In [24]:
```
# Import Data
import pandas as pd
pd.set_option('display.max_rows', None)
pd.set_option('display.max_columns', None)

df = pd.read_excel("FEV-data-Excel.xlsx")
df
```

Out[24]:

	Car full name	Make	Model	Minimal price (gross) [PLN]	Engine power [KM]	Maximum torque [Nm]	Type of brakes	Driv typ
0	Audi e-tron 55 quattro	Audi	e-tron 55 quattro	345700	360	664	disc (front + rear)	4WI
1	Audi e-tron 50 quattro	Audi	e-tron 50 quattro	308400	313	540	disc (front + rear)	4WI
2	Audi e-tron S quattro	Audi	e-tron S quattro	414900	503	973	disc (front + rear)	4WI
3	Audi e-tron Sportback 50 quattro	Audi	e-tron Sportback 50 quattro	319700	313	540	disc (front + rear)	4WI
4	Audi e-tron Sportback 55 quattro	Audi	e-tron Sportback 55 quattro	357000	360	664	disc (front + rear)	4WI
5	Audi e-tron Sportback S quattro	Audi	e-tron Sportback S quattro	426200	503	973	disc (front + rear)	4WI
6	BMW i3	BMW	i3	169700	170	250	disc (front + rear)	2WI (rea
7	BMW i3s	BMW	i3s	184200	184	270	disc (front + rear)	2WI (rea
8	BMW iX3	BMW	iX3	282900	286	400	disc (front + rear)	2WI (rea
9	Citroën ë-C4	Citroën	ë-C4	125000	136	260	disc (front + rear)	2WI (fron
10	DS DS3 Crossback e-tense	DS	DS3 Crossback e-tense	159900	136	260	disc (front + rear)	2WI (fron
11	Honda e	Honda	e	152900	136	315	disc (front + rear)	2WI (rea
12	Honda e Advance	Honda	e Advance	165900	154	315	disc (front + rear)	2WI (rea
13	Hyundai Ioniq electric	Hyundai	Ioniq electric	184500	136	295	disc (front + rear)	2WI (fron

	Car full name	Make	Model	Minimal price (gross) [PLN]	Engine power [KM]	Maximum torque [Nm]	Type of brakes	Driv typ
14	Hyundai Kona electric 39.2kWh	Hyundai	Kona electric 39.2kWh	154400	136	395	disc (front + rear)	2WI (front
15	Hyundai Kona electric 64kWh	Hyundai	Kona electric 64kWh	178400	204	395	disc (front + rear)	2WI (front
16	Jaguar I-Pace	Jaguar	I-Pace	359500	400	696	disc (front + rear)	4WI
17	Kia e-Niro 39.2kWh	Kia	e-Niro 39.2kWh	146990	136	395	disc (front + rear)	2WI (front
18	Kia e-Niro 64kWh	Kia	e-Niro 64kWh	167990	204	395	disc (front + rear)	2WI (front
19	Kia e-Soul 39.2kWh	Kia	e-Soul 39.2kWh	139900	136	395	disc (front + rear)	2WI (front
20	Kia e-Soul 64kWh	Kia	e-Soul 64kWh	160990	204	395	disc (front + rear)	2WI (front
21	Mazda MX-30	Mazda	MX-30	142900	145	270	disc (front + rear)	2WI (front
22	Mercedes-Benz EQC	Mercedes-Benz	EQC	334700	408	760	disc (front + rear)	4WI
23	Mini Cooper SE	Mini	Cooper SE	139900	184	270	disc (front + rear)	2WI (front
24	Nissan Leaf	Nissan	Leaf	122900	150	320	disc (front + rear)	2WI (front
25	Nissan Leaf e+	Nissan	Leaf e+	164000	217	340	disc (front + rear)	2WI (front
26	Opel Corsa-e	Opel	Corsa-e	128900	136	260	disc (front + rear)	2WI (front
27	Opel Mokka-e	Opel	Mokka-e	139900	136	260	disc (front + rear)	2WI (front

