Movement Neuroscience:

Exercise 1(Report)

Name: Harshvardhan Joshi
Idm: og52yvov
M.N.: 23162173

Task 1: Data Preprocessing and Visualization

Load the Dataset:

FILE NAVIGATE CODE ANALYZE SECTION RUN

```
% Task 1: Data preprocessing and visualisation

% Load the dataset
load('Slow_Contraction.mat');
ConversionFactor = 0.02;
Gravity = 9.81; % Acceleration due to gravity g

```

1.1 Convert Force Signal to Newtons

```

% 1.1 Convertion of force signal to Newtons
Force_N_Slow = ref_signal * ConversionFactor * Gravity;
time_vector_slow = (0:1/fsamp:(length(ref_signal)-1)/fsamp);

```

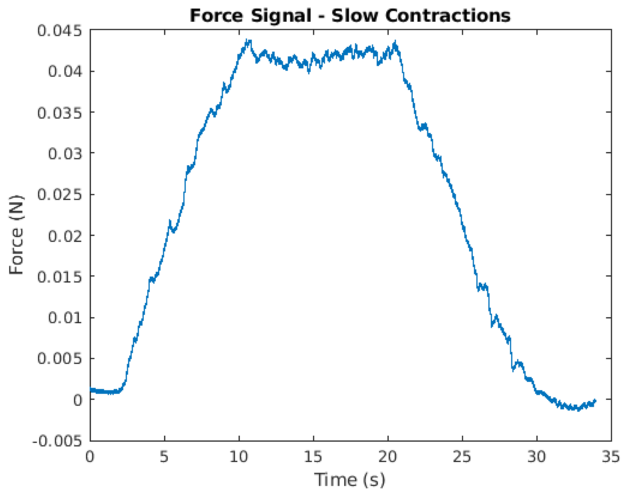
```
Force_N_Slow = ref_signal * ConversionFactor * Gravity;
time_ector_slow = (0:1/fsamp:(length(ref_signal)-1)/fsamp);
```

1.2 Plot Force Signal in Newton

```
% 1.2 Plot of force signal in Newtons
figure;
plot(time_vector_slow, Force_N_Slow);
xlabel('Time (s)');
ylabel('Force (N)');
title('Force Signal - Slow Contractions');

```

```
figure;
plot(time_vector_slow, Force_N_Slow);
xlabel('Time (s)');
ylabel('Force (N)');
title('Force Signal - Slow Contractions');
```



The force signal shows the following patterns:

- No Force (0 N): From 0 to 2.5 seconds, there is no force observed.
- Gradual Increase: Between 2.5 and 10 seconds, there is a gradual increase in force.
- Plateau (Above 1000 N): From 10 to 20 seconds, there is a plateau with a force above 1000 N.
- Fall: Between 20 and 30 seconds, there is a fall in force.
- Gradual decrease: From 30 to 35 seconds, the force pattern is similar to the period from 2.5 to 10 seconds but is decreasing gradually.
- No Force (0 N): Beyond 35 seconds, the force remains at or below zero.

