Task 1: Filter EVs by Criteria and Analyze

a) Filter EVs by Budget and Range:

```
import pandas as pd
df = pd.read_excel(r"C:\Users\ABC\Downloads\FEV-data-excel.xlsx")
filtered_evs = df[(df['Minimal price (gross) [PLN]'] <= 350000) &
(df['Range (WLTP) [km]'] >= 400)]
print(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

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

    Engine power [KM]  Maximum torque [Nm]          Type of brakes
\
0                 360                  664        disc (front + rear)

8                 286                  400        disc (front + rear)

15                204                  395        disc (front + rear)

18                204                  395        disc (front + rear)
```

```
20                  204                   395         disc (front + rear)

22                  408                   760         disc (front + rear)

39                  285                   450         disc (front + rear)

40                  372                   510         disc (front + rear)

41                  480                   639         disc (front + rear)

47                  204                   310  disc (front) + drum (rear)

48                  204                   310  disc (front) + drum (rear)

49                  204                   310  disc (front) + drum (rear)


     Drive type  Battery capacity [kWh]  Range (WLTP) [km]  ...  \
0           4WD                    95.0                438  ...
8    2WD (rear)                    80.0                460  ...
15  2WD (front)                    64.0                449  ...
18  2WD (front)                    64.0                455  ...
20  2WD (front)                    64.0                452  ...
22          4WD                    80.0                414  ...
39   2WD (rear)                    54.0                430  ...
40          4WD                    75.0                580  ...
41          4WD                    75.0                567  ...
47   2WD (rear)                    58.0                425  ...
48   2WD (rear)                    77.0                549  ...
49   2WD (rear)                    77.0                500  ...

    Permissable gross weight [kg]  Maximum load capacity [kg]  \
0                          3130.0                       640.0
8                          2725.0                       540.0
15                         2170.0                       485.0
18                         2230.0                       493.0
20                         1682.0                       498.0
22                         2940.0                       445.0
39                            NaN                         NaN
40                            NaN                         NaN
41                            NaN                         NaN
47                         2270.0                       540.0
48                         2280.0                       412.0
49                         2660.0                       661.0

    Number of seats  Number of doors  Tire size [in]  Maximum speed
[kph]  \
0                 5                5              19
200
8                 5                5              19
```

```
180
15                  5                 5              17
167
18                  5                 5              17
167
20                  5                 5              17
167
22                  5                 5              19
180
39                  5                 5              18
225
40                  5                 5              18
233
41                  5                 5              20
261
47                  5                 5              18
160
48                  5                 5              19
160
49                  5                 5              20
160

    Boot capacity (VDA) [l]  Acceleration 0-100 kph [s]  \
0                     660.0                         5.7
8                     510.0                         6.8
15                    332.0                         7.6
18                    451.0                         7.8
20                    315.0                         7.9
22                    500.0                         5.1
39                    425.0                         5.6
40                    425.0                         4.4
41                    425.0                         3.3
47                    385.0                         7.3
48                    385.0                         7.9
49                    543.0                         8.5

    Maximum DC charging power [kW]  mean - Energy consumption [kWh/100
km]
0                              150
24.45
8                              150
18.80
15                             100
15.40
18                             100
15.90
20                             100
15.70
22                             110
```

```
21.85
39                                150
NaN
40                                150
NaN
41                                150
NaN
47                                100
15.40
48                                125
15.90
49                                125
18.00

[12 rows x 25 columns]
```

b) Group by Manufacturer (Make):

```
grouped_evs = filtered_evs.groupby('Make')
print(grouped_evs.size())

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

c) Calculate Average Battery Capacity:

```
avg_battery = grouped_evs['Battery capacity [kWh]'].mean()
print(avg_battery)

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: Find Outliers in Energy Consumption

