

ECG Data Analysis MATLAB course project

projectek122.m

```
%% Step 1 Load our data
RestingData = load("Resting.mat");
RestingData = RestingData.b1;
ExerciseData = load("Exercise.mat");
ExerciseData = ExerciseData.b1;
BoxBreathingData = load("BoxBreathing.mat");
BoxBreathingData = BoxBreathingData.b1;

%% Step 2: Plot our Signals
% 4) Plot 10s of Resting ECG signals
figure;
plot(RestingData(:, 1), RestingData(:, 2), 'LineWidth', 2);
xlabel('time (s)', 'FontSize', 16);
ylabel('ECG (mV)', 'FontSize', 16);
xlim([10, 20]);
ylim([-0.2, 0.8]);
title('Resting', 'FontSize', 18);
saveas(gcf, 'Resting_ECG_Plot.png'); % Save as PNG
%Plot 10s of BoxBreathing ECG signals
figure;
plot(BoxBreathingData(:, 1), BoxBreathingData(:, 2), 'LineWidth', 2);
xlabel('time (s)', 'FontSize', 16);
ylabel('ECG (mV)', 'FontSize', 16);
xlim([10, 20]);
ylim([-0.2, 0.8]);
title('BoxBreathing', 'FontSize', 18);
saveas(gcf, 'BoxBreathing_ECG_Plot.png');
%Plot 10s of Exercise ECG signals
figure;
plot(ExerciseData(:, 1), ExerciseData(:, 2), 'LineWidth', 2);
xlabel('time (s)', 'FontSize', 16);
ylabel('ECG (mV)', 'FontSize', 16);
xlim([10, 20]);
ylim([-0.3, 0.8]);
title('Exercise', 'FontSize', 18);
saveas(gcf, 'Exercise_ECG_Plot.png'); % Save as PNG

%3x1 Subplot for all 3 datas
figure;
% Plot Resting
subplot(3, 1, 1);
plot(RestingData(:, 1), RestingData(:, 2), 'LineWidth', 2);
xlim([120, 180]);
ylim([-0.2, 0.8]);
title('Resting', 'FontSize', 18);
saveas(gcf, 'Resting_ECG_Plot_120_180s.png')
```

```

% Plot Exercise
subplot(3, 1, 2);
plot(ExerciseData(:, 1), ExerciseData(:, 2), 'LineWidth', 2);
xlim([120, 180]);
ylim([-0.3, 0.8]);
title('Exercise', 'FontSize', 18);
saveas(gcf, 'Exercise_ECG_Plot_120_180s.png');
% Plot BoxBreathing
subplot(3, 1, 3);
plot(BoxBreathingData(:, 1), BoxBreathingData(:, 2), 'LineWidth', 2);
xlim([120, 180]);
ylim([-0.2, 0.8]);
title('Box Breathing', 'FontSize', 18);
saveas(gcf, 'BoxBreathing_ECG_Plot_120_180s.png');

%% Step 4: Use findpeaks() to plot the peaks, using default settings
% Resting Original ECG plot
figure;
subplot(2, 1, 1);
plot(RestingData(:, 1), RestingData(:, 2), 'LineWidth', 2);
xlim([120, 130]);
ylim([-0.2, 0.8]);
title('Resting', 'FontSize', 18);
saveas(gcf, 'Exercise_ECG_Plot_120_130s.png');
% Resting ECG with Peaks plot
subplot(2, 1, 2);
[peaksResting, locsResting] = findpeaks(RestingData(:, 2), RestingData(:, 1));
plot(RestingData(:, 1), RestingData(:, 2), 'LineWidth', 2);
hold on;
plot(locsResting, peaksResting, 'v', 'MarkerFaceColor', 'b', 'LineStyle',
'none');
hold off;
xlim([120, 130]);
ylim([-0.2, 0.8]);
title('Resting with Peaks', 'FontSize', 18);
grid on;
saveas(gcf, 'Resting_ECG_with_Peaks_120_130s.png');

% Exercise Original ECG plot
figure;
subplot(2, 1, 1);
plot(ExerciseData(:, 1), ExerciseData(:, 2), 'LineWidth', 2);
xlim([120, 130]);
ylim([-0.3, 0.8]);
title('Exercise', 'FontSize', 18);
saveas(gcf, 'Exercise_ECG_Plot_120_130s.png');
% ECG with Peaks plot
subplot(2, 1, 2);

```

```

[peaksExercise, locsExercise] = findpeaks(ExerciseData(:, 2), ExerciseData(:,
1));
plot(ExerciseData(:, 1), ExerciseData(:, 2), 'LineWidth', 2);
hold on;
plot(locsExercise, peaksExercise, 'v', 'MarkerFaceColor', 'b', 'LineStyle',
'none');
hold off;
xlim([120, 130]);
ylim([-0.3, 0.8]);
title('Exercise with Peaks', 'FontSize', 18);
grid on;
saveas(gcf, 'Exercise_ECG_with_Peaks_120_130s.png');

```

```

%% Step 5: Redo Step 4 with the appropriate settings of findpeaks
% RLocs is the time (or x-value) of the peaks, not index
% MinPeakHeight for resting and exercise

```

```

% Resting
figure;
subplot(2, 1, 1);
plot(RestingData(:, 1), RestingData(:, 2), 'LineWidth', 2);
xlim([120, 130]);
ylim([-0.2, 0.8]);
title('Resting', 'FontSize', 18);
saveas(gcf, 'Exercise_ECG_Plot_120_130s.png');
% Resting with R-Peaks
minPeakHeightResting = 0.4;
[restingRPeaks, restingRLocs] = findpeaks(RestingData(:, 2), RestingData(:, 1),
'MinPeakHeight', minPeakHeightResting);
subplot(2, 1, 2);
plot(RestingData(:, 1), RestingData(:, 2), 'LineWidth', 2);
hold on;
plot(restingRLocs, restingRPeaks, 'v', 'MarkerFaceColor', 'b', 'LineStyle',
'none');
hold off;
xlim([120, 130]);
ylim([-0.2, 0.8]);
title('Resting with R-Peaks', 'FontSize', 18);
grid on;
saveas(gcf, 'Resting_with_R_Peaks.png');

```

```

% Exercise
figure;
subplot(2, 1, 1);
plot(ExerciseData(:, 1), ExerciseData(:, 2), 'LineWidth', 2);
xlim([121, 131]);
ylim([-0.3, 0.8]);

