Name: Apurbo Sharma

Roll: 220604

Detailed Explanation of my Task:

This Python code is designed to analyze **PPG (Photoplethysmography) signals** for multiple subjects, comparing their physiological responses under **normal** and **stress** conditions. It performs the following tasks step by step:

1. **Data Input and Preprocessing:**
   - Reads two datasets (data_normal.csv and data_stress.csv) containing PPG signal measurements for different subjects.
   - Extracts the **PPG signal data** for each subject, isolating relevant numerical columns while removing missing values to ensure clean signals.
2. **Abnormality Detection:**
   - Implements a custom function (detect_abnormalities) to identify abrupt changes in the PPG signal based on a **threshold** (default is 50).
     - An abnormality is defined as a point where the difference between consecutive signal values exceeds the threshold.
   - Applies this function separately to each subject's **normal** and **stress** PPG signals to detect abnormalities in both conditions.
3. **Visualization of Results:**
   - Creates a **line plot** for each subject to visually compare the PPG signals under normal and stress conditions:
     - The **normal PPG signal** is plotted in blue.
     - The **stress PPG signal** is plotted in red.
   - Highlights detected abnormalities using **scatter points**:
     - Abnormalities in the normal signal are marked in black.

        - Abnormalities in the stress signal are marked in orange.

    - The number of abnormalities in each condition is **calculated and displayed** on the plot title and annotated in the figure.

4. **Insights and Interpretation:**

    - By analyzing these plots, the user can visually and quantitatively assess:

        - Differences in signal behavior between normal and stress conditions.

        - The prevalence of abnormalities, such as sudden spikes or drops, which may indicate physiological changes caused by stress.

        - Patterns in PPG signal variability and irregularities, potentially useful for research in stress monitoring, mental health, or cardiovascular analysis.

5. **Iterative Analysis for All Subjects:**

    - The code loops through all unique subjects in the dataset, performing the above steps for each subject individually.

    - This enables a subject-by-subject comparison of PPG signals and abnormalities under different conditions.

Overall Purpose:

The code provides a **tool for analyzing physiological data** by detecting and highlighting signal abnormalities in PPG data under different conditions. It combines statistical analysis and visualization to offer insights into how stress impacts PPG signals and identifies regions where these signals deviate unexpectedly.

Source Code:

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

# Reading the CSV files
data_normal = pd.read_csv('data_normal.csv')
data_stress = pd.read_csv('data_stress.csv')

# Get all unique subject IDs
subjects = data_normal['subject ID'].unique()

# Function to detect abnormalities (e.g., sudden spikes or drops)
def detect_abnormalities(signal, threshold=50):
    """Detects abnormalities where the signal changes abruptly by a certain threshold."""
    abnormalities = []
    for i in range(1, len(signal)):
        if abs(signal[i] - signal[i - 1]) > threshold:
            abnormalities.append(i)
    return abnormalities



# Loop through each subject and plot the PPG signals with abnormalities marked
for subject in subjects:
    # Extract PPG data for the current subject from both datasets
    ppg_normal = data_normal[data_normal['subject ID'] == subject].iloc[:, 2:].dropna(axis=1).values.flatten()
    ppg_stress = data_stress[data_stress['subject ID'] == subject].iloc[:, 2:].dropna(axis=1).values.flatten()

    # Create a time axis for both signals
    time_normal = range(len(ppg_normal))
    time_stress = range(len(ppg_stress))

    # Detect abnormalities in the signals
    abnormalities_normal = detect_abnormalities(ppg_normal)
    abnormalities_stress = detect_abnormalities(ppg_stress)

    # Count the number of abnormalities
    num_abnormalities_normal = len(abnormalities_normal)
    num_abnormalities_stress = len(abnormalities_stress)

    # Plot the signals
    plt.figure(figsize=(12, 6))
    plt.plot(time_normal, ppg_normal, label='Normal PPG', color='blue', alpha=0.7)
    plt.plot(time_stress, ppg_stress, label='Stress PPG', color='red', alpha=0.7)
```

```
# Mark abnormalities
    plt.scatter([time_normal[i] for i in abnormalities_normal],
                [ppg_normal[i] for i in abnormalities_normal],
                color='black', label='Abnormalities (Normal)', zorder=5)

    plt.scatter([time_stress[i] for i in abnormalities_stress],
                [ppg_stress[i] for i in abnormalities_stress],
                color='orange', label='Abnormalities (Stress)', zorder=5)

