🚗⚡ Electric Vehicle Data Analysis Project

📊 Investigating EV Range, Battery Capacity & Energy Consumption

Welcome! In this project, I analyze **Electric Vehicle (EV) Data** to uncover trends, detect anomalies, and provide actionable insights.

🔹 Loading the Dataset & Rename columns

```
import pandas as pd

# Load FEV dataset
Fev_df = pd.read_excel("FEV-data-Excel.xlsx")

# Rename columns
Fev_df.rename(columns={
    "Minimal price (gross) [PLN]": "Price_PLN",
    "Engine power [KM]": "Engine_Power",
    "Maximum torque [Nm]": "Torque",
    "Battery capacity [kWh]": "Battery_Capacity",
    "Range (WLTP) [km]": "Range_km",
    "mean - Energy consumption [kWh/100 km]": "Energy_Consumption",
}, inplace=True)

# First few rows of dataset
Fev_df.head()
```

	Car full name	Make	Model	Price_PLN	Engine_Power	Torque	Type of brakes	Drive type	Battery_Capacity	Range_km	...	Permissable gross weight [kg]	Maximum load capacity [kg]
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
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
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
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
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 rows × 25 columns

📌 Task 1: Filter EVs Based on Budget & Range

Objective: Identify EVs with: ✅ Budget <= **350,000 PLN** ✅ Minimum range => **400 km**

```
# Filter EVs
filtered_evs = Fev_df[(Fev_df["Price_PLN"] <= 350000) & (Fev_df["Range_km"] >= 400)].reset_index()

# Group by manufacturer & calculate average battery capacity
battery_capacity_avg = filtered_evs.groupby("Make")["Battery_Capacity"].mean().reset_index()

# Select specific columns
filtered_evs = filtered_evs[["Car full name", "Make", "Price_PLN", "Range_km","Engine_Power"]]

# Convert filtered results into markdown format
filtered_evs = filtered_evs.to_markdown()
battery_capacity_avg = battery_capacity_avg.to_markdown()

print(filtered_evs)
```

```
print(battery_capacity_avg)
```

	Car full name	Make	Price_PLN	Range_km	Engine_Power
0	Audi e-tron 55 quattro	Audi	345700	438	360
1	BMW iX3	BMW	282900	460	286
2	Hyundai Kona electric 64kWh	Hyundai	178400	449	204
3	Kia e-Niro 64kWh	Kia	167990	455	204
4	Kia e-Soul 64kWh	Kia	160990	452	204
5	Mercedes-Benz EQC	Mercedes-Benz	334700	414	408
6	Tesla Model 3 Standard Range Plus	Tesla	195490	430	285
7	Tesla Model 3 Long Range	Tesla	235490	580	372
8	Tesla Model 3 Performance	Tesla	260490	567	480
9	Volkswagen ID.3 Pro Performance	Volkswagen	155890	425	204
10	Volkswagen ID.3 Pro S	Volkswagen	179990	549	204
11	Volkswagen ID.4 1st	Volkswagen	202390	500	204

	Make	Battery_Capacity
0	Audi	95
1	BMW	80
2	Hyundai	64
3	Kia	64
4	Mercedes-Benz	80
5	Tesla	68
6	Volkswagen	70.6667

📊 Findings:

1. Filtering **affordable high -range EVs** helps customers choose efficiently.
2. **Battery capacity varies per manufacturer**, affecting range and efficiency.

📌 Task 2: Detecting Energy Consumption Outliers

💡 *EVs with unusually high or low energy consumption using the Interquartile Range (IQR) method.*

```
# Calculate Q1, Q3 & IQR
Q1 = Fev_df["Energy_Consumption"].quantile(0.25)
Q3 = Fev_df["Energy_Consumption"].quantile(0.75)
IQR = Q3 - Q1

# Define outlier thresholds
lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR

# Find outliers
outliers = Fev_df[(Fev_df["Energy_Consumption"] < lower_bound) | (Fev_df["Energy_Consumption"] > upper_bound)]

# Display results
print(outliers[["Car full name", "Make", "Energy_Consumption"]])
```

```
Empty DataFrame
Columns: [Car full name, Make, Energy_Consumption]
Index: []
```

📊 Findings

✅ *Energy consumption across all EVs is fairly consistent—there are no extreme inefficiencies or exceptionally optimized models.* ✅ *No unusual spikes or drops in energy usage mean that manufacturers design EVs with similar efficiency levels.* ✅ *Recommendation: Instead of focusing on energy outliers, we could compare average energy consumption per brand to find the most efficient EVs*