```

```

title('Exercise', 'FontSize', 18);
saveas(gcf, 'Exercise_ECG_Plot_120_130s.png');
% Exercise with R-Peaks
minPeakHeightExercise = 0.4;
[exerciseRPeaks, exerciseRLocs] = findpeaks(ExerciseData(:, 2), ExerciseData(:,
1), 'MinPeakHeight', minPeakHeightExercise);
subplot(2, 1, 2);
plot(ExerciseData(:, 1), ExerciseData(:, 2), 'LineWidth', 2);
hold on;
plot(exerciseRLocs, exerciseRPeaks, 'v', 'MarkerFaceColor', 'b', 'LineStyle',
'none');
hold off;
xlim([121, 131]);
ylim([-0.3, 0.8]);
title('Exercise with R-Peaks', 'FontSize', 18);
grid on;
saveas(gcf, 'Exercise_with_R_Peaks.png');

% BoxBreathing
figure;
subplot(2, 1, 1);
plot(BoxBreathingData(:, 1), BoxBreathingData(:, 2), 'LineWidth', 2);
xlim([120, 130]);
ylim([-0.2, 0.8]);
title('Box Breathing', 'FontSize', 18);
saveas(gcf, 'BoxBreathing_ECG_Plot_120_130s.png');
% BoxBreathing with R-Peaks
minPeakHeightBoxBreathing = 0.4;
[boxBreathingRPeaks, boxBreathingRLocs] = findpeaks(BoxBreathingData(:, 2),
BoxBreathingData(:, 1), 'MinPeakHeight', minPeakHeightBoxBreathing);
subplot(2, 1, 2);
plot(BoxBreathingData(:, 1), BoxBreathingData(:, 2), 'LineWidth', 2);
hold on;
plot(boxBreathingRLocs, boxBreathingRPeaks, 'v', 'MarkerFaceColor', 'b',
'LineStyle', 'none');
hold off;
xlim([122, 130]);
ylim([-0.25, 0.8]);
title('Box Breathing with R-Peaks', 'FontSize', 18);
grid on;
saveas(gcf, 'BoxBreathing_with_R_Peaks.png');

%% Step 6: A more advanced view
%Plot heartbeat center about the trigger(R-peaks) for each condition
samplesBefore = 100;
samplesAfter = 100;
numSamples = samplesBefore + samplesAfter + 1;
% Exercise heartbeat = Red

```

```

exerciseHeartbeats = zeros(length(exerciseRLocs), numSamples);
figure;
hold on;
for i = 1:length(exerciseRLocs)
    peakIndex = find(ExerciseData(:, 1) == exerciseRLocs(i), 1, 'first');
    startIndex = max(1, peakIndex - samplesBefore);
    endIndex = min(size(ExerciseData, 1), peakIndex + samplesAfter);
    segment = ExerciseData(startIndex:endIndex, 2);
    paddingBefore = max(0, samplesBefore - (peakIndex - startIndex));
    paddingAfter = max(0, (peakIndex + samplesAfter) - endIndex);
    heartbeatSegment = [nan(paddingBefore, 1); segment; nan(paddingAfter, 1)];
    exerciseHeartbeats(i, :) = heartbeatSegment;
    plot(0:200, heartbeatSegment, 'Color', [1 0 0 0.5]);
end
title('Exercise Heartbeats Centered on R-Peaks');
xlabel('count');
ylabel('mV');
xlim([0 200]);
ylim([-0.3 0.7]);
hold off;
saveas(gcf, 'Exercise_Heartbeats_Centered_Count.png');
% Resting Heartbeat = Blue
restingHeartbeats = zeros(length(restingRLocs), numSamples);
figure;
hold on;
for i = 1:length(restingRLocs)
    peakIndex = find(RestingData(:, 1) == restingRLocs(i), 1, 'first');
    startIndex = max(1, peakIndex - samplesBefore);
    endIndex = min(size(RestingData, 1), peakIndex + samplesAfter);
    segment = RestingData(startIndex:endIndex, 2);
    paddingBefore = max(0, samplesBefore - (peakIndex - startIndex));
    paddingAfter = max(0, (peakIndex + samplesAfter) - endIndex);
    heartbeatSegment = [nan(paddingBefore, 1); segment; nan(paddingAfter, 1)];
    restingHeartbeats(i, :) = heartbeatSegment;
    plot(0:200, heartbeatSegment, 'Color', [0 0 1 0.5]);
end
title('Resting Heartbeats Centered on R-Peaks');
xlabel('count');
ylabel('mV');
xlim([0 200]);
ylim([-0.15 0.6]);
hold off;
saveas(gcf, 'Resting_Heartbeats_Centered_Count.png');
% BoxBreathing heartbeat = Green
boxBreathingHeartbeats = zeros(length(boxBreathingRLocs), numSamples);
figure;
hold on;
for i = 1:length(boxBreathingRLocs)

```

```

    peakIndex = find(BoxBreathingData(:, 1) == boxBreathingRLocs(i), 1,
'first');
    startIndex = max(1, peakIndex - samplesBefore);
    endIndex = min(size(BoxBreathingData, 1), peakIndex + samplesAfter);
    segment = BoxBreathingData(startIndex:endIndex, 2);
    paddingBefore = max(0, samplesBefore - (peakIndex - startIndex));
    paddingAfter = max(0, (peakIndex + samplesAfter) - endIndex);
    heartbeatSegment = [nan(paddingBefore, 1); segment; nan(paddingAfter, 1)];
    boxBreathingHeartbeats(i, :) = heartbeatSegment;
    plot(0:200, heartbeatSegment, 'Color', [0 1 0 0.5]);
end
title('BoxBreathing Heartbeats Centered on R-Peaks');
xlabel('count');
ylabel('mV');
xlim([0 200]);
ylim([-0.15 0.6]);
hold off;
saveas(gcf, 'BoxBreathing_Heartbeats_Centered_Count.png');

```

```

%% Step 7: Plot a histogram using histogram and boxplot using boxchart
%Histogram of resting vs exercise frequency in bpm (subject's heart rate)
timeWindowStart = 150;
timeWindowEnd = 250;
restingRPeaksInWindow = restingRLocs(restingRLocs >= timeWindowStart &
restingRLocs <= timeWindowEnd);
exerciseRPeaksInWindow = exerciseRLocs(exerciseRLocs >= timeWindowStart &
exerciseRLocs <= timeWindowEnd);
% Way 1: Compute resting intervals numerically
restingIntervals = zeros(length(restingRPeaksInWindow) - 1, 1);
for i = 1:length(restingRPeaksInWindow) - 1
    restingIntervals(i) = restingRPeaksInWindow(i + 1) -
restingRPeaksInWindow(i);
end
% Way 2: Compute exercise interval using diff
exerciseIntervals = diff(exerciseRPeaksInWindow);
% Compute bpm, or frequency of heart rate
restingHeartRate = 60 ./ restingIntervals;
exerciseHeartRate = 60 ./ exerciseIntervals;
% Using histogram plot a histogram of this frequency for all the events in the
limited
figure;
histogram(restingHeartRate, 'BinWidth',0.5, 'Normalization','probability');
hold on
histogram(exerciseHeartRate, 'BinWidth',0.5, 'Normalization','probability');
hold off;
xlabel('Rate (bpm)');
ylabel('Probability');

