Title: PPG Signal Analysis with Abnormality Detection

Objectives: This code performs the analysis of a synthetic PPG signal, detecting normal and abnormal peaks and valleys. It also calculates Heart Rate (BPM), Pulse Rate Variability (PRV), Amplitude Variation, and Peak-to-Valley Ratio.

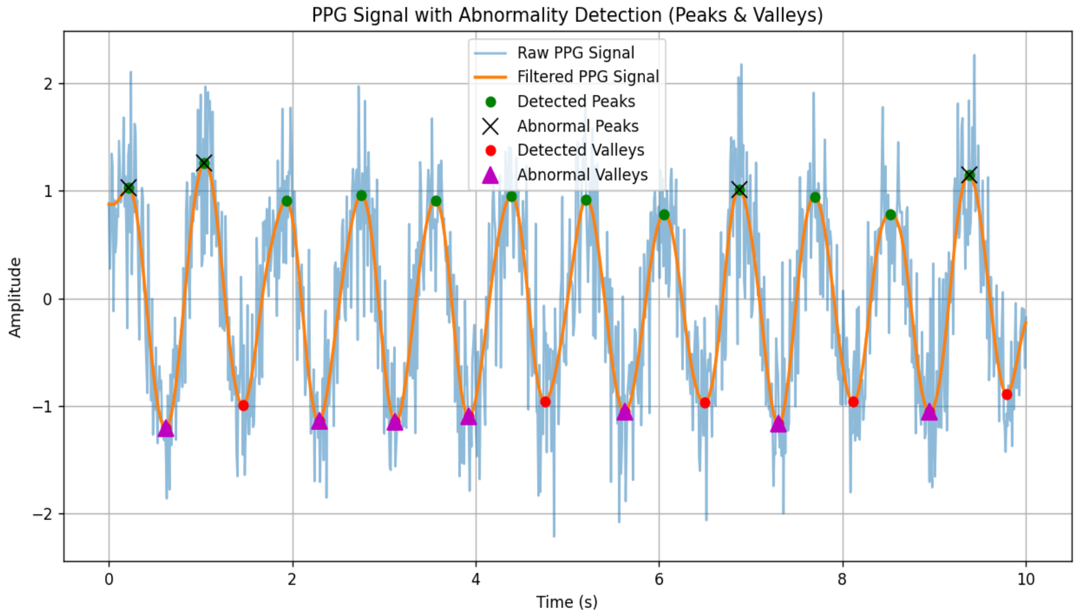
Source code:

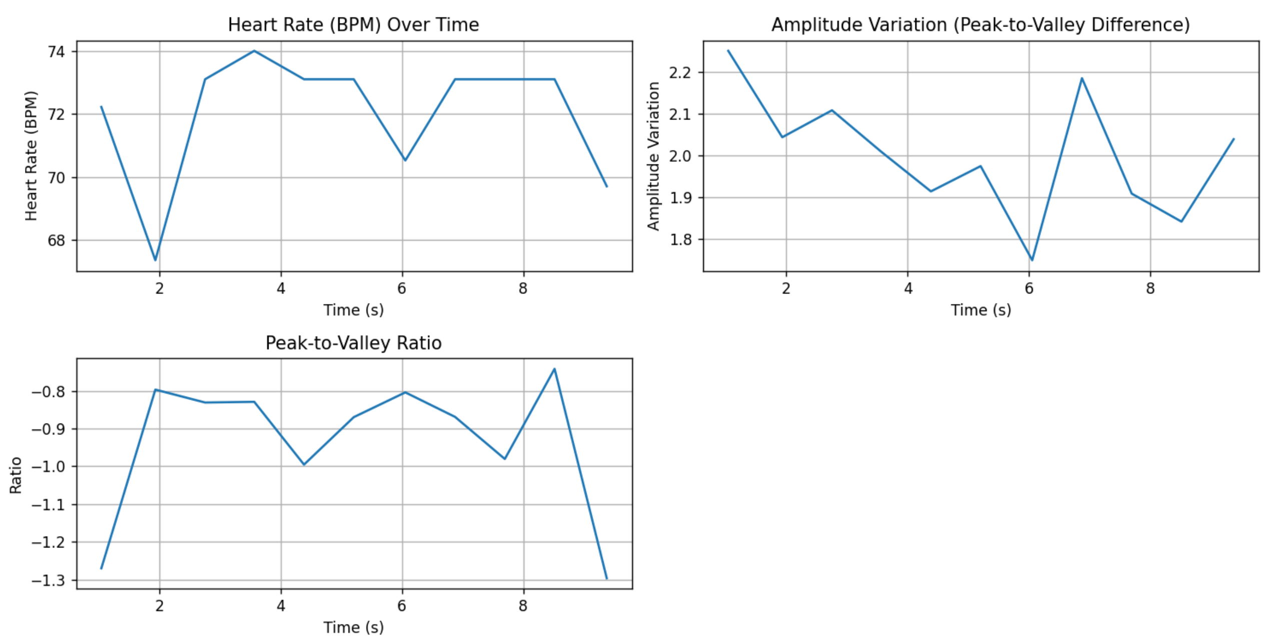
```python
import numpy as np
import matplotlib.pyplot as plt
from scipy.signal import find_peaks, butter, filtfilt

# Generate synthetic PPG data (replace with your actual data)
np.random.seed(0)
time = np.linspace(0, 10, 1000)  # 10 seconds at 100 Hz sampling rate
ppg_signal = np.sin(2 * np.pi * 1.2 * time) + 0.5 *
np.random.normal(size=len(time))

# Butterworth filter to smooth the signal
def butter_lowpass_filter(data, cutoff, fs, order=5):
    nyquist = 0.5 * fs
    normal_cutoff = cutoff / nyquist
    b, a = butter(order, normal_cutoff, btype='low', analog=False)
    y = filtfilt(b, a, data)
    return y

# Filter parameters
fs = 100  # Sampling frequency in Hz
cutoff = 3  # Cutoff frequency in Hz
filtered_ppg = butter_lowpass_filter(ppg_signal, cutoff, fs)

# Detect peaks and valleys
peaks, _ = find_peaks(filtered_ppg, height=0.5, distance=fs//2)
valleys, _ = find_peaks(-filtered_ppg, height=0.5, distance=fs//2)

# Abnormality detection (e.g., irregular intervals, abnormal peak heights)
peak_heights = filtered_ppg[peaks]
valley_heights = filtered_ppg[valleys]

# Abnormal peaks: Threshold for high spikes
abnormal_peaks = peaks[peak_heights > 1.0]
# Abnormal valleys: Threshold for deep valleys
abnormal_valleys = valleys[valley_heights < -1.0]
```

```python
# Calculate heart rate (in beats per minute)
peak_times = time[peaks]
ibi = np.diff(peak_times)  # Inter-beat interval in seconds
hr = 60 / ibi  # Heart rate in beats per minute (BPM)

# Calculate the amplitude variation (peak-to-peak difference)
amplitude_variation = filtered_ppg[peaks] - filtered_ppg[valleys]

# Peak-to-Valley ratio
peak_valley_ratio = filtered_ppg[peaks] / filtered_ppg[valleys]

# Plot the results
plt.figure(figsize=(12, 6))

# Raw and filtered signal plot
plt.plot(time, ppg_signal, label="Raw PPG Signal", alpha=0.5)
plt.plot(time, filtered_ppg, label="Filtered PPG Signal", linewidth=2)

# Normal and abnormal peaks
plt.plot(time[peaks], filtered_ppg[peaks], "go", label="Detected Peaks")
plt.plot(time[abnormal_peaks], filtered_ppg[abnormal_peaks], "kx",
label="Abnormal Peaks", markersize=10)

# Normal and abnormal valleys
plt.plot(time[valleys], filtered_ppg[valleys], "ro", label="Detected Valleys")
plt.plot(time[abnormal_valleys], filtered_ppg[abnormal_valleys], "m^",
label="Abnormal Valleys", markersize=10)

# Title and labels
plt.title("PPG Signal with Abnormality Detection (Peaks & Valleys)")
plt.xlabel("Time (s)")
plt.ylabel("Amplitude")
plt.legend()
plt.grid()

plt.show()

# Display the calculated features
plt.figure(figsize=(12, 6))

# Heart Rate Plot
plt.subplot(2, 2, 1)
plt.plot(peak_times[1:], hr, label="Heart Rate (BPM)")
plt.title("Heart Rate (BPM) Over Time")
plt.xlabel("Time (s)")
```

```python
plt.ylabel("Heart Rate (BPM)")
plt.grid()

# Amplitude Variation Plot
plt.subplot(2, 2, 2)
plt.plot(time[peaks][1:], amplitude_variation[1:], label="Amplitude Variation")
plt.title("Amplitude Variation (Peak-to-Valley Difference)")
plt.xlabel("Time (s)")
plt.ylabel("Amplitude Variation")
plt.grid()

# Peak-to-Valley Ratio Plot
plt.subplot(2, 2, 3)
plt.plot(time[peaks][1:], peak_valley_ratio[1:], label="Peak-to-Valley Ratio")
plt.title("Peak-to-Valley Ratio")
plt.xlabel("Time (s)")
plt.ylabel("Ratio")
plt.grid()

# Main plot
plt.tight_layout()
plt.show()

# Print features
print(f"Calculated Heart Rate (first 5 values): {hr[:5]}")
print(f"Amplitude Variation (first 5 values): {amplitude_variation[:5]}")
print(f"Peak-to-Valley Ratio (first 5 values): {peak_valley_ratio[:5]}")
```

