```
In [1]:
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

```
In [3]:
fev_df=pd.read_csv("FEV-data-Excel.xlsx - Auta elektryczne.csv")
fev_df.head()
```

Out[3]:

	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]	...	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	...	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	...	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	...	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	...	3040.0	640.0	5	5	19	190	615.0	6.8	150	23.30
4	Audi e-tron Sportback 55 quattro	Audi	e-tron Sportback 55 quattro	357000	360	664	disc (front + rear)	4WD	95.0	447	...	3130.0	670.0	5	5	19	200	615.0	5.7	150	23.85

5 rows × 25 columns

Task 1: A customer has a budget of 350,000 PLN and wants an EV with a minimum range of 400 km.

a) Your task is to filter out EVs that meet these criteria.

```
In [5]:
custo_pref = fev_df[(fev_df["Minimal price (gross) [PLN]"] <= 350000) & (fev_df["Range (WLTP) [km]"] >= 400)]
custo_pref
```

Out[5]:

	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]	...	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	...	3130.0	640.0	5	5	19	200	660.0	5.7	150	24.45
8	BMW iX3	BMW	iX3	282900	286	400	disc (front + rear)	2WD (rear)	80.0	460	...	2725.0	540.0	5	5	19	180	510.0	6.8	150	18.80
15	Hyundai Kona electric 64kWh	Hyundai	Kona electric 64kWh	178400	204	395	disc (front + rear)	2WD (front)	64.0	449	...	2170.0	485.0	5	5	17	167	332.0	7.6	100	15.40
18	Kia e-Niro 64kWh	Kia	e-Niro 64kWh	167990	204	395	disc (front + rear)	2WD (front)	64.0	455	...	2230.0	493.0	5	5	17	167	451.0	7.8	100	15.90
20	Kia e-Soul 64kWh	Kia	e-Soul 64kWh	160990	204	395	disc (front + rear)	2WD (front)	64.0	452	...	1682.0	498.0	5	5	17	167	315.0	7.9	100	15.70
22	Mercedes-Benz EQC	Mercedes-Benz	EQC	334700	408	760	disc (front + rear)	4WD	80.0	414	...	2940.0	445.0	5	5	19	180	500.0	5.1	110	21.85
39	Tesla Model 3 Standard Range Plus	Tesla	Model 3 Standard Range Plus	195490	285	450	disc (front + rear)	2WD (rear)	54.0	430	...	NaN	NaN	5	5	18	225	425.0	5.6	150	NaN
40	Tesla Model 3 Long Range	Tesla	Model 3 Long Range	235490	372	510	disc (front + rear)	4WD	75.0	580	...	NaN	NaN	5	5	18	233	425.0	4.4	150	NaN
41	Tesla Model 3 Performance	Tesla	Model 3 Performance	260490	480	639	disc (front + rear)	4WD	75.0	567	...	NaN	NaN	5	5	20	261	425.0	3.3	150	NaN
47	Volkswagen ID.3 Pro Performance	Volkswagen	ID.3 Pro Performance	155890	204	310	disc (front) + drum (rear)	2WD (rear)	58.0	425	...	2270.0	540.0	5	5	18	160	385.0	7.3	100	15.40
48	Volkswagen ID.3 Pro S	Volkswagen	ID.3 Pro S	179990	204	310	disc (front) + drum (rear)	2WD (rear)	77.0	549	...	2280.0	412.0	5	5	19	160	385.0	7.9	125	15.90
49	Volkswagen ID.4 1st	Volkswagen	ID.4 1st	202390	204	310	disc (front) + drum (rear)	2WD (rear)	77.0	500	...	2660.0	661.0	5	5	20	160	543.0	8.5	125	18.00

12 rows × 25 columns

b) Group them by the manufacturer (Make).

