EV_Data_Analysis

July 20, 2025

[14]:
```python
# Load the electric vehicle dataset and required Python libraries

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

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

print(df.head())
```

```
                  Car full name  Make                         Model  \
0            Audi e-tron 55 quattro  Audi            e-tron 55 quattro
1            Audi e-tron 50 quattro  Audi            e-tron 50 quattro
2             Audi e-tron S quattro  Audi             e-tron S quattro
3  Audi e-tron Sportback 50 quattro  Audi  e-tron Sportback 50 quattro
4  Audi e-tron Sportback 55 quattro  Audi  e-tron Sportback 55 quattro

   Minimal price (gross) [PLN]  Engine power [KM]  Maximum torque [Nm]  \
0                       345700                360                  664
1                       308400                313                  540
2                       414900                503                  973
3                       319700                313                  540
4                       357000                360                  664

       Type of brakes Drive type  Battery capacity [kWh]  Range (WLTP) [km]  \
0  disc (front + rear)        4WD                    95.0                438
1  disc (front + rear)        4WD                    71.0                340
2  disc (front + rear)        4WD                    95.0                364
3  disc (front + rear)        4WD                    71.0                346
4  disc (front + rear)        4WD                    95.0                447

   …  Permissable gross weight [kg]  Maximum load capacity [kg]  \
0  …                         3130.0                       640.0
1  …                         3040.0                       670.0
2  …                         3130.0                       565.0
3  …                         3040.0                       640.0
```

```
4   …                        3130.0                       670.0

   Number of seats  Number of doors  Tire size [in]  Maximum speed [kph]  \
0                5                5              19                  200
1                5                5              19                  190
2                5                5              20                  210
3                5                5              19                  190
4                5                5              19                  200

   Boot capacity (VDA) [l]  Acceleration 0-100 kph [s]  \
0                    660.0                         5.7
1                    660.0                         6.8
2                    660.0                         4.5
3                    615.0                         6.8
4                    615.0                         5.7

   Maximum DC charging power [kW]  mean - Energy consumption [kWh/100 km]
0                             150                                   24.45
1                             150                                   23.80
2                             150                                   27.55
3                             150                                   23.30
4                             150                                   23.85

[5 rows x 25 columns]
```

[15]:

```python
# Task 1: Filter EVs under 350,000 PLN with at least 400 km range,
# group the filtered EVs by manufacturer (Make), and calculate
# the average battery capacity per manufacturer.

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

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

filtered_df = df[(df["Minimal price (gross) [PLN]"] <= 350000) & (df["Range␣
↪(WLTP) [km]"] >= 400)]
print("Filtered EVs:\n", filtered_df[["Car full name", "Make", "Minimal price␣
↪(gross) [PLN]", "Range (WLTP) [km]"]])

grouped_by_make = filtered_df.groupby("Make")
print("\nGrouped by Manufacturer:\n", grouped_by_make.size())

avg_battery_by_make = grouped_by_make["Battery capacity [kWh]"].mean().
↪reset_index()
```

```python
print("\nAverage Battery Capacity per Make:\n", avg_battery_by_make)
```

```
Filtered EVs:
                         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

    Minimal price (gross) [PLN]  Range (WLTP) [km]
0                        345700                438
8                        282900                460
15                       178400                449
18                       167990                455
20                       160990                452
22                       334700                414
39                       195490                430
40                       235490                580
41                       260490                567
47                       155890                425
48                       179990                549
49                       202390                500

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

Average Battery Capacity per Make:
             Make  Battery capacity [kWh]
0           Audi               95.000000
1            BMW               80.000000
2        Hyundai               64.000000
3            Kia               64.000000
```

```
4   Mercedes-Benz                 80.000000
5           Tesla                 68.000000
6      Volkswagen                 70.666667
```

[16]:
```
# Task 2: Detect outlier vehicles based on their mean energy consumption
# using the Interquartile Range (IQR) method.
import pandas as pd

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

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 in Energy Consumption:\n", outliers[["Car full name", "mean -␣
↪Energy consumption [kWh/100 km]"]])
```

