Title: Skilled Golfers Exhibit Less Variable Hip and Elbow Movements Than Amateur Golfers

Introduction: Golf performance relies heavily on biomechanical precision, yet most existing golf analysis tools prioritize club and ball data, often neglecting detailed body mechanics. This study investigates biomechanical differences between amateur and professional male golfers by analyzing joint positions, angles, and angular velocities during three key phases of the swing: backswing, downswing, and follow-through. We hypothesized that professional golfers demonstrate greater consistency in joint metrics, with specific joints and swing phases serving as indicators of skill level. By leveraging motion capture and biomechanical analysis from OpenCap and OpenSim software we aim to identify the key kinematic markers that differentiate skill level.

Methods: Data were collected from five amateur and five professional male golfers using OpenCap motion capture technology with two synchronized cameras in an indoor environment. Each subject performed 20 swings using the same 7-iron. The same club was used to make the distance between their bodies and the ball uniform. This created a standardized starting, impact, and follow through point of the joints in the hands by being the same distance away from the ball. Joint angle data was segmented into three swing phases: backswing, downswing, and follow-through. These are the three key segments of the swing and were segmented to get an overall understanding of where the most important and consistent intervals of the swing are. This was done by going through each recording and manually noting the time points at the start of the swing when the player started their motion of their swing, contact of the face of the club with the golf ball, and the end of the swing where the player stopped their motion through the swing at the top of their swing. Data was normalized over time (0% to 100% completion of each phase), and the mean and standard deviation were calculated for each joint angle across trials. This allowed us to create models that projected the accuracy of the players in the parts of their swing and their consistency (ability to create the same points in their swing again and again). Statistical analyses including aggregate standard deviation and significance testing, K-Nearest Neighbors, and Monte Carlo were conducted for joint movements such as hip rotation, knee angle, arm flexion, and rotation. This data allowed us to compare the swings of the beginner versus advanced golfers and determine similarities and differences of the data.

Results & Discussion: Professionals exhibited significantly lower variability in key joint angles, particularly pelvic rotation and right elbow flexion (P-value < 0.05), underscoring their greater consistency compared to amateurs. In contrast, metrics like right knee angle and right arm rotation showed no statistically significant differences between groups. These findings align with the consensus that hip rotation and upper body control are critical for skillful swings, while variability in arm movement is less impactful. Increased standard deviation observed in the follow-through phase highlights its reduced relevance for distinguishing skill levels.

We found that the specific joint movements of importance were hip rotation and knee angle. The professional player pool had significantly higher consistency in these movements. Meanwhile, joint movements such as arm flexion and arm rotation were not of importance. The difference in standard deviation or consistency between swings for these movements were not statistically significant, in fact, professional players actually had lower consistency across trials than amateurs for these movements. This was found in both our K-NN algorithm and Levene's statistical test. This points to the conclusion that precise lower-body movement is the key indicator of skill and a successful golf swing whereas arm movement is not. In a study on the three-dimensional knee joint kinetics during a golf swing, they found that only lead knee flexion and internal rotation moments were significantly correlated with skill level which is consistent with our findings (Gatt CJ, et al.)

Significance: This study reinforces the importance of movement consistency as a hallmark of professional golf performance. Future research should include more diverse subject groups (e.g., male vs. female golfers) and incorporate additional metrics such as joint velocities and angular velocities. Advanced data analysis techniques, including Principal Component Analysis (PCA), Monte Carlo simulations, and K-Nearest Neighbors (KNN), may uncover deeper biomechanical patterns. These findings could inform training protocols and golf simulator design to enhance skill development.

References: Dong R, Ikuno S. Biomechanical Analysis of Golf Swing Motion Using Hilbert-Huang Transform. Sensors (Basel). 2023 Jul 26;23(15):6698. doi: 10.3390/s23156698. PMID: 37571482; PMCID: PMC10422357.

Gatt CJ, Pavol MJ, Parker RD, Grabiner MD. Three-Dimensional Knee Joint Kinetics During a Golf Swing. The American Journal of Sports Medicine. 1998;26(2):285-294. doi:10.1177/03635465980260022101

Skilled Golfers Exhibit Less Variable Hip and Elbow Movements Than Amateur Golfers

Introduction:

Golf is a sport where success is determined by the precision, consistency, and efficiency of body movement. While existing analytical tools emphasize club and ball data—such as swing speed, launch angle, and ball spin—these metrics provide limited insights into the body mechanics driving the swing. The golf swing is a complex, coordinated motion involving multiple joints, with phases like the backswing, downswing, and follow-through requiring fine-tuned control for effective energy transfer and ball striking.

Despite its importance, joint-level biomechanical data is often overlooked in favor of outcome-based metrics, leaving a gap in understanding how skilled golfers achieve consistency and control. This study addresses that gap by analyzing joint angles and their variability in skilled and amateur golfers by using recreational items instead of thousands of dollars worth of equipment. Using OpenCap for motion capture and OpenSim for biomechanical modeling, we compare movements across key joints—hips, elbows, and knees—during critical phases of the swing. We hypothesize that skilled golfers exhibit lower variability in their joint movements, particularly in the hips and upper body, reflecting their ability to generate power and maintain control. With these results we will be able to create a more accessible alternative to collecting data than most players have access to. This work aims to highlight the kinematic markers that differentiate skill levels, offering a more comprehensive understanding of the golf swing.

Methods:

Data were collected from five amateur and five professional male golfers using OpenCap motion capture technology with two synchronized cameras in an indoor environment. The data was then run through OpenSim which took the skeletal analysis of OpenCap and produced matrices of the joint angles throughout the entirety of the video. We then used MATLAB to

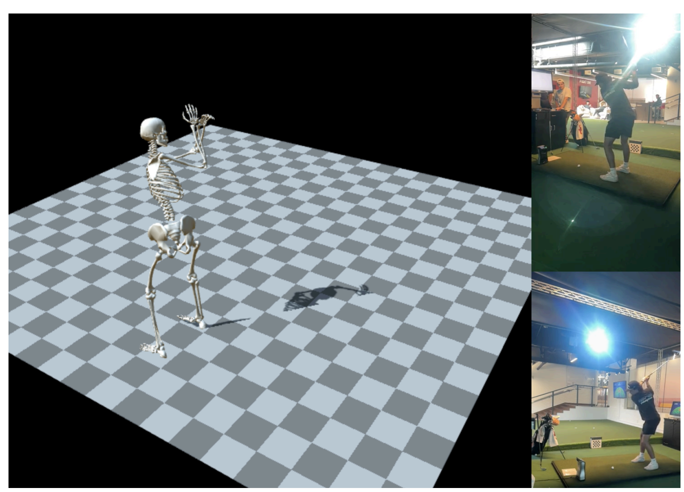

Figure 1 - Motion capture setup using OpenCap. The figure shows the skeletal model generated by OpenCap (left) and synchronized video recordings of a participant performing a golf swing (right) of the backswing. Uses smartphone cameras to track joint movements and create a biomechanical representation for subsequent analysis.

