**Notes for Q2.**

Address the relationship between sleep hours, mood, and activity levels (steps and active minutes).

**Key Steps in the Code**

 **Data Loading and Cleaning:**

* The dataset is loaded using pd.read\_csv.
* Date conversion and numeric coercion are applied to relevant columns.
* Missing values are handled by dropping rows with NaN.

 **Mapping Mood to Numerical Values:**

* Moods (Tired, Neutral, Happy) are mapped to numerical values (1, 2, 3) for easier plotting and analysis.

 **Analysis Logic:**

* Data is filtered by mood for each activity type (active\_minutes, steps).
* Scatter plots are created to visualize the relationship between sleep hours and activity levels for each mood.
* Trendlines (using linear regression via np.polyfit) are added to highlight patterns.

 **Visualization:**

* Subplots are used to create separate scatter plots for each mood.
* An overall scatter plot is also included, with mood indicated by color.

Improved Version of the Code (Snippet)

|  |
| --- |
| # Mapping mood to numerical values with error handling  mood\_mapping = {'Tired': 1, 'Neutral': 2, 'Happy': 3}  df['mood\_numerical'] = df['mood'].map(mood\_mapping)  if df['mood\_numerical'].isna().any():  print("Warning: Some rows have unmapped mood values. Dropping these rows.")  df.dropna(subset=['mood\_numerical'], inplace=True)  # Activity analysis (loop for both activity types)  for activity\_type in ['active\_minutes', 'steps']:  unique\_moods = sorted(df['mood\_numerical'].unique())  num\_moods = len(unique\_moods)  # Dynamic subplot size  fig, axes = plt.subplots(1, num\_moods, figsize=(6\*num\_moods, 5), squeeze=False)  for i, mood\_val in enumerate(unique\_moods):  df\_mood = df[df['mood\_numerical'] == mood\_val]  axes[0, i].scatter(df\_mood['sleep\_hours'], df\_mood[activity\_type], alpha=0.7, label="Data Points")    # Trendline with error handling  if len(df\_mood) > 1: # Ensure enough data for trendline  z = np.polyfit(df\_mood['sleep\_hours'], df\_mood[activity\_type], 1)  p = np.poly1d(z)  axes[0, i].plot(df\_mood['sleep\_hours'], p(df\_mood['sleep\_hours']), "r--", label="Trendline")  axes[0, i].set\_title(f"Mood: {mood\_val} ({activity\_type})")  axes[0, i].set\_xlabel('Sleep Hours')  axes[0, i].set\_ylabel(activity\_type)  axes[0, i].legend()  # Super title and layout adjustment  fig.suptitle(f"Sleep vs. {activity\_type} (by Mood)", fontsize=16)  plt.tight\_layout()  plt.subplots\_adjust(top=0.85)  plt.show()  # Overall scatter plot with mood color coding  plt.scatter(df['sleep\_hours'], df[activity\_type], c=df['mood\_numerical'], cmap='viridis', alpha=0.7)  plt.title(f"Overall Sleep vs. {activity\_type} (Mood-Color Coded)")  plt.xlabel('Sleep Hours')  plt.ylabel(activity\_type)  cbar = plt.colorbar(label="Mood (1: Tired, 2: Neutral, 3: Happy)")  plt.show() |

 **Objective:**

* Investigate the relationship between sleep hours, mood, and activity levels (steps and active\_minutes).

 **Data Preparation:**

* Mapped moods to numerical values for easier analysis.
* Filtered out rows with invalid or missing mood data.

 **Analysis Process:**

* Analyzed data separately for active\_minutes and steps.
* Visualized the relationship between sleep hours and activity levels for each mood.
* Calculated trendlines to summarize the relationship.

 **Findings:**

* Highlight patterns observed in the scatter plots and trendlines (e.g., increased activity with better moods and sufficient sleep).

 **Improvements and Next Steps:**

* Use more sophisticated regression models for non-linear trends.
* Explore combining activity metrics for a more comprehensive analysis.