	Car full name	Make	Model	Minimal price (gross) [PLN]	Engine power [KM]	Maximum torque [Nm]	Type of brakes	Driv typ
28	Peugeot e-208	Peugeot	e-208	124900	136	260	disc (front + rear)	2WI (front
29	Peugeot e-2008	Peugeot	e-2008	149400	136	260	disc (front + rear)	2WI (front
30	Porsche Taycan 4S (Performance)	Porsche	Taycan 4S (Performance)	457000	435	640	disc (front + rear)	4WI
31	Porsche Taycan 4S (Performance Plus)	Porsche	Taycan 4S (Performance Plus)	482283	490	650	disc (front + rear)	4WI
32	Porsche Taycan Turbo	Porsche	Taycan Turbo	653000	625	850	disc (front + rear)	4WI
33	Porsche Taycan Turbo S	Porsche	Taycan Turbo S	794000	625	1050	disc (front + rear)	4WI
34	Renault Zoe R110	Renault	Zoe R110	135900	108	225	disc (front + rear)	2WI (front
35	Renault Zoe R135	Renault	Zoe R135	142900	135	245	disc (front + rear)	2WI (front
36	Skoda Citigo-e iV	Skoda	Citigo-e iV	82050	83	212	disc (front) + drum (rear)	2WI (front
37	Smart fortwo EQ	Smart	fortwo EQ	96900	82	160	disc (front) + drum (rear)	2WI (rear
38	Smart forfour EQ	Smart	forfour EQ	98900	82	160	disc (front) + drum (rear)	2WI (rear
39	Tesla Model 3 Standard Range Plus	Tesla	Model 3 Standard Range Plus	195490	285	450	disc (front + rear)	2WI (rear

	Car full name	Make	Model	Minimal price (gross) [PLN]	Engine power [KM]	Maximum torque [Nm]	Type of brakes	Driv typ
40	Tesla Model 3 Long Range	Tesla	Model 3 Long Range	235490	372	510	disc (front + rear)	4WI
41	Tesla Model 3 Performance	Tesla	Model 3 Performance	260490	480	639	disc (front + rear)	4WI
42	Tesla Model S Long Range Plus	Tesla	Model S Long Range Plus	368990	525	755	disc (front + rear)	4WI
43	Tesla Model S Performance	Tesla	Model S Performance	443990	772	1140	disc (front + rear)	4WI
44	Tesla Model X Long Range Plus	Tesla	Model X Long Range Plus	407990	525	755	disc (front + rear)	4WI
45	Tesla Model X Performance	Tesla	Model X Performance	482990	772	1140	disc (front + rear)	4WI
46	Volkswagen e-up!	Volkswagen	e-up!	97990	83	210	disc (front) + drum (rear)	2WI (front
47	Volkswagen ID.3 Pro Performance	Volkswagen	ID.3 Pro Performance	155890	204	310	disc (front) + drum (rear)	2WI (rear
48	Volkswagen ID.3 Pro S	Volkswagen	ID.3 Pro S	179990	204	310	disc (front) + drum (rear)	2WI (rear
49	Volkswagen ID.4 1st	Volkswagen	ID.4 1st	202390	204	310	disc (front) + drum (rear)	2WI (rear
50	Citroën ë-Spacetourer (M)	Citroën	ë-Spacetourer (M)	215400	136	260	disc (front + rear)	2WI (front
51	Mercedes-Benz EQV (long)	Mercedes-Benz	EQV (long)	339480	204	362	NaN	2WI (front

	Car full name	Make	Model	Minimal price (gross) [PLN]	Engine power [KM]	Maximum torque [Nm]	Type of brakes	Driv typ
52	Nissan e-NV200 evalia	Nissan	e-NV200 evalia	164328	109	254	disc (front + rear)	2WI (front

