**Notes for Q5.**

How weather conditions influence workout type, location, and related metrics while including mood-based distributions.

**Key Steps in the Code**

 **Data Cleaning and Preparation:**

* Ensures necessary columns are numeric.
* Converts categorical data (mood, workout\_type, location, weather\_conditions) to category type for efficiency and clarity.

 **Weather-Based Analysis:**

* Analyzes the distribution of workout types and locations for each weather condition.
* Visualizes these distributions using bar charts.

 **Mood-Based Insights:**

* Analyzes mood distribution for each combination of workout type and weather condition, as well as location and weather condition, using pie charts.

 **Metrics Analysis:**

* Computes and visualizes average metrics (calories\_burned, heart\_rate\_avg, active\_minutes, steps, distance\_km) for each combination of workout type and weather condition, as well as location and weather condition.

Improved Version of the Code (Snippet)

|  |
| --- |
| import pandas as pd  import matplotlib.pyplot as plt  # 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', 'distance\_km']:  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()  # Mapping and Conversion  mood\_mapping = {'Tired': 1, 'Neutral': 2, 'Happy': 3}  df['mood\_numerical'] = df['mood'].map(mood\_mapping)  df['mood'] = df['mood'].astype('category')  df['workout\_type'] = df['workout\_type'].astype('category')  df['location'] = df['location'].astype('category')  df['weather\_conditions'] = df['weather\_conditions'].astype('category')  # Weather-Based Analysis  unique\_weather = df['weather\_conditions'].unique()  for weather in unique\_weather:  df\_weather = df[df['weather\_conditions'] == weather]  print(f"\nAnalyzing data for weather condition: {weather}")  # Workout Type Distribution  plt.figure(figsize=(8, 5))  df\_weather['workout\_type'].value\_counts(normalize=True).plot(kind='bar', color='skyblue', title=f"Workout Type Distribution ({weather})")  plt.ylabel("Proportion")  plt.xlabel("Workout Type")  plt.xticks(rotation=45)  plt.tight\_layout()  plt.show()  # Location Distribution  plt.figure(figsize=(8, 5))  df\_weather['location'].value\_counts(normalize=True).plot(kind='bar', color='coral', title=f"Location Distribution ({weather})")  plt.ylabel("Proportion")  plt.xlabel("Location")  plt.xticks(rotation=45)  plt.tight\_layout()  plt.show()  # Metrics and Mood Analysis by Workout Type  for workout in df\_weather['workout\_type'].unique():  df\_workout\_weather = df\_weather[df\_weather['workout\_type'] == workout]  if not df\_workout\_weather.empty:  print(f"Metrics and Mood Analysis for Workout: {workout} under Weather: {weather}")  # Mood Distribution  plt.figure(figsize=(8, 5))  df\_workout\_weather['mood'].value\_counts(normalize=True).plot(kind='bar', color='limegreen', title=f"Mood Distribution ({workout}, {weather})")  plt.ylabel("Proportion")  plt.xlabel("Mood")  plt.tight\_layout()  plt.show()  # Metrics Distribution  metrics = df\_workout\_weather[['calories\_burned', 'heart\_rate\_avg', 'active\_minutes', 'steps', 'distance\_km']].mean()  plt.figure(figsize=(8, 5))  metrics.plot(kind='bar', color='plum', title=f"Metrics Distribution ({workout}, {weather})")  plt.ylabel("Average Value")  plt.tight\_layout()  plt.show()  # Metrics and Mood Analysis by Location  for location in df\_weather['location'].unique():  df\_location\_weather = df\_weather[df\_weather['location'] == location]  if not df\_location\_weather.empty:  print(f"Metrics and Mood Analysis for Location: {location} under Weather: {weather}")  # Mood Distribution  plt.figure(figsize=(8, 5))  df\_location\_weather['mood'].value\_counts(normalize=True).plot(kind='bar', color='teal', title=f"Mood Distribution ({location}, {weather})")  plt.ylabel("Proportion")  plt.xlabel("Mood")  plt.tight\_layout()  plt.show()  # Metrics Distribution  metrics = df\_location\_weather[['calories\_burned', 'heart\_rate\_avg', 'active\_minutes', 'steps', 'distance\_km']].mean()  plt.figure(figsize=(8, 5))  metrics.plot(kind='bar', color='gold', title=f"Metrics Distribution ({location}, {weather})")  plt.ylabel("Average Value")  plt.tight\_layout()  plt.show() |

 **Objective:**

* Analyze the impact of weather on workouts, locations, and mood.

 **Findings:**

* Certain workout types and locations may be preferred under specific weather conditions.
* Weather influences mood, which in turn could impact metrics like calorie burn and activity levels.

 **Next Steps:**

* Use statistical tests to validate observed trends (e.g., ANOVA for mood across weather conditions).
* Correlate weather conditions with metrics to quantify their effect.