1.3 Filtering the Force Signal

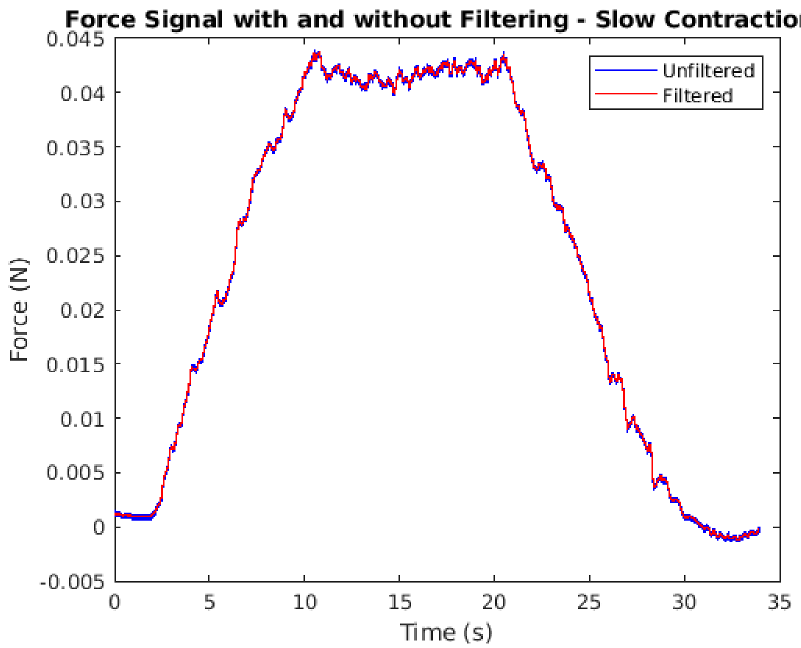
```

% 1.3 Filtering the force signal
cutoff_frequency_slow = 10; % Set cutoff frequency in Hz
[b_slow, a_slow] = butter(4, cutoff_frequency_slow / (fsamp / 2), 'low');
Filtered_Force_N_Slow = filtfilt(b_slow, a_slow, Force_N_Slow);

% Plot of unfiltered and filtered signals
figure;
plot(time_vector_slow, Force_N_Slow, 'b');
hold on;
plot(time_vector_slow, Filtered_Force_N_Slow, 'r');
xlabel('Time (s)');
ylabel('Force (N)');
title('Force Signal with and without Filtering - Slow Contractions');
legend('Unfiltered', 'Filtered');
```

```
cutoff_frequency_slow = 10; % Set cutoff frequency in Hz
[b_slow, a_slow] = butter(4, cutoff_frequency_slow / (fsamp / 2), 'low');
Filtered_Force_N_Slow = filtfilt(b_slow, a_slow, Force_N_Slow);
% Plot of unfiltered and filtered signals
figure;
plot(time_vector_slow, Force_N_Slow, 'b');
hold on;
```

```
plot(time_vector_slow, Filtered_Force_N_Slow, 'r');
xlabel('Time (s)');
ylabel('Force (N)');
title('Forceignal with and without Filtering - Slow Contractions');
legend('Unfiltered', 'Filtered');
```