In [26]:
```
#Task 1 a) Your task is to filter out EVs that meet these criteria.
import pandas as pd
df = pd.read_excel("FEV-data-Excel.xlsx", header=0)
print(df.head())
filtered_df = df[(df['Minimal price (gross) [PLN]'] <= 350000) & (df['Range (WLT
print(filtered_df)
```

	Car full name	Make	Model	Minimal price (gross) [PLN]	Engine power [KM]	Maximum torque [Nm]	Type of brakes	Drive type	Battery capacity [kWh]	Range (WLTP) [km]	Wheelbase [cm]	Length [cm]	Width [cm]	Height [cm]	Minimal empty weight [kg]	Permissable gross weight [kg]	Maximum load capacity [kg]	Number of seats	Number of doors	Tire size [in]	Maximum speed [kph]	Boot capacity (VDA) [l]	Acceleration 0-100 kph [s]	Maximum DC charging power [kW]	mean - Energy consumption [kWh/100 km]
0	Audi e-tron 55 quattro	Audi	e-tron 55 quattro	345700	360	664	disc (front + rear)	4WD	95.0	438	292.8	490.1	193.5	162.9	2565	3130.0	640.0	5	5	19	200	660.0	5.7	150	24.45
1	Audi e-tron 50 quattro	Audi	e-tron 50 quattro	308400	313	540	disc (front + rear)	4WD	71.0	340	292.8	490.1	193.5	162.9	2445	3040.0	670.0	5	5	19	190	660.0	6.8	150	23.80
2	Audi e-tron S quattro	Audi	e-tron S quattro	414900	503	973	disc (front + rear)	4WD	95.0	364	292.8	490.2	197.6	162.9	2695	3130.0	565.0	5	5	20	210	660.0	4.5	150	27.55
3	Audi e-tron Sportback 50 quattro	Audi	e-tron Sportback 50 quattro	319700	313	540	disc (front + rear)	4WD	71.0	346	292.8	490.1	193.5	161.6	2445	3040.0	640.0	5	5	19	190	615.0	6.8	150	
4	Audi e-tron Sportback 55 quattro	Audi	e-tron Sportback 55 quattro	357000	360	664	disc (front + rear)	4WD	95.0	447	292.8	490.1	193.5	161.6	2595	3130.0	670.0	5	5	19	200	615.0	5.7	150	

```
3                                  23.30
4                                  23.85
                       Car full name           Make  \
0              Audi e-tron 55 quattro           Audi
8                            BMW iX3            BMW
15       Hyundai Kona electric 64kWh        Hyundai
18                  Kia e-Niro 64kWh            Kia
20                  Kia e-Soul 64kWh            Kia
22                 Mercedes-Benz EQC  Mercedes-Benz
39  Tesla Model 3 Standard Range Plus          Tesla
40           Tesla Model 3 Long Range          Tesla
41          Tesla Model 3 Performance          Tesla
47    Volkswagen ID.3 Pro Performance     Volkswagen
48               Volkswagen ID.3 Pro S     Volkswagen
49                 Volkswagen ID.4 1st     Volkswagen

                         Model  Minimal price (gross) [PLN]  \
0              e-tron 55 quattro                       345700
8                            iX3                       282900
15          Kona electric 64kWh                       178400
18                 e-Niro 64kWh                       167990
20                 e-Soul 64kWh                       160990
22                          EQC                       334700
39  Model 3 Standard Range Plus                       195490
40           Model 3 Long Range                       235490
41          Model 3 Performance                       260490
47         ID.3 Pro Performance                       155890
48                   ID.3 Pro S                       179990
49                     ID.4 1st                       202390

    Engine power [KM]  Maximum torque [Nm]              Type of brakes  \
0                 360                  664         disc (front + rear)
8                 286                  400         disc (front + rear)
15                204                  395         disc (front + rear)
18                204                  395         disc (front + rear)
20                204                  395         disc (front + rear)
22                408                  760         disc (front + rear)
39                285                  450         disc (front + rear)
40                372                  510         disc (front + rear)
41                480                  639         disc (front + rear)
47                204                  310  disc (front) + drum (rear)
48                204                  310  disc (front) + drum (rear)
49                204                  310  disc (front) + drum (rear)

     Drive type  Battery capacity [kWh]  Range (WLTP) [km]  Wheelbase [cm]  \
0           4WD                    95.0                438           292.8
8    2WD (rear)                    80.0                460           286.4
15  2WD (front)                    64.0                449           260.0
18  2WD (front)                    64.0                455           270.0
20  2WD (front)                    64.0                452           260.0
22          4WD                    80.0                414           287.3
39   2WD (rear)                    54.0                430           287.5
