In [11]:

```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from scipy.stats import ttest_ind
```

In [31]:

```python
# Load the dataset (Assuming a CSV file named 'ev_data.csv
df=pd.read_csv("D:/Bhavesh/Internshala Data Analytics/C5 PYTHON/Project/FEV-data-Excel.xlsx - Auta elektryczne.csv")
df.head()
```

Out[31]:

	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

In [15]:

```python
#Task 1 Part a
filtered_evs = df[(df["Minimal price (gross) [PLN]"] <= 350000) & (df["Range (WLTP) [km]"] >= 400)]
print(filtered_evs[['Make','Model','Minimal price (gross) [PLN]','Range (WLTP) [km]']])
```

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

    Range (WLTP) [km]
0                 438
8                 460
15                449
18                455
20                452
22                414
39                430
40                580
41                567
47                425
48                549
49                500
```

In [102…

```python
#Part b
grouped_evs = filtered_evs.groupby('Make').size()
print(grouped_evs)
```

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

In [126…

```python
#Part c
avg_battery_capacity = filtered_evs.groupby('Make')['Battery capacity [kWh]'].mean()
print(avg_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
```

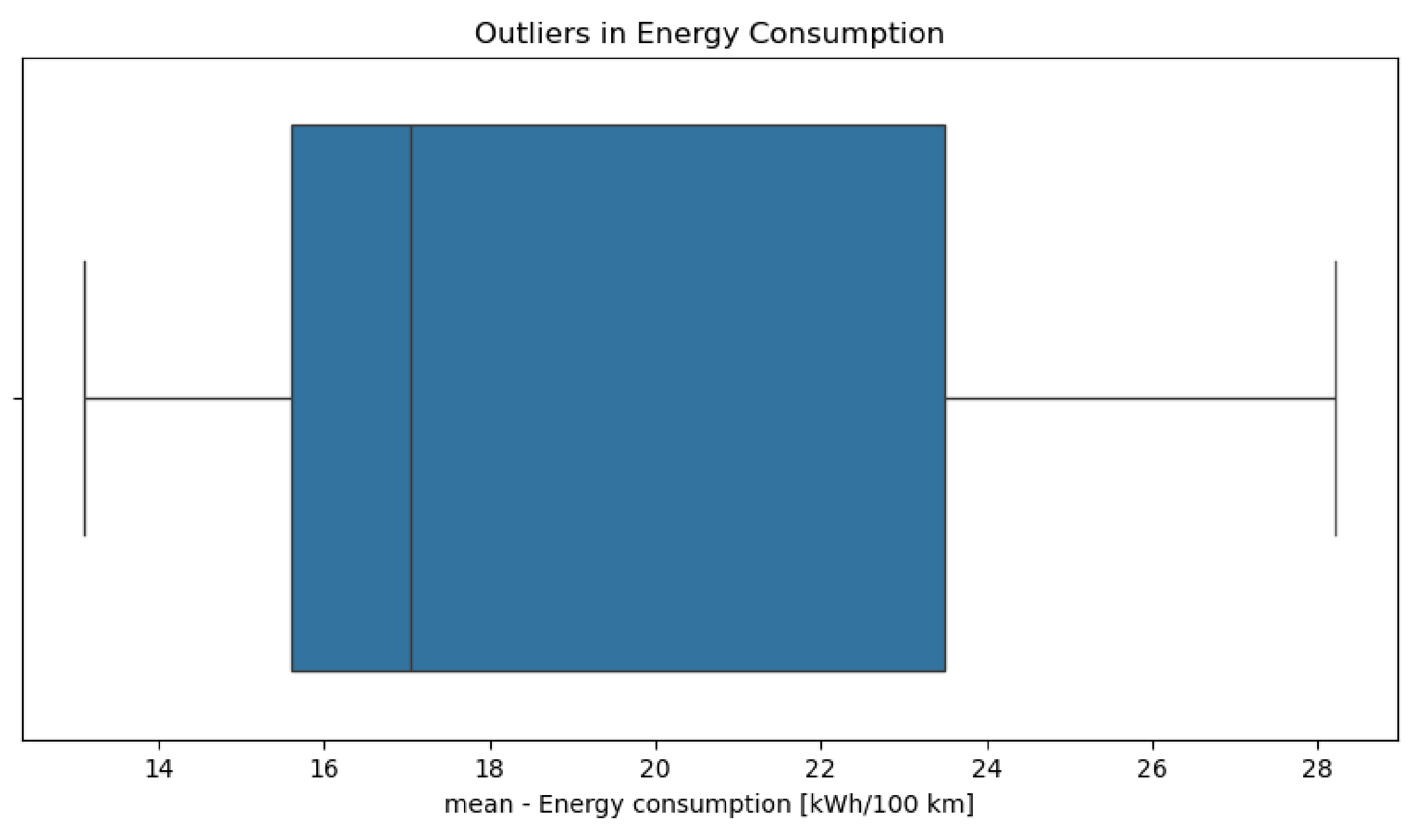
In [124…