```

```

title(['Histogram of Heart Rate      ', 'Time range: ',
num2str(timeWindowStart), '-', num2str(timeWindowEnd)]);
legend({'Resting', 'Exercise'}, "Location", "northwest");
saveas(gcf, 'HeartRate_Histogram_Exercise_vs_Resting.png');
% box chart
% Boxchart of heart rates Resting vs Exercise
combinedHeartRates = [restingHeartRate; exerciseHeartRate];
group = [repmat({'Resting'}, length(restingHeartRate), 1);
        repmat({'Exercise'}, length(exerciseHeartRate), 1)];
group = categorical(group);
figure
boxchart(group, combinedHeartRates);
xlabel('Dataset');
ylabel('Rate (bpm)');
title('Boxchart of Heart Rates: Resting vs Exercise');
saveas(gcf, 'Boxplot Heart_Rate_Comparison_Resting_vs_Exercise.png');

%% Step 8: Compute the derivative of different phases of the signal
% Compute the derivative and speed of Resting R & T pulses
% Define minimum and maximum peak heights for T peaks (since R peaks are
pre-defined)
minPeakHeightRestingT = 0.15;
maxPeakHeightRestingT = 0.4;
[allTPeaks, allTLocs] = findpeaks(RestingData(:, 2), RestingData(:, 1));
validTIndices = (allTPeaks > minPeakHeightRestingT) & (allTPeaks <
maxPeakHeightRestingT);
restingTPeaks = allTPeaks(validTIndices);
restingTLocs = allTLocs(validTIndices);
% 10 samples before and after each R and T pulse
samplesAroundPulse = 10;
rSpeeds = zeros(1, length(restingRLocs));
tSpeeds = zeros(1, length(restingTLocs));
% Loop through each R pulse to compute the speed
for i = 1:length(restingRLocs)
    rIndex = find(RestingData(:, 1) == restingRLocs(i), 1);
    if rIndex > samplesAroundPulse && rIndex + samplesAroundPulse <=
size(RestingData, 1)
        segment = RestingData(rIndex - samplesAroundPulse:rIndex +
samplesAroundPulse, 2);
        voltageChange = max(segment) - min(segment);
        timeInterval = RestingData(rIndex + samplesAroundPulse, 1) -
RestingData(rIndex - samplesAroundPulse, 1);
        speed = voltageChange / timeInterval; % in mV/s
        rSpeeds(i) = abs(speed);
    end
end
% Loop through each T pulse to compute the speed

```

```

for i = 1:length(restingTLocs)
    tIndex = find(RestingData(:, 1) == restingTLocs(i), 1);
    if tIndex > samplesAroundPulse && tIndex + samplesAroundPulse <=
size(RestingData, 1)
        segment = RestingData(tIndex - samplesAroundPulse:tIndex +
samplesAroundPulse, 2);
        voltageChange = max(segment) - min(segment);
        timeInterval = RestingData(tIndex + samplesAroundPulse, 1) -
RestingData(tIndex - samplesAroundPulse, 1);
        speed = voltageChange / timeInterval; % in mV/s
        tSpeeds(i) = abs(speed);
    end
end
combinedSpeeds = [rSpeeds, tSpeeds];
% Plot histogram of speeds
figure;
histogram(rSpeeds, 'BinWidth', 0.03, 'Normalization', 'probability',
'FaceColor', 'b', 'DisplayName', 'R');
hold on;
histogram(tSpeeds, 'BinWidth', 0.03, 'Normalization', 'probability',
'FaceColor', 'r', 'DisplayName', 'T');
hold off;
xlabel('Speed (mV/s)', 'FontSize', 12);
ylabel('Probability', 'FontSize', 12);
title('Rate of change of voltage for different phases', 'FontSize', 14);
legend('show', 'Location', 'best');
saveas(gcf, 'R_and_T_Speeds_Histogram_with_Segments.png');

%% Step 9: Run a two-sided t-test using ttest2
clc
%Run a t-test on the heart rate frequency data during rest and exercise and
compute the output of the test
%From our above plots, it should be pretty clear that our heart rate during
rest and exercise seem pretty different, and there should be strong evidence to
suggest that they are different
x = restingHeartRate;
y = exerciseHeartRate;
result_RestvExercise = ttest2(x,y);
disp(result_RestvExercise)
if result_RestvExercise == 1
    disp('The null hypothesis is resting heart rate is there is no difference
from exercise heart rate. We reject the null hypothesis.')
elseif result_RestvExercise == 0
    disp('The null hypothesis is resting heart rate is there is no difference
from exercise heart rate. We cannot reject the null hypothesis.')
end
end

```



```

%Run a t-test on the heart rate frequency during two separate periods of the
rest data
% 1:10 is from 150 sec to 158 sec period
% 20:30 is from 165 sec to 173 sec period
x2 = restingHeartRate(1:10);
y2 = restingHeartRate(20:30);
result_sep_period = ttest2(x2,y2);
disp(result_sep_period)
if result_sep_period == 1
    disp('The null hypothesis is resting heart rate is there is no difference
among different time period. We reject the null hypothesis.')
elseif result_sep_period == 0
    disp('The null hypothesis is resting heart rate is there is no difference
among different time period. We cannot reject the null hypothesis.')
end

```

Command Window

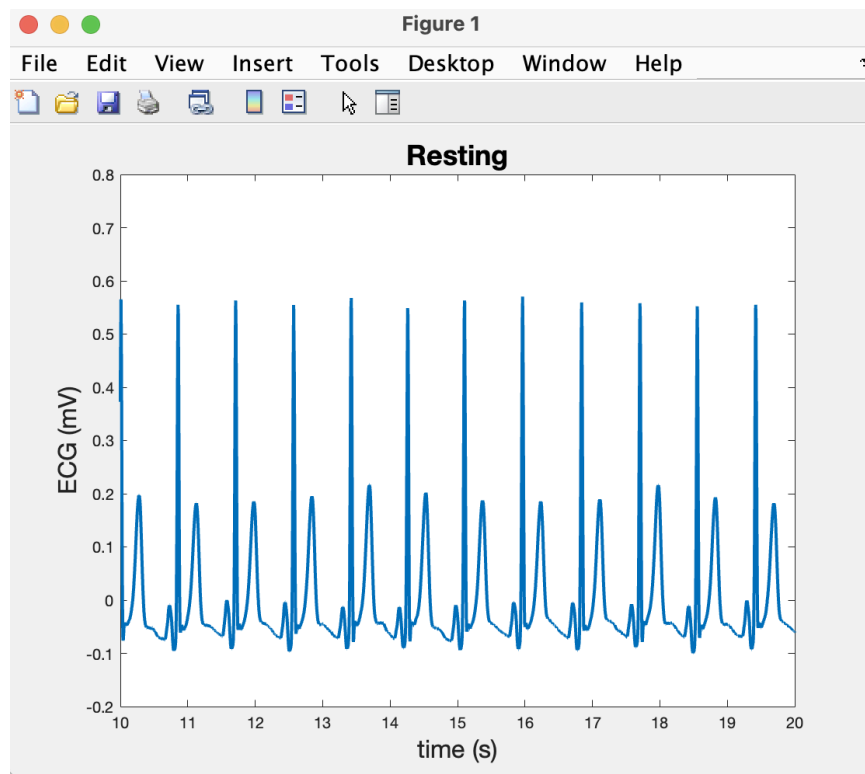
1

The null hypothesis is resting heart rate is there is no difference from exercise heart rate. We reject the null hypothesis.