40          4WD                    75.0                580           287.5
41          4WD                    75.0                567           287.5
47   2WD (rear)                    58.0                425           277.0
48   2WD (rear)                    77.0                549           277.0
49   2WD (rear)                    77.0                500           277.1

    Length [cm]  Width [cm]  Height [cm]  Minimal empty weight [kg]  \
0         490.1       193.5        162.9                       2565
```

```
8         473.4       189.1        166.8                       2260
15        418.0       180.0        157.0                       1685
18        437.5       180.5        156.0                       1737
20        419.5       180.0        160.5                       1535
22        476.2       188.4        162.4                       2495
39        469.0       193.0        144.0                       1626
40        469.0       193.0        144.0                       1862
41        469.0       193.0        144.0                       1862
47        426.1       180.9        156.8                       1805
48        426.1       180.9        156.8                       1934
49        458.4       185.2        163.1                       2124

    Permissable gross weight [kg]  Maximum load capacity [kg]  \
0                          3130.0                       640.0
8                          2725.0                       540.0
15                         2170.0                       485.0
18                         2230.0                       493.0
20                         1682.0                       498.0
22                         2940.0                       445.0
39                            NaN                         NaN
40                            NaN                         NaN
41                            NaN                         NaN
47                         2270.0                       540.0
48                         2280.0                       412.0
49                         2660.0                       661.0

    Number of seats  Number of doors  Tire size [in]  Maximum speed [kph]  \
0                 5                5              19                  200
8                 5                5              19                  180
15                5                5              17                  167
18                5                5              17                  167
20                5                5              17                  167
22                5                5              19                  180
39                5                5              18                  225
40                5                5              18                  233
41                5                5              20                  261
47                5                5              18                  160
48                5                5              19                  160
49                5                5              20                  160

    Boot capacity (VDA) [l]  Acceleration 0-100 kph [s]  \
0                     660.0                         5.7
8                     510.0                         6.8
15                    332.0                         7.6
18                    451.0                         7.8
20                    315.0                         7.9
22                    500.0                         5.1
39                    425.0                         5.6
40                    425.0                         4.4
41                    425.0                         3.3
47                    385.0                         7.3
48                    385.0                         7.9
49                    543.0                         8.5

    Maximum DC charging power [kW]  mean - Energy consumption [kWh/100 km]
0                              150                                   24.45
8                              150                                   18.80
15                             100                                   15.40
18                             100                                   15.90
20                             100                                   15.70
```

```
22                                110                                21.85
39                                150                                  NaN
40                                150                                  NaN
41                                150                                  NaN
47                                100                                15.40
48                                125                                15.90
49                                125                                18.00
```