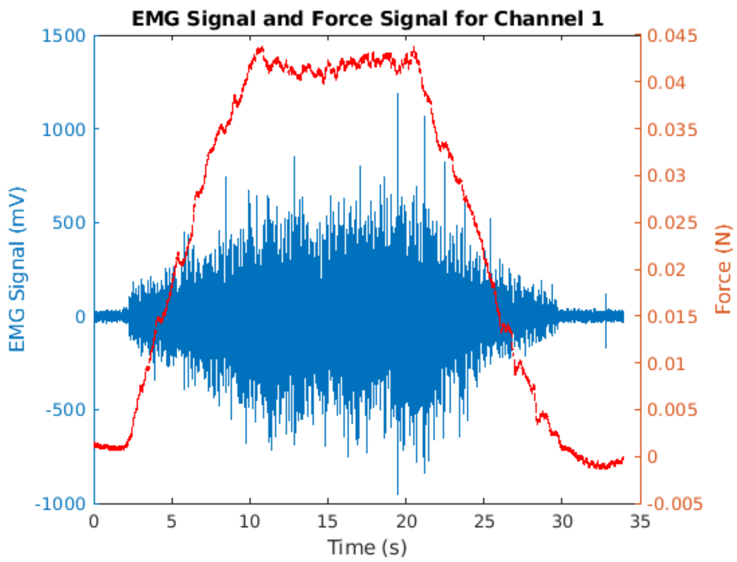


1.4 Plot Exemplary Channel of EMG Data

```
% 1.4 Plot of exemplary channel of the EMG data with force signal
channel_slow = 1;
figure;
yyaxis left;
plot(time_vector_slow, SIG{channel_slow});
ylabel('EMG Signal (mV)');

% Plot of selected channel of the EMG data and the force signal
yyaxis right;
plot(time_vector_slow, Force_N_Slow, 'r--');
ylabel('Force (N)');
xlabel('Time (s)');
title(['EMG Signal and Force Signal for Channel ' num2str(channel_slow)]);
```

```
channel_slow = 1;
figure;
yyaxis left;
plot(time_vector_slow, SIG{channel_slow});
ylabel('EMG Signal (mV)');
% Plot of the selected channel of the EMG data and the force signal
yyaxis right;
plot(time_vector_slow, Force_N_Slow, 'r--');
ylabel('Force (N)');
xlabel('Time (s)');
title(['EMG Signal and Force Signal for Channel ' num2str(channel_slow)]);
```



The relationship between force and EMG signal :

- **Force Plateau (10-25 seconds):** EMG shows sustained or varying patterns, indicating consistent neural activation during the force plateau.
- **Force Rise and Fall (5-10 seconds, 25-30 seconds):** Force rise to increased EMG amplitude, while force fall with decreased EMG activity.
- **Zero Force Intervals (0-5 seconds, 25-30 seconds):** Periods of low force i.e. low or flat EM.
- **EMG Peaks (5-25 seconds):** Peaks in the EMG signal coincide with force variations, reflecting changes in the neural drive.
- **EMG in Constant Force Zones (0-2.5 seconds, 2.5-5 seconds, 30-35 seconds):** Near-zero EMG can be seen during constant force intervals.

Task 2: Force Steadiness

2.1 Calculate the Coefficient of Variation (CV) :

```
% Task 2. Force steadiness
% 2.1 Force Steadiness and coefficient of variation (CV)
plateau_start_time_slow = 10; % seconds
plateau_end_time_slow = 20; % seconds
force_plateau_slow = Force_N_Slow(time_vector_slow >= plateau_start_time_slow & time_vector_slow <= plateau_end_time_slow);
% Calculating CV for Slow Contractions
CV_slow = std(force_plateau_slow) / mean(force_plateau_slow) * 100;
fprintf('Coefficient of Variation (CV) during plateau phase for Slow Contractions: %.2f%%\n', CV_slow);
```

```
plateau_start_time_slow = 10; % seconds
plateau_end_time_slow = 20; % seconds
force_plateau_slow = Force_N_Slow(time_vector_slow >= plateau_start_time_slow &
time_vector_slow <= plateau_end_time_slow);

% Calculating CV for Slow Contractions
CV_slow = std(force_plateau_slow) / mean(force_plateau_slow) * 100;
printf('Coefficient of Variation (CV) during plateau phase for Slow Contractions:
%.2f%%\n', CV_slow);
```

O/P: Coefficient of Variation (CV) during plateau phase for Slow Contractions: 1.75%

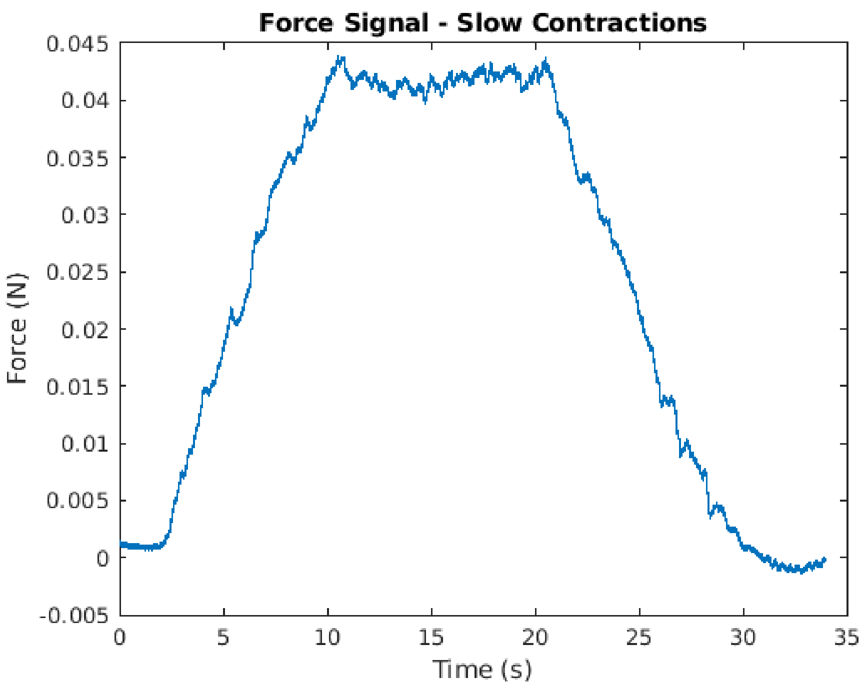
2.2 What does it mean physiologically?