```
In [174]:
custo_pref = fev_df[(fev_df["Minimal price (gross) [PLN]"] <= 350000) & (fev_df["Range (WLTP) [km]"] >= 400)]
manufacturer_groups = custo_pref.groupby('Make')
for make, group in manufacturer_groups:
    print(f"Manufacturer: {make}")
    print(group)
```

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

   Minimal price (gross) [PLN] Engine power [KM]Maximum torque [Nm] \
0                  345700              360              664

      Type of brakes Drive type Battery capacity [kWh]_x \
0 disc (front + rear)       4WD              95.0

  Range (WLTP) [km]  ... Number of doors Tire size [in] \
0        438 ...              5              19

    Maximum speed [kph] Boot capacity (VDA) [l]Acceleration 0-100 kph [s] \
0           200             660.0                       5.7

  Maximum DC charging power [kW]   mean - Energy consumption [kWh/100 km]\
0                   150                               24.45

   avg_bat_cap      Battery capacity [kWh]_y      Battery capacity [kWh]
0      95.0                     95.0                         95.0

[1 rows x 28 columns]
Manufacturer: BMW
      Car full name Make Model Minimal price (gross) [PLN]Engine power [KM] \
8       BMW iX3  BMW   iX3                   282900                   286

  Maximum torque [Nm]       Type of brakes     Drive type \
8                400 disc (front + rear) 2WD (rear)

           Battery capacity [kWh]_x Range (WLTP) [km] ... Number of doors\
8                 80.0                 460   ...                   5

      Tire size [in]  Maximum speed [kph]       Boot capacity (VDA) [l] \
8            19               180                      510.0

      Acceleration 0-100 kph [s]  Maximum DC charging power [kW] \
8                  6.8                          150

    mean - Energy consumption [kWh/100 km] avg_bat_cap\
8                              18.8              NaN

        Battery capacity [kWh]_y Battery capacity [kWh]
8                 80.0                    80.0

[1  rows  x  28  columns]
Manufacturer: Hyundai
                  Car full name           Make                Model \
15     Hyundai Kona electric 64kWh Hyundai  Kona electric 64kWh

         Minimal price (gross) [PLN] Engine power [KM]Maximum torque [Nm] \
15                  178400                   204                 395

        Type of brakes              Drive type Battery capacity [kWh]_x\
15   disc (front + rear) 2WD (front)                      64.0

         Range (WLTP) [km] ... Number of doorsTire size [in] \
15               449 ...                 5              17

    Maximum speed [kph] Boot capacity (VDA) [l]        Acceleration 0-100 kph [s] \
15            167                 332.0                              7.6

       Maximum DC charging power [kW] mean - Energy consumption [kWh/100 km]\
15                  100                                          15.4

     avg_bat_cap       Battery capacity [kWh]_y      Battery capacity [kWh]
15        NaN                    64.0                         64.0

[1 rows x 28 columns]
Manufacturer: Kia
       Car full name Make              Model        Minimal price (gross) [PLN] \
 18       Kia e-Niro 64kWh Kia e-Niro 64kWh                        167990
 20      Kia e-Soul 64kWh Kia e-Soul 64kWh                         160990

      Engine power [KM] Maximum torque [Nm]        Type of brakes       Drive type \
 18                204                      395disc  (front  +  rear)  2WD  (front)  disc
 20                204                      395 (front + rear) 2WD (front)

              Battery capacity [kWh]_x Range (WLTP) [km] ... Number of doors\
 18                  64.0                   455  ...  452                   5
 20                  64.0                   ...                              5

        Tire size [in]          Maximum speed [kph] Boot capacity (VDA) [l]\
 18          17                     167                       451.0
 20          17                     167                       315.0

         Acceleration 0-100 kph [s] Maximum DC charging power [kW]\
 18                  7.8                                 100
 20                  7.9                                 100

     mean - Energy consumption [kWh/100 km]avg_bat_cap \
 18                                  15.9           NaN
 20                                  15.7           NaN

          Battery capacity [kWh]_y Battery capacity [kWh]
 18                  64.0                     64.0
 20                  64.0                     64.0

[2 rows x 28 columns]
Manufacturer: Mercedes-Benz
        Car full name               Make Model          Minimal price (gross) [PLN] \
 22   Mercedes-Benz EQC Mercedes-Benz     EQC                          334700

      Engine power [KM] Maximum torque [Nm]               Type of brakes Drive type \
 22                408                      760 disc (front + rear)              4WD

