// JavaScript code for heart rate assignment to calculate minimum, maximum, and median beats for minute?

To calculate the minimum, maximum, and median beats per minute (BPM) in JavaScript, you can create a function that takes an array of BPM values as input and performs the necessary calculations.

// Function to calculate minimum, maximum, and median BPM

To calculateBPMStats function takes an array of BPM values as input.

It first checks if the array is empty and returns null values for minimum, maximum, and median if it is.

The array is then sorted in ascending order.

The minimum BPM is the first value in the sorted array.

The maximum BPM is the last value in the sorted array.

The median BPM is calculated based on whether the array length is even or odd.

You can replace the bpmValues array with your own array of BPM values for testing.

function calculateBPMStats(bpmArray) {

// Ensure the array is not empty

if (bpmArray.length === 0) {

return {

minBPM: null,

maxBPM: null,

medianBPM: null,

};

}

// Sort the array in ascending order

const sortedBPM = [...bpmArray]. sort ((a, b) => a - b);

// Calculate minimum BPM

const minBPM = sortedBPM[0];

// Calculate maximum BPM

const maxBPM = sortedBPM[sortedBPM.length - 1];

// Calculate median BPM

let medianBPM;

const mid = Math.floor(sortedBPM.length / 2);

if (sortedBPM.length % 2 === 0) {

// If array length is even, take the average of the two middle values

medianBPM = (sortedBPM[mid - 1] + sortedBPM[mid]) / 2;

} else {

// If array length is odd, take the middle value

medianBPM = sortedBPM[mid];

}

return {

minBPM,

maxBPM,

medianBPM,

};

}

// Example usage

const bpmValues = [80, 95, 75, 110, 120, 88, 100];

const bpmStats = calculateBPMStats(bpmValues);

console.log('Minimum BPM:', bpmStats.minBPM);

console.log('Maximum BPM:', bpmStats.maxBPM);

console.log('Median BPM:', bpmStats.medianBPM);