The coefficient of variation is 1.75%.

A low CV means the subject can maintain stable and consistent force during the plateau, Which further indicates control and steadiness in force generation. Also, controlled force production shows the subject's ability to maintain neuromuscular stability.
The analysis focuses on the performance of displaying the subject's ability to maintain a steady force output with accuracy and control.

Task 3: Rate of Force Development for Rapid Contractions

3.1 Preprocessing and visualizing force signals :



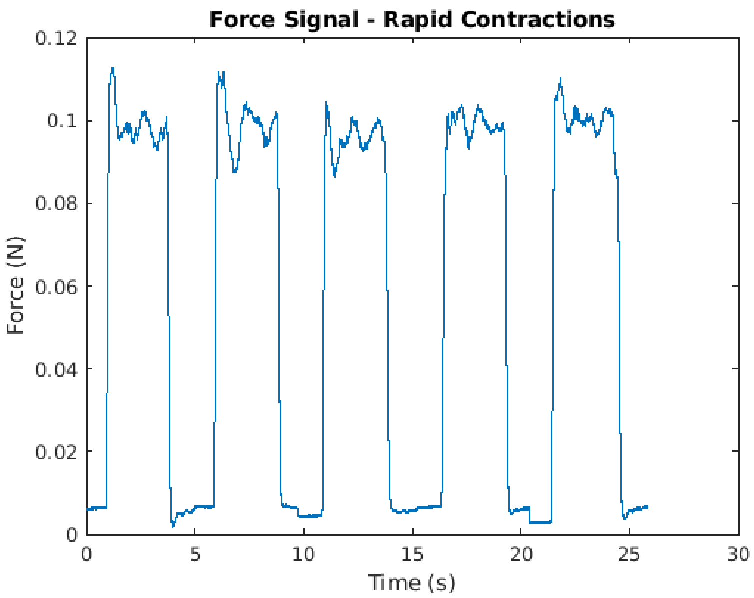
3.2 Plotting Force Signal for Rapid Contractions

```
% Task 3.1: Load the dataset
load('Rapid_Contractions.mat');
ConversionFactor_rapid = 0.02;
Gravity_rapid = 9.81; % Acceleration due to gravity g

% Converting force signal to Newtons
Force_N_Rapid = ref_signal * ConversionFactor_rapid * Gravity_rapid;

% Obtain time vector
time_vector_rapid = (0:1/fsamp:(length(ref_signal)-1)/fsamp);

% Plotting force signal for Rapid Contractions
figure;
plot(time_vector_rapid, Force_N_Rapid);
xlabel('Time (s)');
ylabel('Force (N)');
title('Force Signal - Rapid Contractions');
```