convert these matrices to data points over a normalized time that we configured using the video. Each subject performed 20 swings using a 7-iron. The same golf club was used across subjects for the sake of standardization and reducing confounding variables. The reduction of the variables were that the golfers had to start and get back to the same spot with their bodies to make contact with the ball. Joint angle data were segmented into three swing phases: backswing, when the subject initiates their swing to when they reached the top of the backswing where their momentum stopped in one direction and they were in the process of bringing the club back towards the ball, downswing, where the subject initiated the change of momentum towards the golf ball from the top of the backswing to when they made impact with the golf ball, and follow-through, where the subject made contact until they reach the end of their swing where momentum in the forward direction comes to an end. This was done by going through each recording and manually noting the time points at the start of the swing, contact of the face of the club with the golf ball, and the end of the swing as dictated above. They were normalized over time (0% to 100% completion of each phase) for each subject, the mean and standard deviation were calculated for each joint angle across trials throughout the normalized time. Statistical analyses were done using MATLAB code, including aggregate standard deviation and significance testing, K-Nearest Neighbors, and Monte Carlo were conducted for joint movements such as hip rotation, knee angle, arm flexion, and rotation.

Results & Discussion:

The results of this study reveal distinct differences in joint angle variability between professional and amateur golfers, with significant findings for hip rotation and right elbow flexion. Analysis of the mean and standard deviation of joint angles across swing phases demonstrated that professionals consistently exhibit lower variability in these two joints, highlighting their superior control and precision. Levene's test for statistical significance confirmed these findings, showing that variability in hip rotation and right elbow flexion differed significantly between the two groups (p-values < 0.05). These results underscore the importance

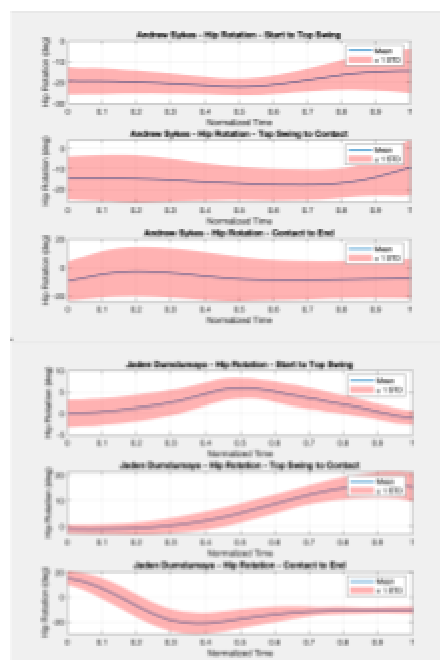

Figure 2: Hip rotation variability across swing phases for an amateur golfer (top) and a professional golfer (bottom). The professional golfer demonstrates visibly lower variability (narrower standard deviation region) across all swing phases—start to top swing, top swing to contact, and contact to end—highlighting greater consistency in hip rotation compared to the amateur golfer.

of lower-body mechanics, particularly hip stability, and controlled upper-body movements, such as elbow flexion, as critical components of a skillful golf swing. In contrast, variability in other joints, including right knee angle and right arm rotation, did not exhibit statistically significant differences, suggesting that these movements play a less prominent role in differentiating skill levels.