```python
#Task 2
q1 = df['mean - Energy consumption [kWh/100 km]'].quantile(0.25)
q3 = df['mean - Energy consumption [kWh/100 km]'].quantile(0.75)
iqr = q3 - q1
lower_bound = q1 - 1.5 * iqr
upper_bound = q3 + 1.5 * iqr
outliers = df[(df['mean - Energy consumption [kWh/100 km]'] < lower_bound) | (df['mean - Energy consumption [kWh/100 km]'] > upper_bound)]
print(outliers[['Make','Model','mean - Energy consumption [kWh/100 km]']])
```

```
Empty DataFrame
Columns: [Make, Model, mean - Energy consumption [kWh/100 km]]
Index: []
```

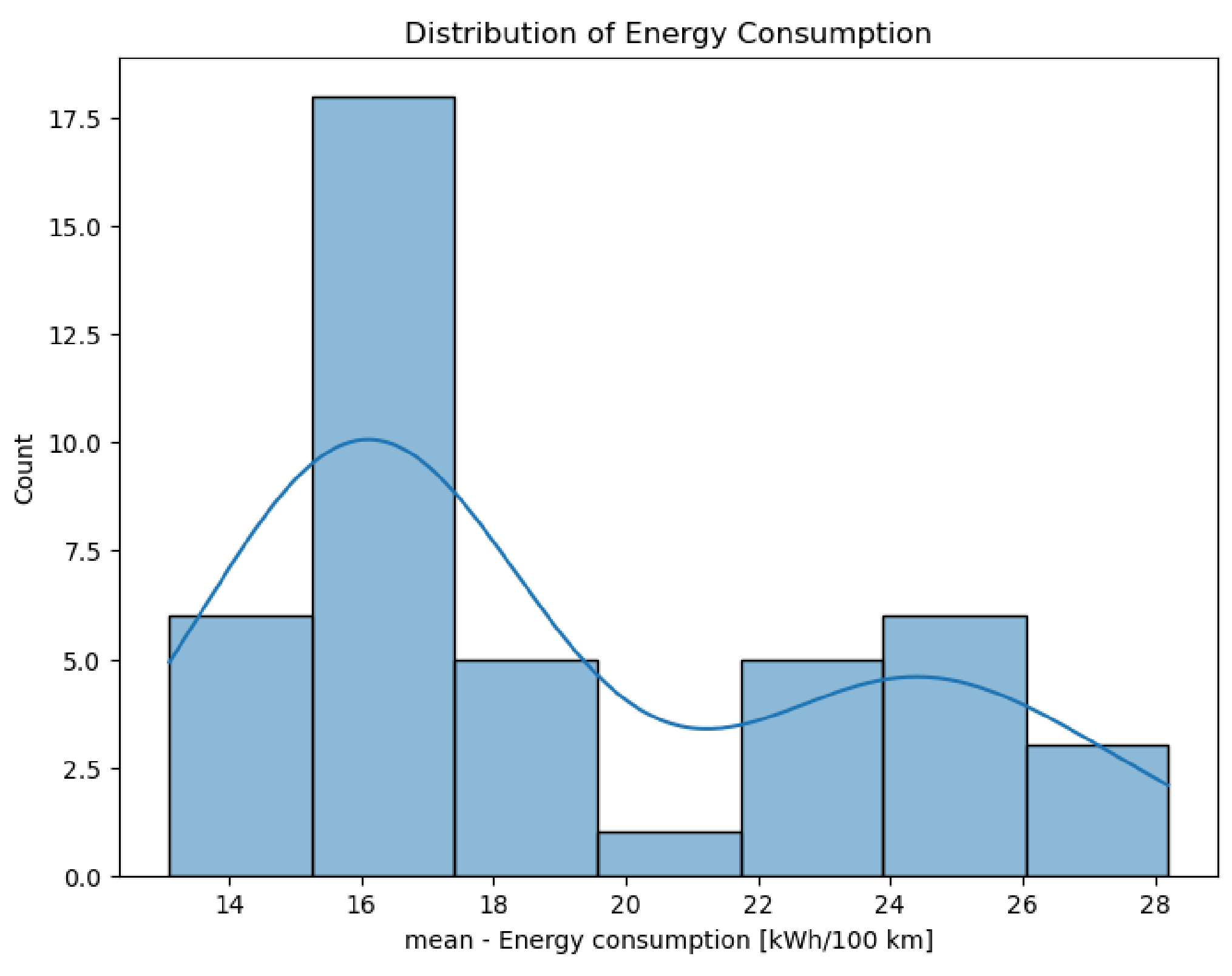
In [136…

