wheel Drive	Type 2 CCS	Hatchback	4	21421	1.3
43	Skoda	CITIGOe iV	12.3	130	195	166	170	Front Wheel Drive	Type 2 CCS	Hatchback	4	24534	1.3
44	SEAT	Mii Electric	12.3	130	195	166	170	Front Wheel Drive	Type 2 CCS	Hatchback	4	20129	1.3
14	Hyundai	IONIQ Electric	9.7	165	250	153	210	Front Wheel Drive	Type 2 CCS	Liftback	5	34459	1.2
36	Lexus	UX 300e	7.5	160	270	193	190	Front Wheel Drive	Type 2 CHAdeMO	SUV	5	50000	1.1

103 rows x 13 columns

```
In [8]: plot_eff = ev["Efficiency"]

plot_Top = ev["Top"]
plt.plot(plot_eff,plot_Top,linestyle=" ",marker=".")
plt.show()
```



```
In [10]: ev["BodyStyle"].unique()
```

```
Out [10]: array(['Sedan', 'Hatchback', 'Liftback', 'SUV', 'Pickup', 'MPV', 'Cabrio',
'SPV', 'Station'], dtype=object)
```

```
In [60]: #####
import csv
from collections import Counter
style_counter=Counter()
model_counter=Counter()
wheel_counter=Counter()

with open("D:\ElectricCarData.csv") as csv_file:
    csv_reader=csv.DictReader(csv_file)
```

```

for row in csv_reader:
    style_counter.update(row["BodyStyle"].split(" "))
    model_counter.update(row["Brand"].split(" "))
    wheel_counter.update(row["PowerTrain"].split("Drive"))

style_counter
model_counter
wheel_counter

```

Out [60]: Counter({'All Wheel ': 41, '': 103, 'Rear Wheel ': 25, 'Front Wheel ': 37})

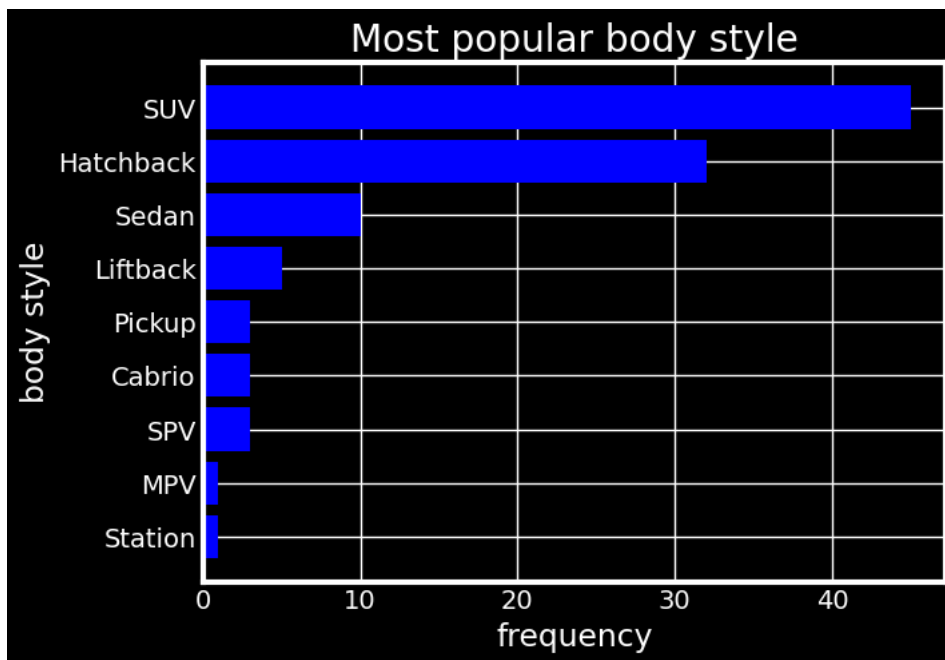
```
In [26]: style_counter.most_common(5)
```

Out [26]: [('SUV', 45), ('Hatchback', 32), ('Sedan', 10), ('Liftback', 5), ('Pickup', 3)]

```
In [27]: stylee=[]
frequency=[]
for item in style_counter.most_common(15):
    stylee.append(item[0])
    frequency.append(item[1])
print(stylee)
print(frequency)
```