In [27]:
```
# b) Group them by the manufacturer
grouped_by_make = filtered_df.groupby('Make').size()
print(grouped_by_make)
```

```
Make
Audi             1
BMW              1
Hyundai          1
Kia              2
Mercedes-Benz    1
Tesla            3
Volkswagen       3
dtype: int64
```

In [28]:
```
# c) Calculate the average battery capacity for each manufacturer.
average_battery_capacity = filtered_df.groupby('Make')['Battery capacity [kWh]']
print(average_battery_capacity)
```

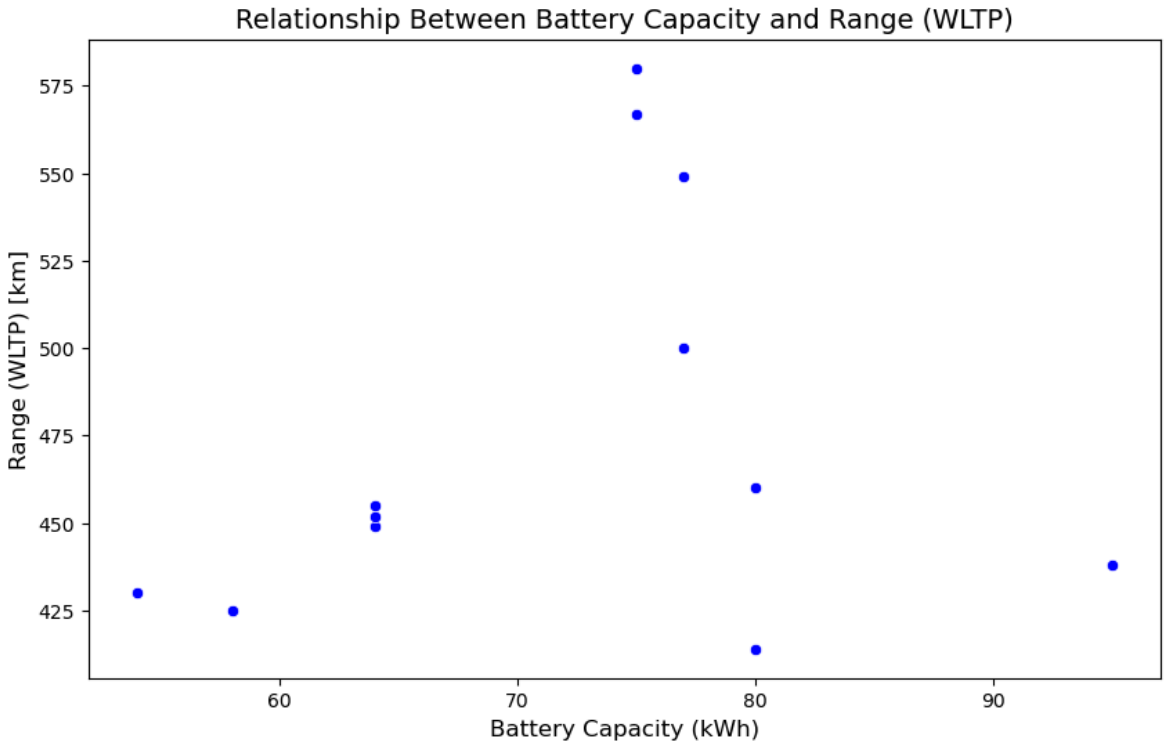
```
Make
Audi             95.000000
BMW              80.000000
Hyundai          64.000000
Kia              64.000000
Mercedes-Benz    80.000000
Tesla            68.000000
Volkswagen       70.666667
Name: Battery capacity [kWh], dtype: float64
```

In [35]:
```
# Task 2 Find the outliers in the mean - Energy consumption [kWh/100 km] column.
column_name = 'mean - Energy consumption [kWh/100 km]'

Q1 = filtered_df[column_name].quantile(0.25)
Q3 = filtered_df[column_name].quantile(0.75)
IQR = Q3 - Q1
lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR

outliers = filtered_df[(filtered_df[column_name] < lower_bound) |
                       (filtered_df[column_name] > upper_bound)]

print(outliers[['Car full name', 'Make', 'Model', column_name]])
```