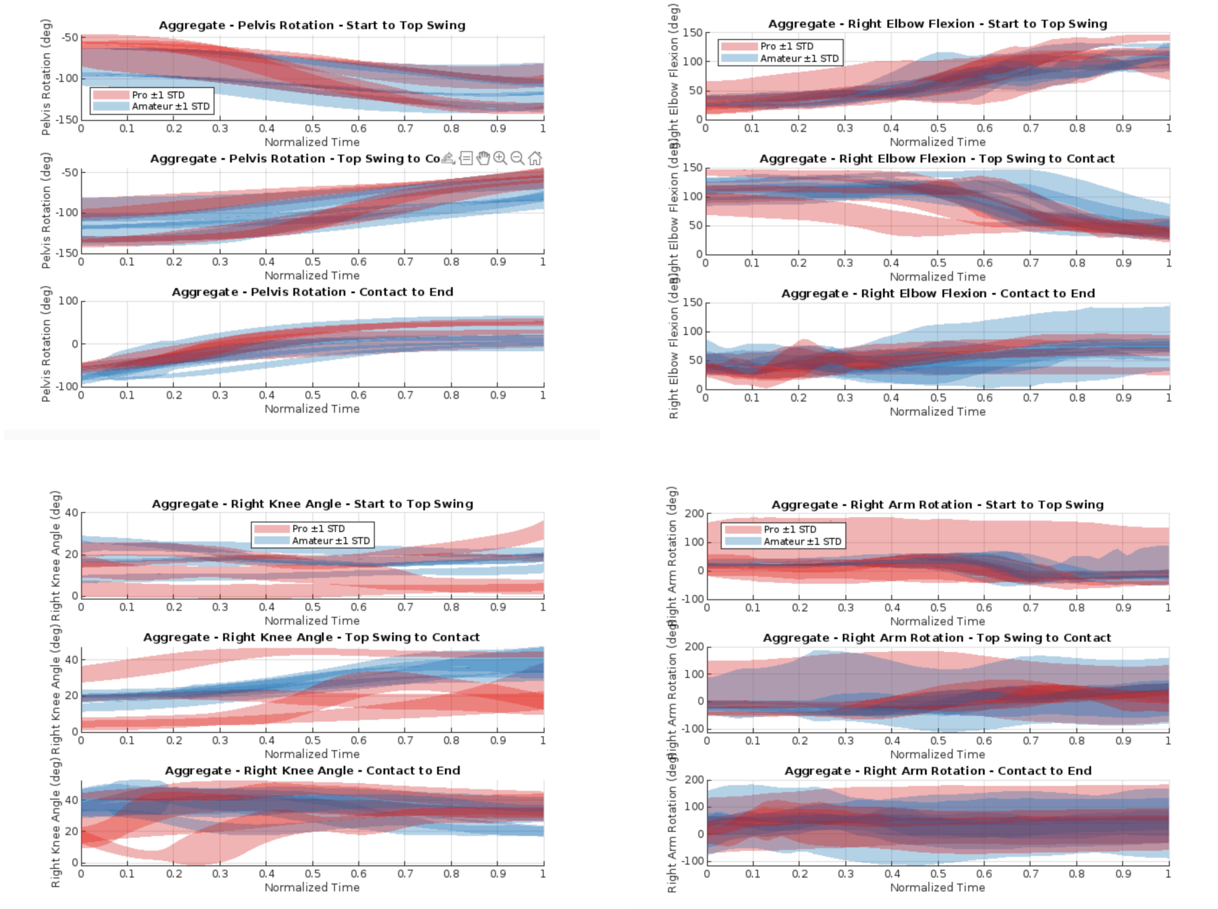

Figure 3: Aggregate variability in joint angles across swing phases for professionals (blue) and amateurs (red). Professionals demonstrate reduced variability in pelvis rotation and right elbow flexion, particularly during critical phases like the downswing and contact. Variability in right knee angle and right arm rotation shows smaller differences, with upper-body movements exhibiting less consistent trends between the two groups.

The Monte Carlo analysis for hip rotation provided additional validation of these findings, reinforcing the role of consistent lower-body mechanics in skilled performance. By generating null distributions through repeated random shuffling of group labels, the analysis confirmed that the observed group differences in hip rotation variability were unlikely to occur by chance. For both left hip rotation and right hip rotation (Figure 4), the observed differences lay far outside the range of shuffled group differences, highlighting their statistical significance. This result validates the hypothesis that professionals achieve greater consistency in hip movement, particularly during critical swing phases such as the downswing and the moment of contact, where energy transfer and stability are paramount.

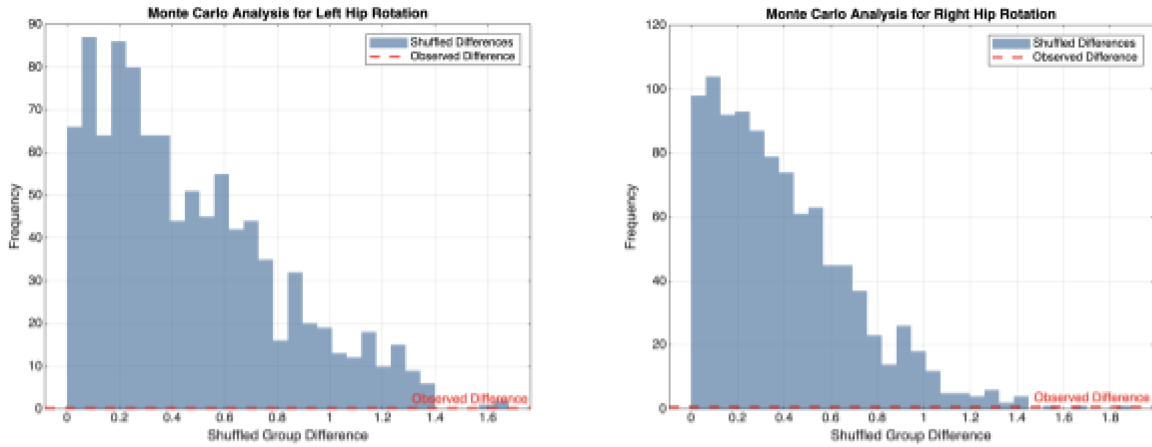

Figure 4: Monte Carlo analysis for left (left) and right (right) hip rotation variability. The observed group differences (red dashed line) lie far outside the shuffled distributions, confirming significantly lower variability in professional golfers.

In contrast to hip rotation, variability in right knee angle did not show statistically significant differences between groups, as confirmed by Levene's test. However, K-Nearest Neighbors (KNN) classification provided some additional insight into the role of knee angle in skilled performance. Although the variability was not significant overall, professionals demonstrated lower mean squared error (MSE) values for knee angle when classified using the KNN algorithm. This suggests that while knee variability alone may not be sufficient for statistical differentiation, its stabilization likely plays a supportive role in maintaining balance and facilitating the efficient transfer of energy during the swing.

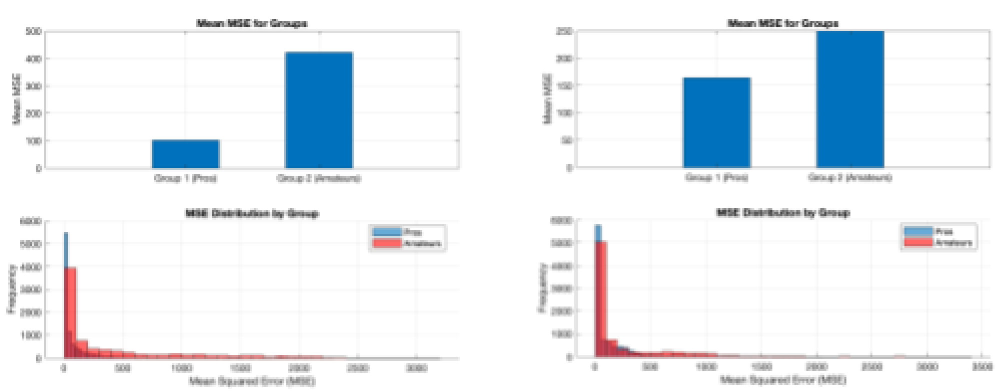

Figure 5: Mean squared error (MSE) analysis for **hip rotation** (left) and **knee angle** (right) using K-Nearest Neighbors (KNN). Professionals (Group 1) exhibit lower mean MSE compared to amateurs (Group 2) for both joints, indicating greater consistency in lower-body movements among professionals. The MSE distribution further highlights tighter variability in professional golfers.

Upper-body movements, including right elbow flexion and right arm rotation, exhibited mixed trends in variability. Elbow flexion emerged as a significant differentiator based on both Levene's test and KNN classification. Professionals showed lower variability and MSE values for elbow flexion, suggesting that control of elbow movement is a relevant factor in skillful swings, particularly during the downswing and contact phases, where precision is key. In contrast, right arm rotation displayed an unexpected trend, with amateurs achieving lower MSE values compared to professionals. This finding indicates that greater variability in arm rotation may not be detrimental to performance, as professionals appear to tolerate or even leverage variability in upper-body movements. This observation highlights the relative flexibility of upper-body mechanics compared to the more rigid and consistent demands of lower-body stability.