Calculating Rate of Force Development (RFD)

```
% Task 3.2 :

% Onset times for each contraction
total_contractions_rapid = 5;
onset_times_rapid = linspace(5, 25, total_contractions_rapid);

rfd_window_rapid = 0.5; % 5 seconds
rfd_values_rapid = zeros(1, length(onset_times_rapid));

for i = 1:length(onset_times_rapid)
    window_indices_rapid = time_vector_rapid >= onset_times_rapid(i) & time_vector_rapid <= (onset_times_rapid(i) + rfd_window_rapid);
    rfd_values_rapid(i) = mean(diff(Force_N_Rapid(window_indices_rapid))) / mean(diff(time_vector_rapid(window_indices_rapid)));
end

% Visualizing RFD values for Rapid Contractions
figure;
bar(rfd_values_rapid);
xlabel('Contractions');
ylabel('Rate of Force Development (N/s)');
title('Rate of Force Development for Each Contraction - Rapid Contractions');
```

3.3 In what demographics/populations could the RFD be important?

The relevance of calculating the Rate of Force Development is its ability to offer insights, into the performance of the neuromuscular system. It specifically helps us understand how quickly muscles can generate force.

Athletes:
RFD is essential in sports that require rapid and explosive movements (e.g. weightlifting, high jump).
Improved RFD improves an athlete's ability to generate quick and powerful movements, resulting in better overall performance.

Elderly Population:
RFD in old people can be used for understanding age-related changes in muscle function. Enhancing RFD through interventions can play a role, in preserving the elderly's functional independence and reducing the risk of falls.

Clinical Populations:
Individuals with neuromuscular disorders or neurological conditions may experience deficits in RFD.
Monitoring RFD can improve the designing of strategies(like therapy) to improve overall motor function.

Strength and Conditioning Programs:
In strength and conditioning, RFD plays a role, in designing training programs aimed to optimize the development of explosive strength.
Customized interventions informed by RFD data can majorly benefit individuals who want to enhance their power and speed capabilities.

Task 4: EMG Data Analysis

4.1 Convert SIG Cell Array to 2D Array and Compute Average

```
% Task 4.1: Converting SIG cell array to a 2D array and compute average across channels

% Function to normalize data
normalize_data = @(data) (data - mean(data)) / std(data);

% Slow Contractions
data_size_slow = size(SIG{1});
emg_data_slow = zeros(length(SIG), data_size_slow(2));

% Convert SIG cell array to a 2D array
for i = 1:length(SIG)
    current_size = size(SIG{i});
    emg_data_slow(i, :) = [SIG{i}, zeros(1, data_size_slow(2) - current_size(2))];
end

% Computing average across channels
average_emg_slow = mean(emg_data_slow, 1);

% Rapid Contractions
data_size_rapid = size(SIG{1});
emg_data_rapid = zeros(length(SIG), data_size_rapid(2));

% Assuming the same size for rapid contractions
for i = 1:length(SIG)
    emg_data_rapid(i, :) = [SIG{i}(1:current_size(2)), zeros(1, data_size_rapid(2) - current_size(2))];

end

% Computing average across channels
average_emg_rapid = mean(emg_data_rapid, 1);

```

```
normalize_data = @(data) (data - mean(data)) / std(data);
% Slow Contractions
data_size_slow = size(SIG{1});
emg_data_slow = zeros(length(SIG), data_size_slow(2));
for i = 1:length(SIG)
 current_size = size(SIG{i});
 emg_data_slow(i, :) = [SIG{i}, zeros(1, data_size_slow(2) - current_size(2))];
end
average_emg_slow = mean(emg_data_slow, 1);
% Rapid Contractions
data_size_rapid = size(SIG{1});
emg_data_rapid = zeros(length(SIG), data_size_rapid(2));
for i = 1:length(SIG)
 emg_data_rapid(i, :) = [SIG{i}(1:current_size(2)), zeros(1, data_size_rapid(2) -
current_size(2))];
end
average_emg_rapid = mean(emg_data_rapid, 1);
```