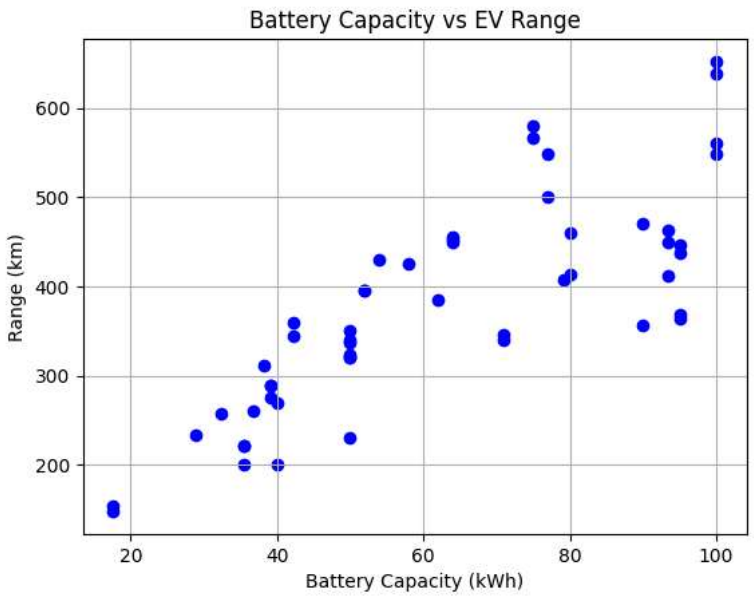
📌 Task 3: Battery Capacity vs Range

🔍 *Does higher battery capacity lead to longer range?*

```
import matplotlib.pyplot as plt

plt.scatter(Fev_df["Battery_Capacity"], Fev_df["Range_km"], color='blue')
plt.xlabel("Battery Capacity (kWh)")
plt.ylabel("Range (km)")
plt.title("Battery Capacity vs EV Range")
```

```
plt.grid(True)
plt.show()
```



📊 Findings:

1. Higher battery capacity generally extends range
2. Some high-capacity EVs have short ranges, indicating inefficiency

📌 Task 4: EV Recommendation System

Enter **budget, range, and battery capacity**, and get **top 3 matching EVs**!

```
from tabulate import tabulate

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

    def recommend(self, budget, min_range, min_battery_capacity):
        # Filter EVs based on criteria
        eligible_evs = self.df[
            (self.df["Price_PLN"] <= budget) &
            (self.df["Range_km"] >= min_range) &
            (self.df["Battery_Capacity"] >= min_battery_capacity)
        ]

        # Select relevant columns & sort by range
        top_evs = eligible_evs.sort_values(by="Range_km", ascending=False).head(3)
        return top_evs[["Car full name", "Make", "Price_PLN", "Range_km", "Battery_Capacity"]]

# Instantiate & use the recommendation system
ev_recommender = EVRecommendation(Fev_df)
recommended_evs = ev_recommender.recommend(350000, 400, 50)

# Display results in a table format using Tabulate
print(tabulate(recommended_evs, headers="keys", tablefmt="github"))
```

```
|    | Car full name             | Make       |   Price_PLN |   Range_km |   Battery_Capacity |
|----|---------------------------|------------|-------------|------------|--------------------|
| 40 | Tesla Model 3 Long Range  | Tesla      |      235490 |        580 |                 75 |
| 41 | Tesla Model 3 Performance | Tesla      |      260490 |        567 |                 75 |
| 48 | Volkswagen ID.3 Pro S     | Volkswagen |      179990 |        549 |                 77 |
```

📊 Findings

✅ Customers can **instantly identify the best EVs** based on range, price, and battery efficiency.

✅ **Sorting by range** ensures optimal battery performance.

✅ Tesla & Volkswagen manufacturers Brand offer **high-performance EVs within budget.**

📌 Task 5: Hypothesis Testing

⚡ Comparing Tesla vs Audi Engine Power using a Two-Sample T-Test

```
from scipy.stats import ttest_ind

# Select engine power data for Tesla & Audi
tesla_power = Fev_df[Fev_df["Make"] == "Tesla"]["Engine_Power"]
audi_power = Fev_df[Fev_df["Make"] == "Audi"]["Engine_Power"]

# Perform t-test
t_stat, p_value = ttest_ind(tesla_power, audi_power)

# Display results
print(f"T-Statistic: {t_stat:.2f}, P-Value: {p_value:.4f}")
```

```
T-Statistic: 1.70, P-Value: 0.1167
```

📊 Findings from the T-Test (Tesla vs Audi Engine Power)

Since the **p-value = 0.11**, which is greater than 0.05, **we fail to reject the null hypothesis.**

This means there is no statistically significant difference in the average engine power between Tesla and Audi— **any variation might be due to random chance rather than an actual performance difference.**

The **T-statistic = 1.70** suggests Tesla's engine power might be slightly higher than Audi's, but not enough to be statistically confirmed.

🔍 **Business Insight:** ✅ Performance differences might not be drastic, making Audi and Tesla comparable in engine power. ✅ Tesla may still focus on acceleration, while Audi might prioritize efficiency and balanced performance. ✅ If considering engine power alone, both brands offer similar levels, so buyers should look at other features like range, battery efficiency, and technology.

🎥 Project Explanation Video

🔗 https://drive.google.com/file/d/1MfrUJW01THwOr-5nYpF7guQqqXidCD8y/view?usp=drivesdk