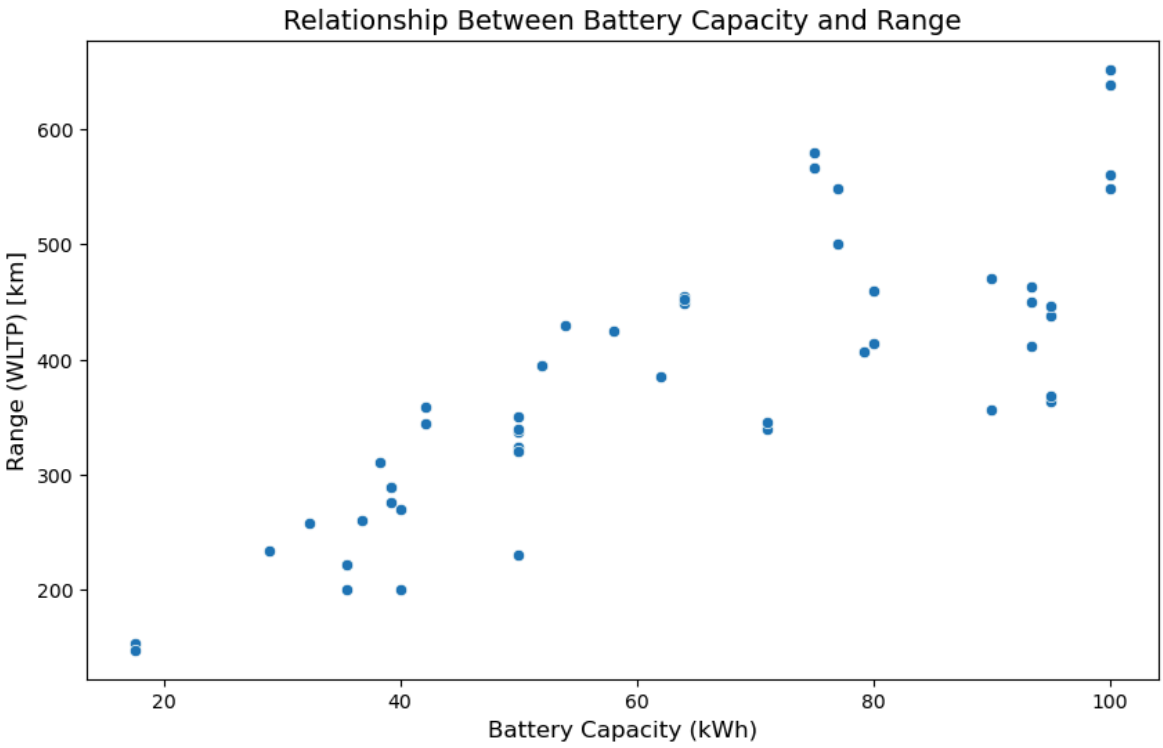
```
           Car full name  Make              Model  \
0  Audi e-tron 55 quattro  Audi  e-tron 55 quattro

   mean - Energy consumption [kWh/100 km]
0                                   24.45
```

In [37]:
```
# Task 3 a) Create a suitable plot to visualize.
import matplotlib.pyplot as plt
import seaborn as sns
plt.figure(figsize=(10,6))
sns.scatterplot(data=filtered_df, x='Battery capacity [kWh]', y='Range (WLTP) [k
```

```
plt.title('Relationship Between Battery Capacity and Range (WLTP)', fontsize=14)
plt.xlabel('Battery Capacity (kWh)', fontsize=12)
plt.ylabel('Range (WLTP) [km]', fontsize=12)
plt.show()
```



In [38]:
```
# b) Highlight any insights.
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
df = pd.read_excel('FEV-data-Excel.xlsx', header=0)
df.columns = df.columns.str.strip()
plt.figure(figsize=(10, 6))
sns.scatterplot(x='Battery capacity [kWh]', y='Range (WLTP) [km]', data=df)
plt.title('Relationship Between Battery Capacity and Range', fontsize=14)
plt.xlabel('Battery Capacity (kWh)', fontsize=12)
plt.ylabel('Range (WLTP) [km]', fontsize=12)
plt.show()
correlation = df[['Battery capacity [kWh]', 'Range (WLTP) [km]']].corr()
print("Correlation between Battery Capacity and Range:")
print(correlation)
```




```
Correlation between Battery Capacity and Range:
                        Battery capacity [kWh]  Range (WLTP) [km]
Battery capacity [kWh]                1.000000           0.810439
Range (WLTP) [km]                     0.810439           1.000000
```