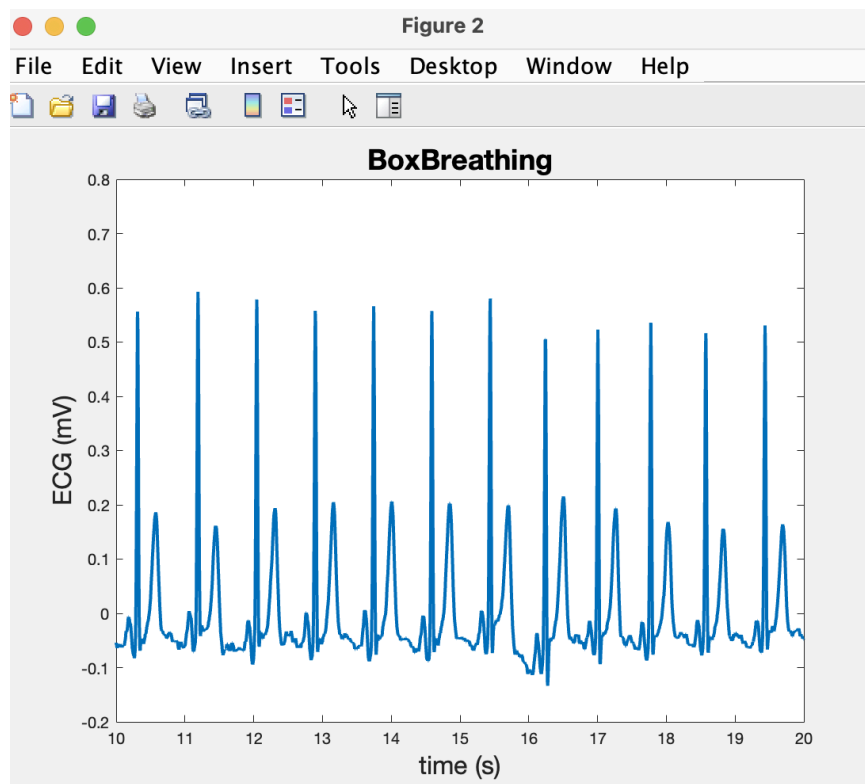
0

The null hypothesis is resting heart rate is there is no difference among different time period. We cannot reject the null hypothesis.