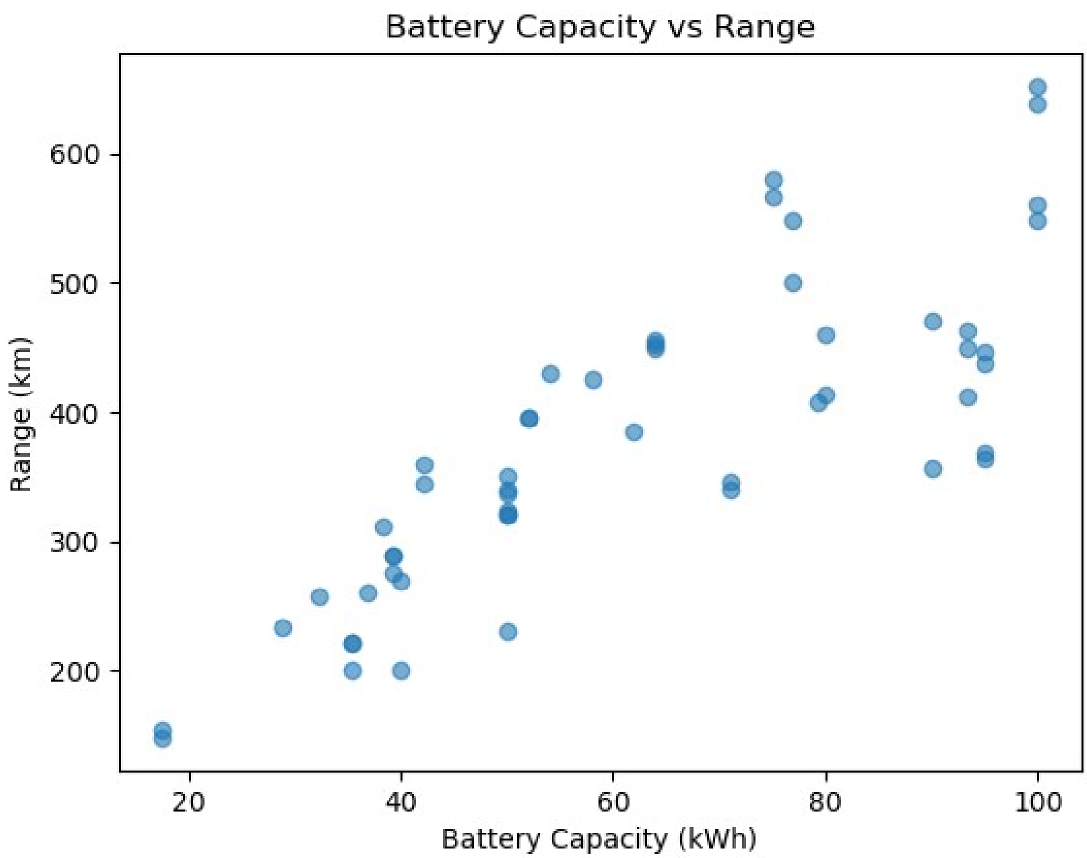
```
from scipy.stats import zscore

df['Z-score'] = zscore(df['mean - Energy consumption [kWh/100 km]'])

# Filter out rows with Z-scores greater than 3 or less than -3
(typical threshold for outliers)
outliers = df[(df['Z-score'] > 3) | (df['Z-score'] < -3)]

print(outliers)
```

```
Empty DataFrame
Columns: [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], Wheelbase [cm], Length
[cm], Width [cm], Height [cm], Minimal empty weight [kg], 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], Z-score]
Index: []

[0 rows x 26 columns]
```

Task 3: Relationship Between Battery Capacity and Range

a) Create a Visualization:

```
import matplotlib.pyplot as plt

# Scatter plot
plt.scatter(df['Battery capacity [kWh]'], df['Range (WLTP) [km]'],
alpha=0.6)
plt.xlabel('Battery Capacity (kWh)')
plt.ylabel('Range (km)')
plt.title('Battery Capacity vs Range')
plt.show()
```



(b): Highlight Insights from Scatter Plot and Correlation Coefficient

```
correlation = df['Battery capacity [kWh]'].corr(df['Range (WLTP)
[km]'])
print("Correlation Coefficient:", correlation)

Correlation Coefficient: 0.8104385771936845
```

Based on the scatter plot provided (Battery Capacity vs Range) and the correlation coefficient (0.8104385771936845), here are the observations and insights:

Key Insights: Strong Positive Correlation:

The correlation coefficient of 0.81 indicates a strong positive relationship between battery capacity and range. This means that electric vehicles with higher battery capacities tend to achieve longer ranges, which aligns with expectations—larger batteries store more energy, enabling greater distances.