Output:


PPG Signal with Abnormality Detection (Peaks & Valleys)
Raw PPG Signal
Filtered PPG Signal
Detected Peaks
Abnormal Peaks
Detected Valleys
Abnormal Valleys
Amplitude
Time (s)


Heart Rate (BPM) Over Time
Heart Rate (BPM)
Time (s)
Amplitude Variation (Peak-to-Valley Difference)
Amplitude Variation
Time (s)
Peak-to-Valley Ratio
Ratio
Time (s)

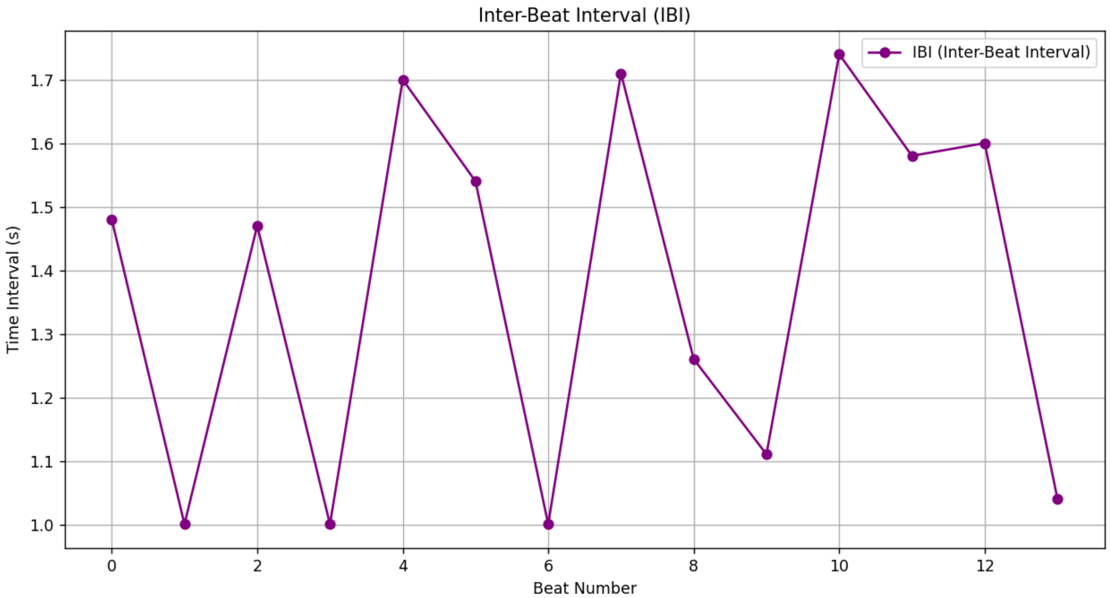
Title: Inter-Beat Interval (IBI) Calculation from PPG Signal.

Objectives: The objective of this task is to detect peaks in a PPG signal, calculate the Inter-Beat Interval (IBI), which represents the time difference between consecutive heartbeats, and visualize the IBI values over time for analysis.

Source Code:

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.signal import find_peaks

# Synthetic PPG Signal (replace with actual data if needed)
np.random.seed(0)
time = np.linspace(0, 20, 2000)  # 20 seconds at 100 Hz sampling rate
ppg_signal = np.sin(2 * np.pi * 1.2 * time) + 0.5 *
np.random.normal(size=len(time))

# Peak Detection
peaks, _ = find_peaks(ppg_signal, height=0.5, distance=100)  # Detecting Peaks

# Inter-Beat Interval (IBI) Calculation (Time difference between consecutive
peaks)
ibi_intervals = np.diff(time[peaks])  # Time difference between consecutive peaks

# Plot IBI (Inter-Beat Interval)
plt.figure(figsize=(12, 6))
plt.plot(ibi_intervals, label="IBI (Inter-Beat Interval)", marker='o',
linestyle='-', color="purple")
plt.title("Inter-Beat Interval (IBI)")
plt.xlabel("Beat Number")
plt.ylabel("Time Interval (s)")
plt.legend()
plt.grid()
plt.show()

# Print Inter-Beat Interval values
print(f"Calculated Inter-Beat Intervals (IBI): {ibi_intervals[:5]} seconds")
```

Output:



Inter-Beat Interval (IBI)
IBI (Inter-Beat Interval)
Time Interval (s)
1.0
1.1
1.2
1.3
1.4
1.5
1.6
1.7
Beat Number
0
2
4
6
8
10
12