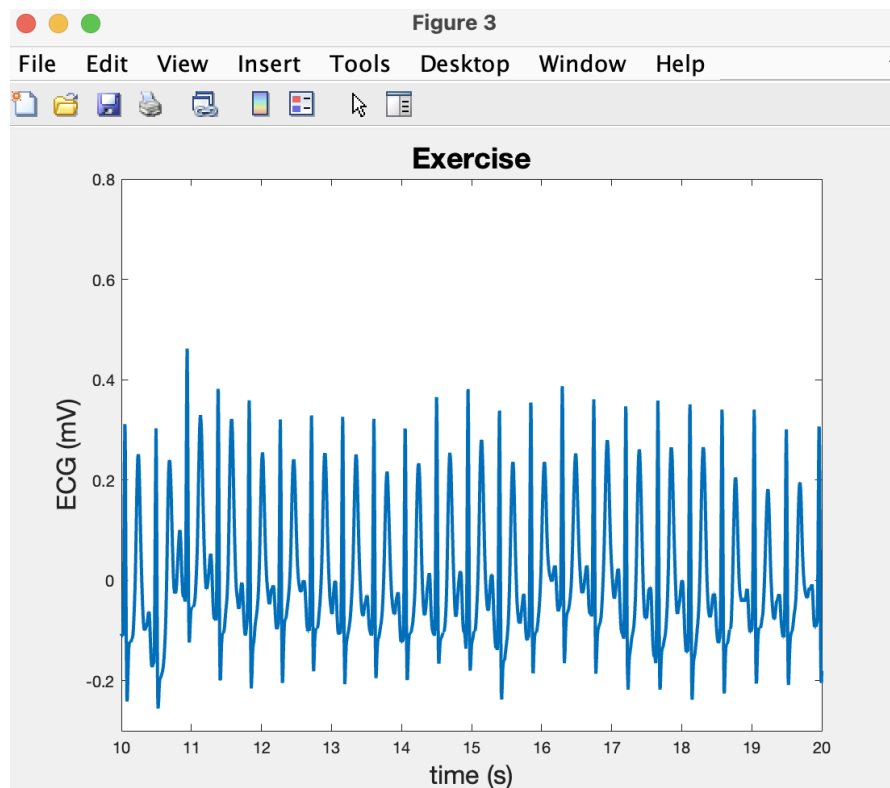
Plot 10s of Resting ECG signals



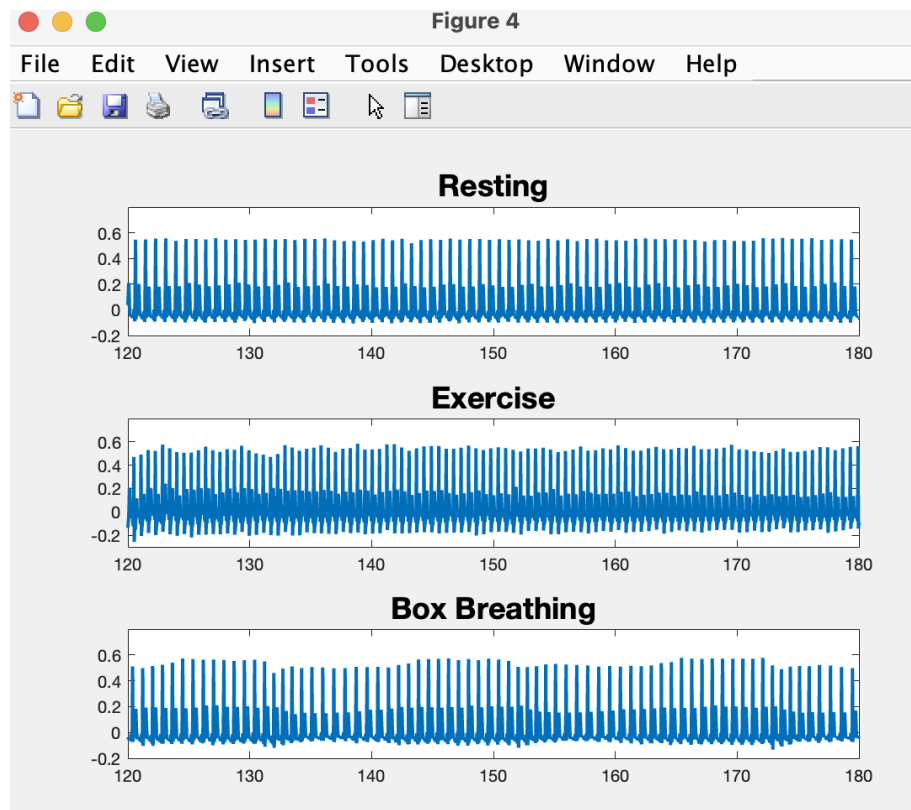
Plot 10s of BoxBreathing ECG signals



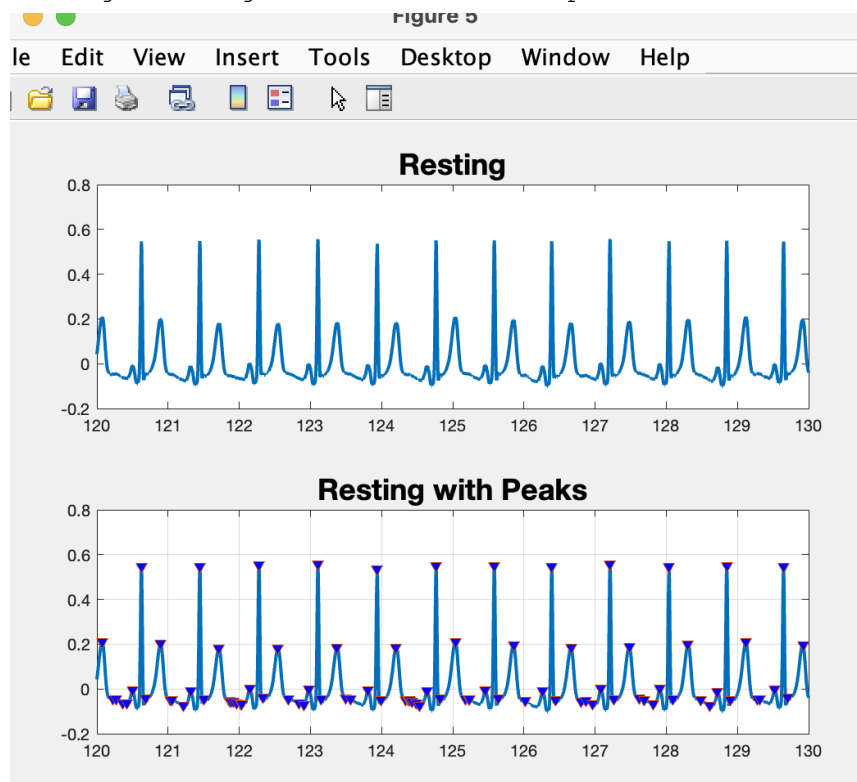
Plot 10s of Exercise ECG signals



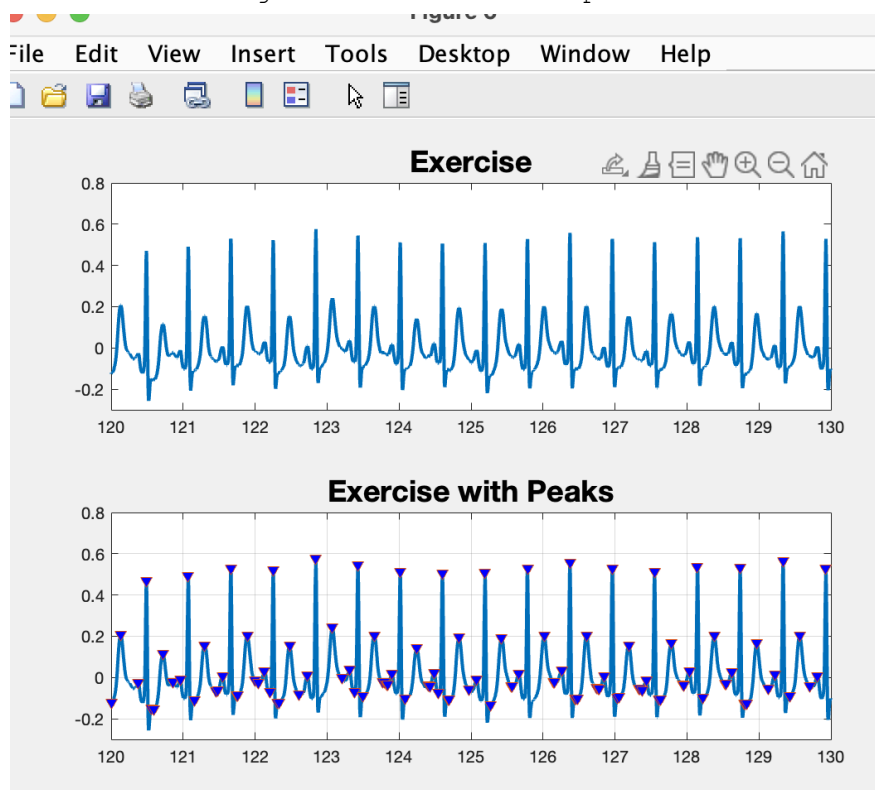
3x1 Subplot for all 3 datas



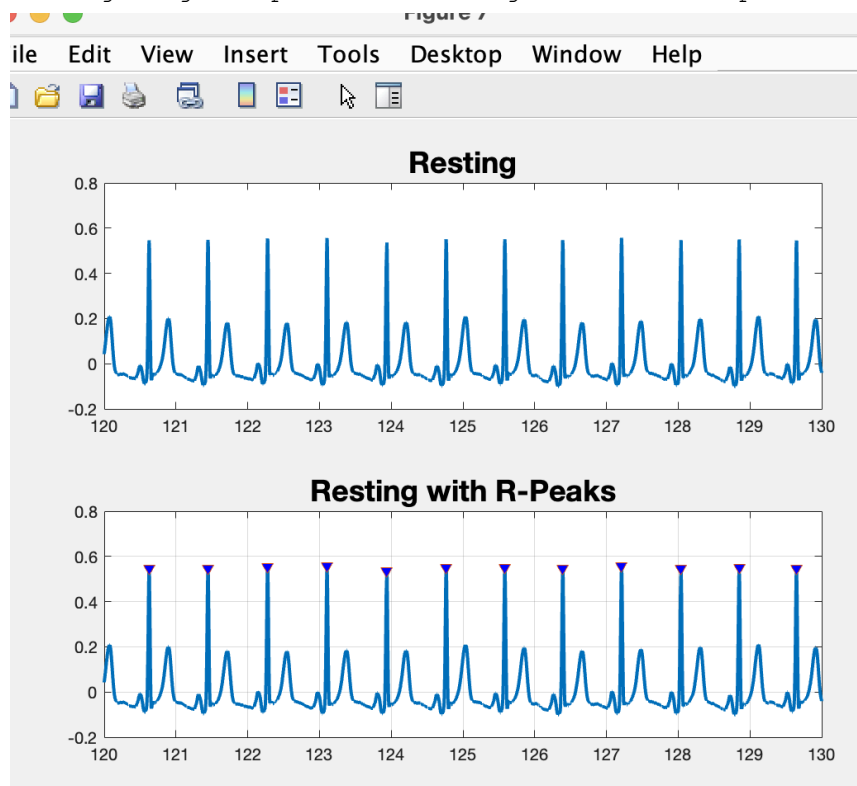
Resting ECG Original and with Peaks plot