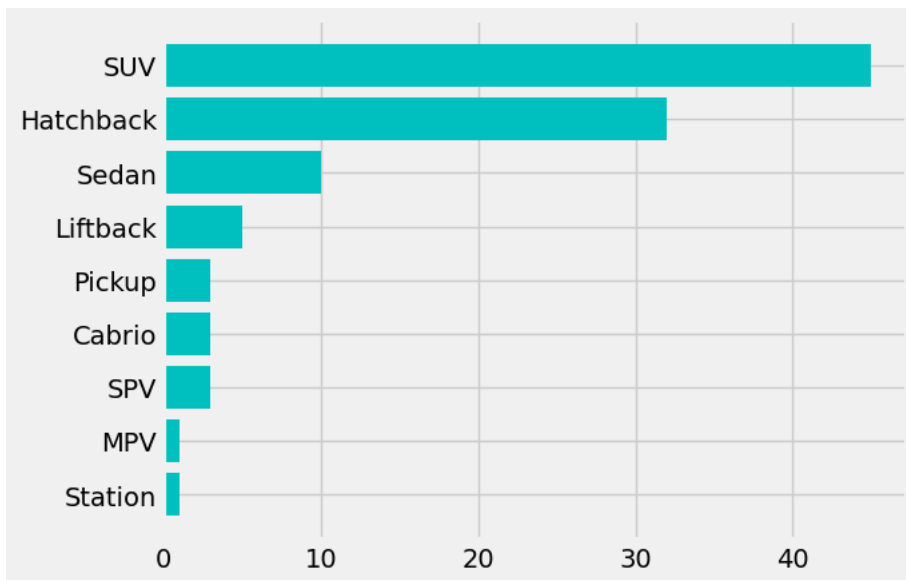
```
['SUV', 'Hatchback', 'Sedan', 'Liftback', 'Pickup', 'Cabrio', 'SPV', 'MPV', 'Station']
[45, 32, 10, 5, 3, 3, 3, 1, 1]
```

```
In [34]: plt.style.use("dark_background")
plt.barh(stylee, frequency, color="b")
plt.title("Most popular body style")
plt.ylabel("body style")
plt.xlabel("frequency")
plt.show()
```



```
In [30]: stylee.reverse()
frequency.reverse()
plt.barh(stylee, frequency, color="c")
```

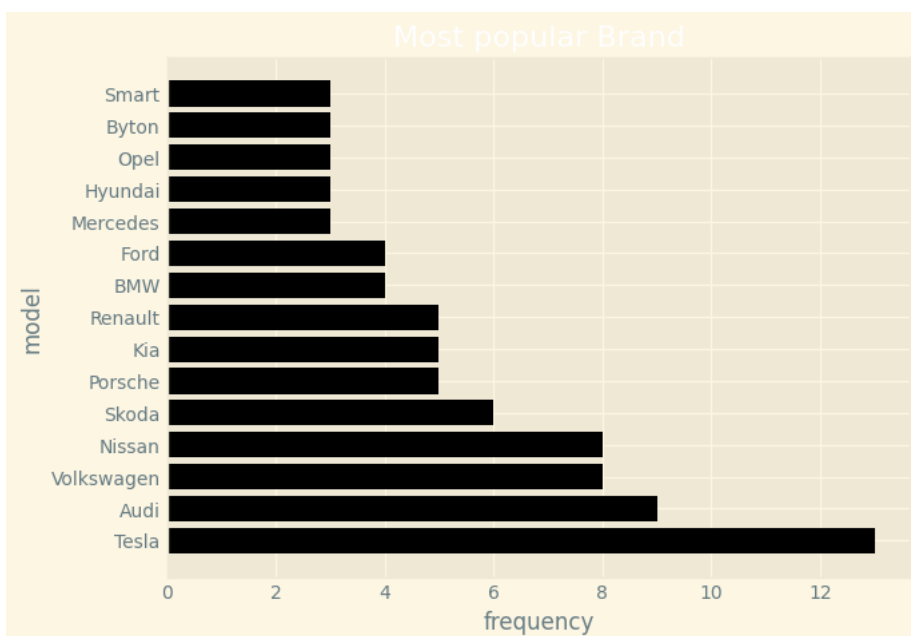
Out [30]: <BarContainer object of 9 artists>



```
In [44]: model_counter[""]=0
model_counter
model_counter.most_common(10)
```

```
Out [44]: [('Tesla', 13),
('Audi', 9),
('Volkswagen', 8),
('Nissan', 8),
('Skoda', 6),
('Porsche', 5),
('Kia', 5),
('Renault', 5),
('BMW', 4),
('Ford', 4)]
```

```
In [58]: plt.style.use('Solarize_Light2')
plt.barh(model,frequency,color="k")
plt.title("Most popular Brand")
plt.ylabel("model")
plt.xlabel("frequency")
plt.show()
```



```
In [66]: wheel=[]
frequency=[]
for item in wheel_counter.most_common(15):
    wheel.append(item[0])
    frequency.append(item[1])
del frequency[0]
del wheel[0]
print(wheel)
print(frequency)
```

```
['All Wheel ', 'Front Wheel ', 'Rear Wheel ']  
[41, 37, 25]
```

```
In [67]: plt.plot(wheel,frequency, color="k", linestyle="--", marker=" ", linewidth=2, label="Python Developers")  
  
plt.title('Median Salary (USD) by Age')  
plt.xlabel('Ages')  
plt.ylabel('Median Salary (USD)')  
plt.legend()  
plt.grid(True)  
plt.show()
```