```python
#Task 2 Plotting Boxplot
plt.figure(figsize=(10,5))
sns.boxplot(x=df['mean - Energy consumption [kWh/100 km]'])
plt.title("Outliers in Energy Consumption")
plt.show()
```



In [138…

```python
#Task 2 Plotting Histogram
plt.figure(figsize=(8, 6))
sns.histplot(df['mean - Energy consumption [kWh/100 km]'], kde=True)
plt.title('Distribution of Energy Consumption')
plt.show()
```



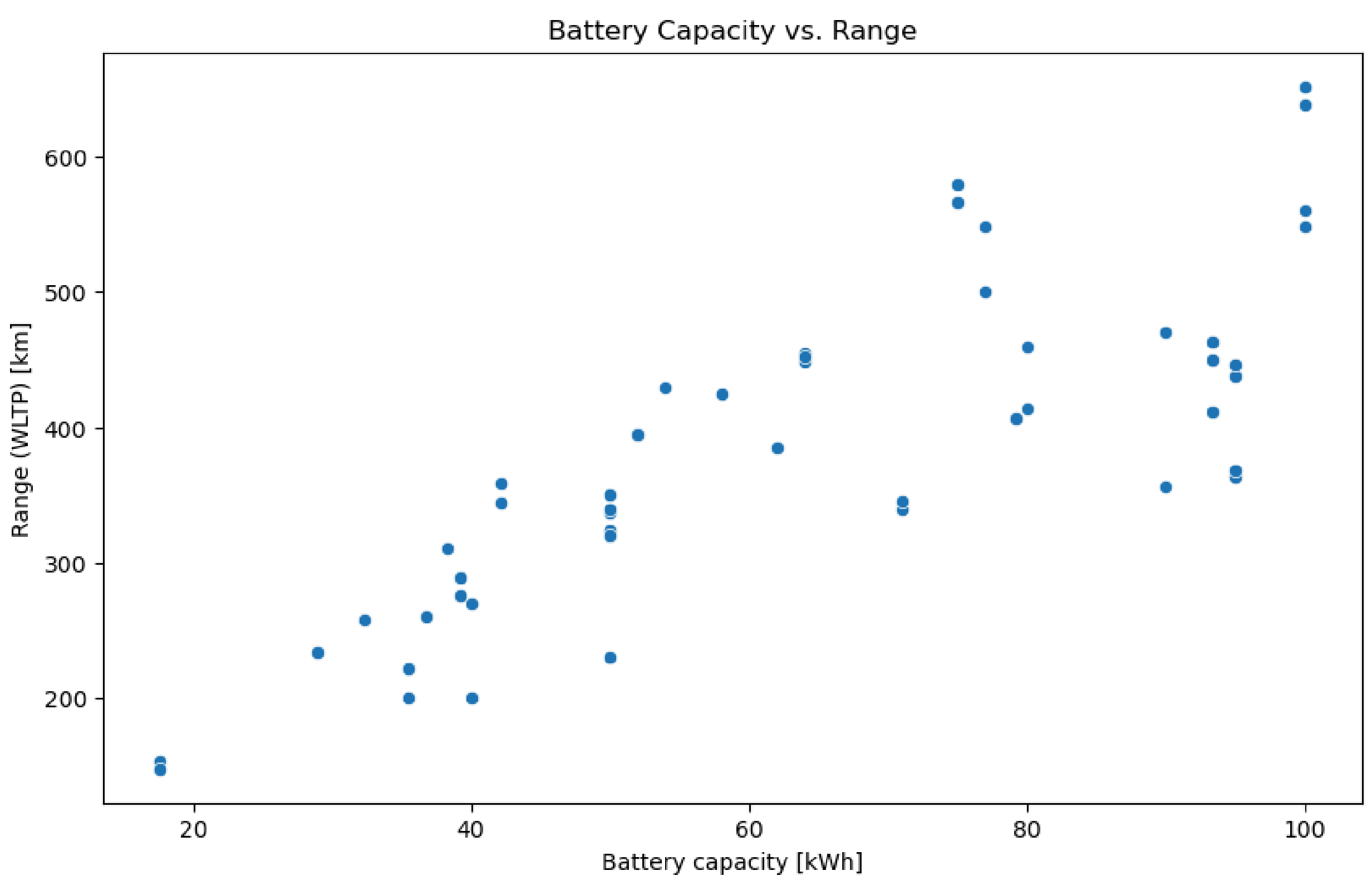
In [158…