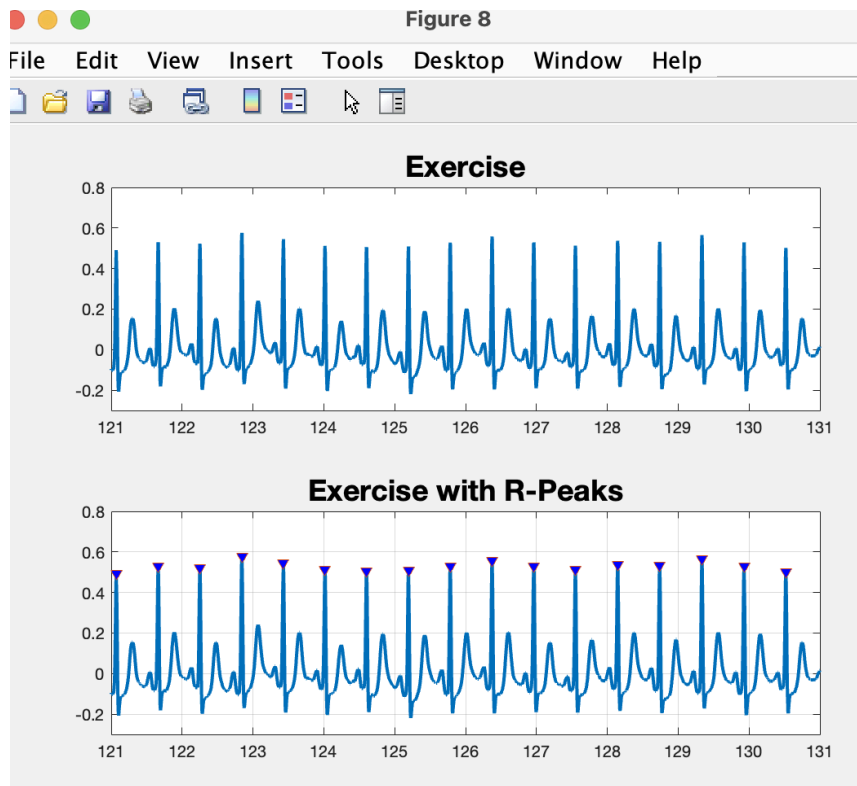
Exercise ECG Original and with Peaks plot



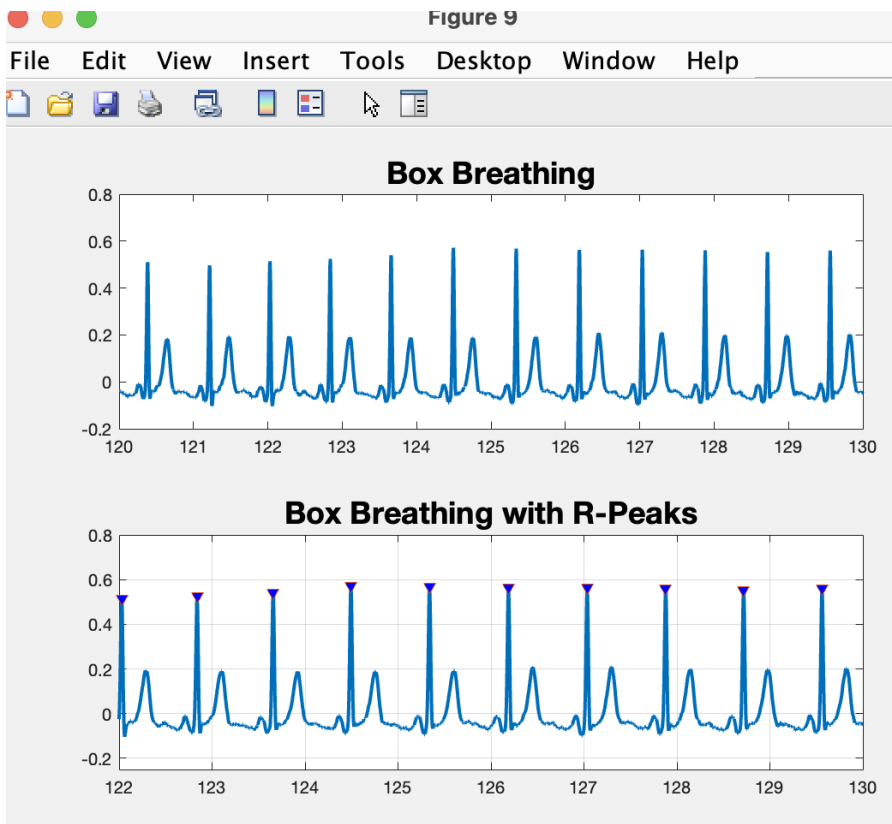
Resting Original plot and Resting with R-Peaks plot



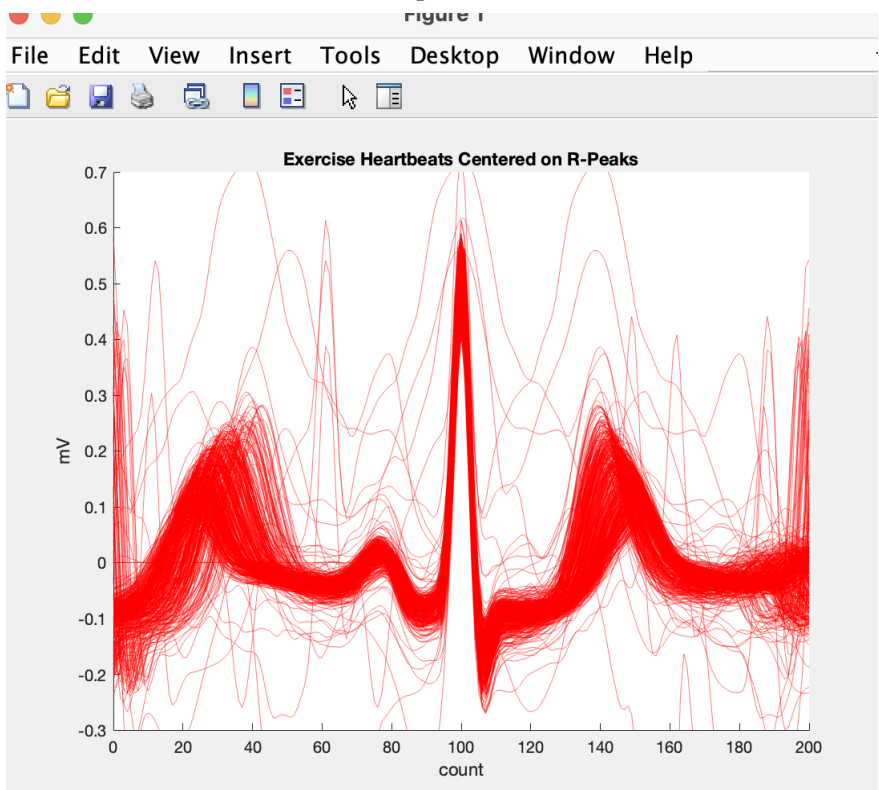
Exercise Original plot and Resting with R-Peaks plot