The variability trends for pelvis rotation, elbow flexion, and knee angle across normalized swing phases (Figure 3) further clarify these findings. Professionals exhibited visibly narrower standard deviation regions for hip and pelvic rotation, particularly during the downswing and top swing to contact phases. These phases are biomechanically critical, as they involve the coordinated transfer of energy from the lower body to the upper body and the club. Meanwhile, increased variability observed in the follow-through phase for both groups suggests that this phase plays a less central role in skill differentiation. In contrast, the variability trends for arm rotation and elbow flexion revealed greater variability in amateurs during critical phases, aligning with the broader conclusion that upper-body movements are less predictive of skill level.

Taken together, these results emphasize the biomechanical importance of hip rotation and elbow flexion as key differentiators of skill level in golfers. Professionals achieve superior consistency in these joints, reinforcing the critical role of lower-body stability and controlled upper-body movement in producing a repeatable and efficient swing. The lack of significant differences in knee angle and arm rotation variability highlights the secondary importance of these joints, suggesting that variability in upper-body movements may be tolerated or even beneficial within certain limits. These findings provide practical implications for training programs, which should prioritize improving hip rotation consistency and elbow control during critical swing phases to enhance overall performance. Future work incorporating additional kinematic metrics, such as joint velocities and rotational accelerations, may further elucidate the relationships between variability, skill level, and swing outcomes.

Significance:

This study highlights the critical role of biomechanical consistency in differentiating skilled golfers from amateurs, moving beyond traditional club and ball data. Our findings show that reduced variability in hip rotation and elbow flexion is a key marker of skill, emphasizing the importance of lower-body stability and upper-body control in generating power and precision. These insights have practical applications for improving training methodologies and enhancing accessibility to players who want to understand their swing by integrating biomechanical feedback to provide actionable insights for players and coaches.

Several limitations may have influenced our results, including slight misalignment in camera angles, variability in participant execution, and environmental factors such as lighting inconsistencies. Additionally, timing errors during swing segmentation, particularly for the follow-through phase, may have introduced minor inaccuracies in data normalization.

Future research will address these limitations by increasing the sample size to include a more diverse group of golfers, segmented by gender and refined skill categories based on handicap. We also plan to incorporate additional metrics, such as joint velocities and angular velocities, to analyze the acceleration and rotational efficiency of specific joints. Further analysis will focus on linking joint variability and movement efficiency to swing outcomes, such as ball speed, accuracy, and energy transfer, to provide deeper insights into how biomechanics influence overall performance. Finally, developing affordable real-time motion feedback tools based on these findings could enable golfers to make targeted corrections during practice, bridging the gap between biomechanical analysis and practical skill development.

References:

Dong R, Ikuno S. Biomechanical Analysis of Golf Swing Motion Using Hilbert-Huang Transform. Sensors (Basel). 2023 Jul 26;23(15):6698. doi: 10.3390/s23156698. PMID: 37571482; PMCID: PMC10422357.

Gatt CJ, Pavol MJ, Parker RD, Grabiner MD. Three-Dimensional Knee Joint Kinetics During a Golf Swing. The American Journal of Sports Medicine. 1998;26(2):285-294. doi:10.1177/03635465980260022101

Appendix:

MATLAB Code with KNN:

```
%% Golf Group Curve Code
clc;
clear;
close all;
% Define the column index and name for the joint to process
joint_column = 10; % Change this value to the desired column in the .mot file
joint_name = 'Hip Rotation'; % Change this to the name of the joint
% Read segmentation data from the file
segmentation_file = 'GolfTimeSegment.txt';
segmentation_data = readtable(segmentation_file, 'Delimiter', '\t',
'ReadVariableNames', false, 'TextType', 'string');
% Filter valid rows (those with complete segmentation data)
valid_rows = all(~ismissing(segmentation_data{:, 4:7}), 2);
segmentation_data = segmentation_data(valid_rows, :);
% Extract unique subjects
subjects = unique(segmentation_data.Var2);
% Initialize parameters
num_points = 100; % Number of normalized points per segment
aggregate_data = containers.Map; % To store aggregate data
% Classify subjects as Pro or Amateur
classification = containers.Map; % Map for subject classification
pro_color = [0, 0.4470, 0.7410]; % Blue for pros
amateur_color = [1, 0, 0]; % Red for amateurs
% Process each subject
for s = 1:length(subjects)
   subject_name = subjects{s};

   % Determine if the subject is a Pro or Amateur
   subject_prefix = segmentation_data.Var1(strcmp(segmentation_data.Var2,
subject_name));
   if contains(subject_prefix(1), 'P')
       classification(subject_name) = 'Pro';
   else
       classification(subject_name) = 'Amateur';
   end

   % Extract rows for the current subject
   subject_rows = strcmp(segmentation_data.Var2, subject_name);
   subject_trials = segmentation_data(subject_rows, :);
   num_trials = sum(subject_rows);

   % Preallocate storage for subject data
   joint_segments = zeros(num_trials, 3, num_points);

   % Process each trial for the current subject
   for t = 1:num_trials
       trial_number = subject_trials.Var3(t);
       start_time = subject_trials.Var4(t);
```

```
        top_swing_time = subject_trials.Var5(t);
        contact_time = subject_trials.Var6(t);
        end_time = subject_trials.Var7(t);

        % Generate filename for the trial
        filename = sprintf('%s%d.mot', replace(subject_name, " ", ""),
trial_number);

        % Load .mot file data
        data = readmatrix(filename, 'FileType', 'text');
        time = data(:, 1); % Time column
        joint_angle = data(:, joint_column); % Use specified column for joint
angle

        % Interpolate data for each segment
        segment1_time = linspace(start_time, top_swing_time, num_points);
        segment2_time = linspace(top_swing_time, contact_time, num_points);
        segment3_time = linspace(contact_time, end_time, num_points);

        joint_segments(t, 1, :) = interp1(time, joint_angle, segment1_time);
        joint_segments(t, 2, :) = interp1(time, joint_angle, segment2_time);
        joint_segments(t, 3, :) = interp1(time, joint_angle, segment3_time);
    end

    % Calculate mean and standard deviation
    mean_values = squeeze(mean(joint_segments, 1));
    std_values = squeeze(std(joint_segments, 1));

    % Store aggregate data
    aggregate_data(subject_name) = struct('mean', mean_values, 'std',
std_values);

    % Plot individual subject data
    segment_labels = {'Start to Top Swing', 'Top Swing to Contact', 'Contact to
End'};
    figure;
    for segment = 1:3
        subplot(3, 1, segment);
        x = linspace(0, 1, num_points);
        plot(x, mean_values(segment, :), 'LineWidth', 1.5); hold on;
        fill([x, fliplr(x)], ...
             [mean_values(segment, :) + std_values(segment, :), ...
              fliplr(mean_values(segment, :) - std_values(segment, :))], ...
             'r', 'FaceAlpha', 0.3, 'EdgeColor', 'none');
        title([subject_name ' - ' joint_name ' - ' segment_labels{segment}]);
        xlabel('Normalized Time');
        ylabel([joint_name ' (deg)']);
        legend('Mean', '± 1 STD');
        grid on;
    end
    saveas(gcf, sprintf('%s_%s_analysis.png', replace(subject_name, " ", ""),
replace(joint_name, " ", "_")));
end
% Create aggregate plots for all subjects with Pro and Amateur color coding
```

```
figure;
segment_labels = {'Start to Top Swing', 'Top Swing to Contact', 'Contact to
End'};
% Loop through each segment
for segment = 1:3
   subplot(3, 1, segment);
   hold on;
   x = linspace(0, 1, num_points); % Normalized time
   for s = 1:length(subjects)
       subject_name = subjects{s};
       data = aggregate_data(subject_name);

       % Assign color based on classification
       if strcmp(classification(subject_name), 'Pro')
           plot_color = pro_color;
       else
           plot_color = amateur_color;
       end

       % Plot the mean curve for the subject
       plot(x, data.mean(segment, :), 'Color', plot_color, 'LineWidth', 1.5);
   end

   % Customize subplot
   title(['Aggregate - ' joint_name ' - ' segment_labels{segment}]);
   xlabel('Normalized Time');
   ylabel([joint_name ' (deg)']);
   grid on;
end
% Add legend for classification
subplot(3, 1, 1);
legend({'Pro', 'Amateur'}, 'Location', 'best', 'TextColor', 'black');
hold off;
% Save the aggregate plot
saveas(gcf, sprintf('aggregate_overlay_color_coded_%s.png', replace(joint_name,
" ", "_")));
% KNN Analysis for Pros and Amateurs
% Group definitions
group1_names = ["Jack Buchanan", "Charlie Adams", "Jack Jerge", "Jaden
Dumdumaya", "Nico Dominguez"];
group2_names = ["Andrew Sykes", "Nate Goss", "Ryan Becton", "Max Miesen"];
% Initialize storage for KNN results
group1_results = [];
group2_results = [];
% Process each subject
for s = 1:length(subjects)
   subject_name = subjects{s};
   % Check if the subject is in Group 1 (Pros) or Group 2 (Amateurs)
   if ismember(subject_name, group1_names)
       group = 1;
   elseif ismember(subject_name, group2_names)
       group = 2;
   else
       continue; % Skip subjects not in either group
```

```
    end
    % Extract rows for the current subject
    subject_rows = strcmp(segmentation_data.Var2, subject_name);
    subject_trials = segmentation_data(subject_rows, :);
    num_trials = sum(subject_rows);
    % Process each trial for the current subject
    for t = 1:num_trials
        trial_number = subject_trials.Var3(t);
        start_time = subject_trials.Var4(t);
        top_swing_time = subject_trials.Var5(t);
        contact_time = subject_trials.Var6(t);
        end_time = subject_trials.Var7(t);
        % Generate filename for the trial
        filename = sprintf('%s%d.mot', replace(subject_name, " ", ""),
trial_number);
        try
            % Check if file exists before attempting to read
            if ~isfile(filename)
                fprintf('Warning: File does not exist: %s\n', filename);
                continue;
            end
            % Load .mot file data
            data = readmatrix(filename, 'FileType', 'text');
            % Ensure data is non-empty and has sufficient columns
            if isempty(data) || size(data, 2) < joint_column
                error('File is missing data or has incorrect format: %s',
filename);
            end
            time = data(:, 1); % Time column
            joint_angle = data(:, joint_column); % Use specified column for
joint angle
            % Check if time and joint_angle have the same length
            if length(time) ~= length(joint_angle)
                error('X and Y do not have the same number of observations.');
            end
            % Interpolate joint angles for the entire trial
            trial_time = linspace(start_time, end_time, num_points);
            interpolated_angles = interp1(time, joint_angle, trial_time,
'linear', 'extrap');
            % Ensure interpolated angles are non-empty
            if isempty(interpolated_angles)
                error('Interpolated angles are empty for file: %s', filename);
            end
            % Prepare features and target for KNN (example feature extraction)
            features = [mean(interpolated_angles), std(interpolated_angles)];
            target = interpolated_angles; % Use actual interpolated angles as
target
            % Perform KNN
            mdl = fitcknn(repmat(features, length(target), 1), target,
'NumNeighbors', 5, 'Standardize', true);
            predicted = predict(mdl, repmat(features, length(target), 1));
            % Calculate mean squared error for this trial
            trial_mse = mean((predicted - target).^2);
        catch ME
```

```
            fprintf('Warning: Error processing file %s: %s\n', filename,
ME.message);
            continue; % Skip this trial if there is an error
        end
        % Append the KNN result to the respective group
        if group == 1
            group1_results = [group1_results, trial_mse];
        else
            group2_results = [group2_results, trial_mse];
        end
    end
end
% Analyze results
if isempty(group1_results)
    fprintf('Warning: No valid results for Group 1 (Pros).\n');
    group1_mean = NaN;
else
    group1_mean = mean(group1_results);
end
if isempty(group2_results)
    fprintf('Warning: No valid results for Group 2 (Amateurs).\n');
    group2_mean = NaN;
else
    group2_mean = mean(group2_results);
end
% Display results
fprintf('Group 1 (Pros) Mean MSE: %.4f\n', group1_mean);
fprintf('Group 2 (Amateurs) Mean MSE: %.4f\n', group2_mean);
if ~isnan(group1_mean) && ~isnan(group2_mean)
    if group1_mean < group2_mean
        fprintf('Group 1 (Pros) has better performance (lower MSE).\n');
    else
        fprintf('Group 2 (Amateurs) has better performance (lower MSE).\n');
    end
else
    fprintf('Comparison could not be completed due to missing data.\n');
end
```