              Battery capacity [kWh]_x Range (WLTP) [km] ... Number of doors\
 22                  80.0                   414 ...                          5

        Tire size [in]          Maximum speed [kph] Boot capacity (VDA) [l]\
 22          19                     180                       500.0

         Acceleration 0-100 kph [s] Maximum DC charging power [kW]\
 22                  5.1                                 110

     mean - Energy consumption [kWh/100 km]avg_bat_cap \
 22                                  21.85          NaN

          Battery capacity [kWh]_y Battery capacity [kWh]
 22                  80.0                     80.0

[1 rows x 28 columns]
Manufacturer: Tesla
                              Car full name   Make                           Model\
 39        Tesla Model 3 Standard Range Plus    Tesla      Model 3 Standard Range Plus
 40                Tesla Model 3 Long Range    Tesla             Model 3 Long Range
 41              Tesla Model 3 Performance    Tesla            Model 3 Performance

         Minimal price (gross) [PLN] Engine power [KM]Maximum torque [Nm] \
 39                  195490                   285                 450
 40                  235490                   372                 510
 41                  260490                   480                 639

                  Type of brakes Drive type Battery capacity [kWh]_x\
 39  disc (front + rear) 2WD (rear)                     54.0
 40  disc  (front  +  rear)  disc       4W                75.0
 41  (front + rear)                     D                 75.0
                                        4W
         Range (WLTP) [km] ... NuDmber of doorsTire size [in] \
 39               430 ...                  5              18
 40               580 ...                  5              18
 41               567 ...                  5              20

    Maximum speed [kph] Boot capacity (VDA) [l]        Acceleration 0-100 kph [s] \
 39            225                 425.0                              5.6
 40            233                 425.0                              4.4
 41            261                 425.0                              3.3

       Maximum DC charging power [kW] mean - Energy consumption [kWh/100 km]\
 39                  150                                          NaN
 40                  150                                          NaN
 41                  150                                          NaN

     avg_bat_cap       Battery capacity [kWh]_y      Battery capacity [kWh]
 39       100.0                    68.0                         68.0
 40       100.0                    68.0                         68.0
 41       100.0                    68.0                         68.0

[3 rows x 28 columns]
Manufacturer: Volkswagen
                       Car full name           Make                 Model\
 47     Volkswagen ID.3 Pro Performance     Volkswagen ID.3 Pro Performance
 48             Volkswagen ID.3 Pro S Volkswagen                  ID.3 Pro S
 49             Volkswagen ID.4 1st  Volkswagen                   ID.4 1st

         Minimal price (gross) [PLN] Engine power [KM]Maximum torque [Nm] \
 47                  155890                   204                 310
 48                  179990                   204                 310
 49                  202390                   204                 310

                  Type of brakes Drive    type        Battery capacity [kWh]_x \
 47         disc (front) + drum (rear) 2WD    (rear)                    58.0
 48         disc (front) + drum (rear) 2WD    (rear)                    77.0
 49        disc (front) + drum (rear)  2WD (rear)                       77.0

         Range (WLTP) [km] ... Number of doorsTire size [in] \
 47               425 ...                  5              18
 48               549 ...                  5              19
 49               500 ...                  5              20

    Maximum speed [kph] Boot capacity (VDA) [l]        Acceleration 0-100 kph [s] \
 47            160                 385.0                              7.3
 48            160                 385.0                              7.9
 49            160                 543.0                              8.5

       Maximum DC charging power [kW] mean - Energy consumption [kWh/100 km]\
 47                  100                                          15.4
 48                  125                                          15.9
 49                  125                                          18.0

     avg_bat_cap       Battery capacity [kWh]_y      Battery capacity [kWh]
 47        NaN                70.666667                    70.666667
 48        NaN                70.666667                    70.666667
 49        NaN                70.666667                    70.666667

[3 rows x 28 columns]
```