Efficient EVs and Trends:

The majority of points on the scatter plot follow an upward trend, confirming that most EVs adhere to this correlation. However, some vehicles exhibit exceptional efficiency or inefficiency:

Highly Efficient EVs: Vehicles with lower battery capacities but longer ranges suggest optimized energy consumption.

Inefficient EVs: Vehicles with higher battery capacities but shorter ranges might indicate heavier weight or less efficient energy usage.

Outliers:

There could be a few outliers visible in the plot:

Outliers with High Battery Capacity but Low Range: These might represent vehicles designed for performance rather than distance.

Outliers with Low Battery Capacity but High Range: These showcase advanced technology or lightweight designs.

General Observation:

The data supports the hypothesis that battery capacity is a significant determinant of range. However, external factors like aerodynamics, vehicle weight, and powertrain efficiency also play crucial roles.

Task 4: Build an EV Recommendation Class

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

    def recommend(self, budget, desired_range, min_battery_capacity):
        recommendations = self.data[(self.data['Minimal price (gross)
[PLN]'] <= budget) &
                                     (self.data['Range (WLTP) [km]']
>= desired_range) &
                                     (self.data['Battery capacity
[kWh]'] >= min_battery_capacity)]
        return recommendations.nlargest(3, 'Range (WLTP) [km]')  #
Return top 3 EVs by range


recommender = EVRecommender(df)
print(recommender.recommend(350000, 400, 50))
```

```
                Car full name        Make                Model  \
40   Tesla Model 3 Long Range       Tesla   Model 3 Long Range
41  Tesla Model 3 Performance       Tesla  Model 3 Performance
48       Volkswagen ID.3 Pro S  Volkswagen          ID.3 Pro S

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

```
48                      179990                204
310

                Type of brakes  Drive type  Battery capacity [kWh]  \
40         disc (front + rear)         4WD                    75.0
41         disc (front + rear)         4WD                    75.0
48  disc (front) + drum (rear)  2WD (rear)                    77.0

    Range (WLTP) [km]  ...  Maximum load capacity [kg]  Number of
seats  \
40                580  ...                         NaN
5
41                567  ...                         NaN
5
48                549  ...                       412.0
5

    Number of doors  Tire size [in]  Maximum speed [kph]  \
40                5              18                  233
41                5              20                  261
48                5              19                  160

    Boot capacity (VDA) [l]  Acceleration 0-100 kph [s]  \
40                    425.0                         4.4
41                    425.0                         3.3
48                    385.0                         7.9

    Maximum DC charging power [kW]  mean - Energy consumption [kWh/100
km]  \
40                             150
NaN
41                             150
NaN
48                             125
15.9

    Z-score
40      NaN
41      NaN
48      NaN

[3 rows x 26 columns]
```

Task 5 : Hypothesis testing

```
from scipy.stats import ttest_ind

tesla_power = df[df['Make'] == 'Tesla']['Engine power [KM]']
```

```
audi_power = df[df['Make'] == 'Audi']['Engine power [KM]']

t_stat, p_value = ttest_ind(tesla_power, audi_power, equal_var=False)
print(f'T-Statistic: {t_stat}, P-Value: {p_value}')

if p_value < 0.05:
    print('There is a significant difference in average engine power
between Tesla and Audi.')
else:
    print('No significant difference in average engine power between
Tesla and Audi.')
```

```
T-Statistic: 1.7939951827297178, P-Value: 0.10684105068839565
No significant difference in average engine power between Tesla and
Audi.
```

Video Link

Watch the Project Video

```
https://drive.google.com/file/d/1loZcDgKfCFg5cOpMcvUJ1pbxnb5JdKMN/
view?usp=drive_link
```