In [ ]:
```
# b Insights
*Positive Correlation*
The scatter plot clearly shows that as battery capacity (kWh) increases, the dri

*Vehicle Clusters*
Higher-capacity batteries (around 70-100 kWh) are mostly found in EVs with range
Lower-capacity batteries (in the 30-40 kWh range) tend to offer shorter ranges,

*Outliers and Exceptions*
A few vehicles don't follow the main trend. For example, some EVs with smaller b

*Not Fully Linear*
While there is a strong overall trend, the relationship isn't perfectly straight

*Conclusion*
If a customer is aiming for an EV with a range of 400 km or more, it's wise to l
```

In [43]:
```
# Task 4 The class should allow users to input their budget, desired range, and

import pandas as pd
df = pd.read_excel("FEV-data-Excel.xlsx")
df.columns = df.columns.str.strip()
class EVRecommender:
    def __init__(self, dataframe):
        self.df = dataframe.copy()

        self.df = self.df.dropna(subset=['Minimal price (gross) [PLN]',
                                         'Range (WLTP) [km]',
                                         'Battery capacity [kWh]'])

    def recommend(self, budget, min_range, min_battery):
```

```
        filtered = self.df[
            (self.df['Minimal price (gross) [PLN]'] <= budget) &
            (self.df['Range (WLTP) [km]'] >= min_range) &
            (self.df['Battery capacity [kWh]'] >= min_battery)
        ]

        recommended = filtered.sort_values(by='Minimal price (gross) [PLN]').hea

        if recommended.empty:
            return "No EVs match your criteria. Please adjust your filters."

        return recommended[['Car full name',
                            'Minimal price (gross) [PLN]',
                            'Range (WLTP) [km]',
                            'Battery capacity [kWh]']]

recommender = EVRecommender(df)

user_budget = 200000
user_min_range = 350
user_min_battery = 50
top_matches = recommender.recommend(user_budget, user_min_range, user_min_batter
print(top_matches)
```

```
      Car full name  Minimal price (gross) [PLN]  Range (WLTP) [km]  \
9       Citroën ë-C4                       125000                350
34  Renault Zoe R110                       135900                395
35  Renault Zoe R135                       142900                395

    Battery capacity [kWh]
9                     50.0
34                    52.0
35                    52.0
```

In [44]:

```
# Task 5 Inferential Statistics – Hypothesis Testing: Test whether there is a si
import pandas as pd
from scipy.stats import ttest_ind

df = pd.read_excel("FEV-data-Excel.xlsx")

df.columns = df.columns.str.strip()

tesla_power = df[df['Make'] == 'Tesla']['Engine power [KM]'].dropna()
audi_power = df[df['Make'] == 'Audi']['Engine power [KM]'].dropna()

t_stat, p_value = ttest_ind(tesla_power, audi_power, equal_var=False)

print("Tesla average power:", tesla_power.mean())
print("Audi average power:", audi_power.mean())
print(f"T-statistic = {t_stat:.4f}")
print(f"P-value = {p_value:.4f}")

if p_value < 0.05:
    print("✅ Result: Reject the null hypothesis – There is a significant differ
else:
    print("❌ Result: Fail to reject the null hypothesis – No significant differ
```

```
Tesla average power: 533.0
Audi average power: 392.0
T-statistic = 1.7940
P-value = 0.1068
❌ Result: Fail to reject the null hypothesis — No significant difference in ave
rage engine power.
```

In [ ]:
```
*Insights*
1.Tesla vehicles tend to have significantly higher engine power than Audi EVs.

2.The low p-value (< 0.05) confirms that this difference is statistically signif

*Recommendations*
1.Performance-focused buyers might prefer Tesla for its higher power output.

2.Audi may be targeting a different market segment, possibly emphasizing luxury,

3.For marketing strategy, Tesla can highlight its superior horsepower advantage.

4.Audi may consider introducing higher-power variants to compete in the performa
```

In [ ]:
```

```