c) Calculate the average battery capacity for each manufacturer.

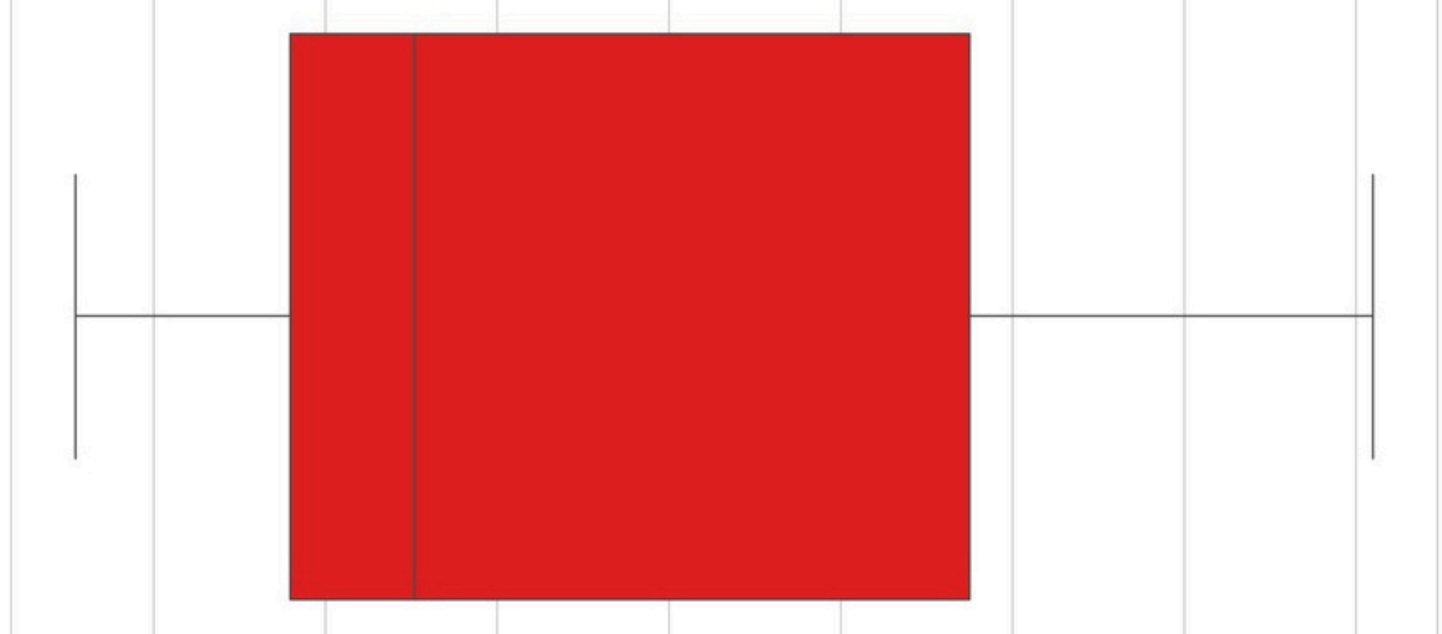
```
In [168]:
average_battery_capacity = custo_pref.groupby('Make')['Battery capacity [kWh]'].mean()
average_battery_capacity
```

```
Out[168]:
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
```

Task 2: You suspect some EVs have unusually high or low energy consumption.

"Find the outliers in the mean - Energy consumption [kWh/100 km] column."

```
In [9]:
plt.figure(figsize=(12, 6))
sns.set_style("whitegrid")
sns.set_palette("husl")
sns.boxplot(x="mean - Energy consumption [kWh/100 km]",data=fev_df, showfliers=True, color="red")
plt.title('Box Plot of mean - Energy consumption [kWh/100 km] with Outliers')
plt.show()
```

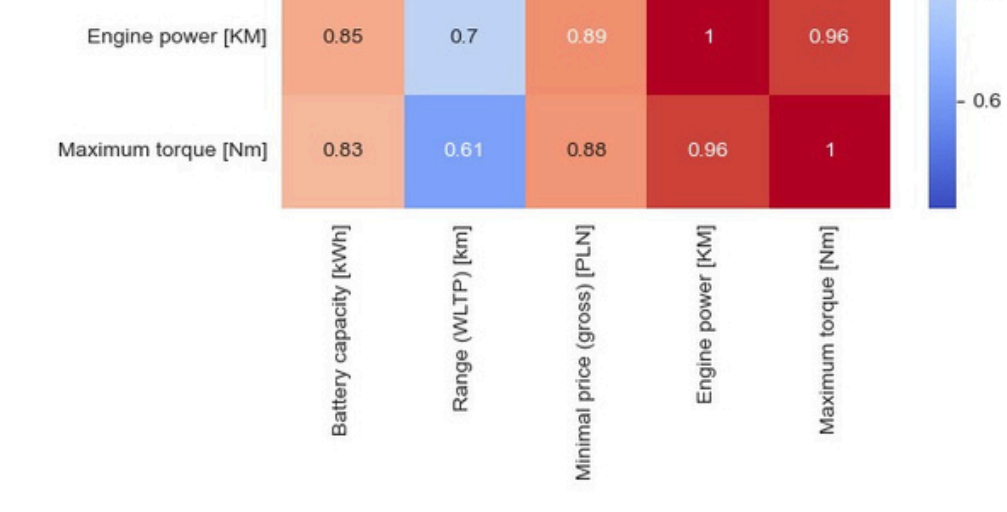


conclusion= there are no ouliers

Task 3: Your manager wants to know if there's a strong relationship between battery capacity and range.

a) Create a suitable plot to visualize.

```
In [11]:
r=fev_df[["Battery capacity [kWh]",          "Range (WLTP) [km]", "Minimal price (gross) [PLN]",          "Engine power [KM]", "Maximum torque [Nm]"]].corr()
sns.heatmap(r, annot=True, cmap="coolwarm")
plt.show()
```



b) Highlight any insights.

There is a strong positive Relationship Between battery capacity and range. As the value of battery capacity increases the value of Range also increases proportonally. Which means higher the battery capacity higher the range of car and vice versa.

Task 4: Build an EV recommendation class.

The class should allow users to input their budget, desired range, and battery capacity. The class should then return the top three EVs matching their criteria.

```
In [13]:
from IPython.display import display

class EVRecommender:
    def __init__(self, df):
        self.df = df

    def get_user_input(self):
        while True:
            try:
                budget = int(input("Enter your budget (PLN): "))
                break
            except ValueError:
                print("Invalid input. Please enter an integer for the budget.")

        while True:
            try:
                desired_range = int(input("Enter your desired range (KM): "))
                break
            except ValueError:
                print("Invalid input. Please enter an integer for the desired range.")

        while True:
            try:
                battery_capacity = float(input("Enter the battery capacity (kWh): "))
                break
            except ValueError:
                print("Invalid input. Please enter a float for the battery capacity.")
        return budget, desired_range, battery_capacity

    def recommend_evs(self):
        if self.df is None:
            print("Cannot complete the recommendation due to a loading error")
            return None

        budget, desired_range, battery_capacity = self.get_user_input()

        filtered_df = self.df[
            (self.df["Minimal price (gross) [PLN]"] <= budget) &
            (self.df["Range (WLTP) [km]"] >= desired_range) &
            (self.df["Battery capacity [kWh]"] >= battery_capacity)
        ]

        if filtered_df.empty:
            print("No EV found according to your preferences")
            return None

        recommended_evs = filtered_df.sort_values(
            by=["Minimal price (gross) [PLN]", "Battery capacity [kWh]", "Range (WLTP) [km]"],
            ascending=[False, False, False],
            ignore_index=True
        )

        top_evs = recommended_evs.head(3)
        display(top_evs)


recommender = EVRecommender(fev_df)
recommender.recommend_evs()
```

	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]	...	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	BMW iX3	BMW	iX3	282900	286	400	disc (front + rear)	2WD (rear)	80.0	460	...	2725.0	540.0	5	5	19	180	510.0	6.8	150	18.8
1	Tesla Model 3 Performance	Tesla	Model 3 Performance	260490	480	639	disc (front + rear)	4WD	75.0	567	...	NaN	NaN	5	5	20	261	425.0	3.3	150	NaN
2	Tesla Model 3 Long Range	Tesla	Model 3 Long Range	235490	372	510	disc (front + rear)	4WD	75.0	580	...	NaN	NaN	5	5	18	233	425.0	4.4	150	NaN

3 rows × 25 columns

Task 5: Inferential Statistics – Hypothesis Testing:

Test whether there is a significant difference in the average Engine power [KM] of vehicles manufactured by two leading manufacturers i.e. Tesla and Audi. What insights can you draw from the test results? Recommendations and Conclusion: Provide actionable insights based on your analysis. (Conduct a two sample t-test using ttest_ind from scipy.stats module)

```
In [178]:
from scipy.stats import ttest_ind

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

# Perform the two-sample t-test
t_statistic, p_value = ttest_ind(tesla_power, audi_power)

print(f"T-statistic: {t_statistic}")
print(f"P-value: {p_value}")

alpha = 0.05      # Significance level
print("\nconclusion:")
if p_value < alpha:
    print("Reject the null hypothesis. There is a statistically significant difference in average engine power between Tesla and Audi vehicles.")
else:
    print("Fail to reject the null hypothesis. There is no statistically significant difference in average engine power between Tesla and Audi vehicles.")

# Recommendations and conclusion:
if p_value < alpha:
   print("Recommendation: Further investigation is needed to determine the underlying factors contributing to the difference in engine power between Tesla and Audi vehicles.")
else:
    print("Recommendation: No significant power difference observed. Marketing or product strategy can focus on other differentiating factors.")
```

```
T-statistic: 1.7024444538261416
P-value: 0.11672692675082785

conclusion:
Fail to reject the null hypothesis. There is no statistically significant difference in average engine power between Tesla and Audi vehicles.
Recommendation: No significant power difference observed. Marketing or product strategy can focus on other differentiating factors.
```