```python
#Task 3
def analyze_battery_range_relationship(df: pd.DataFrame):
    plt.figure(figsize=(10, 6))
    sns.scatterplot(x=df['Battery capacity [kWh]'], y=df['Range (WLTP) [km]'])
    plt.xlabel("Battery capacity [kWh]")
    plt.ylabel("Range (WLTP) [km]")
    plt.title("Battery Capacity vs. Range")
    plt.show()


    correlation = df['Battery capacity [kWh]'].corr(df['Range (WLTP) [km]'])
    return correlation


if __name__ == "__main__":

    df = pd.read_csv("D:/Bhavesh/Internshala Data Analytics/C5 PYTHON/Project/FEV-data-Excel.xlsx - Auta elektryczne.csv")
    correlation = analyze_battery_range_relationship(df)
    print(f"\nCorrelation between Battery Capacity and Range: {correlation:.2f}")
```



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

In [23]:

```python
# Task 4
class EVRecommendation:
    def __init__(self, df):
        self.df = df

    def recommend_ev(self, budget, min_range, min_battery_capacity):

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


        if filtered_evs.empty:
            return "No EVs match your criteria. Try adjusting your budget or requirements."


        top_evs = filtered_evs.sort_values(by=['Range (WLTP) [km]', 'Battery capacity [kWh]'], ascending=[False, False]).head(3)

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


ev_recommender = EVRecommendation(df)


budget = float(input("Enter your budget in PLN: "))
min_range = float(input("Enter the minimum range you need (in km): "))
min_battery_capacity = float(input("Enter the minimum battery capacity (in kWh): "))


recommended_evs = ev_recommender.recommend_ev(budget, min_range, min_battery_capacity)


print("\nTop Recommended EVs:")
print(recommended_evs)
```

```
Top Recommended EVs:
                Car full name  Minimal price (gross) [PLN]  Range (WLTP) [km]  \
40   Tesla Model 3 Long Range                       235490                580
41  Tesla Model 3 Performance                       260490                567
48      Volkswagen ID.3 Pro S                       179990                549

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

In [162…

```python
#Task 5
import scipy.stats as stats

tesla_power = df[df['Make'] == 'Tesla']['Engine power [KM]']
audi_power = df[df['Make'] == 'Audi']['Engine power [KM]']


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


print("="*50)
print("Hypothesis Testing: Tesla vs. Audi Engine Power")
print("="*50)
print(f"T-statistic: {t_stat:.4f}")
print(f"P-value: {p_value:.4f}")
print("-"*50)


alpha = 0.05
if p_value < alpha:
    print("Conclusion: We reject the Null Hypothesis (H₀).")
    print("There is a significant difference in the average engine power between Tesla and Audi EVs.")
else:
    print("Conclusion: We fail to reject the Null Hypothesis (H₀).")
    print("There is NO significant difference in the average engine power between Tesla and Audi EVs.")

print("-"*50)


if p_value < alpha:
    if tesla_power.mean() > audi_power.mean():
        print("Insight: Tesla's EVs have significantly higher average engine power than Audi.")
        print("Recommendation: Tesla can market its EVs as high-performance vehicles, targeting customers who value power.")
    else:
        print("Insight: Audi's EVs have significantly higher average engine power than Tesla.")
        print("Recommendation: Audi should highlight its power advantage in marketing and product positioning.")
else:
    print("Insight: There is no significant difference in the engine power of Tesla and Audi EVs.")
    print("Recommendation: Both brands are competing closely in performance. Customers should focus on other factors like range, price, and charging speed when making pu

print("="*50)
```

```
==================================================
Hypothesis Testing: Tesla vs. Audi Engine Power
==================================================
T-statistic: 1.7940
P-value: 0.1068
--------------------------------------------------
Conclusion: We fail to reject the Null Hypothesis (H₀).
There is NO significant difference in the average engine power between Tesla and Audi EVs.
--------------------------------------------------
Insight: There is no significant difference in the engine power of Tesla and Audi EVs.
Recommendation: Both brands are competing closely in performance. Customers should focus on other factors like range, price, and charging speed when making purchasing dec
```

```
isions.
==================================================
```

```
In [ ]:
```