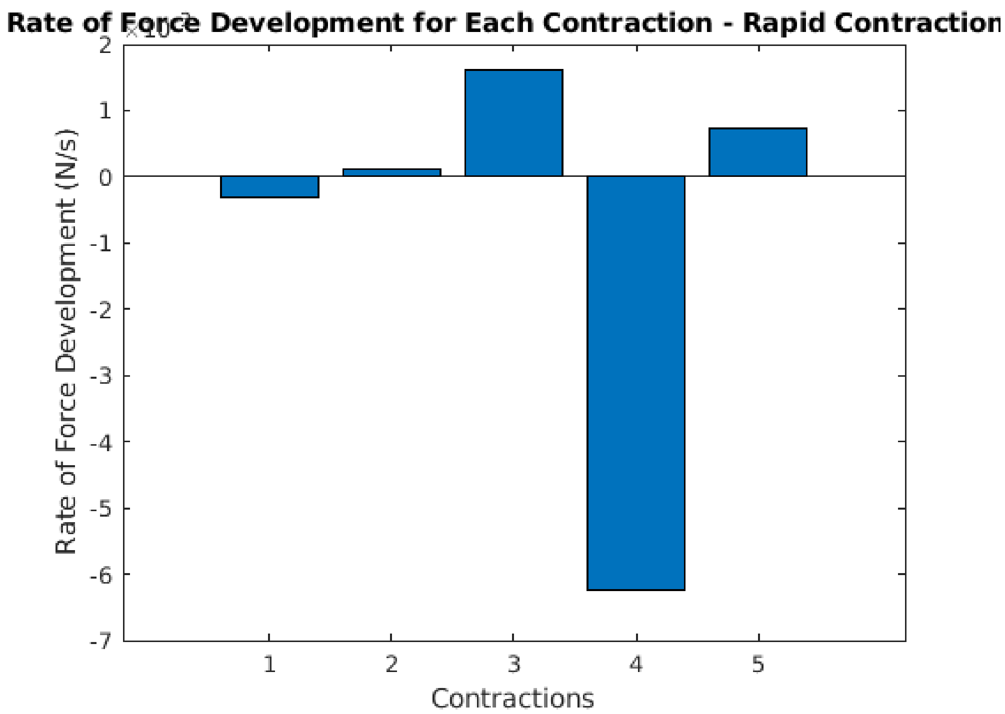
4.2 Compute RMS with Moving Average

```
% Task 4.2: Computing the RMS of the average as a moving average with a window length of 200 ms

% RMS window length in samples
rms_window_length = round(0.2 * fsamp);

% Computing the RMS for both datasets
rms_slow = rms(movmean(average_emg_slow .^ 2, rms_window_length));
rms_rapid = rms(movmean(average_emg_rapid .^ 2, rms_window_length));
```

```
rms_window_length = round(0.2 * fsamp);
rms_slow = rms(movmean(average_emg_slow .^ 2, rms_window_length));
rms_rapid = rms(movmean(average_emg_rapid .^ 2, rms_window_length));
```



