In [5]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import ttest_ind
```

In [11]:
```
data_path = 'FEV.xlsx'
df = pd.read_excel(data_path)
```

In [33]:
```
df.head(5) #checking the file whether imported correctly or not
```

Out[33]:

	Car full name	Make	Model	Minimal price (gross) [PLN]	Engine power [KM]	Maximum torque [Nm]	Type of brakes	Drive type	Battery capacity [kWh]	(
0	Audi e-tron 55 quattro	Audi	e-tron 55 quattro	345700	360	664	disc (front + rear)	4WD	95.0	
1	Audi e-tron 50 quattro	Audi	e-tron 50 quattro	308400	313	540	disc (front + rear)	4WD	71.0	
2	Audi e-tron S quattro	Audi	e-tron S quattro	414900	503	973	disc (front + rear)	4WD	95.0	
3	Audi e-tron Sportback 50 quattro	Audi	e-tron Sportback 50 quattro	319700	313	540	disc (front + rear)	4WD	71.0	
4	Audi e-tron Sportback 55 quattro	Audi	e-tron Sportback 55 quattro	357000	360	664	disc (front + rear)	4WD	95.0	

5 rows × 25 columns

In [45]:
```
#Task 1
filtered_df = df[(df['Minimal price (gross) [PLN]'] <= 350000) & (df['Range (WLT
grouped_by_make = filtered_df.groupby('Make')
avg_battery_capacity = grouped_by_make['Battery capacity [kWh]'].mean()
print("Average Battery Capacity by Manufacturer")
print(avg_battery_capacity)
```

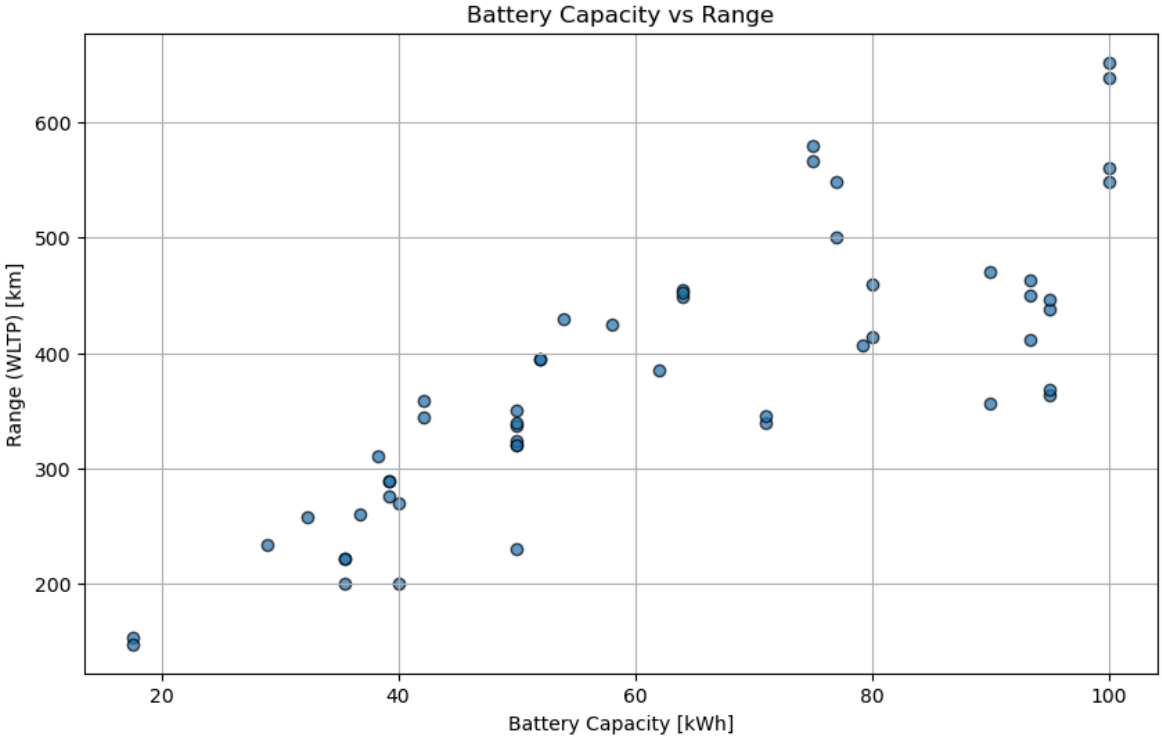
```
Average Battery Capacity by Manufacturer
Make
Audi              95.000000
BMW               80.000000
Hyundai           64.000000
Kia               64.000000
Mercedes-Benz     80.000000
Tesla             68.000000
Volkswagen        70.666667
Name: Battery capacity [kWh], dtype: float64
```

In [41]:
```python
#Task 2
energy_consumption = df['mean - Energy consumption [kWh/100 km]']
z_scores = (energy_consumption - energy_consumption.mean()) / energy_consumption
outliers = df[(z_scores.abs() > 3)]
print("\n Outliers in Energy Consumption")
print(outliers[['Car full name', 'mean - Energy consumption [kWh/100 km]']])
```