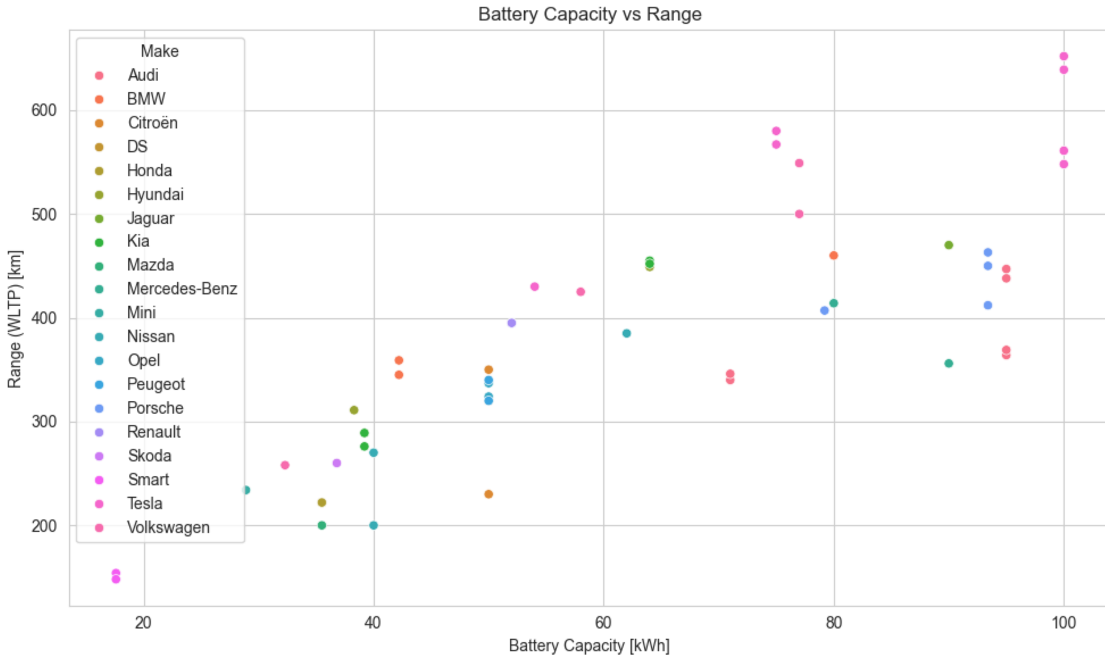
```
Outliers in Energy Consumption:
 Empty DataFrame
Columns: [Car full name, mean - Energy consumption [kWh/100 km]]
Index: []
```

[17]:
```
# Task 3: Visualize the relationship between battery capacity and vehicle range
# using a scatter plot with manufacturer-based coloring.

import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

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

plt.figure(figsize=(10, 6))
sns.scatterplot(data=df, x="Battery capacity [kWh]", y="Range (WLTP) [km]",␣
↪hue="Make")
plt.title("Battery Capacity vs Range")
plt.xlabel("Battery Capacity [kWh]")
plt.ylabel("Range (WLTP) [km]")
plt.grid(True)
plt.tight_layout()
plt.show()
```



[18]:
```
# Task 4: Build a recommendation system class that filters the top 3 EVs
# based on user's budget, minimum range, and battery capacity.

import pandas as pd

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

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

    def recommend(self, budget, min_range, min_battery):
        filtered = self.data[
            (self.data["Minimal price (gross) [PLN]"] <= budget) &
            (self.data["Range (WLTP) [km]"] >= min_range) &
            (self.data["Battery capacity [kWh]"] >= min_battery)
        ]
        top = filtered.sort_values(by="Range (WLTP) [km]", ascending=False).
 ↪head(3)
        return top[["Car full name", "Make", "Minimal price (gross) [PLN]",␣
 ↪"Range (WLTP) [km]", "Battery capacity [kWh]"]]

recommender = EVRecommender(df)
top3 = recommender.recommend(budget=330000, min_range=350, min_battery=60)
print("Top 3 Recommended EVs:\n", top3)
```

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

    Range (WLTP) [km]  Battery capacity [kWh]
40                580                    75.0
41                567                    75.0
48                549                    77.0
```

[19]:

```python
# Task 5: Perform a two-sample t-test to determine whether the average
# engine power of Tesla and Audi vehicles differ significantly.

import pandas as pd
from scipy.stats import ttest_ind

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

tesla = df[df["Make"].str.lower() == "tesla"]["Engine power [KM]"]
audi = df[df["Make"].str.lower() == "audi"]["Engine power [KM]"]

t_stat, p_value = ttest_ind(tesla, audi, equal_var=False)

print("Tesla Engine Power:\n", tesla.describe())
print("\nAudi Engine Power:\n", audi.describe())
print("\nT-statistic:", t_stat)
print("P-value:", p_value)
```

```
Tesla Engine Power:
 count      7.000000
mean     533.000000
std      184.663658
min      285.000000
25%      426.000000
50%      525.000000
75%      648.500000
max      772.000000
Name: Engine power [KM], dtype: float64

Audi Engine Power:
 count      6.000000
mean     392.000000
std       88.512146
min      313.000000
25%      324.750000
50%      360.000000
75%      467.250000
```

```
max      503.000000
Name: Engine power [KM], dtype: float64
```

```
T-statistic: 1.7939951827297178
P-value: 0.10684105068839565
```

Google Drive Link to Project Video Explanation - https://drive.google.com/drive/folders/14Ch77-29g_9UH9fMBHQ7kSa7JzV-h1ob?usp=sharing

By, Utkarsh Anand