MEAN AND ST. DEV GRAPHS

```
% Segmentation data for 20 trials
segmentation_data = [
   3.25, 4.08, 4.33, 4.8;
   1.00, 1.82, 2.07, 2.75;
   3.50, 4.28, 4.53, 5.08;
   3.13, 3.92, 4.18, 4.72;
   3.37, 4.17, 4.38, 4.93;
   2.42, 3.18, 3.40, 3.93;
   2.85, 3.63, 3.85, 4.55;
   2.40, 3.18, 3.45, 4.02;
   3.32, 4.20, 4.38, 3.70;
   2.33, 3.08, 3.37, 4.32;
   2.47, 3.27, 3.52, 3.98;
   3.15, 3.88, 4.13, 4.75;
   2.63, 3.30, 3.55, 4.13;
   2.90, 3.62, 3.87, 4.48;
   2.27, 3.02, 3.28, 3.83;
   1.68, 2.43, 2.68, 3.13;
   4.67, 5.45, 5.70, 6.55;
   2.42, 3.23, 3.43, 3.98;
   1.93, 2.73, 2.99, 3.85;
   3.20, 3.97, 4.23, 4.98
];
% Initialize parameters
num_trials = size(segmentation_data, 1);
num_points = 100; % Normalized time points per segment
num_files = 20; % Total number of files
knee_segments = zeros(num_trials, 3, num_points);
% Loop through each file and load data
for trial = 1:num_files
   % Load the .mot file
   filename = sprintf('JackBuchanan%d.mot', trial);
   data = readmatrix(filename, 'FileType', 'text');

   % Extract time and right knee angle data
   time = data(:, 1);
   knee_angle_r = data(:, 10); % 10th column: Right Knee Angle

   % Segmentation times for this trial
   start_time = segmentation_data(trial, 1);
   top_swing_time = segmentation_data(trial, 2);
   contact_time = segmentation_data(trial, 3);
   end_time = segmentation_data(trial, 4);

   % Normalize time spans and interpolate knee angle data
   segment1_time = linspace(start_time, top_swing_time, num_points);
   segment2_time = linspace(top_swing_time, contact_time, num_points);
```

```
    segment3_time = linspace(contact_time, end_time, num_points);

    % Interpolate data for each segment
    knee_segments(trial, 1, :) = interp1(time, knee_angle_r, segment1_time);
    knee_segments(trial, 2, :) = interp1(time, knee_angle_r, segment2_time);
    knee_segments(trial, 3, :) = interp1(time, knee_angle_r, segment3_time);
end
% Calculate statistics across all trials
mean_values = squeeze(mean(knee_segments, 1));
std_values = squeeze(std(knee_segments, 1));
% Plot results
segment_labels = {'Start to Top Swing', 'Top Swing to Contact', 'Contact to
End'};
figure;
for segment = 1:3
    subplot(3, 1, segment);
    x = linspace(0, 1, num_points); % Normalized time
    plot(x, mean_values(segment, :), 'LineWidth', 1.5); hold on;
    fill([x, fliplr(x)], ...
         [mean_values(segment, :) + std_values(segment, :), ...
          fliplr(mean_values(segment, :) - std_values(segment, :))], ...
         'r', 'FaceAlpha', 0.3, 'EdgeColor', 'none');
    title(segment_labels{segment});
    xlabel('Normalized Time');
    ylabel('Knee Angle (deg)');
    legend('Mean', '± 1 STD');
    grid on;
end
% Save the plot
saveas(gcf, 'knee_angle_analysis_across_trials.png');
```

```
% Define column titles for 19 columns
column_titles = {
    'Pelvis Tilt', 'Pelvis List', 'Pelvis Rotation', ...
    'Hip Flexion L', 'Hip Flexion R', ...
    'Hip Adduction L', 'Hip Adduction R', ...
    'Knee Flexion L', 'Knee Flexion R', ...
    'Ankle Dorsiflexion L', 'Ankle Dorsiflexion R', ...
    'Subtalar Angle L', 'Subtalar Angle R', ...
    'MTJ Angle L', 'MTJ Angle R', ...
    'Toe Angle L', 'Toe Angle R', ...
    'Hip Rotation L', 'Hip Rotation R'
};
% Process only the first 19 columns (excluding "Time" column)
for col = 2:20
    segment_data = zeros(num_trials, 3, num_points);
    for trial = 1:num_files
        filename = sprintf('JackBuchanan%d.mot', trial);
        if ~isfile(filename)
```

```
            warning('File %s does not exist. Skipping...', filename);
            continue;
        end
        data = readmatrix(filename, 'FileType', 'text');
        if size(data, 2) < col
            warning('Column %d missing in file %s. Skipping...', col, filename);
            continue;
        end
        time = data(:, 1);
        current_data = data(:, col);
        % Interpolation
        segment1_time = linspace(segmentation_data(trial, 1),
segmentation_data(trial, 2), num_points);
        segment2_time = linspace(segmentation_data(trial, 2),
segmentation_data(trial, 3), num_points);
        segment3_time = linspace(segmentation_data(trial, 3),
segmentation_data(trial, 4), num_points);
        segment_data(trial, 1, :) = interp1(time, current_data, segment1_time);
        segment_data(trial, 2, :) = interp1(time, current_data, segment2_time);
        segment_data(trial, 3, :) = interp1(time, current_data, segment3_time);
    end
    mean_values = squeeze(mean(segment_data, 1));
    std_values = squeeze(std(segment_data, 1));
    % Plot results
    figure;
    segment_labels = {'Start to Top Swing', 'Top Swing to Contact', 'Contact to
End'};
    for segment = 1:3
        subplot(3, 1, segment);
        x = linspace(0, 1, num_points);
        plot(x, mean_values(segment, :), 'LineWidth', 1.5); hold on;
        fill([x, fliplr(x)], ...
             [mean_values(segment, :) + std_values(segment, :), ...
              fliplr(mean_values(segment, :) - std_values(segment, :))], ...
             'r', 'FaceAlpha', 0.3, 'EdgeColor', 'none');
        title([column_titles{col - 1}, ': ', segment_labels{segment}]);
        xlabel('Normalized Time');
        ylabel(column_titles{col - 1});
        legend('Mean', '± 1 STD');
        grid on;
    end
end


% Segmentation data for 20 trials
segmentation_data = [
  3.25, 4.08, 4.33, 4.8;
  1.00, 1.82, 2.07, 2.75;
  3.50, 4.28, 4.53, 5.08;
  3.13, 3.92, 4.18, 4.72;
  3.37, 4.17, 4.38, 4.93;
  2.42, 3.18, 3.40, 3.93;
  2.85, 3.63, 3.85, 4.55;
  2.40, 3.18, 3.45, 4.02;
```

```
  3.32, 4.20, 4.38, 3.70;
  2.33, 3.08, 3.37, 4.32;
  2.47, 3.27, 3.52, 3.98;
  3.15, 3.88, 4.13, 4.75;
  2.63, 3.30, 3.55, 4.13;
  2.90, 3.62, 3.87, 4.48;
  2.27, 3.02, 3.28, 3.83;
  1.68, 2.43, 2.68, 3.13;
  4.67, 5.45, 5.70, 6.55;
  2.42, 3.23, 3.43, 3.98;
  1.93, 2.73, 2.99, 3.85;
  3.20, 3.97, 4.23, 4.98
];
% Define column titles for the last 14 columns
column_titles = {
   'Pelvis Rotation', 'Trunk Flexion', 'Head Rotation', ...
   'Shoulder Flexion L', 'Shoulder Flexion R', ...
   'Elbow Flexion L', 'Elbow Flexion R', ...
   'Wrist Flexion L', 'Wrist Flexion R', ...
   'Pelvis Acceleration', 'Trunk Acceleration', ...
   'Hip Torque L', 'Hip Torque R', 'Knee Torque R'
};
% Initialize parameters
num_trials = size(segmentation_data, 1);
num_points = 100; % Normalized time points per segment
num_files = 20; % Total number of files
% Process the last 14 columns
for col = 20:33
   % Initialize storage for segmented data
   segment_data = zeros(num_trials, 3, num_points);
   for trial = 1:num_files
       % Load the .mot file
       filename = sprintf('JackBuchanan%d.mot', trial);
       data = readmatrix(filename, 'FileType', 'text');
       % Extract time and the current column data
       time = data(:, 1);
       current_data = data(:, col);
       % Segmentation times for this trial
       start_time = segmentation_data(trial, 1);
       top_swing_time = segmentation_data(trial, 2);
       contact_time = segmentation_data(trial, 3);
       end_time = segmentation_data(trial, 4);
       % Normalize time spans and interpolate data
       segment1_time = linspace(start_time, top_swing_time, num_points);
       segment2_time = linspace(top_swing_time, contact_time, num_points);
       segment3_time = linspace(contact_time, end_time, num_points);
       % Interpolate data for each segment
       segment_data(trial, 1, :) = interp1(time, current_data, segment1_time);
       segment_data(trial, 2, :) = interp1(time, current_data, segment2_time);
       segment_data(trial, 3, :) = interp1(time, current_data, segment3_time);
   end
   % Calculate statistics across all trials
   mean_values = squeeze(mean(segment_data, 1));
   std_values = squeeze(std(segment_data, 1));
```

```
    % Plot results
    segment_labels = {'Start to Top Swing', 'Top Swing to Contact', 'Contact to
End'};
    figure;
    for segment = 1:3
        subplot(3, 1, segment);
        x = linspace(0, 1, num_points); % Normalized time
        plot(x, mean_values(segment, :), 'LineWidth', 1.5); hold on;
        fill([x, fliplr(x)], ...
             [mean_values(segment, :) + std_values(segment, :), ...
              fliplr(mean_values(segment, :) - std_values(segment, :))], ...
             'r', 'FaceAlpha', 0.3, 'EdgeColor', 'none');
        % Access column_titles correctly
        title([column_titles{col - 19}, ': ', segment_labels{segment}]);
        xlabel('Normalized Time');
        ylabel(column_titles{col - 19});
        legend('Mean', '± 1 STD');
        grid on;
    end
    % Save the plot with the column title
    saveas(gcf, sprintf('%s_analysis_across_trials.png', column_titles{col -
19}));
end
```

MONTE CARLO

```
% Monte Carlo Analysis with Data Shuffling
clc; clear; close all;
% PARAMETERS
num_iterations = 1000; % Number of Monte Carlo iterations
joint_column = 16; % Example: Change to desired joint column
num_points = 100; % Normalized time points per segment
joint_name = 'Left Hip Rotation'; % Example joint name
% Load Data
% Assumes 'segmentation_file' and joint data files (.mot) are already available
segmentation_file = 'GolfTimeSegment.txt';
segmentation_data = readtable(segmentation_file, 'Delimiter', '\t', ...
   'ReadVariableNames', false, 'TextType', 'string');
% Extract subjects
subjects = unique(segmentation_data.Var2);
num_subjects = length(subjects);
% Initialize storage for group means
pro_means = [];
amateur_means = [];
% Classify subjects as Pro or Amateur
classification = containers.Map;
for s = 1:num_subjects
   subject_name = subjects{s};
   subject_prefix = segmentation_data.Var1(strcmp(segmentation_data.Var2, subject_name));
   if contains(subject_prefix(1), 'P') % 'P' for pro
       classification(subject_name) = 'Pro';
   else
       classification(subject_name) = 'Amateur';
   end
end
% Extract Original Group Data
data_all = []; group_labels = []; % Stores all data and labels for shuffling
```

```matlab
for s = 1:num_subjects
   subject_name = subjects{s};
   subject_rows = strcmp(segmentation_data.Var2, subject_name);
   subject_trials = segmentation_data(subject_rows, :);
   num_trials = sum(subject_rows);

   for t = 1:num_trials
       trial_number = subject_trials.Var3(t);
       start_time = subject_trials.Var4(t);
       end_time = subject_trials.Var7(t);

       % Load joint angle data for the trial
       filename = sprintf('%s%d.mot', replace(subject_name, " ", ""), trial_number);
       if ~isfile(filename), continue; end

       data = readmatrix(filename, 'FileType', 'text');
       time = data(:, 1);
       joint_angle = data(:, joint_column);

       % Normalize and interpolate the joint angle over 100 points
       normalized_time = linspace(start_time, end_time, num_points);
       interpolated_angle = interp1(time, joint_angle, normalized_time, 'linear', 'extrap');

       % Store the joint angle data and labels
       data_all = [data_all; interpolated_angle];
       if strcmp(classification(subject_name), 'Pro')
           group_labels = [group_labels; 1]; % 1 = Pro
           pro_means = [pro_means; mean(interpolated_angle)];
       else
           group_labels = [group_labels; 2]; % 2 = Amateur
           amateur_means = [amateur_means; mean(interpolated_angle)];
       end
   end
end
% Observed Difference in Means
observed_diff = abs(mean(pro_means) - mean(amateur_means));
% Monte Carlo Simulation
shuffled_diffs = zeros(num_iterations, 1);
for i = 1:num_iterations
   % Shuffle group labels
   shuffled_labels = group_labels(randperm(length(group_labels)));

   % Recompute group means with shuffled labels
   shuffled_pro_means = mean(data_all(shuffled_labels == 1, :), 'all');
   shuffled_amateur_means = mean(data_all(shuffled_labels == 2, :), 'all');

   % Store the shuffled difference
   shuffled_diffs(i) = abs(shuffled_pro_means - shuffled_amateur_means);
end
% P-value Calculation
p_value = sum(shuffled_diffs >= observed_diff) / num_iterations;
% Plot Results
figure;
histogram(shuffled_diffs, 30, 'FaceColor', [0.2, 0.4, 0.6], 'EdgeColor', 'none');
hold on;
yline(observed_diff, 'r--', 'LineWidth', 2, 'Label', 'Observed Difference');
title(['Monte Carlo Analysis for ', joint_name]);
xlabel('Shuffled Group Difference');
ylabel('Frequency');
legend('Shuffled Differences', 'Observed Difference');
grid on;
% Display Results
fprintf('Observed Difference: %.4f\n', observed_diff);
fprintf('Monte Carlo P-value: %.4f\n', p_value);
if p_value < 0.05
   fprintf('Statistically significant difference detected.\n');
else
   fprintf('No statistically significant difference detected.\n');
end
```

Plots - aggregate plots with STD regions

```
%% Golf Group Curve Code
clc;
clear;
close all;
% Define the column index and name for the joint to process
joint_column = 27; % Change this value to the desired column in the .mot file
joint_name = 'Right Arm Rotation'; % Change this to the name of the joint
% Read segmentation data from the file
segmentation_file = 'GolfTimeSegment.txt';
segmentation_data = readtable(segmentation_file, 'Delimiter', '\t',
'ReadVariableNames', false, 'TextType', 'string');
% Filter valid rows (those with complete segmentation data)
valid_rows = all(~ismissing(segmentation_data{:, 4:7}), 2);
segmentation_data = segmentation_data(valid_rows, :);
% Extract unique subjects
subjects = unique(segmentation_data.Var2);
% Initialize parameters
num_points = 100; % Number of normalized points per segment
aggregate_data = containers.Map; % To store aggregate data
% Classify subjects as Pro or Amateur
classification = containers.Map; % Map for subject classification
pro_color = [0, 0.4470, 0.7410]; % Blue for pros
amateur_color = [1, 0, 0]; % Red for amateurs
% Process each subject
for s = 1:length(subjects)
   subject_name = subjects{s};

   % Determine if the subject is a Pro or Amateur
   subject_prefix = segmentation_data.Var1(strcmp(segmentation_data.Var2,
subject_name));
```

```
    if contains(subject_prefix(1), 'P')
        classification(subject_name) = 'Pro';
    else
        classification(subject_name) = 'Amateur';
    end

    % Extract rows for the current subject
    subject_rows = strcmp(segmentation_data.Var2, subject_name);
    subject_trials = segmentation_data(subject_rows, :);
    num_trials = sum(subject_rows);

    % Preallocate storage for subject data
    joint_segments = zeros(num_trials, 3, num_points);

    % Process each trial for the current subject
    for t = 1:num_trials
        trial_number = subject_trials.Var3(t);
        start_time = subject_trials.Var4(t);
        top_swing_time = subject_trials.Var5(t);
        contact_time = subject_trials.Var6(t);
        end_time = subject_trials.Var7(t);

        % Generate filename for the trial
        filename = sprintf('%s%d.mot', replace(subject_name, " ", ""),
trial_number);

        % Load .mot file data
        data = readmatrix(filename, 'FileType', 'text');
        time = data(:, 1); % Time column
        joint_angle = data(:, joint_column); % Use specified column for joint
angle

        % Interpolate data for each segment
        segment1_time = linspace(start_time, top_swing_time, num_points);
        segment2_time = linspace(top_swing_time, contact_time, num_points);
        segment3_time = linspace(contact_time, end_time, num_points);

        joint_segments(t, 1, :) = interp1(time, joint_angle, segment1_time);
        joint_segments(t, 2, :) = interp1(time, joint_angle, segment2_time);
        joint_segments(t, 3, :) = interp1(time, joint_angle, segment3_time);
    end

    % Calculate mean and standard deviation
    mean_values = squeeze(mean(joint_segments, 1));
    std_values = squeeze(std(joint_segments, 1));

    % Store aggregate data
    aggregate_data(subject_name) = struct('mean', mean_values, 'std',
std_values);

    % Plot individual subject data
    segment_labels = {'Start to Top Swing', 'Top Swing to Contact', 'Contact to
End'};
    figure;
```

```matlab
    for segment = 1:3
        subplot(3, 1, segment);
        x = linspace(0, 1, num_points);
        plot(x, mean_values(segment, :), 'LineWidth', 1.5); hold on;
        fill([x, fliplr(x)], ...
             [mean_values(segment, :) + std_values(segment, :), ...
              fliplr(mean_values(segment, :) - std_values(segment, :))], ...
             'r', 'FaceAlpha', 0.3, 'EdgeColor', 'none');
        title([subject_name ' - ' joint_name ' - ' segment_labels{segment}]);
        xlabel('Normalized Time');
        ylabel([joint_name ' (deg)']);
        legend('Mean', '± 1 STD');
        grid on;
    end
    saveas(gcf, sprintf('%s_%s_analysis.png', replace(subject_name, " ", ""),
replace(joint_name, " ", "_")));
end
% Create aggregate plots for all subjects with Pro and Amateur color coding
figure;
segment_labels = {'Start to Top Swing', 'Top Swing to Contact', 'Contact to
End'};
% Loop through each segment
for segment = 1:3
    subplot(3, 1, segment);
    hold on;
    x = linspace(0, 1, num_points); % Normalized time

    for s = 1:length(subjects)
        subject_name = subjects{s};
        data = aggregate_data(subject_name);

        % Assign color based on classification
        if strcmp(classification(subject_name), 'Pro')
            fill_color = pro_color; % Blue for Pros
        else
            fill_color = amateur_color; % Red for Amateurs
        end

        % Plot the ±1 STD region using fill
        upper_bound = data.mean(segment, :) + data.std(segment, :);
        lower_bound = data.mean(segment, :) - data.std(segment, :);

        fill([x, fliplr(x)], [upper_bound, fliplr(lower_bound)], ...
             fill_color, 'FaceAlpha', 0.3, 'EdgeColor', 'none');
    end

    % Customize subplot
    title(['Aggregate - ' joint_name ' - ' segment_labels{segment}]);
    xlabel('Normalized Time');
    ylabel([joint_name ' (deg)']);
    grid on;
end
% Add legend for classification
subplot(3, 1, 1);
```

```
legend({'Pro ±1 STD', 'Amateur ±1 STD'}, 'Location', 'best', 'TextColor',
'black');
hold off;
% Save the aggregate plot
saveas(gcf, sprintf('aggregate_overlay_color_coded_%s.png', replace(joint_name,
" ", "_")));
```