4.3 Visualising the Correlation between RMS and Force

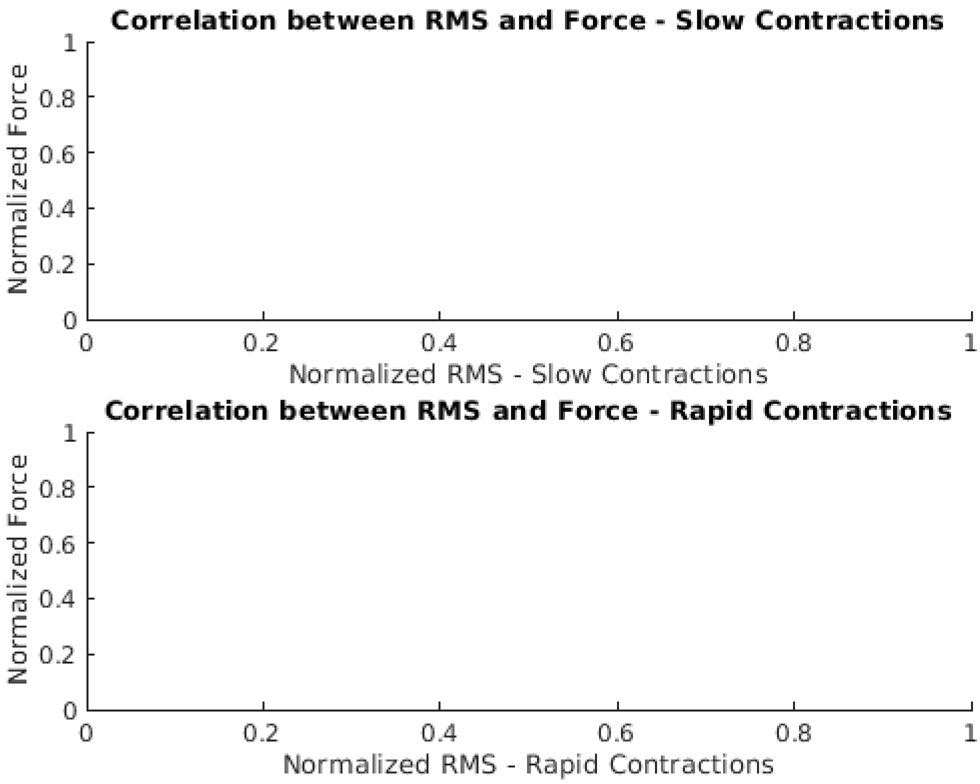
```
% Task 4.3: Visualizing the correlation between RMS and the respective force signali

% Slow Contractions
normalized_rms_slow = normalize_data(rms_slow);
normalized_force_slow = normalize_data(Force_N_Slow(1:length(rms_slow))); % Ensure lengths match

% Rapid Contractions
normalized_rms_rapid = normalize_data(rms_rapid);
normalized_force_rapid = normalize_data(Force_N_Rapid(1:length(rms_rapid))); % Ensure lengths match


% Scatter plot
figure;
subplot(2, 1, 1);
scatter(normalized_rms_slow, normalized_force_slow);
xlabel('Normalized RMS - Slow Contractions');
ylabel('Normalized Force');
title('Correlation between RMS and Force - Slow Contractions');

subplot(2, 1, 2);
scatter(normalized_rms_rapid, normalized_force_rapid);
xlabel('Normalized RMS - Rapid Contractions');
ylabel('Normalized Force');
title('Correlation between RMS and Force - Rapid Contractions');
```



Report Correlation Coefficients

```
% Compute and report the correlation coefficient R
correlation_slow = corrcoef(normalized_rms_slow, normalized_force_slow, 'Rows', 'complete');
disp('Size of correlation_slow:');
disp(size(correlation_slow));

% Check if correlation_slow is a scalar
if isscalar(correlation_slow)
    correlation_slow = correlation_slow(1);
end

correlation_rapid = corrcoef(normalized_rms_rapid, normalized_force_rapid, 'Rows', 'complete');
disp('Size of correlation_rapid:');
disp(size(correlation_rapid));

% Check if correlation_rapid is a scalar
if isscalar(correlation_rapid)
    correlation_rapid = correlation_rapid(1);
end

fprintf('Correlation coefficient R - Slow Contractions: %.2f\n', correlation_slow);
fprintf('Correlation coefficient R - Rapid Contractions: %.2f\n', correlation_rapid);
```

```
low = corrcoef(normalized_rms_slow, normalized_force_slow, 'Rows', 'complete');
disp('Size of correlation_slow:'nd

ation_rapid = corrcoef(normalized_rms_rapid, normalized_force_rapid, 'Rows',
'complete');
disp('Size of correlation_rapid:');
disp(size(correlation_rapid)

if isscalar(correlation_rapid)
 correlation_rapid = correlation_rapid(1);
end

fprintf('Correlation coefficient R - Slow Contractions: %.2f\n', correlation_slow
```