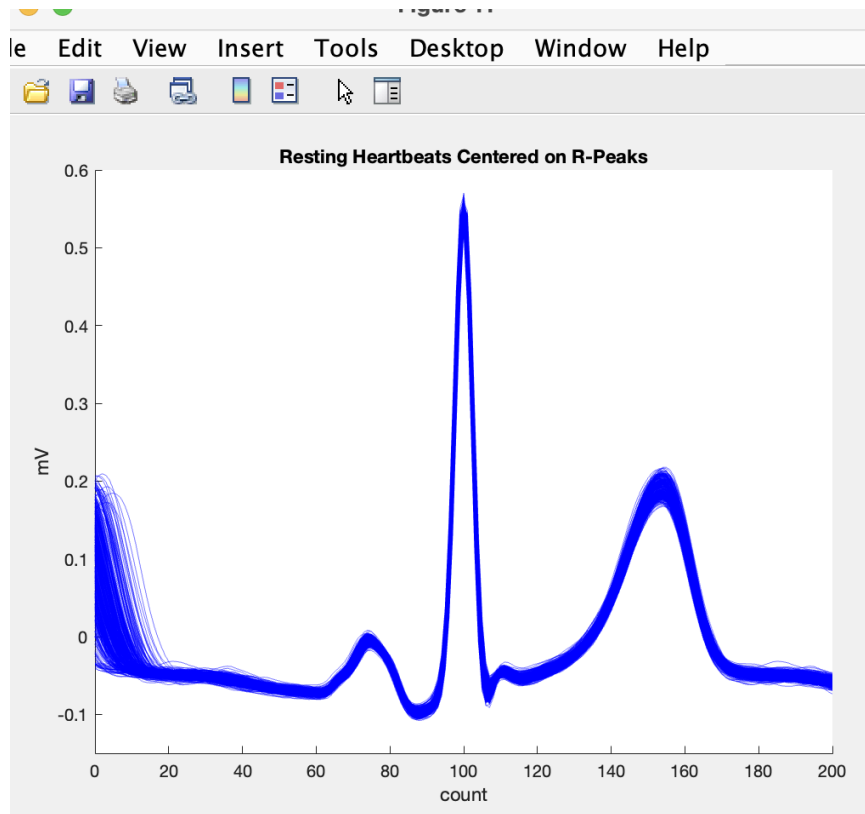
BoxBreathing Original plot and Resting with R-Peaks plot



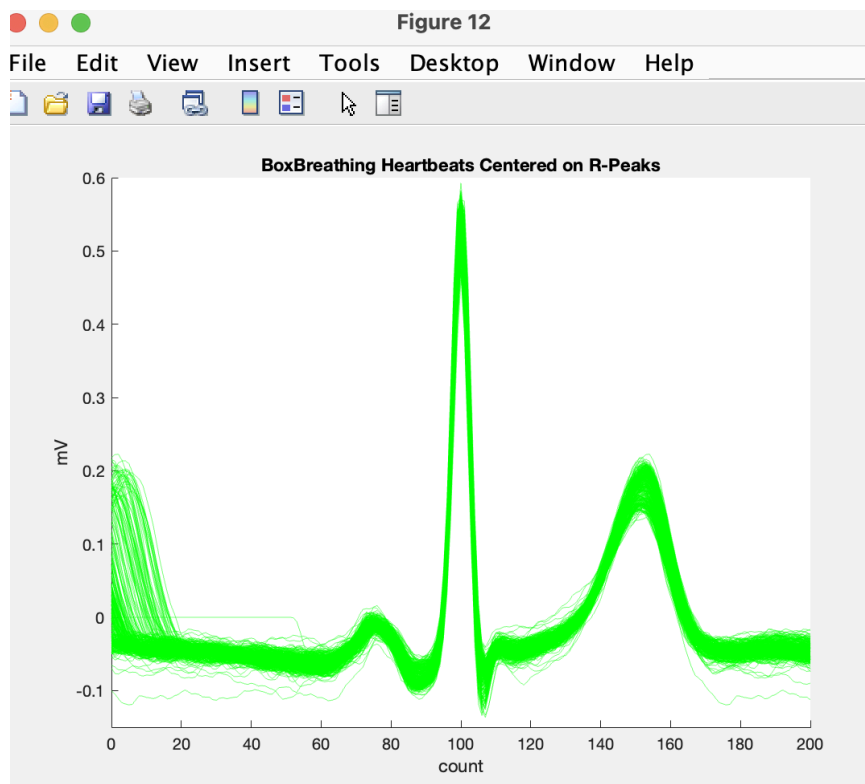
Exercise heartbeat red plot



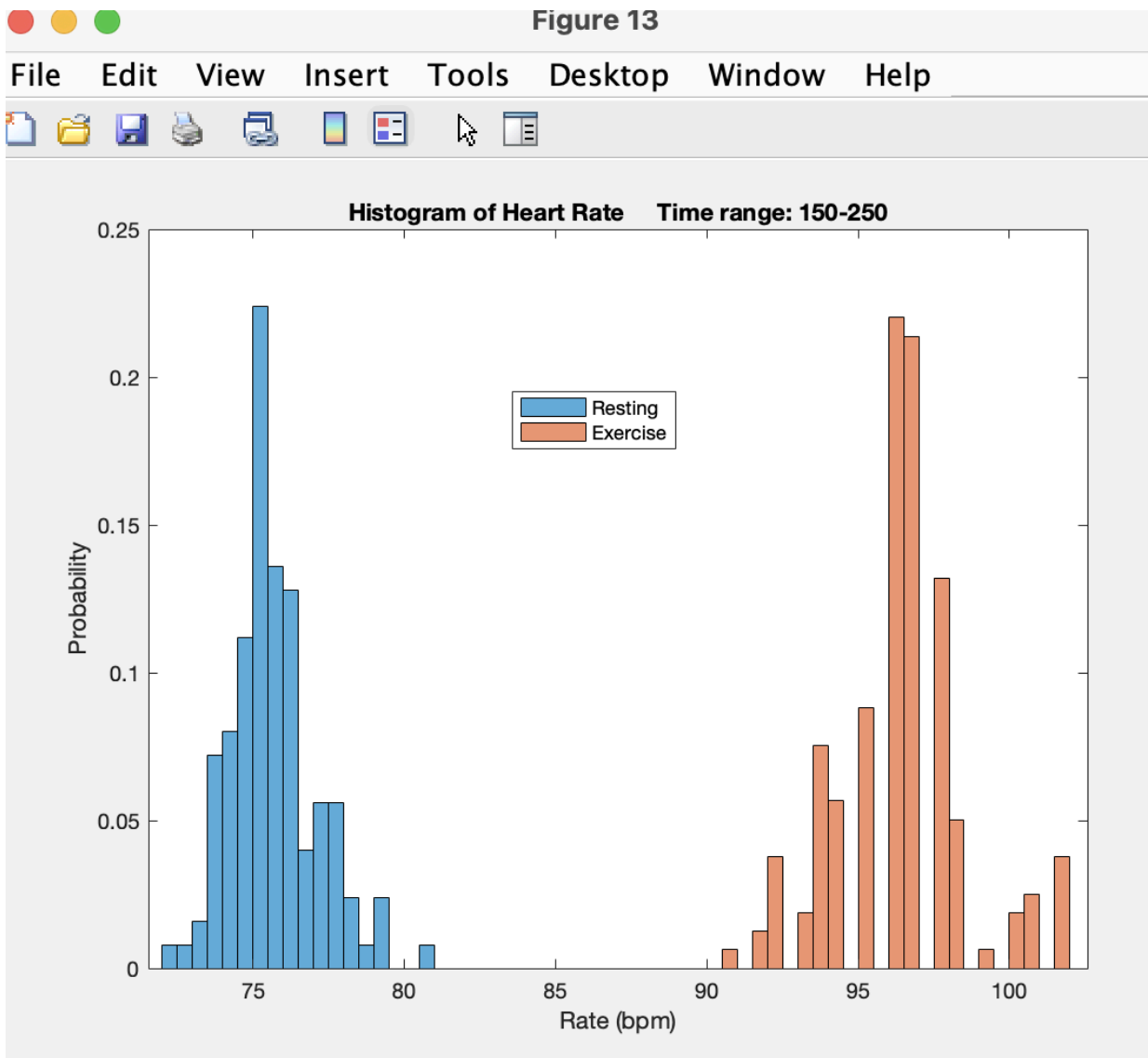
Resting Heartbeat blue plot



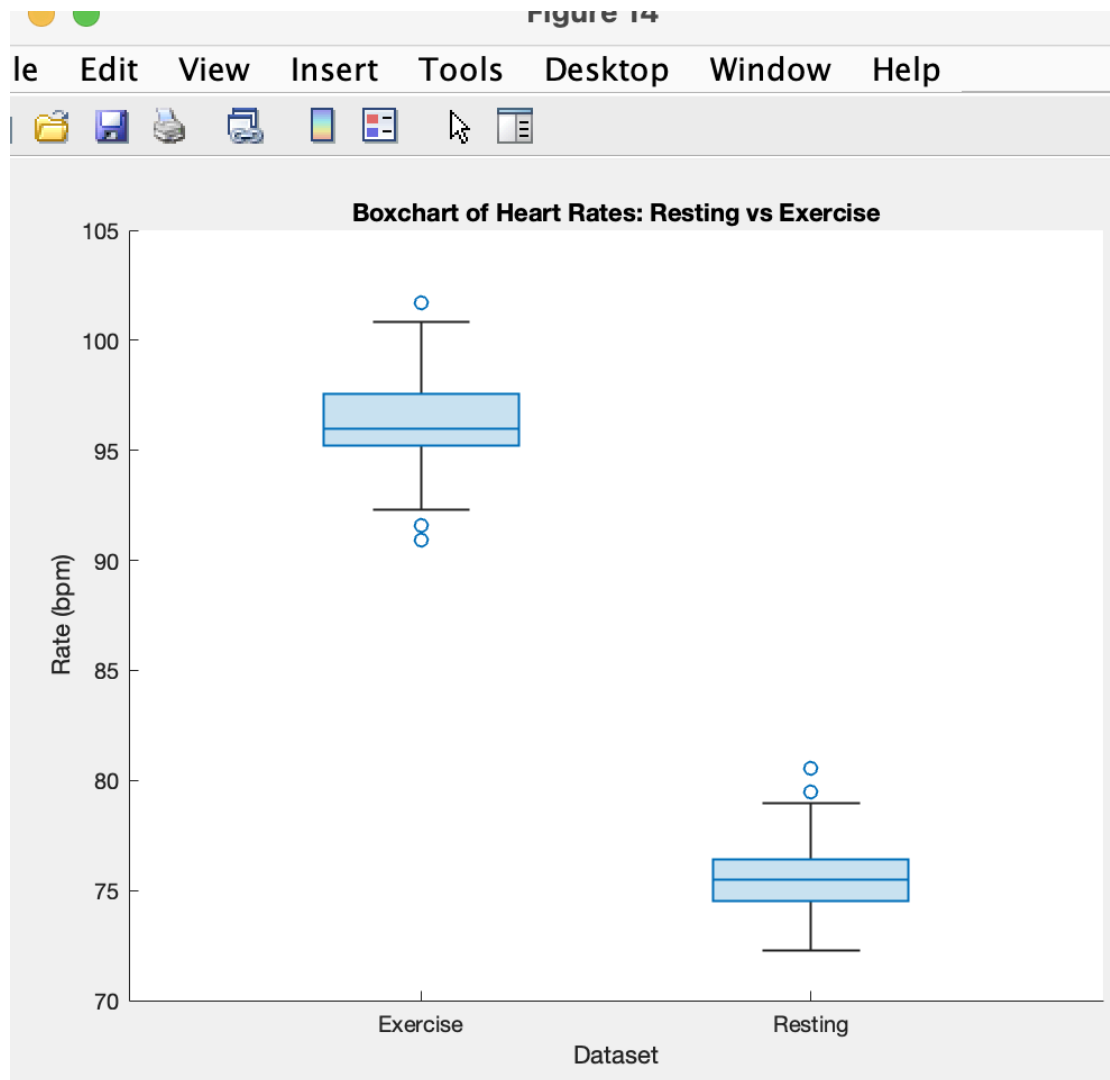
Resting Heartbeat green plot



Histogram of heart rate frequency in bpm Resting vs Exercise



Boxplot of heart rate Resting vs Exercise



R & T pulse Speed histogram of Resting data



Figure 1

File Edit View Insert Tools Desktop Window Help