    # Add titles and labels
    plt.title(f'PPG Signal Comparison for {subject}\n'
              f'Abnormalities: Normal ({num_abnormalities_normal}), Stress ({num_abnormalities_stress})')
    plt.xlabel('Time (arbitrary units)')
    plt.ylabel('PPG Amplitude')
    plt.legend()
    plt.grid(True)

    # Add text annotations for the counts
    plt.text(0.05, 0.95, f'Normal Abnormalities: {num_abnormalities_normal}',
             transform=plt.gca().transAxes, color='blue', fontsize=10, verticalalignment='top')
    plt.text(0.05, 0.90, f'Stress Abnormalities: {num_abnormalities_stress}',
             transform=plt.gca().transAxes, color='red', fontsize=10, verticalalignment='top')
```

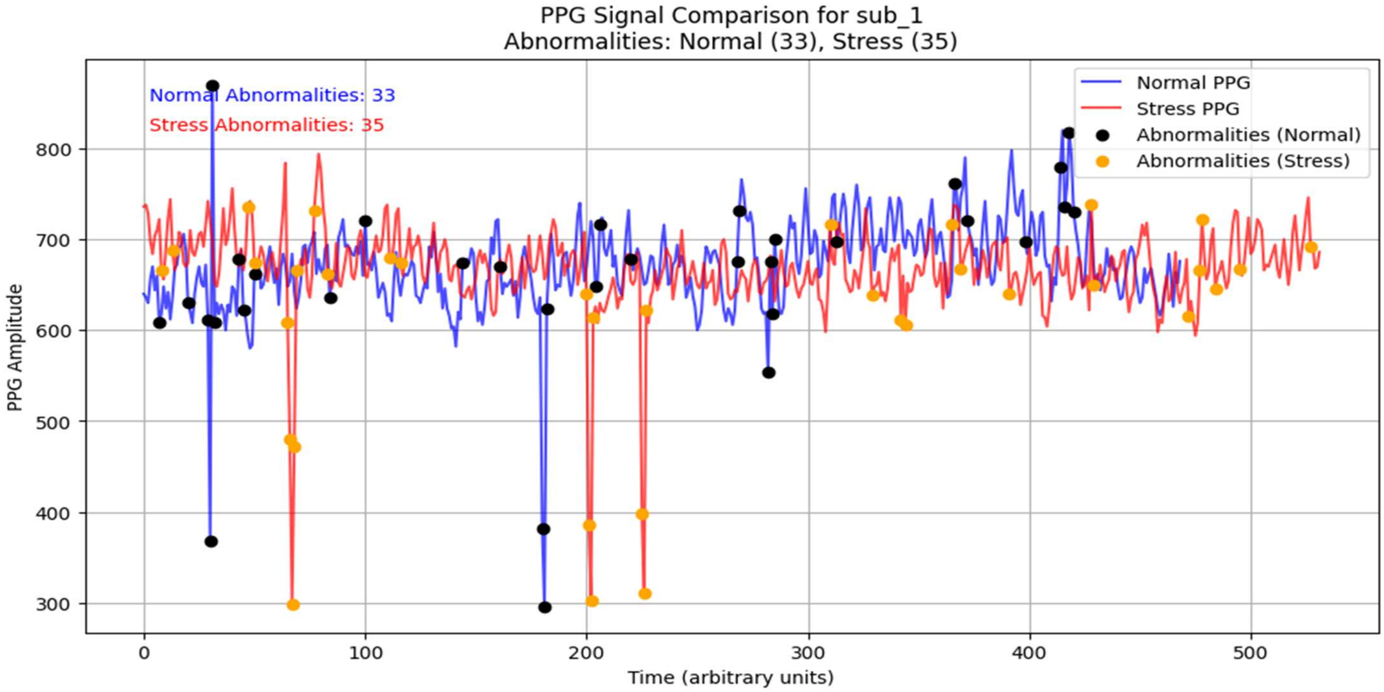
```
 # Display the plot

   plt.show()
```

Input:

attached with the mail

Output:


PPG Signal Comparison for sub_1
Abnormalities: Normal (33), Stress (35)
Normal Abnormalities: 33
Stress Abnormalities: 35
Normal PPG
Stress PPG
Abnormalities (Normal)
Abnormalities (Stress)
PPG Amplitude
Time (arbitrary units)