**Notes for Q4.**

Focuses on finding the optimal steps or active minutes for maximizing calorie burn while staying below a maximum heart rate.

**Key Steps in the Code**

 **Data Cleaning and Filtering:**

* The data is cleaned to ensure numerical columns (active\_minutes, steps, calories\_burned, heart\_rate\_avg) are valid.
* Rows with missing values are removed.

 **Activity-Based Analysis:**

* For each workout type and activity type (active\_minutes, steps), scatter plots are generated to visualize the relationship between:
  + Activity type and calories burned.
  + Activity type and average heart rate.

 **Trendline:**

* A trendline is calculated for the relationship between the activity type and calories burned using polynomial fitting.

 **Optimal Range Calculation:**

* Data is binned into ranges of activity type values, and the bin with the highest average calories burned (without exceeding the maximum heart rate) is identified.

 **Visualization:**

* Scatter plots include a horizontal line for the maximum heart rate and a trendline for calories burned.

Improved Version of the Code (Snippet)

|  |
| --- |
| import pandas as pd  import matplotlib.pyplot as plt  import numpy as np  # Constants  MAX\_HEART\_RATE = 170 # User-configurable maximum heart rate  # Data Loading and Cleaning  try:  df = pd.read\_csv('fitness\_tracker\_dataset.csv')  df['date'] = pd.to\_datetime(df['date'], format='%d-%m-%Y', errors='coerce')  for col in ['sleep\_hours', 'active\_minutes', 'steps', 'calories\_burned', 'heart\_rate\_avg']:  df[col] = pd.to\_numeric(df[col], errors='coerce')  df.dropna(inplace=True)  except Exception as e:  print(f"Error in data loading or cleaning: {e}")  exit()  # Analysis: Optimal Steps/Minutes for Calorie Burn by Workout Type  for workout in df['workout\_type'].unique():  df\_workout = df[df['workout\_type'] == workout]    for activity\_type in ['active\_minutes', 'steps']:  plt.figure(figsize=(10, 6))  df\_filtered = df\_workout[df\_workout['heart\_rate\_avg'] <= MAX\_HEART\_RATE]  if df\_filtered.empty:  print(f"No data for {workout} with {activity\_type} under heart rate limit ({MAX\_HEART\_RATE}).")  continue  # Scatter Plot  plt.scatter(df\_filtered[activity\_type], df\_filtered['calories\_burned'], label='Calories Burned', marker='o', alpha=0.7, color='blue')  plt.scatter(df\_filtered[activity\_type], df\_filtered['heart\_rate\_avg'], label='Heart Rate', marker='x', alpha=0.7, color='green')  # Trendline for Calories Burned  z = np.polyfit(df\_filtered[activity\_type], df\_filtered['calories\_burned'], 1)  p = np.poly1d(z)  plt.plot(df\_filtered[activity\_type], p(df\_filtered[activity\_type]), "b--", label="Calories Burn Trend")  # Add max heart rate line  plt.axhline(y=MAX\_HEART\_RATE, color='r', linestyle='--', label=f'Max Heart Rate ({MAX\_HEART\_RATE})')  plt.title(f'{activity\_type.capitalize()} vs. Calories/Heart Rate ({workout})')  plt.xlabel(activity\_type.capitalize())  plt.ylabel('Value')  plt.legend()  plt.grid(True)  plt.tight\_layout()  plt.show()  # Optimal Range Calculation  bins = np.linspace(df\_filtered[activity\_type].min(), df\_filtered[activity\_type].max(), 10)  df\_filtered['activity\_bins'] = pd.cut(df\_filtered[activity\_type], bins=bins, include\_lowest=True)  optimal\_range = df\_filtered.groupby('activity\_bins')['calories\_burned'].mean()  if not optimal\_range.empty:  best\_bin = optimal\_range.idxmax()  print(f"Optimal {activity\_type} range for {workout}: {best\_bin} for maximum calorie burn (Heart rate ≤ {MAX\_HEART\_RATE}).")  else:  print(f"No optimal {activity\_type} range found for {workout} within the heart rate limit.") |

 **Objective:**

* Identify the optimal range of steps or active minutes for maximizing calorie burn without exceeding a specified heart rate.

 **Approach:**

* Filter data based on heart rate.
* Visualize the relationship between activity type and calorie burn/heart rate.
* Determine the range of activity values that maximize calorie burn.

 **Findings:**

* Present notable trends (e.g., optimal steps or active minutes for different workout types).

 **Next Steps:**

* Incorporate dynamic heart rate limits based on user-specific metrics (e.g., age).
* Explore additional factors influencing calorie burn, such as weather or mood.