```
 Outliers in Energy Consumption
Empty DataFrame
Columns: [Car full name, mean - Energy consumption [kWh/100 km]]
Index: []
```

In [39]:
```python
#Task 3
# a) Create a scatter plot
plt.figure(figsize=(10, 6))
plt.scatter(df['Battery capacity [kWh]'], df['Range (WLTP) [km]'], alpha=0.7, ed
plt.title('Battery Capacity vs Range')
plt.xlabel('Battery Capacity [kWh]')
plt.ylabel('Range (WLTP) [km]')
plt.grid(True)
plt.show()

# b) Highlight Insights
correlation = df['Battery capacity [kWh]'].corr(df['Range (WLTP) [km]'])
print(f"\n Correlation between Battery Capacity and Range: {correlation:.2f}")
```




```
Correlation between Battery Capacity and Range: 0.81
```

In [35]:
```
#Task 4
class EVRecommendation:
    def __init__(self, df):
        self.df = df

    def recommend(self, budget, desired_range, min_battery_capacity):
        recommendations = self.df[(self.df['Minimal price (gross) [PLN]'] <= bud
                                   (self.df['Range (WLTP) [km]'] >= desired_rang
                                   (self.df['Battery capacity [kWh]'] >= min_bat
        return recommendations.nlargest(3, 'Range (WLTP) [km]')[['Car full name'

recommender = EVRecommendation(df)
budget = 350000
desired_range = 400
min_battery_capacity = 50
recommendations = recommender.recommend(budget, desired_range, min_battery_capac
print("\n Top 3 EV Recommendations")
print(recommendations)
```

```
 Top 3 EV Recommendations
                Car full name        Make  Range (WLTP) [km]  \
40   Tesla Model 3 Long Range       Tesla                580
41  Tesla Model 3 Performance       Tesla                567
48      Volkswagen ID.3 Pro S  Volkswagen                549

    Battery capacity [kWh]
40                    75.0
41                    75.0
48                    77.0
```

In [37]:
```
#Task 5
# Comparing Engine power between Tesla and Audi
tesla_power = df[df['Make'] == 'Tesla']['Engine power [KM]']
audi_power = df[df['Make'] == 'Audi']['Engine power [KM]']

# Performing two-sample t-test
stat, p_value = ttest_ind(tesla_power.dropna(), audi_power.dropna(), equal_var=F
print("\n Hypothesis Testing")
print(f"T-statistic: {stat:.2f}, P-value: {p_value:.4f}")
if p_value < 0.05:
    print("There is a significant difference in the average engine power between
else:
    print("No significant difference in the average engine power between Tesla a
```

```
 Hypothesis Testing
T-statistic: 1.79, P-value: 0.1068
No significant difference in the average engine power between Tesla and Audi.
```

In [ ]: