Quantifying Asymmetry and Performance of Lower Limb Mechanical Muscle Function in Varsity Athletes

by

Meredith Stadnyk

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Science

Department of Biomedical Engineering

University of Alberta

© Meredith Stadnyk, 2019

Abstract

In varsity sports, athletes are more susceptible to lower limb injury than the general population due to their continuous involvement in physically-demanding activity. Additionally, these athletes participate in high-level strength and conditioning training programs to ensure they are reaching their optimal performance in their sport. To better understand what should be expected when athletes either (1) have a lower limb injury, or (2) are not performing to their full lower limb function potential, we first need to understand what "normal" or "optimal" lower limb mechanical muscle function is in each sport. This project looked at a means of quantifying the lower limb mechanical muscle function instantaneously to provide information for rehabilitation and performance purposes in athletes of specific sports.

Male and female varsity athletes from swim, volleyball, rugby, and soccer completed five trials of a counter movement jump (CMJ) on dual force plates. An analysis program was written in Wolfram Mathematica to analyze force-time jump data with minimal equipment and labor required. Various parameters of interest were generated, including: peak force, force-time curve shape classification, jump phase lengths (for eccentric and concentric phases), phase-specific kinetic impulse, asymmetry index, takeoff velocity, jump height, phase-specific center of mass displacements, and reactive strength index modified (RSImod).

Male participants had a higher peak force than females, but no significant difference when peak force was normalized to body mass. Males took longer than females to takeoff, spent more time in the air, and had a higher takeoff velocity. Additionally, males had larger kinetic impulses compared to females. Males had a higher jump height and RSImod, which represents jump "explosiveness." There were no differences between asymmetry indexes observed between the genders.

The soccer athletes were able to exert the highest peak force per kilogram of body mass out of all the groups. The soccer group also spent the shortest time in the eccentric and concentric jump phases, and had the smallest impulses. The swim athletes had the lowest peak force out of all the group, and spent the most time in the jump phases. The swim athletes had the lowest RSImod out of all the sport groups. Volleyball players had the highest absolute peak force, spent the longest time in the air, and had the highest jump height and takeoff velocity out of all the groups. Volleyball players also had the largest kinetic impulses in both jump phases. The rugby group did not have any parameters that were significantly different from all the other sport groups, indicating that it may be difficult to determine what a "normal" rugby players' jump performance is.

A follow-up study was conducted to incorporate non-counter movement jumps (NCMJs) into assessing jump characteristics of the same sports as the CMJ study. In the NCMJ, swimmers had the lowest peak force normalized to body mass, and took the longest time to complete the jumps. Volleyball players spent the longest time in the air, had the highest takeoff velocity, the highest jump height, and the highest RSImod (although these were not significant with all other sport groups). Soccer players had the lowest body mass (although only significant with volleyball and rugby). Rugby players again lacked statistically significant results from the all other sport groups for any one specific variable for many of the NCMJ variables, and many of their results averaged somewhere in the middle of the other sport groups'. The results of these studies do not necessarily indicate that some athletes are "better" jumpers or have "stronger" lower limbs, but that differences do exist between sports for lower limb muscle usage.

Preface

This thesis is an original work by Meredith Stadnyk. No part of this thesis has been previously published. The research project, of which this thesis is a part, received research ethics approval from the University of Alberta Research Ethics Board, Project Name "Jump Phases in Athletes", Pro00071204, March 7, 2017.

Acknowledgements

I would like to thank my supervisor, Dr. Samer Adeeb, for his patience, guidance and advice throughout this project. Samer was a major factor in my opportunities to partake in research projects during my undergraduate degree—which translated into a strong interest in the area of biomechanical engineering, and eventually, my enrolment in this program. Samer has continually believed in me when I doubted myself, and constantly challenges me to grow and achieve more. I am also grateful to Dr. Lindsey Westover for her willingness to provide advice, clarity and another perspective throughout this project when I need it most.

I would like to express my sincere appreciation to Michael Cook and Eric Golberg of the University of Alberta Sports Performance Centre for all their help in data collection, definition of the project, and insight into the world of strength and conditioning. I would like to thank all the athletes who participated in this project and made it possible to collect data from so many individuals, as well as their coaches for permitting them to participate.

I would like to thank Joseph Sabbagh, Yvonne Qiu, and Anthony Pham for their help in exploring the use of Mathematica to analyze counter movement jumps and non-counter movement jumps. I would also like to specifically thank Joseph Sabbagh for guidance in writing the VBA code to quickly format datasets—something that I'm sure has saved me many hours of work.

In addition, I would like to thank the many individuals in my life for providing support throughout all these years of my studies (and the years to come). I would especially like to thank my parents, John and Janet, to whom I owe an endless debt of gratitude for providing me with so many opportunities in life.

Thank you to the Natural Sciences and Engineering Research Council for the funding through the Canada Graduate Scholarships – Master's Program, which allowed me focus on completing my degree with a much lesser financial burden.

Table of Contents

Abstract ii

Preface iv

Acknowledgements v

Table of Contents vi

List of Tables ix

List of Figures x

Chapter 1: Introduction 1

- 1.1 The Counter Movement Jump 1
- 1.2 Jump Phases 3
- 1.3 Equations 3
  - 1.3.1 Kinetic Impulse 3
  - 1.3.1 Kinematic equations 4
  - 1.3.3 Jump Height 4
  - 1.3.4 Reactive Strength Index Modified 7
  - 1.3.5 Peak Force Percent Difference 7
- 1.4 Description of Rugby, Soccer, Swim and Volleyball 7
  - 1.4.1 Rugby 7
  - 1.4.2 Soccer 8
  - 1.4.3 Swimming 8
  - 1.4.4 Volleyball 9
- 1.5 Objectives 9

Chapter 2: Quantifying Lower Limb Mechanical Muscle Function during Counter Movement Jumps 11

- 2.1 Methods 11
  - 2.1.1 Participants 11
  - 2.1.2 Experimental Procedures 11

2.1.3 Mathematica Analysis 12

2.2 Results 18

2.2.1 Mass and Force 18

2.2.2 Force-Time Curve Shape 22

2.2.3 Phase Lengths 23

2.2.4 Impulse 27

2.2.5 Displacement and Velocity 31

2.3 Discussion 35

2.3.1 Rugby 35

2.3.2 Soccer 38

2.3.3 Swimming 41

2.3.4 Volleyball 46

2.3.5 Gender 51

2.3.6 Mass and Force 53

2.3.7 Jump Phases 55

2.3.8 Force-Time Curve Shape 58

2.3.9 Impulse 61

2.3.10 Jump Height 62

2.3.11 RSImod 63

2.3.12 Asymmetry 64

2.3.13 Depth of CMJ 66

2.3.14 Plyometric Training 67

2.3.15 Limitations 69

Chapter 3: Use of the Non-Counter Movement Jump in Quantifying Lower Limb Mechanical Muscle Function 72

3.1 Methods 72

3.2 Results 75

3.2.1 Mass and Force........76

3.2.2 Force-Time Curve Shape........79

3.2.3 Phase Lengths........79

3.2.4 Displacement and Velocity........82

3.3 Discussion........84

Chapter 4: Applications and Conclusion........90

4.1 Applications to Injury Rehabilitation........90

4.2 Applications to Performance Improvements........91

4.3 Conclusion........92

Bibliography........94

Appendices........98

Appendix A. CMJ Mass and Force Results........98

Appendix B. CMJ Jump Class Results........100

Appendix C. CMJ Phase Lengths Results........101

Appendix D. CMJ Impulse Results........106

Appendix E. CMJ Displacement and Velocity Results........110

Appendix F. Visual Basic Code for Formatting Data (for CMJ and NCMJ)........115

Appendix G. CMJ Mathematica Code........120

Appendix H. NCMJ Mass and Force Results........157

Appendix I. NCMJ Jump Class Results........160

Appendix J. NCMJ Phase Length Results........161

Appendix K. NCMJ Displacement and Velocity Results........163

Appendix L. NCMJ Mathematica Code........166

List of Tables

Table 1: Participants' Sport and Gender 11
Table 2: Counter Movement Jump Time Point Definitions 13
Table 3: Counter Movement Jump Interval Definitions 13
Table 4: Leg Exerting the Greater Peak Force Compared to the Participant's Dominant Leg: Number and Percent 21
Table 5: Peak Force Difference calculated between Legs (%) 22
Table 6: Jump Classification of Majority of Jump Trials 22
Table 7: Participants' Sport and Gender 73
Table 8: Non-Counter Movement Jump Time Point Definitions 74
Table 9: Jump Classification of Majority of Jump Trials 79

List of Figures

Figure 1: Counter Movement Jump - Time Points and Intervals 13

Figure 2: Examples of Jumps with a) One Peak – Class 1 and b) Two Peaks – Class 2 16

Figure 3: The Normal Force Function, First Derivative and Second Derivatives of the Examples in Figure 2 from t_3 to t_4 for a a) Class 1 Jump, and b) Class 2 Jump 17

Figure 4: Mass by a) gender and b) sport. 19

Figure 5: Peak Force by a) gender and b) sport. 19

Figure 6: Normalized Peak Force by a) gender and b) sport. 20

Figure 7: Peak Force Difference between Legs by a) gender and b) sport. 21

Figure 8: Eccentric Phase Time by a) gender and b) sport. 23

Figure 9: Concentric Phase Time by a) gender and b) sport. 24

Figure 10: Unloading Phase Time by a) gender and b) sport. 24

Figure 11: Loading Phase Time by a) gender and b) sport. 25

Figure 12: Final Phase Time by a) gender and b) sport. 25

Figure 13: Time to Takeoff by a) gender and b) sport. 26

Figure 14: Time in Air by a) gender and b) sport. 27

Figure 15: Eccentric Phase Impulse by a) gender and b) sport. 27

Figure 16: Concentric Phase Impulse by a) gender and b) sport. 28

Figure 17: Normalized Eccentric Phase Impulse by a) gender and b) sport. 29

Figure 18: Normalized Concentric Phase Impulse by a) gender and b) sport. 29

Figure 19: Asymmetry Index – Eccentric Phase by a) gender and b) sport. 30

Figure 20: Asymmetry Index – Concentric Phase by a) gender and b) sport. 30

Figure 21: Takeoff Velocity by a) gender and b) sport. 31

Figure 22: Jump Height – TIA Method by a) gender and b) sport. 32

Figure 23: Jump Height – IMM Method by a) gender and b) sport. 32

Figure 24: Peak (Negative) Eccentric Displacement by a) gender and b) sport. 33

Figure 25: Peak (Positive) Concentric Displacement by a) gender and b) sport. 34

Figure 26: RSImod – Calculated with TIA by a) gender and b) sport. 34

Figure 27: RSImod – Calculated with IMM by a) gender and b) sport. 35

Figure 28: Example of a Participant with a Class 1 (in Yellow, Right Leg) and Class 2 (in Blue, Left Leg) Jump 59

Figure 29: Example of two counter movement jumps with the same eccentric phase impulse, but vastly different jumps 61

Figure 30: Non-Counter Movement Jump Time Points 75
Figure 31: Mass by a) gender and b) sport. 76
Figure 32: Peak Force by a) gender and b) sport. 77
Figure 33: Normalized Peak Force by a) gender and b) sport. 77
Figure 34: Peak Force Difference Between Legs by a) gender and b) sport. 78
Figure 35: Squat Deviation by a) gender and b) sport. 78
Figure 36: Eccentric Phase Time by a) gender and b) sport. 80
Figure 37: Concentric Phase Time by a) gender and b) sport. 80
Figure 38: Time to Takeoff by a) gender and b) sport. 81
Figure 39: Time in Air by a) Gender and b) sport. 82
Figure 40: Takeoff Velocity by a) gender and b) sport. 82
Figure 41: Jump Height, Calculated with the TIA Method by a) gender and b) sport. 83
Figure 42: RSImod, Calculated with the TIA Jump Height by a) gender and b) sport. 84
Figure 43: Example of a NCMJ with Unsteady Force during the Squat Portion, and a Small Countermovement Dip when Leaving the Squat Phase. 85
Figure 44: CMJ Peak Force by Leg over Four Dates for a Soccer Player Post-Left ACL Surgery 90
Figure 45: CMJ Jump Height over Four Dates for a Soccer Player Post-ACL Surgery 91

Chapter 1: Introduction

High-level athletes are highly susceptible to body injury due to their continuous involvement in physically-demanding activity, including training and competition. Injuries in these athletes can be particularly devastating, as they may restrict the athlete's ability to participate in their sport for extended periods of time. To better understand how a lower limb injury may affect an athlete's lower limb muscle function and their subsequent rehabilitation, we must first understand lower limb muscle function in uninjured athletes. This study's initial goal was to develop a tool to instantaneously quantify an athlete's lower left- and right-limb muscle function with minimal equipment and labor involved. Dual force plates are a relatively inexpensive tool to use, and many training centers already have some version of a force plate system. Athletes can perform different jumps on these force plates as a way to quickly quantify the force exerted by their lower limbs. Many athletes are already familiar with jumps used on force plates, such as a counter movement jump, a non-counter movement jump (also known as a squat jump), or a drop jump. The quantification of muscle function during these jumps can be used as a tool to help athletes assess and improve their performance over time, including post-injury. It is anticipated that athletes from different sports may exhibit differences in their muscle function when performing these jumps, due to the large variation in muscle usage and development across sport types.

Additionally, with the use of dual force plates rather than a single force plate, we can also assess differences between the left and right lower limbs. Muscle asymmetry can be prevalent in athletes due to a variety of causes such as the preference of a dominant side of the body (laterality), previous or current injuries, the nature of the athlete's particular sport, or the specific position within the sport. An athlete simply favoring one side of their body may lead to asymmetry in an athlete's muscle function over time. Injuries to a lower limb may hinder its ability to exert forces, even after return to sport following rehabilitation. By quantifying the asymmetry in muscle usage, the progress of injury rehabilitation can be assessed and corrective action can be taken. It is important to understand the various asymmetries that may naturally exist in sports in order to establish normative data for each sport.

1.1 The Counter Movement Jump

The Counter Movement Jump (CMJ) involves the jumper beginning in an upright standing position with their hands on their hips, squatting down (flexing at knees and hips) and immediately extending their knees and hips to jump vertically into the air. CMJs are simple, practical, and a reliable measure of lower body muscle power. CMJs are used by many strength and conditioning coaches and are frequently used in literature to quantify lower limb strength. Athletes from a variety of backgrounds and levels are often familiar with CMJs as it is a common part of strength testing and/or training. CMJs have also been shown to correlate with maximal strength, maximal speed, and explosive strength.

A meta-analysis conducted by Claudino et al (2017) suggested that using CMJ height averaged over trials was more sensitive than the single highest CMJ in monitoring neuromuscular status (Claudino, et al., 2017). Additionally, using the CMJ performance without arm swing is recommended (Claudino, et al., 2017). The current study used the CMJ performance averaged over the 5 trials, and experimental protocol required that the participants kept their hands on their hips throughout the whole jump to eliminate arm swing.

Plyometric training, also known as reactive training, includes exercises with a quick, powerful movement where eccentric contraction is immediately followed by an explosive concentric contraction (Clark & Lucett, 2010). Jumps, including CMJs, and any other exercises involving an explosive movement are classified as plyometric exercises. Plyometric training uses the stretch-shortening cycle to develop explosive, fast movements to exert maximum force in a short period of time, which improves athletes' rate of force production and allows muscles to generate forces at faster speeds (Clark & Lucett, 2010). Plyometric exercises are one of the most effective ways to increase speed and power (and thus, explosiveness) in sport (Clark & Lucett, 2010). The ability of an athlete's muscles to exert a maximum force in a minimum amount of time is known as the rate of force production and is important in sport to gain advantage in performance (Clark & Lucett, 2010). CMJs are useful both in plyometric training for athletes as well as to be used as a tool to assess the rate of force production.

The muscle action in plyometric exercises (such as the counter movement jump) happens sequentially: the forces are loaded (eccentrically), stabilized (isometrically), and then unloaded (concentrically) (Clark & Lucett, 2010). Plyometric training helps to shorten the reaction time of this sequence of muscle actions by improving the neuromuscular coordination in athlete's bodies (Clark & Lucett, 2010). Plyometric training has three phases: the eccentric phase, the amortization phase, and the concentric phase. In a countermovement jump, the amortization phase should theoretically be equal to zero, and there should be a direct transition between the eccentric to concentric phase. Thus, the amortization phase was not a variable of interest in the CMJ study.

1.2 Jump Phases

"Eccentric" and "Concentric" jump phases have been described in literature (Jordan, Aagaard, & Herzog, 2015; McMahon, Jones, Suchomel, Lake, & Comfort, 2018) to separate a counter movement jump into segments. In a CMJ, the eccentric jump phase represents the braking movement: it starts at the point of maximum negative velocity (with the upward direction being positive) and ends at the point of zero velocity (where the subject's center of mass is in its deepest position). The concentric phase of the jump represents the propulsion upward: it starts at the point of zero velocity and ends at the instant of jump takeoff. During a CMJ, the squat down before taking off for the jump stores elastic energy by eccentrically contracting the muscles. During eccentric contractions, muscles lengthen (while in concentric contractions, muscles shorten). As the jumper generates motion vertically, concentric contraction extends the lower limb muscles and propels the jumper upwards.

An alternative method to separate CMJs into phases is by dividing jumps into unloading and loading phases. These may also be referred to as unweighting and weighting phases. This terminology is somewhat easier to comprehend, as one can imagine the muscles being loaded or unloaded with body weight during the jump. In this method, the "Unloading Phase" starts at the initiation of the counter movement jump and represents the interval where the normal force measured is below that of the jumper's body weight. Once the force exerted by the jumper is equal to their body weight, the jump then transitions into the "Loading Phase," where the force is greater than the jumper's body weigh. At the point where the maximum force occurs, the jump transitions into the final phase, the "Takeoff Phase," which occurs from the point of maximum force to the point of jump takeoff. The main difference between the eccentric/concentric terminology and the unloading/loading/takeoff terminology, is that the transition between eccentric and concentric phases occurs where the subject's center of mass velocity is equal to zero, whereas the loading phase transitions to the takeoff phase where the subject has exerted the maximum normal force.

1.3 Equations

1.3.1 Kinetic Impulse

While peak force during a jump can be an important variable, it may have limitations due to its measurement occurring at a single point in time. Kinetic impulse ("impulse"), represented by the area underneath the force-time curve, may better characterize functional muscle performance than a time point analysis like peak force. Jordan et al. (Jordan, Aagaard, & Herzog, 2015) used the phase-specific kinetic impulse in the eccentric and concentric jump phases to calculate kinetic impulse asymmetry, termed “asymmetry index” and is calculated using (Eq. 1).

$$AI = \frac{|Left\ Leg\ Impulse - Right\ Leg\ Impulse|}{\max\{Left\ Leg\ Impulse, Right\ Leg\ Impulse\}} \times 100\% \qquad \textit{(Eq. 1)}$$

Kinetic impulse may aid in measuring lower extremity explosiveness. Impulse represents the change in momentum of an object, and is equal to the force applied to an object for an amount of time. Given a force-time curve for the normal force exerted, impulse is equal to the time integration of the force minus the impulse due to body weight alone.

1.3.1 Kinematic equations

The kinematics for motion of an object that are relevant to calculations in this study are as follows. Assuming that an objective is moving with a constant acceleration:

$$\mathrm{x} = \mathrm{x_0} + \mathrm{v_i t} + \frac{1}{2}\mathrm{at}^2 \qquad \textit{(Eq. 2)}$$

$$\mathrm{v_f}^2 = \mathrm{v_i}^2 + 2\mathrm{ad} \qquad \textit{(Eq. 3)}$$

Where x is the displacement of the object, x_0 is the initial position of the object relative to a fixed reference, v_i is the initial velocity of the object, t is the time interval, a is the object’s constant acceleration, v_f is the object’s final velocity, and d is the displacement. In the case of an object moving under gravity, the acceleration, a, would be equal to gravity.

1.3.3 Jump Height

Several methods to calculate the jump height from force plate data have been used in the literature (Moir, 2008; Linthorne, 2001). These include the “time in air” method, the “impulse momentum” method, and the “work-energy” method. Based on the study by Linthorne (2001), the work-energy method is the least reliable and often is subject to compounding errors (Linthorne, 2001). Additionally, there is not sufficient literature to support the work-energy method’s reliability, and so this method was excluded from use in this study.

The Time in Air (TIA) method, also known as flight time method, is based on the assumption that the jumper’s center of mass at takeoff is the same as the center of mass at landing. This may overestimate the jump height by 0.5-2 cm (Linthorne, 2001). This method also relies on the assumption that the jumper is accelerating uniformly (constant acceleration). The reference point for the jumper's center of mass is set during the neutral standing position, such that the initial displacement of the jumper is zero ($x_0=0$). Initial velocity is assumed to be zero in the neutral standing position ($v_i=0$), and the acceleration acting on the jumper is equal to gravity ($a=g$). The displacement of interest is in the vertical direction ($x=y$). Substituting into (Eq. 2), the resulting formula for the vertical displacement of the jumper using the time in air method (y_{TIA}) is:

$$y_{TIA} = \frac{1}{2} g t^2 \qquad \text{(Eq. 4)}$$

Assuming the maximum height occurs at one-half the time of flight ($t=t_{flight}/2$) and subbing into (Eq. 4), the calculation for the jump height using the TIA method is:

$$y_{TIA} = \frac{1}{2} g \left(\frac{t_{flight}}{2}\right)^2 \qquad \text{(Eq. 5)}$$

The Impulse-Momentum Method (IMM) (Linthorne, 2001), also known as the vertical velocity at takeoff (TOV) method, is based off of the impulse-momentum theorem. Impulse is equal to the change in momentum, which is equivalent to the integral of force with respect to time (Eq. 6).

$$\mathrm{J} = \int \mathrm{F}\,\mathrm{dt} = \mathrm{m}\Delta\mathrm{v} \qquad \text{(Eq. 6)}$$

The force measured by force plates can be separated into the ground reaction force (F_{GRF}) and the force exerted due to gravity acting on the jumper. Integrating these forced from the onset of a counter movement jump (t_i) to the point at which the jumper takes off (t_{to}):

$$\int_{t_i}^{t_{to}} (F_{GRF} - mg)dt = m(v_{to} - v_i) \qquad \text{(Eq. 7)}$$

Where v_{to} is the vertical velocity at jump takeoff and v_i is the initial vertical velocity. Assuming that the initial velocity in the neutral standing position is zero ($v_i=0$) and rearranging:

$$\int_{t_i}^{t_{to}} (F_{GRF})dt - \int_{t_i}^{t_{to}} (mg)dt = J_{GRF} - J_{BW} = mv_{to} \qquad \text{(Eq. 8)}$$

Giving:

$$v_{to} = \frac{J_{GRF} - J_{BW}}{m} \qquad \textit{(Eq. 9)}$$

Where J_{GRF} is the impulse due to ground reaction force and J_{BW} is the impulse due to the jumper's body weight. The impulse due to body weight can easily be calculated by multiplying the jumper's mass by the length of the time interval for which the impulse is being calculated. Assuming the initial vertical velocity (v_i) is equal to zero in the quiet standing position and taking the final velocity to be the velocity at takeoff ($v_f = v_{to}$) and the acceleration to be equal to gravity (a=g), (Eq. 3) becomes:

$$\mathrm{d} = \frac{{\mathrm{v}_{\mathrm{to}}}^2}{2\mathrm{g}} \qquad \textit{(Eq. 10)}$$

Subbing (Eq. 9) into (Eq. 10):

$$y_{IM} = \frac{\left(\frac{J_{GRF} - J_{BW}}{m}\right)^2}{2g} \qquad \textit{(Eq. 11)}$$

Where y_{IM} is the vertical jump height calculated using the Impulse Momentum method. The IMM method does not account for the change in vertical position of the jumper's center of mass before and after jump takeoff. The jump height calculated will vary based on the takeoff velocity as well as the center of mass position at takeoff.

When using a force platform to calculate vertical jump height, Moir (2008) recommends using the Impulse Momentum method (Moir, 2008). While both the IMM method and the TIA method are logically valid, the IMM method removes many confounding variables associated with the TIA method (Moir, 2008). However, both the IMM method and the TIA method found the same associations between vertical jump height in men and women in Moir's study (Moir, 2008). Linthorne (2001) also compared the different methods of measuring vertical jump height from force platform analysis (Linthorne, 2001). While the TIA method was more straightforward (simply calculate by determining the time spent in the airborne phase and using kinematic equations for one dimensional motion under constant acceleration), the IMM method (using integration) was determined to be more accurate (Linthorne, 2001).

1.3.4 Reactive Strength Index Modified

The Reactive Strength Index Modified (RSImod) is a variable used by McMahon et al. (McMahon, Rej, & Comfort, 2017; McMahon, Jones, Suchomel, Lake, & Comfort, 2018) to quantify the relative jump capacity of athletes performing a counter movement jump. The RSImod is calculated using (Eq. 12).

$$\text{RSImod} = \frac{\text{jump height (m)}}{\text{time to takeoff (s)}} \qquad \text{(Eq. 12)}$$

1.3.5 Peak Force Percent Difference

The percent difference between the peak force exerted by the left and right leg was also calculated as a means to help quantify asymmetry. The Peak Force Percent Difference is calculated by (Eq. 13).

$$\text{Peak Force Percent Difference} = \frac{|Peak\ L\ force - Peak\ R\ force|}{Average\ of\ Peak\ L\ and\ R\ forces} \qquad \textit{(Eq. 13)}$$

1.4 Description of Rugby, Soccer, Swim and Volleyball

Participants from four different sports (rugby, soccer, swim and volleyball) were involved in this study. Brief descriptions of the nature of the sport, lower limb muscle usage in the sport (as it pertains to counter movement jumps), and the common lower limb injuries occurring in each sport are provided in the following sections.

1.4.1 Rugby

Rugby (also known as rugby union) is one of the world’s most popular sports. Rugby is a field sport played with 15 players on each team, with two 40-minute halves. Rugby combines running and endurance (similar to sports such as soccer) with contact and tackling (similar to American football). Points are scored by carrying the ball to the opponent’s end of the field, or by kicking the ball between posts positioned at the end of each field.

Rugby is fast-paced and physically demanding, and every position must be able to run, pass, kick and tackle to be effective on both offense and defense. Jumping motions in rugby that may resemble the CMJs used in this study include jumps made during catches, or jumps during lineouts (the means by which gameplay is restarted after the ball has gone out of bounds). During lineout jumping, two players from the team throwing the ball in help lift a third player (the jumper) upwards. This tactic isn’t always employed during lineouts, but provides another option for a player to throw the ball into other than the players remaining on the ground.

1.4.2 Soccer

Soccer (also known as football) is the world's most popular sport. The sport is highly aerobic, and integrates balance, agility and coordination with cardiovascular fitness. Soccer is a field sport played with 11 players on each team (10 outfield players and one goalie), and the games consist of two 45-minute halves. Goals are scored by driving the ball into the opposing team's net, using mainly feet to pass or strike the ball.

Jumping movements in soccer mainly include challenging for a ball in the air, such as when a player performs a vertical jump to head the ball. Goalies in soccer perform frequent jumping movements. The position which participants play in soccer may impact their jump performance. In a study by Jezdimirovic et al., the goalkeepers could jump the highest when performing a CMJ, followed by forwards, then defenders, and midfielders jumped the lowest (Jezdimirovic, Joksimovic, Stankovic, & Bubanj, 2013). These differences were significant for the midfielders, goalkeepers and forwards.

1.4.3 Swimming

Swimming is a popular low-impact highly aerobic activity that requires high levels of endurance, muscle strength and cardiovascular fitness. At the varsity level, swimmers compete in distances ranging from 50 m to 800 m in typically 50 m long pools. Four different strokes are often used in competitive swimming: butterfly, backstroke, breaststroke, and freestyle. This study includes swimmers that primarily compete in indoor swimming, not open water swimming.

Swimming is a full body workout, and depending on the stroke used, different muscles may be used more than others. Movements in swimming that resemble jumps or plyometric movements include the swim start (performed on a starting block, swimmers project themselves into the water) and turns (performed when the swimmer reaches the end of the swimming pool and has remaining pool lengths to complete). Turns can be classified into three groups: open turns, flip turns, or backward flip turns. Open turns include swimmers touching the wall with their hands, tucking their legs in and turning them against the wall, then pushing off the wall in a streamline position toward the opposite end of the pool. During flip turns, swimmers approach the wall, tuck their legs and flip forwards in the water, then push off the wall in the direction of the opposite end of the pool and twist to reposition themselves face-down in the pool again. Backward flip turns are used in individual medley competitions, when swimmers switch from backstroke to breaststroke. In a backward flip turn, the swimmer touches the wall while still in backstroke, then performs a backwards flip, and pushes off the wall to start breaststroke. All of these turns include a component similar to CMJs: swimmers use movement in the direction of the pool wall (a countermovement) to store potential energy in their lower body, and immediately use this energy to propel themselves in the opposite direction.

1.4.4 Volleyball

Indoor volleyball is a team sport played on a court separated by a net. Each team has six players which rally the ball among themselves for a maximum of three touches, before sending the ball over the net to the other team. Points are won by causing the ball to touch the ground on the opponent's side of the court.

Vertical jumping movements are very frequent in volleyball, both in defense (blocking) as well as in offence (attacking, passing, serving) (Ziv & Lidor, 2010). It is highly advantageous to be able to jump higher as a volleyball player in order to successfully carry out attack hits or defensive blocks. The frequency of jumping movements performed in gameplay can depend on the players' position (setter, hitter, outside, middle, blocker, etc.), but all positions benefit from strong jumping technique. A 2011 review conducted by Ziv and Lidor (Ziv & Lidor, 2010) of observational and experimental studies found that players of better performing teams did indeed have higher vertical jumps, and that strength and conditioning programs that include plyometric training can increase and maintain optimal vertical jump performance. Plyometric training is commonly used in strength and conditioning for volleyball players to help increase jump height as well as decrease reaction time.

1.5 Objectives

The goals of this study were to create a mathematical tool to automatically analyze counter movement jumps performed on dual force plates, to develop a better understanding of how athletes of different sports may exert lower limb muscle power differently, and to quantify asymmetries that may exist in lower limbs due to injury or natural phenomenon. The need for a means to quantify "good performance" in athletes exists, but it is difficult with the large variability in patterns of force application and the labor-intensive nature of analyzing large amounts of data (Dowling & Vamos, 1993; Turner, et al., 2015). It is crucial that coaches and practitioners are able to perform the analysis necessary to uncover trends or associations, make predictions, and assess efficacy of training programs (Turner, et al., 2015).

The mathematical tool was designed to analyze jumps instantaneously and automatically, such that the burden of numerical analysis of large datasets was eliminated. While much of current literature performs force-time jump analysis numerically, this tool would allow for analytical analysis. Additionally, the ability to automatically analyze jumps would eliminate inter-observer variability, and would allow for a wide variety of individuals (e.g. strength and conditioning coaches) to perform analysis easily and immediately after the jump is performed with minimal software required. This tool was designed to evaluate both the jumper's performance, as well as output variables that help quantify asymmetry between the lower limbs.

Comparison of athletes from different sports can highlight the differences in the way that athletes develop lower limb muscle or exert lower limb power during jumps. It is anticipated that different sports will show difference in their CMJ analysis results. Additionally, by better understanding the asymmetries typically exhibited for each sport, gender, or leg dominance, injury rehabilitation can be better understood and athlete performance can be enhanced. The mathematical tool can help to quantify lower limb asymmetries in the context of injury rehabilitation or natural asymmetry of sport.

Chapter 2: Quantifying Lower Limb Mechanical Muscle Function during Counter Movement Jumps

2.1 Methods

2.1.1 Participants

Participants for this study consisted of athletes from the University of Alberta’s various varsity sports teams. Participation was done on a voluntary basis, and so several sports teams elected not to participate. Due to the varying team sizes and willingness of participants, it was difficult to get an equal number of participants from all teams. Additionally, sports that have a high variability in individual athletes’ roles (such as track and field athletes or football players) did not make good candidates for this study and were not included. Players with current lower limb injuries were omitted from this study. All participants were familiar with counter movement jumps, as these jumps are commonly used in strength training and testing at the varsity level. Table 1 outlines the number of participants from each varsity team. Sports included swimmers, indoor volleyball players, rugby union players, and soccer players.

Table 1: Participants’ Sport and Gender

Sport Team	**Gender**	**Number of Participants**	
Swim	Male	5	$\Sigma = 10$
	Female	5	
Volleyball	Male	12	$\Sigma = 12$
Rugby	Female	20	$\Sigma = 20$
Soccer	Male	7	$\Sigma = 28$
	Female	21	
Total	Male	24	$\boldsymbol{\Sigma = 70}$
	Female	46	

2.1.2 Experimental Procedures

After a brief warm up, participants performed five trials of a counter movement jump on dual force plates (PASCO PASPORT 2-Axis Force Platform, Roseville, California, USA). Participants were instructed to stand upright on the force plates with their hands on their hips to help isolate lower limbs from upper limbs. Upon a verbal command, participants squatted down (flexing at the knees and hips), then immediately extended knees and hips to jump vertically into the air. After participants landed back on the force plates, they remained in the squatted position for a minimum of 2 seconds, until the forces exerted on the dual force plates stabilized and they were instructed to relax. Adequate resting time (minimum of 20-30 seconds) was provided between each jump trial. Capstone analysis software (PASCO, 2013) recorded the normal force exerted on the force plates as a function of time at 0.001-second increments (a sampling rate of 1000 Hz). Data was then exported into a customized Mathematica program for analysis. The leg dominance of each participant was noted; in the case of subjects who felt they were equally as strong with both legs, subjects were asked which leg they "kicked a soccer ball with." Additionally, any current injuries were noted to ensure participants met inclusion criteria.

2.1.3 Mathematica Analysis

A custom template was created in Wolfram Mathematica 10 (Wolfram Research, Inc., 2015) to analyze the counter movement jumps. Jumps were analyzed first as left and right legs separated, and then as both legs combined. A VBA macro in Microsoft Excel was used to convert raw data points from the PASCO Capstone software into the appropriate format to be inputted into Mathematica.

Data points were inputted into Mathematica as a matrix, and each dataset was trimmed down to keep only every fourth data point. The goal of trimming down the data was to speed up the analysis program run in Mathematica. Every fourth point was retained as this was found to be a good balance between maintaining data integrity while allowing the program to run efficiently. The original data was collected at a sampling rate of 1000 Hz and so after trimming the data down, each data point was still sufficiently closely spaced at 0.004 seconds. These data points would be converted into a function before analyzing, and the trimming down of data points is not expected to have had an impact on overall results. A 10th order Gaussian filter was applied to the normal force using the Mathematica built-in Gaussian filter function. The purpose of this filter was to reduce noise in raw data collection in order to produce a smooth force-time curve and subsequent curves used in analysis. The appropriate order of filter was determined experimentally: the filter order was selected to be high enough to sufficiently smooth data, but not so high as to unnecessarily remove natural spikes in the data.

Mathematica's third order spline interpolation function was then applied to the data to analytically describe the force as a function of time from the data points. Converting the data points into a function allowed for flexibility in further analysis, such as taking the integral of the function. All necessary time points, intervals, and variables of interest were automatically found for each jump by the program code. To do this, time points were found in an incremental fashion, going from the easiest to find time points, to the most difficult time points that relied on other points of reference to obtain. Table 2 and Table 3 define the acquired time points and jump intervals for the counter movement jumps, and Figure 1 displays these points and intervals on a typical normal force as a function of time curve.

Table 2: Counter Movement Jump Time Point Definitions

Time Point	**Definition**
t_1	Time at onset of movement (jump begins)
t_2	Time at peak negative center of mass velocity; where acceleration=0 (onset of braking)
t_3	Time where center of mass velocity=0 (onset of propulsion)
t_4	Time at takeoff
t_5	Time at landing
t_{max}	Time at overall maximum normal force

Table 3: Counter Movement Jump Interval Definitions

Time Segment	**Jump Phase**	**Indicated in Figure 1**
	Eccentric/Concentric Method	
t_2-t_3	Eccentric	a
t_3-t_4	Concentric	b
	Unloading/Loading Method	
t_1-t_2	Unloading	c
t_2-t_{max}	Loading	d
t_{max}-t_4	Final/Takeoff	e

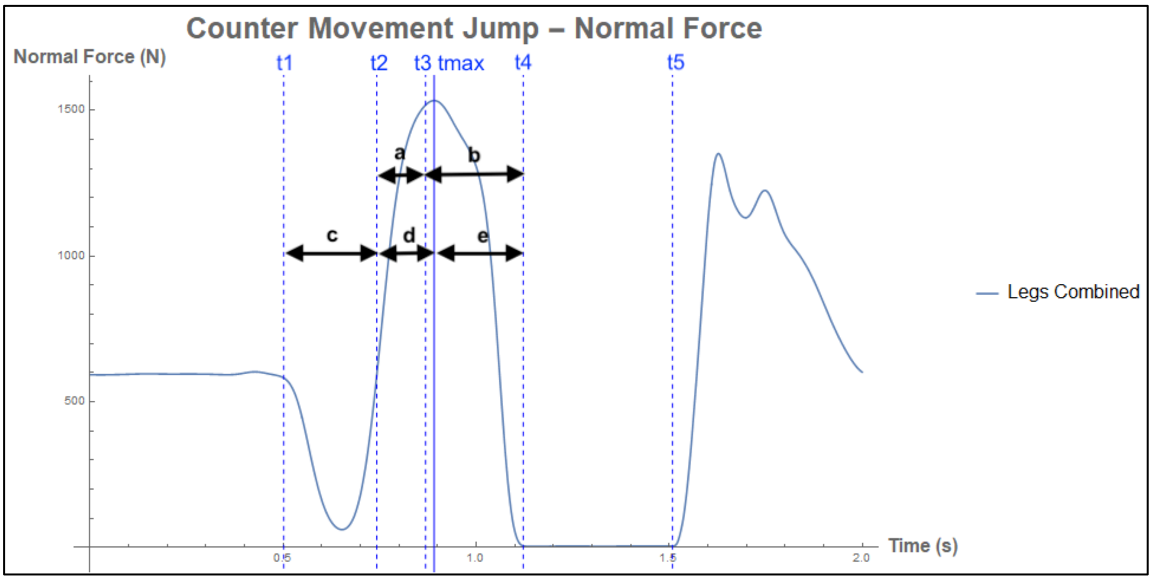


Figure 1: Counter Movement Jump - Time Points and Intervals

The first time point found was where the maximum force of the jump occurred (t_{max}). The maximum force and the time at which it occurs can easily be found using a built-in function in Mathematica. The next time point obtained was t_4, the time at which the jumper takes off from the force plate platform. This was found using an iterative approach over the interval between the point at which the maximum force occurred (t_{max}) and the end of collected data points. Then, t_4 was set to the time at which the function was a minimum over this interval and was iteratively moved towards time zero, until the first point where the force function equalled zero. The time at landing, t_5, was found by solving for the point between t_4 and the end of the jump data collection where the force-time function was equal to 5% of the force exerted at time zero. The criteria of 5% of the force exerted at time zero was experimentally determined to be appropriate for t_5: the force exerted at time zero quickly approximates the subject's body weight, and 5% of this was found to be a good balance for a variety of subjects without accidentally picking a point while the subject was still in the air (if the criteria was too low) or a point after the subject had already landed (if the criteria was too high).

The time point at which the jump started, t_1, was found using a slope criteria method. The lowest point before the maximum time (the dip between t_1 and t_2 in Figure 1) was found, and the slope between this lowest point and the point at time zero was calculated. The point at which this slope reached a threshold value that was determined to be sufficient experimentally was taken as t_1.

Body weight while subjects were in a quiet stance, F_{BW}, was approximated by taking the average of the force function from 0.05 seconds after recording had started to 0.1 seconds before t_1. Quiet stance mass was then calculated from this force by dividing by acceleration due to gravity, and then this was used as body mass in subsequent analysis. Jumps that did not have a consistent quiet stance force from 0.05 seconds to 0.1 seconds before t_1 (e.g. if in individual leg analysis, the subject was slightly shifting weight between legs) would output an error message and were treated on a case-by-case basis to determine a better range to calculate average quiet stance force.

As a check to ensure force plates were functioning properly during the jump trials and that the analysis program was performing correctly, the program checked to see if the force plates were picking up force data from t_4 to t_5 (when the subject was supposedly in the air). The average force over the interval of t_4 plus 0.05 seconds to t_5 minus 0.05 seconds was calculated. If the absolute value of this average force while the subject was in the air was greater than 5% of the subject's body weight, the program outputted an error message that prompted user intervention. The absolute value of the force while the subject was in the air exceeded 5% of subjects' body weight on one testing date in particular (November 6, 2017) where it is assumed that the force plates were not calibrated properly prior to beginning testing. This occurred for five individuals for the left force plate only, where the data collected was approximately -20 N while the subjects were in the air. To adjust for this for these five subjects, the force while in the air was calculated for these individuals, then subtracted from all data points during the jump, and then analysis program was run again from the beginning with the adjusted data.

To calculate the remaining time points, t_2 and t_3 (the time at peak negative center of mass velocity and the time at which center of mass velocity is equal to zero, respectively), the velocity was then derived from the function. The acceleration function, a(t), was derived from the normal force function, F(t), by subtracting force due to body weight, F_{BW}, and dividing the function by mass (Eq. 14).

$$\mathrm{a(t)} = \frac{\mathrm{F(t)} - \mathrm{F_{BW}}}{\mathrm{m}} \qquad \textit{(Eq. 14)}$$

The velocity function, v(t), and the displacement function, d(t), were then calculated from the acceleration function (Eq. 15) and (Eq. 16). Using Mathematica's built-in solve functions, v(t) and d(t) were calculated by setting up a differential equation with the assumptions that the velocity at t_1 and the displacement at t_1 were both equal to zero (Eq. 17).

$$\mathrm{v(t)} = \int \mathrm{a(t)dt} \qquad \textit{(Eq. 15)}$$

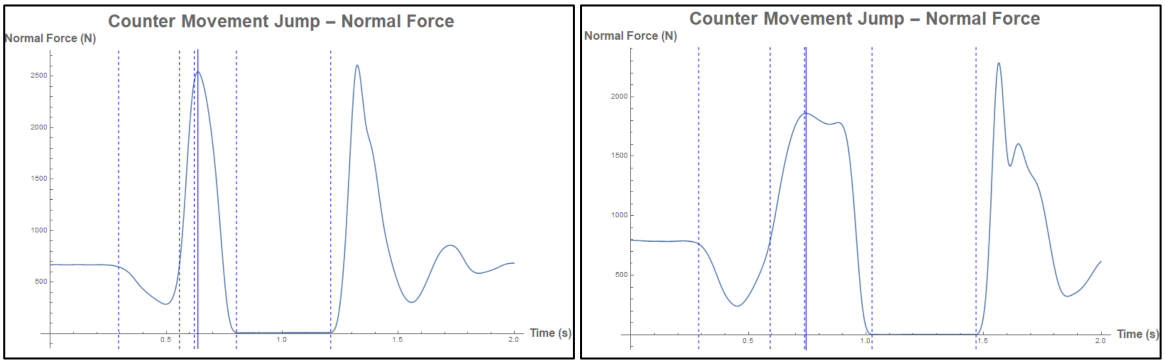
$$\mathrm{d(t)} = \int \mathrm{v(t)dt} = \iint \mathrm{a(t)dtdt} \qquad \textit{(Eq. 16)}$$

$$\mathrm{v(t_1)} = 0;\ \mathrm{d(t_1)} = 0 \qquad \textit{(Eq. 17)}$$

The peak negative center of mass velocity, t_2, was determined by finding the time at which the velocity function was a minimum between t_1 and t_{max}. The time at which the center of mass velocity was equal to zero, t_3, was determined by finding the time at which the displacement function was a minimum between t_2 and t_4. Power was then calculated by multiplying the normal force function, F(t), by the velocity function, v(t).

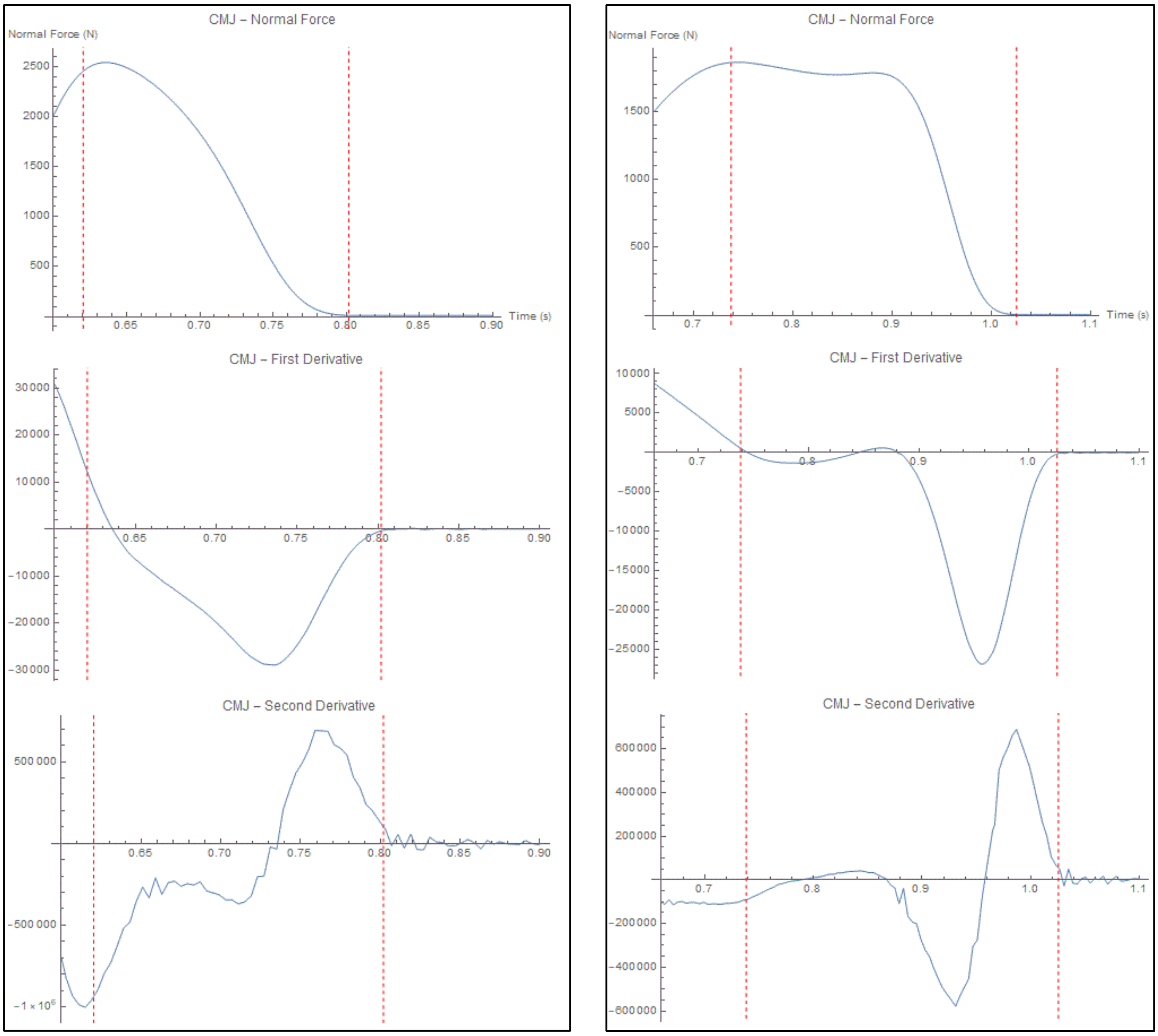
Using all of the previously created functions, maximum and minimum values for force, velocity, displacement, and power were calculated and outputted for each segment and at each time point. Kinetic impulse for each segment of the jump was calculated by integrating the force function between time point bounds for that specific segment, and subtracting the impulse due to body weight alone.

The program classified the CMJs into two categories of typical force-time jump curves observed: jumps with one peak (“Class 1”) and jumps with two peaks (“Class 2”) (Figure 2). A Class 1 jump was defined as a jump in which the normal force function had only one inflection point between t_3 and t_4. A Class 2 jump was defined as a jump in which the normal force function had two or more inflection points between t_3 and t_4. The number of inflection points of $F(t)$ was determined by calculating how many times the second derivative, $F''(t)$, was equal to zero over the interval from t_3 to t_4 (Figure 3). Participants were classified as either Class 1 or Class 2 based on which class the majority of their jumps (i.e. three or more out of their five trials) fell under.



(a) *(b)*

Figure 2: Examples of Jumps with a) One Peak – Class 1 and b) Two Peaks – Class 2



(a) *(b)*

Figure 3: The Normal Force Function, First Derivative and Second Derivatives of the Examples in Figure 2 from t_3 to t_4 for a a) Class 1 Jump, and b) Class 2 Jump

The program outputted values and graphs (with time points plotted) intermittently such that the user could visually confirm the correct points were being selected as it ran, and could intervene if there were any issues or error messages. The analysis was run for the left leg and right leg separately, then the normal force datasets for the separate legs were combined and the analysis was run again to include both legs at once. This was done for each of the five jump trials for each subject. Graphs with the timing points and the desired numerical outputs were automatically outputted into CSV format for subsequent statistical analysis. The visual basic code used to format the raw data into matrix form for analysis, and the Mathematica code used to analyze this matrix, are found in Appendices F and G, respectively.

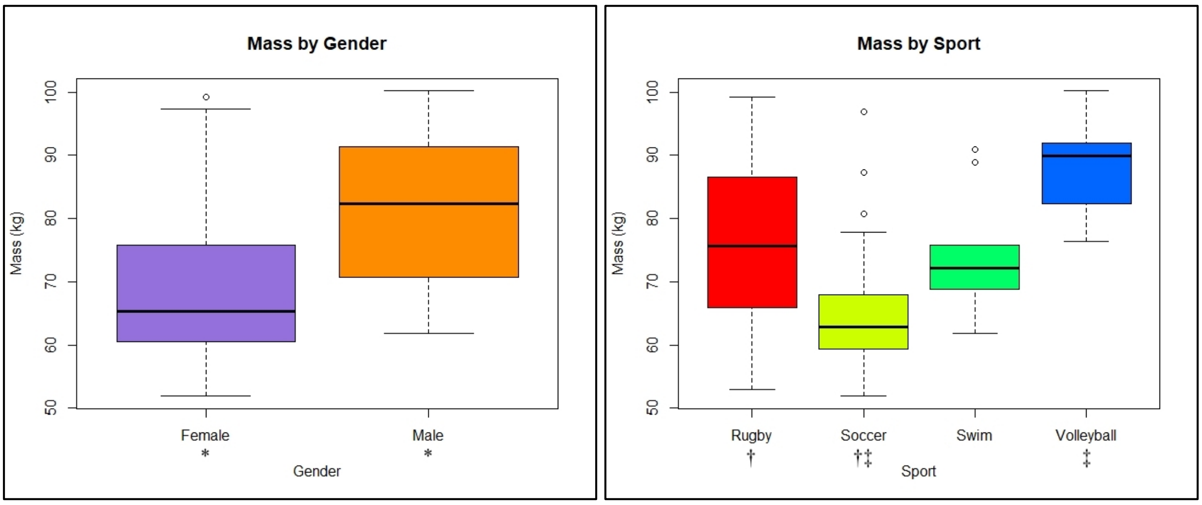
2.2 Results

Variables outputted from the Mathematica analysis for each participant were averaged over the five jump trials. Two-way ANOVA was performed in R (R Core Team, 2019) to determine if there was a significant interaction effect between gender and sport. There were no statistically significant interaction effects ($\alpha < 0.05$) between gender and sport for the variables of interest in this study. Tukey honest significance difference (HSD) post hoc tests were then performed in R to determine which groups had statistically significant differences for variables of interest. As the dependent variables compared were all continuous (except for the run class), and the independent variables (gender and sport) were categorical and independent groups, Two-Way ANOVA was suitable for statistical analysis. Additionally, the following assumptions were made when performing this statistical analysis: each sample was drawn independently of other samples, the variance of the data between groups was the same, and samples were taken from normally distributed populations. Appendix A-E detail the outcomes of the Two-way ANOVA and Tukey HSD tests. Appendix A-E also include further details on the results for the variables of interest.

Results are reported in the form of (average±standard deviation [unit], n=number of participants in group). A significance level of $\alpha=0.05$ is used for all results described in this section.

2.2.1 Mass and Force

Figure 4 contains the results for the mass measured for subjects while in the quiet standing position on the force plates. Significant differences existed between mass for females (69.4±12.3 kg, n=46) and males (81.5±11.3 kg, n=24). Significant differences were also found between rugby (76.2±13.4 kg, n=20) and soccer (65.4±10.2 kg, n=28), and between soccer and volleyball (88.0±6.6 kg, n=12). Female soccer players (61.9±6.4 kg, n=21) had the lowest mass out of all the groups, while male volleyball players (88.0±6.6 kg, n=12) had the highest mass. For the groups where data was available for both male and female athletes of the same sport, male swimmers (73.9±10.8 kg, n=5) had a similar mass to female swimmers (73.7±9.9 kg, n=5). However, male soccer players (75.8±12.7 kg, n=7) had a much larger mass than female soccer players.

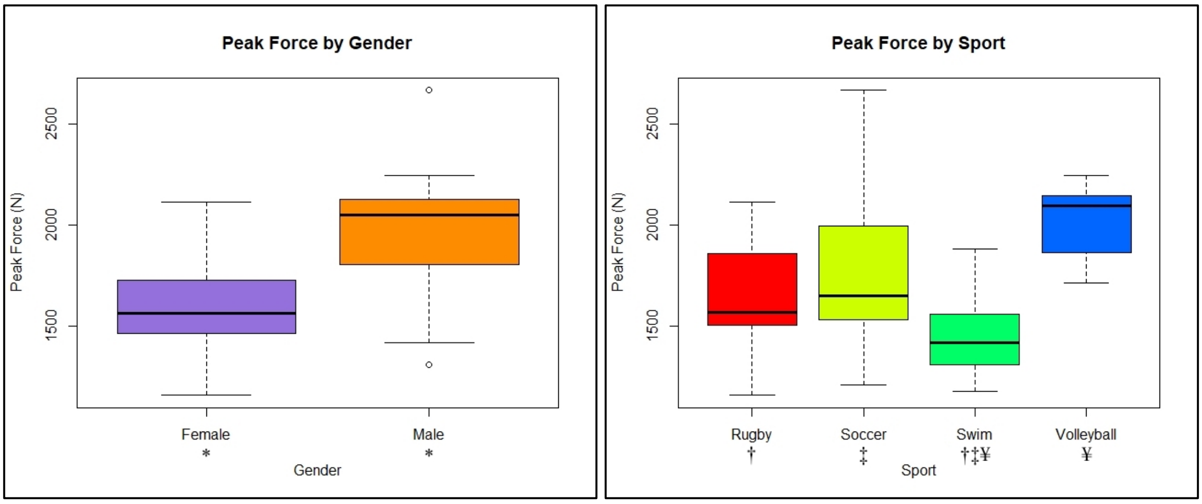


(a) *(b)*

Figure 4: Mass by a) gender and b) sport.

Note: matching symbols under group title indicates a significant difference at the 0.05 level.

Figure 5 displays the peak force measured for both legs combined for all participants. There was a significant difference between females (1608.2±248.5 N, n=46) and males (1945.8±304.6 N, n=24). There were also significant differences between the swim athletes (1469.9±227.1 N, n=10) and all the other sport groups: rugby (1661.0±262.8 N, n=20), soccer (1728.8±319.5 N, n=28), and volleyball (2029.4±174.3 N, n=12). When comparing all the sport groups, swim had the lowest peak force and volleyball had the highest.

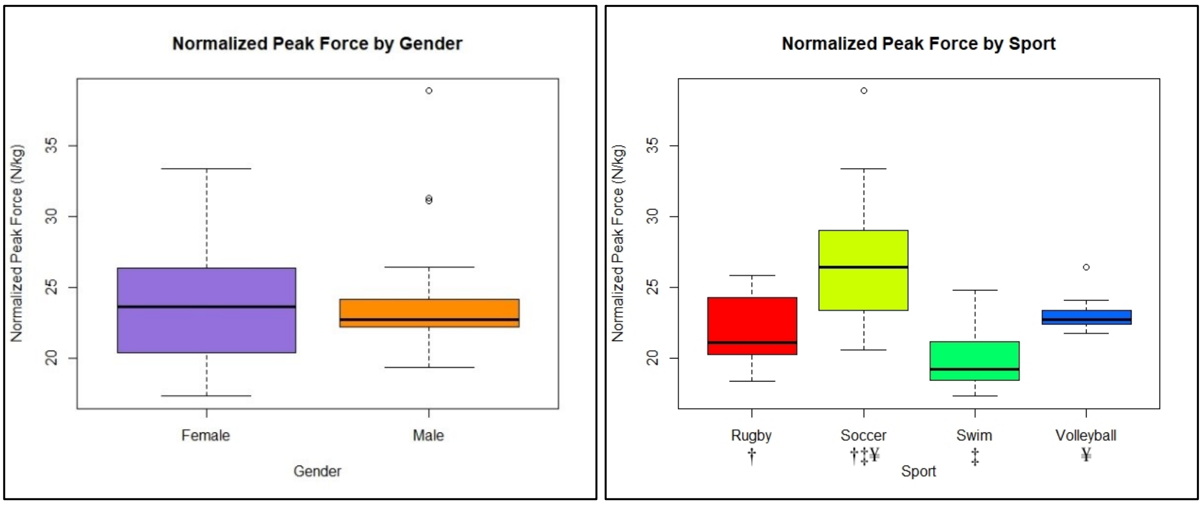


(a) *(b)*

Figure 5: Peak Force by a) gender and b) sport.

Note: matching symbols under group title indicates a significant difference at the 0.05 level.

Normalized peak force (peak force divided by body mass) was also compared (Figure 6). In this case, there was no difference between males and females at this significance level. However, when comparing sport groups, soccer players (26.60±4.13 N/kg, n=28) had a significantly greater normalized peak force compared with all of the other three groups: rugby (21.97±2.36 N/kg, n=20), swim (19.98±2.36 N/kg, n=10), and volleyball (23.08±1.22 N/kg, n=12).



(a) *(b)*

Figure 6: Normalized Peak Force by a) gender and b) sport.
Note: matching symbols under group title indicates a significant difference at the 0.05 level.

The percent difference in peak force between left and right legs was calculated (Figure 7). In this case, no differences at this significance level were found between male and female participants, or between any of the sport groups.

(a) *(b)*

Figure 7: Peak Force Difference between Legs by a) gender and b) sport.
Note: matching symbols under group title indicates a significant difference at the 0.05 level.

Each athletes' self-declared dominant leg was compared with the leg that exerted the greater peak force during the CMJs (Table 4). For all groups, the majority of participants exerted a greater peak force with their dominant leg, except for the male swim group where only 40.0% of participants did. Overall, 67.1% of participants exerted a greater peak force with their dominant leg. For females, 71.7% exerted a greater peak force with their dominant leg, while for males, 58.3% did.

Table 4: Leg Exerting the Greater Peak Force Compared to the Participant's Dominant Leg: Number and Percent

		Leg Exerting Greater Peak Force Matches Dominant Leg?				
Gender	**Sport**	**Yes**		**No**		**N**
		#	**(% of N)**	**#**	**(% of N)**	
Male	Swim	2	(40.0)	3	(60.0)	5
	Volleyball	7	(58.3)	5	(41.7)	12
	Soccer	5	(71.4)	2	(28.6)	7
	Total	14	(58.3)	10	(41.7)	24
Female	Swim	4	(80.0)	1	(20.0)	5
	Rugby	16	(80.0)	4	(20.0)	20
	Soccer	13	(61.9)	8	(38.1)	21
	Total	33	(71.7)	13	(28.3)	46
Total	Swim	6	(60.0)	4	(40.0)	10
	Volleyball	7	(58.3)	5	(41.7)	12
	Rugby	16	(80.0)	4	(20.0)	20
	Soccer	18	(64.3)	10	(35.7)	28
	Total	47	(67.1)	23	(32.9)	70

The difference in the peak force exerted by the left and right legs was calculated as a percentage (Table 5). Overall, participants had a 6.8±4.8% difference between the peak force of their legs, while males had a 5.3±43.4% difference and females had a 7.5±5.2% difference. There was no significant difference found between any of the groups compared for the peak force difference between legs.

Table 5: Peak Force Difference calculated between Legs (%)

Peak Force Difference between Legs (%)

Gender	Sport	Mean	Standard Deviation	N
Male	Swim	5.403	2.494	5
	Volleyball	5.760	4.030	12
	Soccer	4.540	3.091	7
	Total	5.330	3.410	24
Female	Swim	5.135	2.439	5
	Rugby	8.405	5.027	20
	Soccer	7.261	5.760	21
	Total	7.527	5.192	46
Total	Swim	5.269	2.330	10
	Volleyball	5.760	4.030	12
	Rugby	8.405	5.027	20
	Soccer	6.580	5.305	28
	Total	6.774	4.750	70

2.2.2 Force-Time Curve Shape

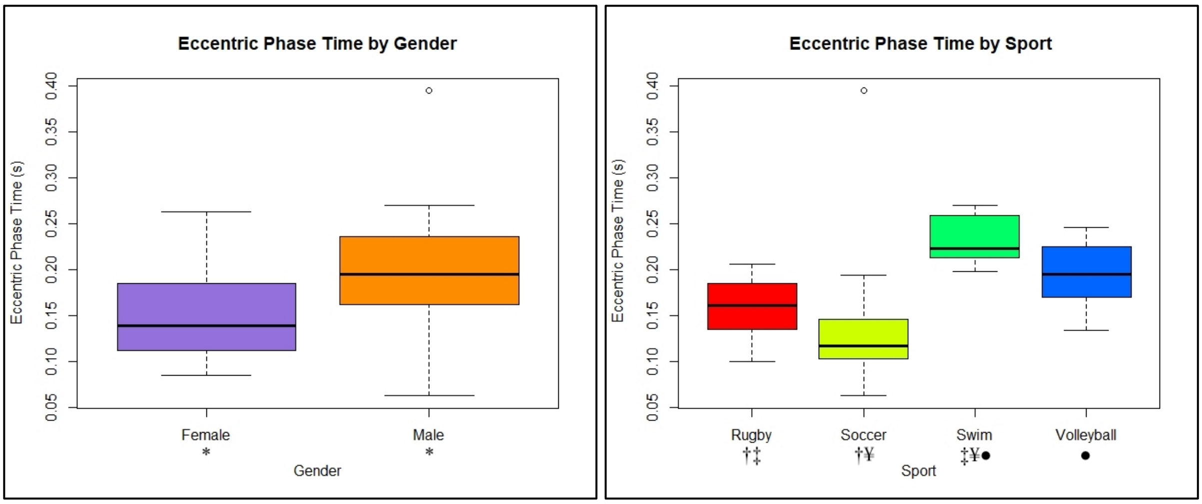
As the classification of the force-time curve shape is a categorical outcome rather than a continuous outcome, participants' jump results in Table 6 are classified by the majority out of the five jump trials (i.e. three or more of their jump trials). For the soccer groups, 42.9% of both male and female participants exhibited Class 1 jumps, while for the other groups, participants exhibited entirely Class 2 jumps (except for one participant female rugby).

Table 6: Jump Classification of Majority of Jump Trials

		Classification of Majority of CMJs				
Gender	Sport	Class 1		Class 2		N
		#	(% of N)	#	(% of N)	
Male	Swim	0	(0.0)	5	(100.0)	5
	Volleyball	0	(0.0)	12	(100.0)	12
	Soccer	3	(42.9)	4	(57.1)	7
	Total	3	(12.5)	21	(87.5)	24
Female	Swim	0	(0.0)	5	(100.0)	5
	Rugby	1	(5.0)	19	(95.0)	20
	Soccer	9	(42.9)	12	(57.1)	21
	Total	10	(21.7)	36	(78.3)	46
Total	Swim	0	(0.0)	10	(100.0)	10
	Volleyball	0	(0.0)	12	(100.0)	12
	Rugby	1	(5.0)	19	(95.0)	20
	Soccer	12	(42.9)	16	(57.1)	28
	Total	13	(18.6)	57	(81.4)	70

2.2.3 Phase Lengths

Figure 8 compares the time participants spent in the eccentric jump phase. Significant differences for the time spent in the eccentric jump phase exist between female (0.150±0.042 s, n=46) and male (0.197±0.066 s, n=24) groups. The swim group (0.232±0.027 s, n=10) spent significantly longer in the eccentric jump phase compared to the other three sport groups: rugby (0.161±0.030 s, n=20), soccer (0.133±0.060 s, n=28) and volleyball (0.197±0.034 s, n=12). Rugby participants also spent significantly longer in the eccentric time phase compared to soccer participants.

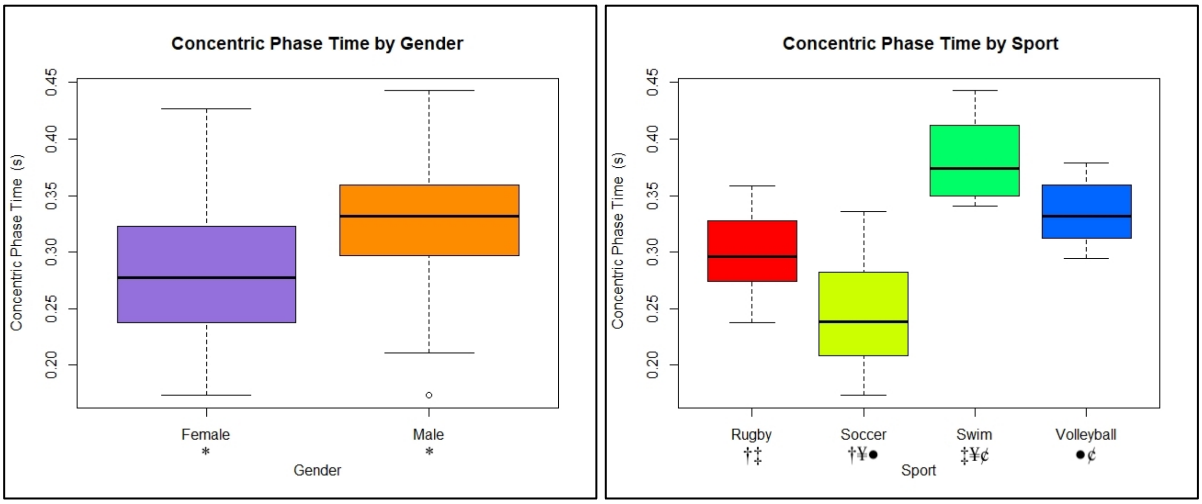


(a) *(b)*

Figure 8: Eccentric Phase Time by a) gender and b) sport.

Note: matching symbols under group title indicates a significant difference at the 0.05 level.

Figure 9 shows the time that participants spent in the concentric jump phase. Male participants (0.323±0.060 s, n=24) spent significantly longer in the concentric jump phase than female participants (0.279±0.058, n=46). Additionally, the soccer participants (0.244±0.044 s, n=28) spent significantly less time than the other three groups, rugby (0.297±0.036 s, n=20), swim (0.382±0.036 s, n=10), and volleyball (0.335±0.028 s, n=12). The swim group spent a significantly longer time in the concentric jump phase compared to all the other groups.

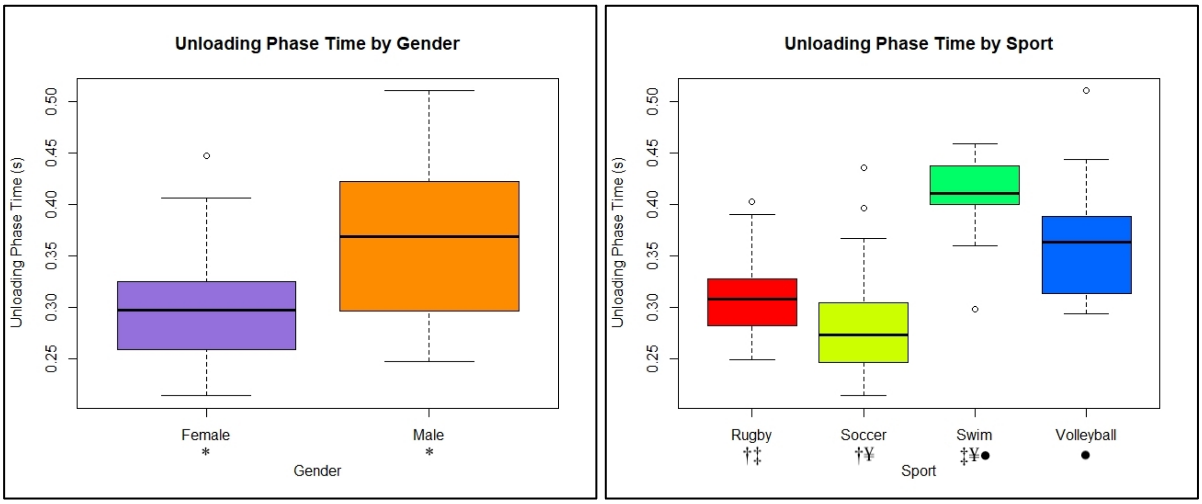


(a) *(b)*

Figure 9: Concentric Phase Time by a) gender and b) sport.

Note: matching symbols under group title indicates a significant difference at the 0.05 level.

Time spent in the unloading phase is compared in Figure 10. Males (0.366±0.069 s, n=24) also spent significantly longer in the unloading phase than females (0.300±0.055 s, n=46). The swim group (0.406±0.047 s, n=10) spent significantly longer in the loading phase compared to all the other groups: rugby (0.311±0.039 s, n=20), soccer (0.282±0.054 s, n=28), and volleyball (0.367±0.064 s, n=12). There was also a significant difference between rugby and soccer, with rugby spending a longer time in the unloading phase.

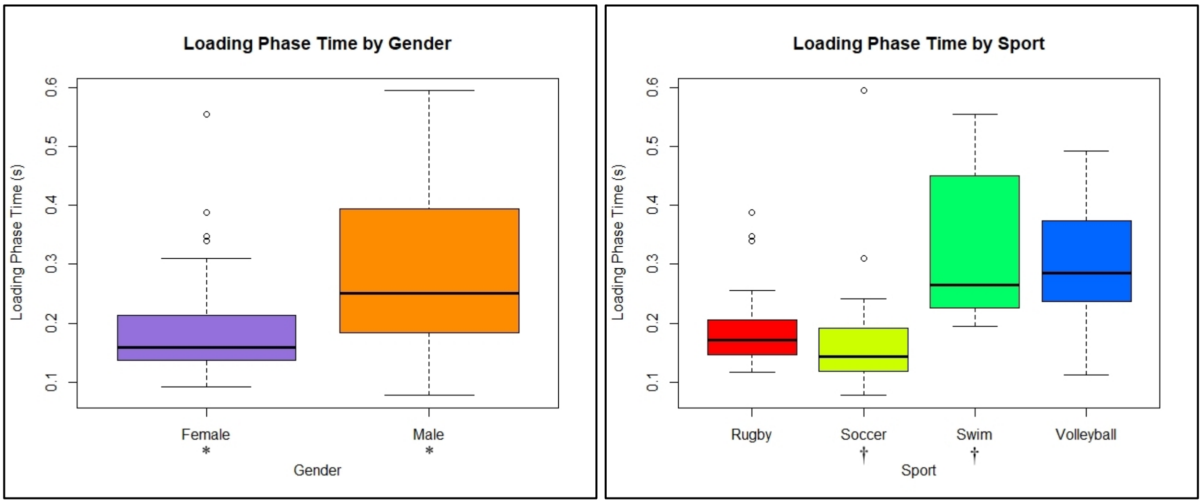


(a) *(b)*

Figure 10: Unloading Phase Time by a) gender and b) sport.

Note: matching symbols under group title indicates a significant difference at the 0.05 level.

Loading phase time is shown in Figure 11. Males (0.293±0.140 s, n=24) spent significantly longer in the loading phase compared to females (0.188±0.087 s, n=46), and swim participants (0.336±0.135 s, n=10) spent significantly longer in this phase compared to soccer participants (0.170±0.098 s, n=28).

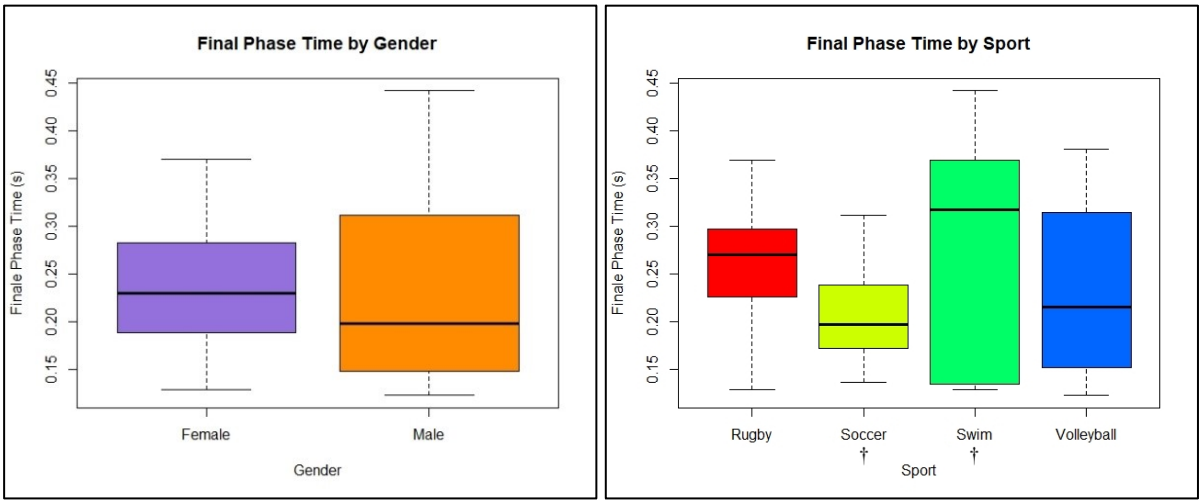


(a) *(b)*

Figure 11: Loading Phase Time by a) gender and b) sport.

Note: matching symbols under group title indicates a significant difference at the 0.05 level.

The results for time spent in the final phase are shown in Figure 12.There were no significant difference between male and female found for final phase time. Between sports, the only significant difference was found between soccer (0.206±0.046 s, n=28) and swim (0.278±0.118 s, n=10).

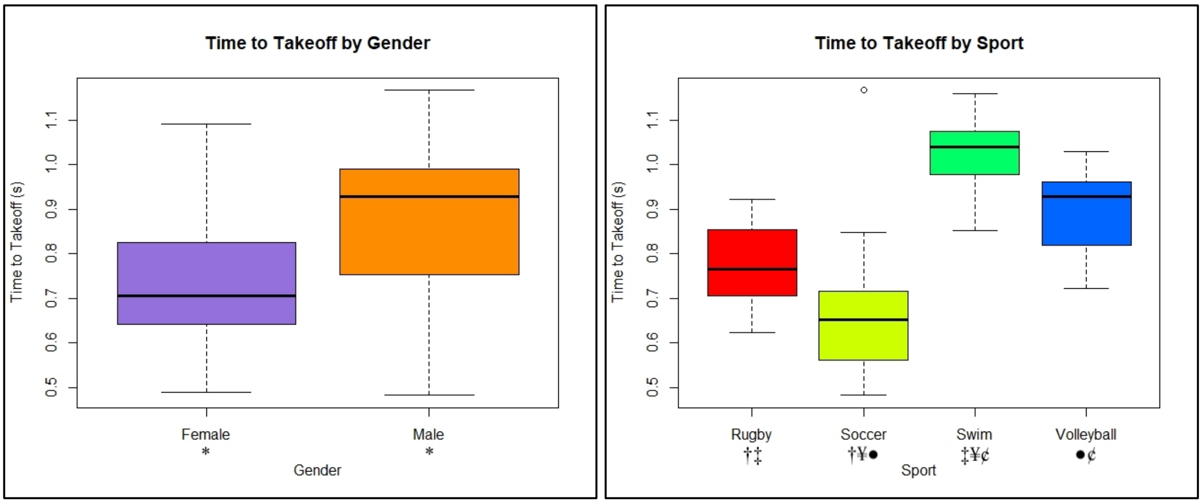


(a) *(b)*

Figure 12: Final Phase Time by a) gender and b) sport.

Note: matching symbols under group title indicates a significant difference at the 0.05 level.

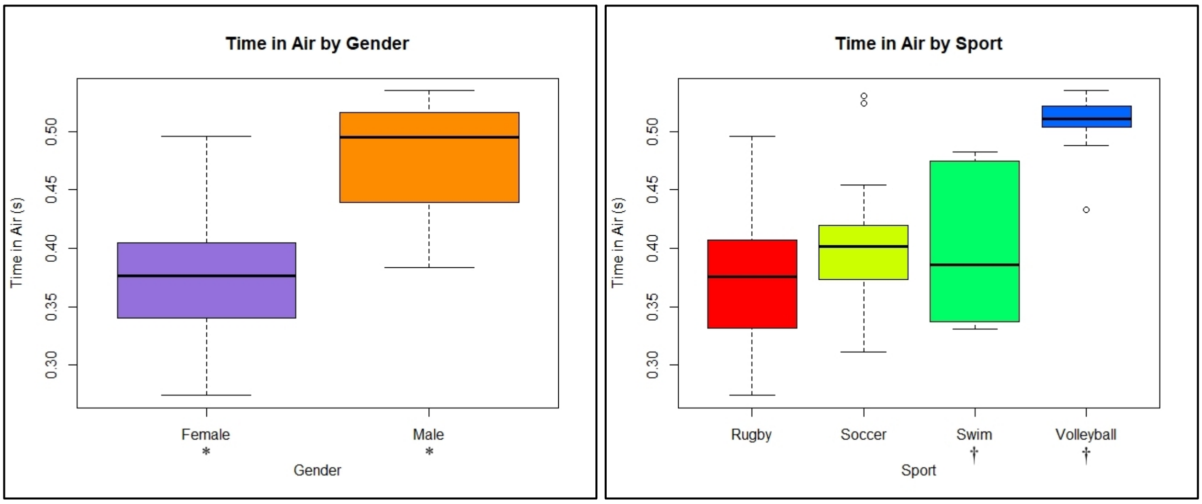
Time to takeoff is compared by group in Figure 13. Males (0.886±0.171 s, n=24) took significantly longer than females (0.729±0.146 s, n=46) to takeoff during their jumps. Soccer (0.659±0.142 s, n=28) took significantly less time to takeoff compared to all other groups—rugby (0.769±0.090 s, n=20), volleyball (0.899±0.096 s, n=12) and swim (1.019±0.089 s, n=10). Swim took significantly more time to takeoff compared to all other groups.



(a) *(b)*

Figure 13: Time to Takeoff by a) gender and b) sport.
Note: matching symbols under group title indicates a significant difference at the 0.05 level.

Figure 14 shows the results of the time spent in air by the participants. Males (0.480±0.046 s, n=24) spent a significantly longer amount of time in the air compared to females (0.374±0.043 s, n=46), and volleyball participants (0.506±0.026 s, n=12) spent significantly longer in the air than swim participants (0.397±0.062 s, n=10).

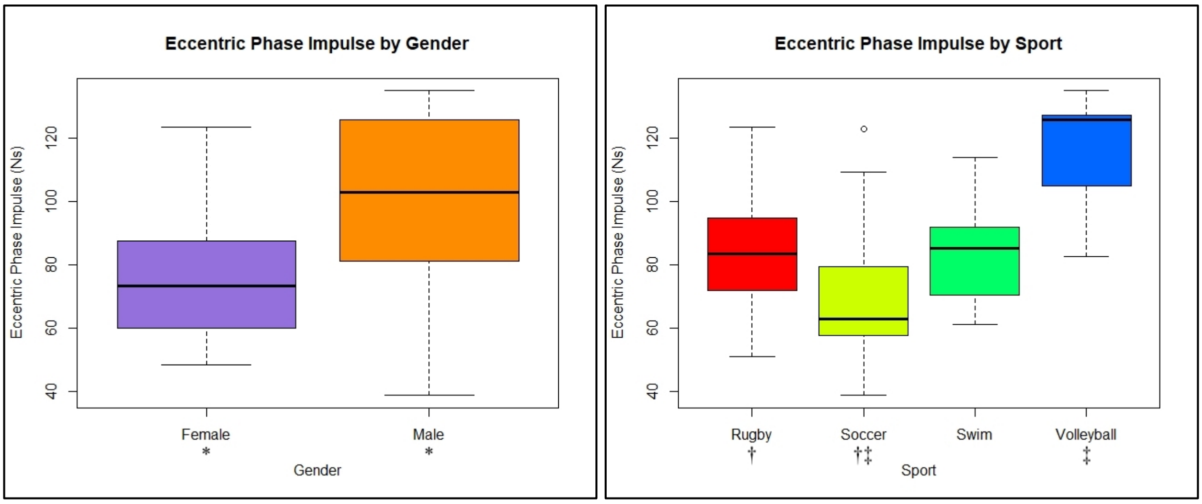


(a) *(b)*

Figure 14: Time in Air by a) gender and b) sport.
Note: matching symbols under group title indicates a significant difference at the 0.05 level.

2.2.4 Impulse

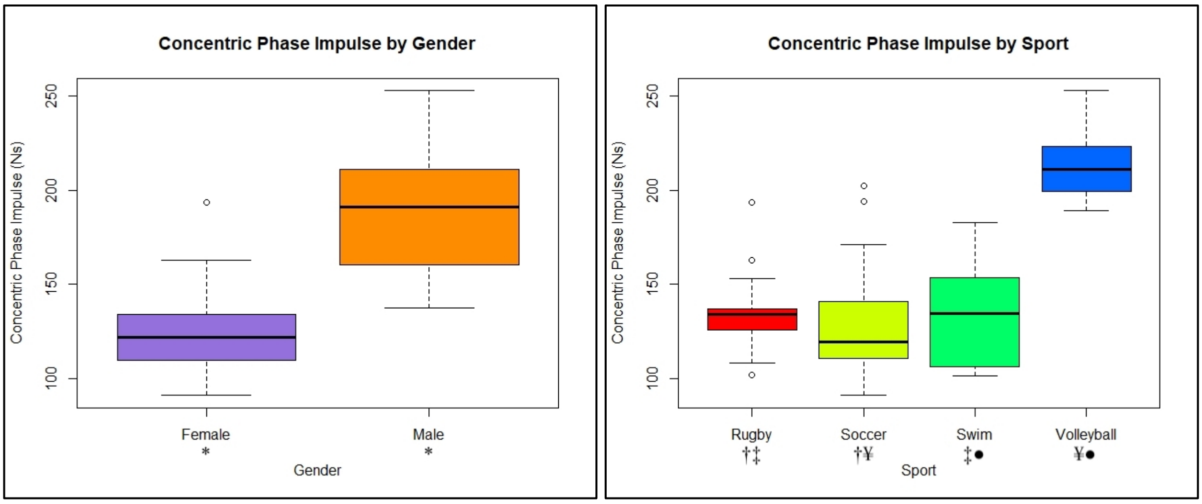
The kinetic impulse during the eccentric jump phase is compared in Figure 15. Males (99.44±27.15 N·s, n=24) had a larger kinetic impulse compared to females (74.94±17.25 N·s, n=46). Additionally, the soccer group (69.38±18.90 N·s, n=28) had a lower eccentric impulse compared to the rugby group (83.39±18.47 N·s, n=20) and the volleyball group (115.78±17.90 N·s, n=12).



(a) *(b)*

Figure 15: Eccentric Phase Impulse by a) gender and b) sport.
Note: matching symbols under group title indicates a significant difference at the 0.05 level.

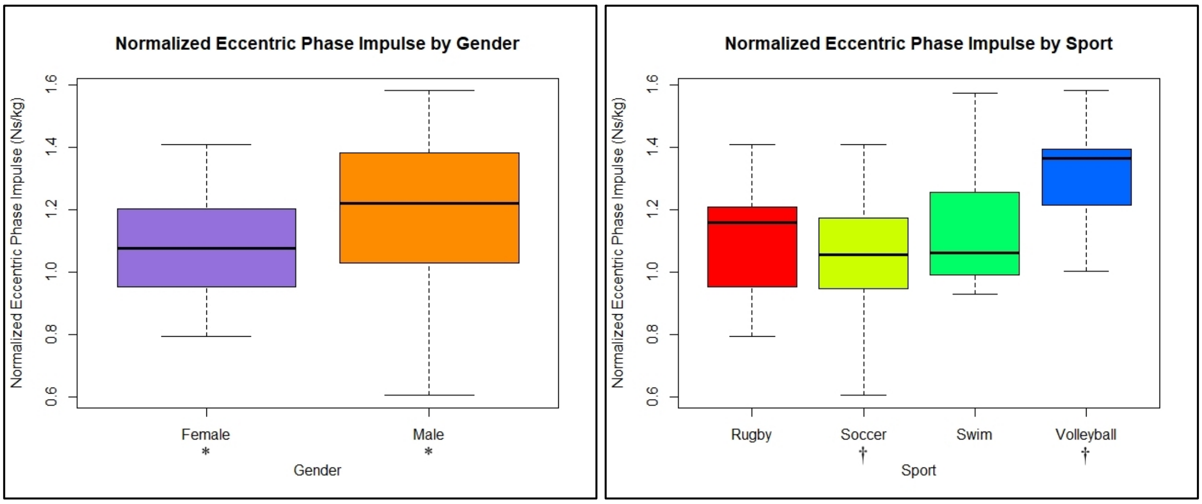
The kinetic impulse during the concentric phase is shown in Figure 16. Males (187.22±32.82 N·s, n=24) had a significantly greater concentric phase impulse compared to females (122.83±19.12 N·s, n=46). Additionally, rugby (133.64±20.38 N·s, n=20) had a significantly greater concentric impulse than soccer (127.29±27.97 N·s, n=28) and swim (135.61±27.45 N·s, n=10). Volleyball (212.54±18.72 N·s, n=12) had a significantly larger concentric impulse compared to soccer and swim, but not rugby. Note here that according to these reported values, the swim group had a slightly larger average concentric impulse than the rugby group (although the average are very close). However, the Tukey HSD test results indicate that the rugby had statistically significant larger concentric phase impulses compared to the swim group, with the adjustment based on the variance estimated from the pooled data of all four sport groups.



(a) *(b)*

Figure 16: Concentric Phase Impulse by a) gender and b) sport.
Note: matching symbols under group title indicates a significant difference at the 0.05 level.

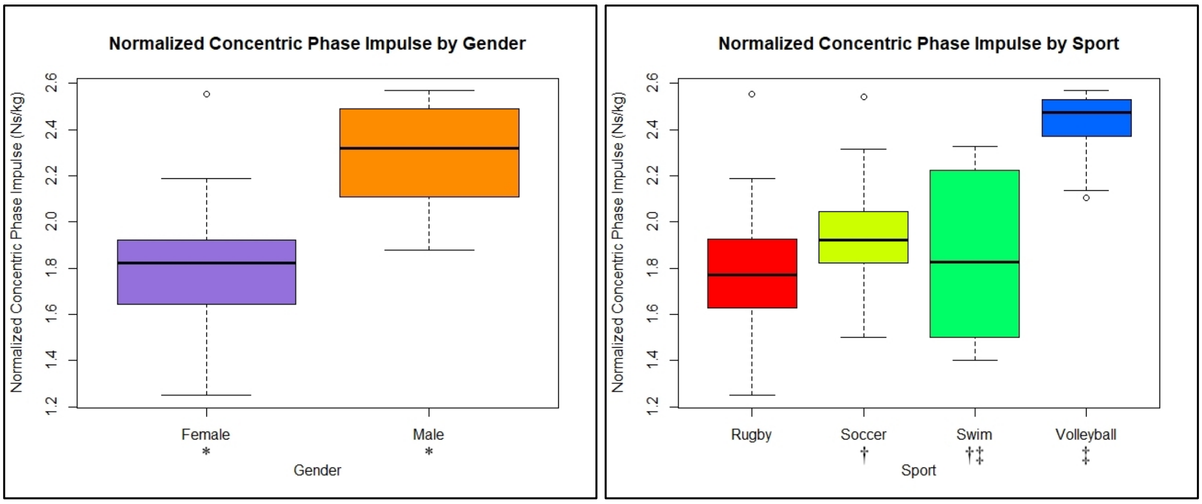
After normalizing the eccentric phase impulse to participants' mass (Figure 17), there was still statistically significant differences between males (1.21±0.24 N·s/kg, n=24) and females (1.08±0.16 N·s/kg, n=46), but for the sport groups, there were only differences between soccer (1.05±0.18 N·s/kg, n=28) and volleyball (1.31±0.16 N·s/kg, n=12).



(a) *(b)*

Figure 17: Normalized Eccentric Phase Impulse by a) gender and b) sport.
Note: matching symbols under group title indicates a significant difference at the 0.05 level.

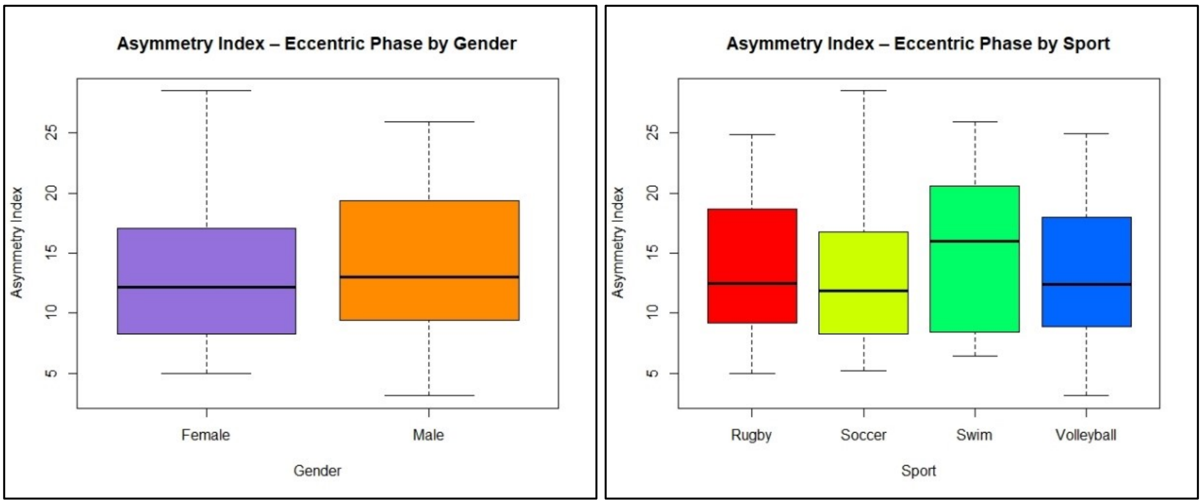
Figure 18 shows the concentric phase impulse normalized to participants' body mass. Males (2.29±0.12 N·s/kg, n=24) had larger normalized concentric phase impulse compared to females (1.79±0.23 N·s/kg, n=46). The swim group (1.85±0.13 N·s/kg, n=10) had statistically significantly smaller normalized concentric phase impulses compared to the soccer group (1.94±0.21 N·s/kg, n=28) and the volleyball group (2.42±0.15 N·s/kg, n=12).



(a) *(b)*

Figure 18: Normalized Concentric Phase Impulse by a) gender and b) sport.
Note: matching symbols under group title indicates a significant difference at the 0.05 level.

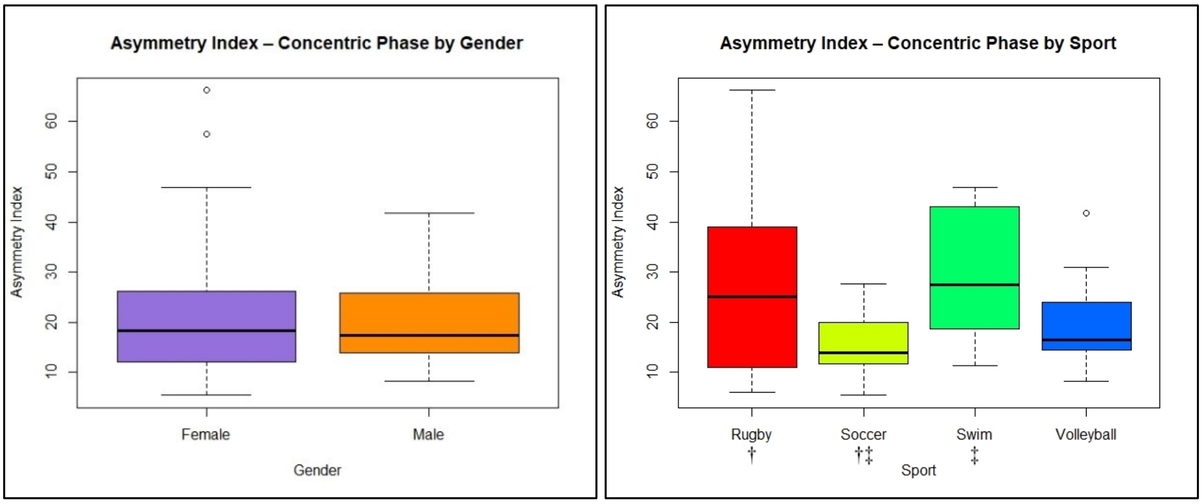
The asymmetry index in the eccentric phase is shown in Figure 19. No significant differences were found between any of the compared groups (male vs female, and comparing the four sport types) for the asymmetry index during the eccentric phase.



(a) *(b)*

Figure 19: Asymmetry Index – Eccentric Phase by a) gender and b) sport.
Note: matching symbols under group title indicates a significant difference at the 0.05 level.

Figure 20 shows the asymmetry index for participants during the concentric phase. No significant differences between male and female were found for the asymmetry index in the concentric phase, but significant differences were found between soccer (15.64±6.10, n=28) and rugby (26.25±17.51, n=20), and soccer and swim (28.53±13.07, n=10).

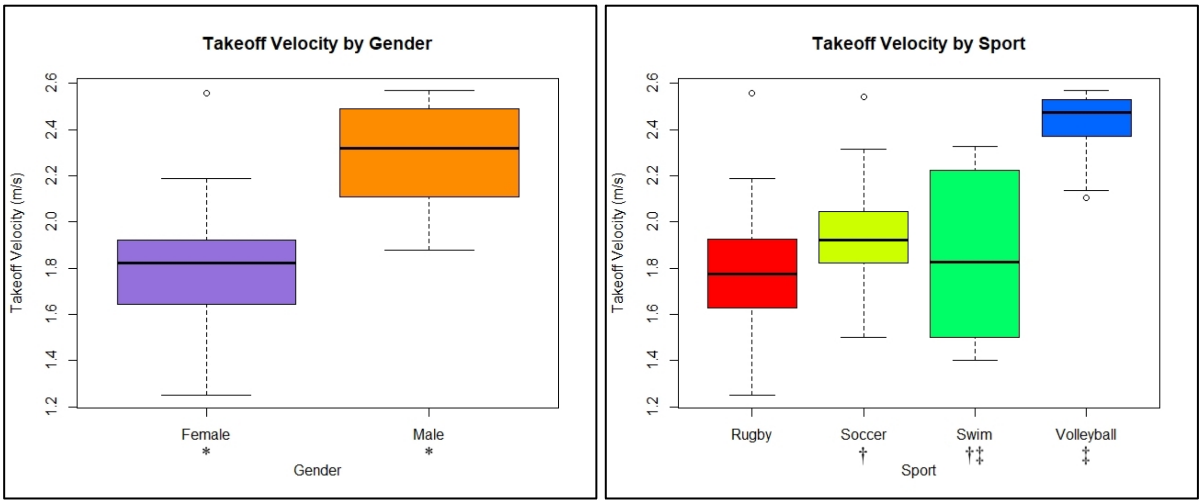


(a) *(b)*

Figure 20: Asymmetry Index – Concentric Phase by a) gender and b) sport.
Note: matching symbols under group title indicates a significant difference at the 0.05 level.

2.2.5 Displacement and Velocity

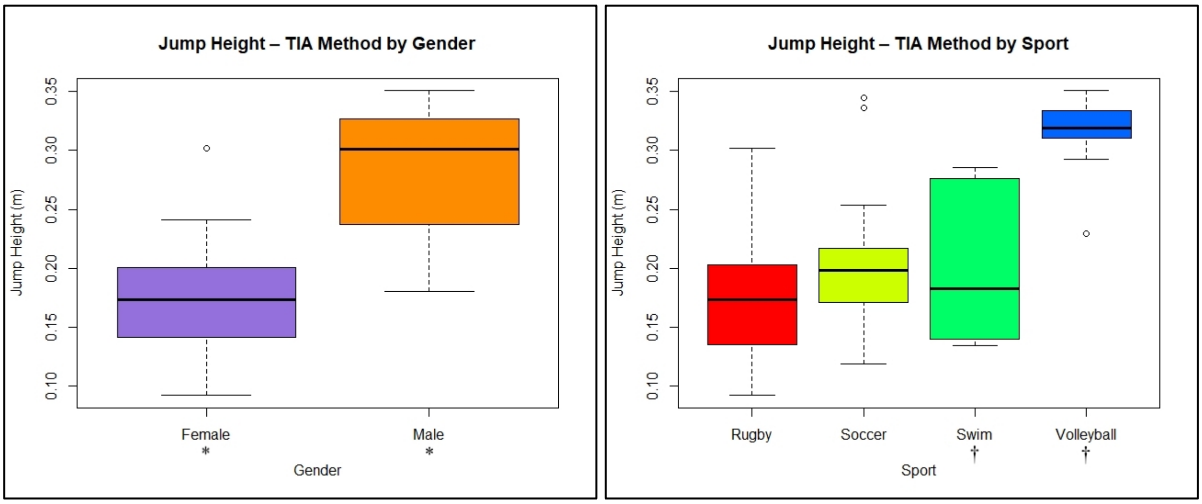
The takeoff velocity (the velocity at t_4) is compared in Figure 21. There was a significant difference between male (2.29±0.21 m/s, n=24) and female (1.79±0.23 m/s, n=46) participants, and a significant difference between the swim (1.85±0.34 m/s, n=10) and soccer (1.94±0.21, n=28) groups, as well as the swim and volleyball (2.42±0.15 m/s, n=12) groups.



(a) *(b)*

Figure 21: Takeoff Velocity by a) gender and b) sport.
Note: matching symbols under group title indicates a significant difference at the 0.05 level.

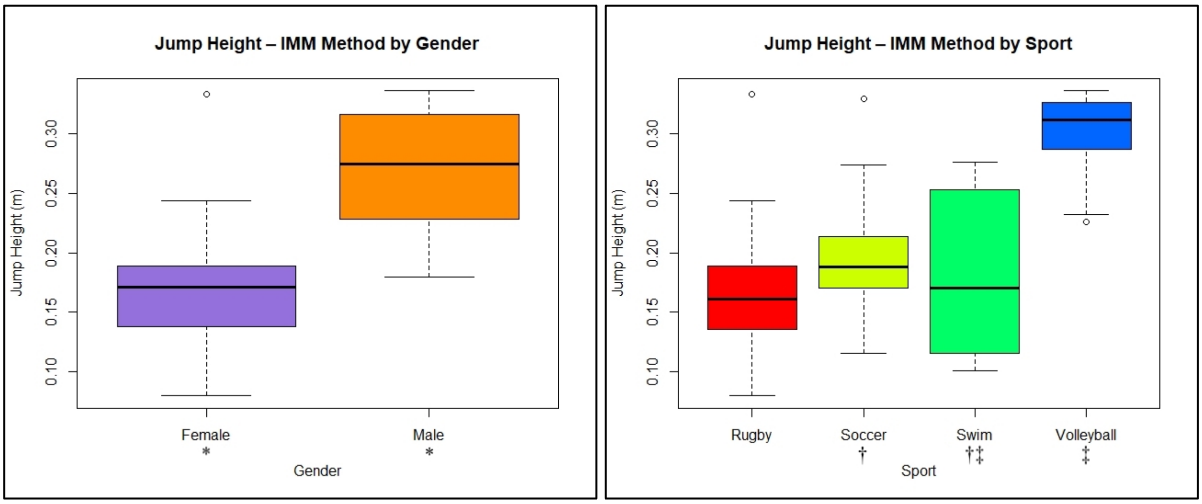
The jump height was first compared using the height calculated with the Time in Air (TIA) method (Figure 22). Males (0.285±0.052 m, n=24) jump significantly higher than females (0.174±0.040 m, n=46). Volleyball participants (0.315±0.031 m, n=12) jumped significantly higher than swim participants (0.198±0.062 m, n=10).



(a) *(b)*

Figure 22: Jump Height – TIA Method by a) gender and b) sport.
Note: matching symbols under group title indicates a significant difference at the 0.05 level.

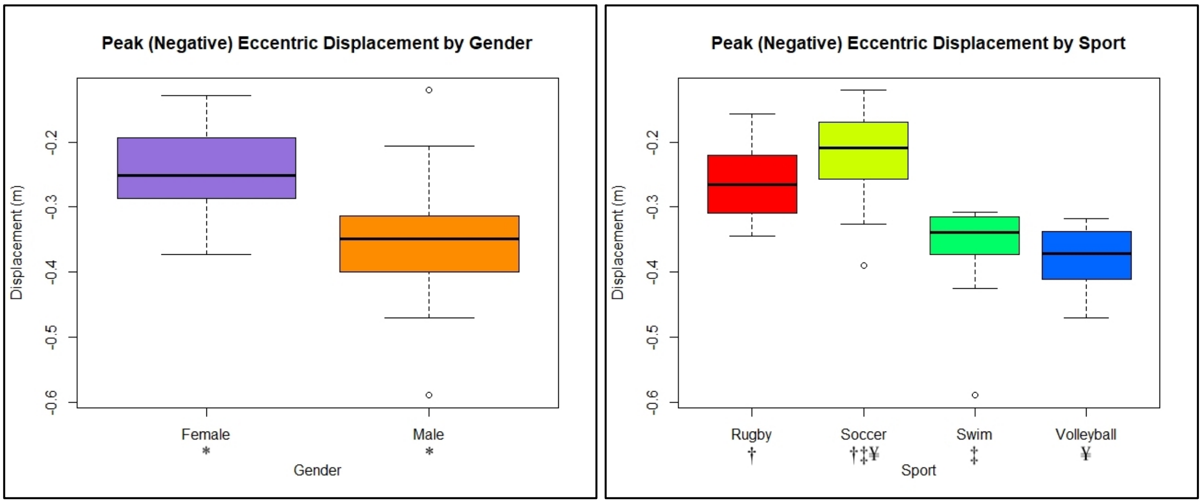
The jump height was also calculated and compared using the Impulse Momentum (IMM) method (Figure 23). Similar to the jump height found using the TIA method, males (0.271±0.024 m, n=24) had significantly higher jump heights than females (0.167±0.043 m, n=46). Swim (0.180±0.065 m, n=10) also had significantly lower jump height than volleyball (0.300±0.037, n=12) again, but with the IMM method, there was also a significant difference between swim and soccer (0.194±0.043 m, n=28).



(a) *(b)*

Figure 23: Jump Height – IMM Method by a) gender and b) sport.
Note: matching symbols under group title indicates a significant difference at the 0.05 level.

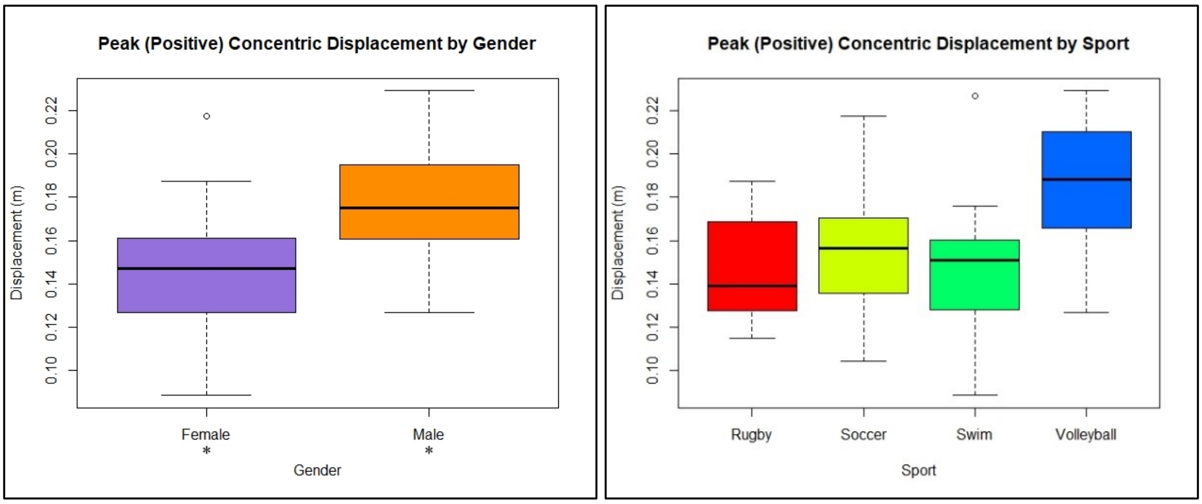
Peak eccentric displacement (in the negative direction) is shown in Figure 24. The female group (-0.243±0.062 m, n=46) exhibited a lesser change in displacement during the eccentric phase compared to the male group (-0.348±0.094 m, n=24). When comparing sports, the soccer group (-0.217±0.061 m, n=28) had significantly less of a change in displacement in the negative direction compared to the other three groups: rugby (-0.263±0.052 m, n=20), swim (-0.368±0.086 m, n=10) and volleyball (-0.377±0.046 m, n=12).



(a) *(b)*

Figure 24: Peak (Negative) Eccentric Displacement by a) gender and b) sport.
Note: matching symbols under group title indicates a significant difference at the 0.05 level.

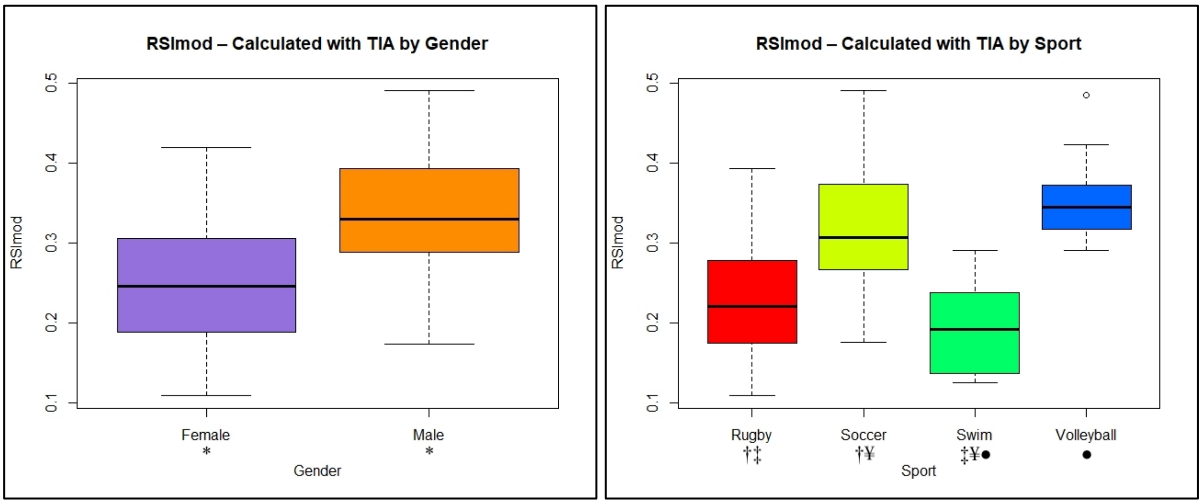
Peak concentric displacement (in the positive direction) is compared in Figure 25. The peak concentric displacement for males (0.179±0.027 m, n=24) was significantly greater than that of females (0.144±0.026 m, n=26). There was no significant difference found between sport groups or the peak concentric displacement.



(a) *(b)*

Figure 25: Peak (Positive) Concentric Displacement by a) gender and b) sport.
Note: matching symbols under group title indicates a significant difference at the 0.05 level.

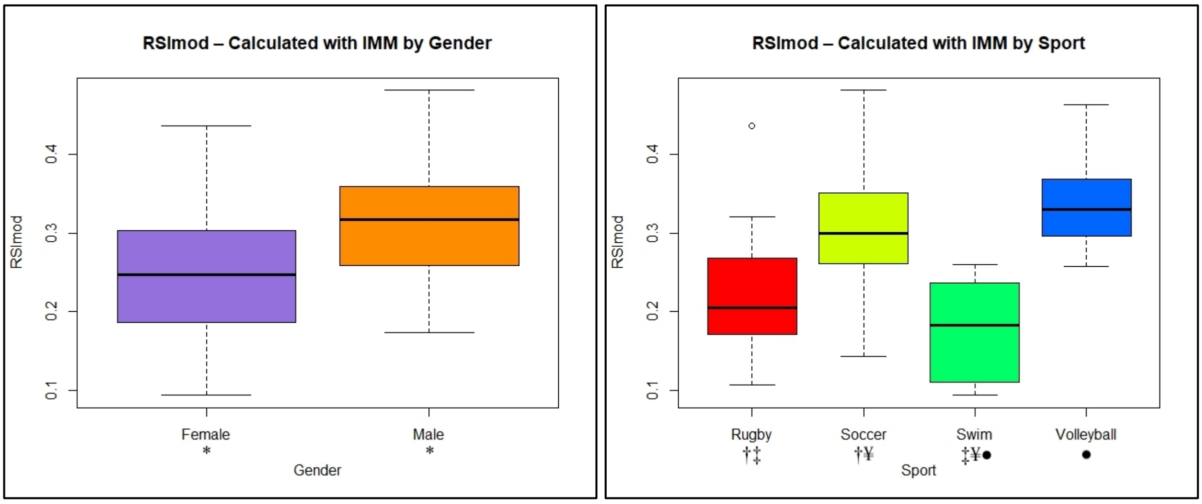
RSImod was first calculated and compared using the jump height calculated with the TIA method (Figure 26). Males (0.333±0.082, n=24) had a significantly higher RSImod using the TIA method compared with females (0.252±0.081, n=46). The swim group (0.196±0.058, n=10) had a significantly lower RSImod calculated with TIA compared to the other three groups: rugby (0.225±0.071, n=20), soccer (0.316±0.077, n=28), and volleyball (0.355±0.055, n=12). Additionally, rugby had a significantly smaller RSImod than soccer.



(a) *(b)*

Figure 26: RSImod – Calculated with TIA by a) gender and b) sport.
Note: matching symbols under group title indicates a significant difference at the 0.05 level.

The RSImod was then calculated with the jump height using the IMM method (Figure 27). Using the IMM jump height resulted in the same significant differences found using the TIA jump height between the following groups: male (0.318±0.079, n=24) and female (0.242±0.085, n=46), swim (0.177±0.060, n=10) and the other three sport groups (rugby (0.221±0.079, n=20), soccer (0.304±0.074, n=28), volleyball (0.338±0.057, n=12)), and between rugby and soccer.



(a) *(b)*

Figure 27: RSImod – Calculated with IMM by a) gender and b) sport.
Note: matching symbols under group title indicates a significant difference at the 0.05 level.

2.3 Discussion

2.3.1 Rugby

The body mass of rugby players was significantly greater than that of soccer players, with no difference when compared to swim or volleyball. In elite rugby players, there can be a large variation in body size and composition between the different positions. Traditionally forward have been heavier and taller with a higher percentage of body fat compared to backs, but recently, forwards have started developing a greater total mass and higher muscle percentage (Duthie, Pyne, & Hooper, 2003). In professional rugby athletes, a relationship between body mass and absolute peak force in a CMJ has been found (Crewther, et al., 2012). This indicates that in elite athletes, body mass likely comes from force-producing body mass (muscle) rather than non-force producing body-mass (body fat) (Crewther, et al., 2012). In rugby, where body-to-body contact is frequent and of a high intensity, a greater body size (especially muscle) may be of advantage in gameplay.

Rugby had a significantly greater peak force than swim, and a significantly lower normalized peak force compared to soccer. The forces produced during vertical jumps have been found to correlate with scrummaging (scrum) forces in rugby (Duthie, Pyne, & Hooper, 2003; Robinson & Mills, 2000). A scrum is a means to restart play in rugby, and requires players to closely pack together with their heads down to try to win possession of the ball. Leg power is important in a scrum, for both maintaining force as well as generating explosive force to push against one another once the ball is put in (Robinson & Mills, 2000). The counter movement jump, with a strong validity and reliability, have been shown to be helpful in assessing rugby players' power (Duthie, Pyne, & Hooper, 2003). This suggests that rugby players could benefit from plyometric exercises in their training, so that they can quantifiably improve their CMJ performance (relative to other sports) and thus, improve their gameplay. Rugby athletes that are high-performing in CMJs generate a deeper and faster countermovement, and apply higher ground reaction forces in the eccentric and concentric phases (Floria, Gomez-Landero, Suarez-Arrones, & Harrison, 2014). Some literature has found that there are differences between peak forces of professional rugby athletes, depending on their position, while other studies have found that there is no significant difference between positions (Crewther, et al., 2012). However, CMJs have been established to be helpful in quantifying, assessing and monitoring the needs of specific positions in rugby, and should be incorporated into training and monitoring (Crewther, et al., 2012).

The rugby athletes spent significantly longer than soccer and significantly shorter than swim in the eccentric phase, concentric phase, unloading phase, and total time to takeoff. However, there were no statistical difference between the rugby group and other groups for the loading phase time, the final phase time, and the time spent in the air.

Rugby had a significantly greater eccentric impulse compared to soccer, and a significantly greater concentric impulse compared to swim and soccer. Once the eccentric and concentric impulses were normalized to body mass, there were no significant differences between rugby and any of the other sport groups.

Rugby had no difference in asymmetry index in the eccentric phase compared to the other groups, but the rugby group did have a significantly larger asymmetry index in the concentric phase compared to the soccer group. Rugby's eccentric asymmetry index was 13.6, and concentric asymmetry index was 26.3. One study proposed that in rugby players, measuring asymmetry may identify individuals predisposed to injury, but found that normative values of asymmetry in rugby players are lacking (Marshall, et al., 2015). The study found that rugby players tend to exhibit bilateral symmetry across a variety of testes, including single leg drop jumps, single leg hurdle hops, and running cuts (Marshall, et al., 2015). All of the tests in the study, however, were measuring performance on one leg at a time, and then comparing them. In this current study, both legs are being used to perform the CMJ, but the difference in how the body distributes the applied force was being quantified. This may explain why this current study found a difference in leg asymmetry.

There was no significant difference between the other sport groups and the rugby group for the takeoff velocity. This is surprising, as rugby had a greater concentric impulse than both swim and soccer, and a greater concentric impulse should lead to a greater takeoff velocity (based on the impulse-momentum theorem). Also interestingly, there was no significant differences between rugby players in this study and the jump height calculated with either method, the TIA or IMM method. One study found that jump height was correlated to absolute peak force in rugby, but not the peak force normalized to body mass (Tillin, Pain, & Folland, 2013). In terms of vertical jump height, rugby position may influence performance. One study found that backs jumped higher than forwards, and out of the forwards, loose forwards jumped higher than props and hookers (Duthie, Pyne, & Hooper, 2003).

For eccentric displacement, rugby players squatted down significantly more than soccer players. However, there was no difference between the rugby group and the other sport groups for concentric displacement. In rugby, high-performing jumpers have been shown to have a lower squat position during CMJs (Floria, Gomez-Landero, Suarez-Arrones, & Harrison, 2014). This larger range of motion can be translated to a greater jump height (Floria, Gomez-Landero, Suarez-Arrones, & Harrison, 2014).

The rugby group had a significantly lower RSImod than the soccer group, but a significantly higher RSImod than the swim group (using both the TIA and IMM jump height). In a study of professional male rugby league players, it was found that a players with a high RSImod jumped higher with a shorter time to takeoff—which was achieved by a more rapid unweighting phase—and greater eccentric and concentric forces (McMahon, Jones, Suchomel, Lake, & Comfort, 2018). However, the high- and low-RSImod rugby players still had similar countermovement depths (McMahon, Jones, Suchomel, Lake, & Comfort, 2018). This means that the higher RSImod was likely attributed to kinetic and kinematic parameters (McMahon, Jones, Suchomel, Lake, & Comfort, 2018). The same study found that a high RSImod correlated with a taller but thinner impulse on the force-time curve, which would suggest a high force being applied over a short period of time—an explosive jump.

In a study that observed rugby players' shape of their force-time curve during CMJs and sorted them into unimodal (one peak) or bimodal (two peak) curves, it was found that bimodal peak groups performed larger net impulses and slightly larger heights than the unimodal group (McMaster, 2016). Since all the rugby athletes in this current study exhibited two peak (Class 2) force-time curves, it is possible that the jump heights observed were higher than if there were also Class 1 jumpers present. A separate study on rugby players' force-time curve shape found that there was no difference between unimodal and bimodal groups for jump height and RSImod, but that the bimodal force-time curve may indicate an inefficient stretch-shortening cycle (Kennedy & Drake, 2018). However their definition of bimodal and unimodal does not exactly match the definition of Class 1 and Class 2 in the current study, and likely grouped more players into their unimodal group (52% of their jumpers) than the current study would.

Experience and level of play have also been shown to correlate with performance in rugby players. In one study, the senior players had a larger jump height, RSImod, concentric displacement, and normalized concentric impulse compared to more junior players (McMahon, Murphy, Rej, & Comfort, 2017). This large concentric impulse applied by the senior players transferred into a greater velocity in the concentric phase and at takeoff, allowing them to jump higher (McMahon, Murphy, Rej, & Comfort, 2017). The senior players also had greater eccentric and concentric displacement compared to the more junior players, but with similar movement time (McMahon, Murphy, Rej, & Comfort, 2017). This suggests that the more junior players could benefit from greater flexibility and angular velocity to perform better (McMahon, Murphy, Rej, & Comfort, 2017).

One limitation to the rugby analysis was that the group was made up of entirely female rugby athletes. At this point, there is no male rugby varsity league in U Sports, the Canadian national university sport organization, and so there is no male rugby varsity team to collect data from.

2.3.2 Soccer

Soccer players had the lowest mass (significant with swim and rugby) out of all the groups. The soccer group only had a significantly larger peak force than the swim group, but after normalizing peak force to body mass, soccer significantly had the largest normalized peak force compared to the three other sport groups. This is an interesting finding, as soccer players likely don’t perform movements similar to CMJs in gameplay as often as volleyball players would. However, since soccer is a highly aerobic sport, soccer players must have a good balance between maintaining enough muscle mass to be strong, and also having a low enough body mass for adequate endurance throughout gameplay.

The soccer players had the lowest phase times for all the phases measured, but only some of these were significantly different. Soccer’s eccentric phase time was less than rugby and swim; soccer’s concentric phase time was less than all the groups compared. The soccer group’s unloading phase time was less than rugby and swim, and both the loading phase time and the final phase time were less than that of swim. For the total time to takeoff, the soccer group significantly had the lowest time out of all the sport groups. However, there was no difference for time spent in the air. This means that the soccer players’ jumps were the quickest, but the time in the air wasn’t different from the rest of the sports.

The soccer athletes had a significantly lower eccentric and concentric impulse compared to rugby and volleyball (and lower than swim too, but not significantly so). Soccer’s normalized eccentric impulse was significantly lower than that of volleyball (and lowest overall), and their normalized concentric impulse was larger than that of swim. While swim had lower impulses likely due to the low force the swimmers were generating, the soccer athletes likely have low impulses due to the short period of time over which they were jumping.

The soccer group exhibited no difference in eccentric asymmetry index from the other sports, and had the lowest concentric asymmetry index (only significant with rugby and swim). The asymmetry index for soccer for the eccentric and concentric phases were 13.8 and 15.6, respectively. Leg dominance may cause asymmetry in the other leg’s muscle groups in soccer players, which may predispose that leg to injury (Zakas, 2006). Soccer may be especially susceptible to this, since players rarely use legs equally for shooting and other technical skills (Zakas, 2006). However, in a study on the effects of leg dominance and strength parameters in soccer players, there was no association found between the preferred leg and strength parameters between the left and right side of the body (Zakas, 2006). This may be because soccer training sessions and matches have created a strength balance such that there is little or no difference between the dominant and non-dominant leg. It appears that the asymmetry index between the two phases (eccentric and concentric) were much more similar for soccer compared to the other three groups, where concentric was usually much larger. One study found that the asymmetry between strength of the left and right leg didn't influence the jump height, but players with better strength symmetry would have better jumping capacity in usage of the stretch-shortening cycle (Krizaj, Rauter, Vodicar, Hadzic, & Simenko, 2019).

There was no difference in soccer's jump height calculated using the TIA method, but using the IMM method, soccer was found to have jumped significantly higher then swim. Vertical jumping is important in soccer to gain advantage over opponents in both offensive and defensive play (Krizaj, Rauter, Vodicar, Hadzic, & Simenko, 2019). Soccer's takeoff velocity was also greater than that of the swimming group's.

In this current study, soccer's RSImod (with both TIA and IMM jump height) was significantly greater than that of rugby and swim. This suggests that the soccer players are jumping more explosively than rugby and swim, and utilizing the strength-shortening cycle more effectively. In high-level male soccer players, significant correlations were found for RSImod with jump time, unloading phase force, eccentric rate of force development, average concentric force, and concentric displacement (Barker, Harry, & Mercer, 2017).

The eccentric displacement in the soccer group was significantly the least compared to the other three groups—indicating that the soccer group is not squatting down as deeply as the rest. For the concentric displacement, the soccer group didn't have any significant differences with the other sport groups. In high-level male soccer players, significant correlations have been found for jump height with RSImod and concentric displacement (Barker, Harry, & Mercer, 2017). However, this current study did not find these same correlations.

The most common soccer injuries involve the ankle and knee joints, as well as the muscles and ligaments in thigh and calves (Fried & Lloyd, 1992). Rehabilitation is crucial to restore lower limb strength and endurance, and to prevent injury recurrence—as re-injuries are often more severe than the initial injury (Fried & Lloyd, 1992). Additionally, if proper lower limb strengthening is not incorporated into soccer athlete's training, the risk of injury may increase. The ratio for strength of quadriceps to strength of hamstring is about 3:2, but in neglect of proper hamstring training, it can be as high as 6:1 (Fried & Lloyd, 1992).

Plyometric training has been shown to improve both cardiovascular and neuromuscular fitness in soccer players: it increases VO2 max, maximal strength, sprinting speed, kicking, endurance, agility, and vertical jumping ability of both male and female soccer players at any level (Wang & Zhang, 2016). Additionally, plyometric training is important for improving strength of muscles and tendons to help avoid injuries (Wang & Zhang, 2016). Higher concentric quad strength has been shown to significantly correlate with higher vertical jump height in soccer players, indicating that strong quads allow for a better jumping capacity (Krizaj, Rauter, Vodicar, Hadzic, & Simenko, 2019). This reinforces the importance of plyometric training and proper lower limb strengthening in soccer, both to prevent injury and to optimize performance. Additionally, jumping capacity has been shown to correlate to sprint performance in soccer players (explained by the strength-shortening cycle) (Krizaj, Rauter, Vodicar, Hadzic, & Simenko, 2019). This suggests that if a player is not performing as well in CMJs, they may also be lower performing in sprints during games.

2.3.3 Swimming

Swimming is unique compared to the other sports in this study, as it is the only group that competes in water rather than in an open air environment on a court or field. The requirement to perform movements through water means that swimmers are met with a greater resistance to muscle movements compared to soccer, volleyball or rugby. Additionally, swimming doesn't involve any upright running and/or vertical jumping movements, while the other three sports do. This may help to explain some of the observations in the swimming group during this study.

The swimming group had no significant difference between any of the other groups for body mass. The swim group significantly had the lowest absolute peak force out of all of the sport groups. When peak force was normalized to body mass, although the swim group still had the lowest normalized peak force, swimmers were only significantly different from the soccer group, which had the highest normalized peak force. There are no true vertical jumping movements made in swimming competitions, whereas vertical jumps can be found in the three other groups, which may help to explain the low peak force found in the swimmers. The two movements in swimming that mostly closely resemble a counter movement jump would likely be starts (entering the water from a block) and turns (performed against the pool wall at the end of a lap). Both of these motions require an explosive takeoff motion to propel the athlete's body in a specific direction, as do CMJs.

Swim starts have been identified to be significant components of competition, especially in short distance competitions (De La Fuente, Garcia, & Arellano, 2003; Benjanuvatra, Edmunds, & Blanksby, 2007; Lee, Huang, Wang, & Lin, 2001; West, Owen, Cunningham, Cook, & Kilduff, 2011; Garcia-Ramos, et al., 2016). Different swim starts are used depending on athlete preference, the stroke used, or the competition level, and two common options include the grab start and the track start. In a track start, the swimmer begins with knees and hips flexed, with one foot at the front of the block with toes curled over the edge, the other foot at the back of the block (sometimes on a wedge). Both hands grab the front edge of the block. Both the back and front leg push off the block to project the entire body in a straight line with arms in the streamline position into the water. A grab start is similar to a track start, but instead of having feet staggered, both feet are at the front of the block with toes curled over the edge. In a track start, the swimmer uses a coordinated two-legged takeoff to enter the water. The takeoff during a grab start appears very similar to the two-legged jump performed in a CMJ, except with the body's projectile motion in mostly a horizontal direction rather than a vertical one. Both track and grab starts require explosive movements paralleling the explosive movement necessary in vertical jumps, and the two activities appear to share similar muscle actions (Benjanuvatra, Edmunds, & Blanksby, 2007).

Swim start performance is typically measured as the time it takes to reach 15 meters, and includes time on the block, flight, entry into water, gliding through water, and underwater propulsion phases (Garcia-Ramos, et al., 2016). For an efficient start, swimmers must have both a quick reaction to the start signal, as well as a powerful force impulse generated on the starting block (Garcia-Ramos, et al., 2016). In competition, if swimmers have any momentum or "rocking" while on the blocks in the start position, or if the swimmer moves too early before the cue, referees will call it as a false start. The start depends on the ability to exert force against the block, and so intuitively, it seems as though swim starts should correlate with CMJ performance.

Some studies have found that there is no relationship between elite swimmer's starts and CMJ performance, which is typically quantified by CMJ jump height (De La Fuente, Garcia, & Arellano, 2003; Benjanuvatra, Edmunds, & Blanksby, 2007; Garcia-Ramos, et al., 2016). This suggests that swim starts are independent of vertical jumping techniques, and there must be other factors at play that contribute to the performance of starts (De La Fuente, Garcia, & Arellano, 2003; Benjanuvatra, Edmunds, & Blanksby, 2007; Garcia-Ramos, et al., 2016). CMJs may be limited to measuring the force during vertical jumping, and not necessarily predicting success in a swim start (De La Fuente, Garcia, & Arellano, 2003). Additionally, swim start performance has been shown to have no correlation with CMJ takeoff velocity, peak force, or normalized peak force (Garcia-Ramos, et al., 2016). A factor in the difference between swim start performance and CMJ performance may be that CMJs begin in the upright position, and momentum is gained during the squat downwards while flexing at the knees and hips, allowing elastic energy to be stored. However, in a swim start, individuals begin with knees and hips already flexed, and at the start cue, there is minimal countermovement before they propel themselves off the block. This means that there is less momentum during the jump—as if the swimmer was performing a counter movement jump but had to pause at the bottom of the squat before jumping.

In one study looking at male international sprint swimmers, start time was found to be related to jump height (West, Owen, Cunningham, Cook, & Kilduff, 2011). In the same study, peak vertical force and peak horizontal force during the swim start was also correlated to CMJ height (West, Owen, Cunningham, Cook, & Kilduff, 2011). This finding emphasizes that lower body strength is essential to start time in swimmers, but recognizes that in correlation studies, it's difficult to determine if relationships are causative or just markers of one another. The disagreement of this study with other literature looking at relationships between CMJs and start time may be due to the population used (male international sprint swimmer), or the small sample size of eleven swimmers.

While CMJs and swim starts have been shown to be separate motor tasks and vertical performance may not correlate with swim starts, lower limb strength and power are still important for a successful start (Benjanuvatra, Edmunds, & Blanksby, 2007; West, Owen, Cunningham, Cook, & Kilduff, 2011). Elite swimmers performed better in starts than recreational swimmers, due to greater horizontal impulse (Benjanuvatra, Edmunds, & Blanksby, 2007), and it is reasonable to assume that higher-level swimmers would have greater lower limb strength than lower-level swimmers. In noncompetitive female swimmers, jumping ability has been shown to correlate to flight distance using multiple different starting techniques (Breed & Young, 2003). When these noncompetitive swimmers underwent resistance training, their CMJ vertical height performance improved, but no improvement in flight distance was found—indicating that although athletes can improve their CMJs, this may not transfer directly to swim starts (Breed & Young, 2003). This may be due to a more complicated pattern of net muscle joint moment required during swim starts compared to a CMJ (Breed & Young, 2003; Lee, Huang, Wang, & Lin, 2001).

The swim group spent significantly longer than the other three sports groups in the eccentric phase, the concentric phase, and the unloading phase, as well as took longer for the total time to takeoff. Swimmers also spent the most time in the loading phase and final phase, but only significantly longer than the soccer group. Although the swimming group spent the most time in all the jump phases, and took the longest to takeoff during the CMJ as a whole, this may actually be desirable or expected in the sport. Assuming that it takes longer to “jump” or apply explosive muscle force through water than air, applying a force over a longer period of time in a CMJ might be expected for swimmers. Swimmers are accustomed to water resistance when completing flip turns, or when lower limb muscles are used during swim strokes. If the stretch-shortening cycle is what athletes rely on to perform quick, explosive movements (such as jumps) then it makes sense that swimmers may not have this as developed as sports that are constantly jumping, such as volleyball.

The swimming group spent the least amount of time in the air during CMJs, but this was only significant with volleyball. The swimming group’s jump height calculated with both the IMM and TIA method was significantly lower than that of volleyball, and the swimmers’ jump height calculated with the IMM was additionally lower than that of soccer. Since jump height and time in air are likely related (or in the case of the jump height calculated with the TIA method, directly calculated using the time in air), it is logical that since the swimming group spent the least amount of time in the air, they also had the lowest jumps.

The swim group had no significant differences with the other sport groups for eccentric impulse or normalized eccentric impulse. For concentric impulse, however, the swimming group was significantly lower than rugby and volleyball. When normalized to body mass, there was no longer a significant difference between rugby and swim, but instead, the swim group had a lower normalized concentric impulse when compared with soccer and volleyball. Considering that the swimming group spent the longest in all the phase times, yet had low concentric impulse, this suggests that the average concentric force applied for swimmers was lower than other sports. The concentric phase is the phase in which jumpers are propelling themselves upwards, so a low concentric force and impulse would mean poorer performance in propulsion upward to achieve a greater height. This may coincide with the lower jump height that the swimming group exhibited compared to the other groups. Additionally, the swimming group’s takeoff velocity was significantly lower than the soccer and volleyball groups, and the lower than rugby, but not significantly so. This also agrees with the lower peak force, concentric impulse, and jump height found in the swimmers.

The swimming group had no significant differences with the other sport groups for their displacement during the concentric phase. In the eccentric phase, the swimmers had a larger displacement (in the negative direction) compared to the soccer group, which had the least eccentric displacement. Since the eccentric phase is the phase in which participants are "braking," this indicates that the soccer group did not brake into as deep of a squatted position compared to the swim group. This may be more of a reflection on the soccer group, as soccer was significantly different from all of the sport groups, but none of rugby, swim or volleyball were significant within each other.

The swimming group also significantly had the lowest RSImod calculated with both the TIA and IMM methods. Since RSImod is calculated directly as jump height over time to takeoff, and the swimming group had the lowest jump height but the longest time to takeoff, this meant their RSImod was also the lowest out of the sport groups. Since RSImod is typically used as a way to quantify how explosive a plyometric movement is, this suggests that swimmers' CMJs are not very explosive compared to the other sports. The majority of swimming movements in competition being done horizontally may explain this poor RSImod, which was calculated using a vertical jump. Future studies could perhaps place swimmers in an environment more similar to their own (e.g. performing a flip turn in a swimming pool) and measure the RSImod based on the contact time with the horizontal wall and the horizontal distance the swimmer was able to achieve with their force applied during the turn.

To maximize propulsion and minimize resistive drag in swimming, both legs should be contributing optimally during swim strokes (Sanders, Thow, & Fairweather, 2011). However, physical asymmetries can cause one side of the body to do more work than the other, as these asymmetries can affect both technique as well as the ideal body posture to reduce resistive drag (Sanders, Thow, & Fairweather, 2011). The swim group's asymmetry index of 14.7 in the eccentric phase and 28.5 in the concentric phase suggests that one leg is applying a larger impulse than the other, especially during propulsion. There was no difference in asymmetry index during the eccentric phase between any of the sport groups, however, the swim group had the largest concentric phase asymmetry index (but only significant with soccer). Using the asymmetry index to help quantify asymmetry, this suggests that swimmers have less functional lower limb muscle symmetry than soccer players. This may be surprising, as swim strokes appear to equally use both sides of the body (either at the same time, or continuously alternating), while soccer players tend to prefer a dominant lower limb. Examples of one asymmetrical movement in swimming would be the track start, which is commonly used by swimmers. However, asymmetries may also be a result of training, not just competition. A possible explanation for the difference in asymmetry between swim and soccer may be in the training regime used by the two athlete groups. The training used by the soccer group may have compensated for any preference of one leg use over the other, as lower body is the main area of focus in soccer strength and conditioning. It is a possibility that the swim group didn't undergo training that strengthened both legs equally to account for any natural asymmetries, and so there was still a preferential leg during swim kicking. Additionally, since swimming is a full-body workout while soccer does not involve the upper body as heavily, there may be a smaller percentage of training time spent just on lower limbs in swimming.

It is important to keep in mind that the apparent "poorer" jump performance exhibited by the swimming group (based off lower peak force, concentric impulse and jump height) is not a reflection swimmers as poor athletes, but rather that counter movement jumps may not be the best performance measurement tool for swimmers. Ideally, CMJs should be used in conjunction with other performance testing methods, and performance testing expectations and goals should be modified to fit each athlete's sport.

The small group size in swimming of only ten participants may also account for some of the results in this study. Future work should include a larger sample size so that stronger comparisons can be made confidently between sport groups.

2.3.4 Volleyball

Volleyball players in this study had the largest body mass, but this was only statistically significant with the soccer group. Due to the nature of volleyball, it is likely that this larger body mass is from the players being a larger height compared to the other sports, as height is a great advantage in volleyball (Forthomme, Croisier, Ciccarone, Crieland, & Cloes, 2005). The only volleyball position that may not have a strong advantage with height is the libero--a specialized defense-only player that can replace back row players without counting as a substitution. Since the libero is used for defensive and passing skills only, they rarely play at the net, and so being tall may not necessarily help them. Volleyball player size and mass also varies with position: middle players are typically tallest and heaviest, then opposites and receivers, and liberos are the shortest and lightest of all (Sattler, Sekulic, Hadzic, Uljevic, &

Dervisevic, 2012). Volleyball players had the highest absolute peak force, but their peak force was only significantly greater than the swim absolute peak force. When peak force was normalized to body mass, the volleyball group's normalized peak force was significantly smaller compared to that of the soccer group. Due to their large body mass, the volleyball group lost their advantage in peak force once it was normalized to their body mass.

The volleyball group's eccentric phase time, concentric phase time, and unloading phase time were all significantly smaller than that of the swim group. Additionally, the volleyball concentric phase length was significantly greater than that of soccer, which is an interesting finding. Intuitively, in a sport that uses vertical jumping constantly throughout gameplay, such as in volleyball, seems as though they should have the quickest, highest, most explosive jumps. The longer time spent in the concentric phase by the volleyball players compared to soccer players suggests that they are taking a long time to propel themselves upwards during CMJs. Spending longer in the concentric phase compared to soccer could mean that volleyball players are potentially using this extra propulsion time compared to soccer in order to jump higher, but trading it off for a slightly slower jump time. It's possible that in this study, the soccer athletes' CMJs were not as deep and forceful as volleyball's, but were quicker.

Volleyball's time phases being significantly shorter than the swim group, but not significantly different from the other groups (except for the concentric phase time with soccer) may just be a reflection on the swim group performing the CMJs at a slower rate than all the athlete groups, and not necessarily that the volleyball players are jumping faster than an average athlete. However, it indicates that volleyball players are able to perform quicker jumps than swimmers, perhaps because these are common in volleyball gameplay but not in swim competition. For total time to takeoff, volleyball took significantly longer than soccer, but significantly less time than swim.

Volleyball players had no significant difference with any of the other sport groups for loading phase time or final phase time. Volleyball players had the highest time in air out of all the groups, but they were only significantly larger that of the swim group. Additionally, the volleyball group had the highest takeoff velocity, but again, was only significant with the swimmers. The slightly longer time that it took for volleyball players to takeoff compared to soccer players could be preferable for volleyball players, since they were able to jump at a higher takeoff velocity.

The volleyball group had the largest impulse for all of absolute and normalized eccentric and concentric phases. However, the eccentric phase impulse was only significantly larger compared to soccer, and the concentric phase impulse significantly larger compared to soccer and swim. For normalized impulse, volleyball was significantly larger than soccer in the eccentric phase, and swim in the concentric phase.

Volleyball is a sport that very obviously benefits from a strong jumping performance, as jumps are a regular occurrence in training and competition. On defense, blocks are more effective when player can jump to a height that the opponent cannot hit over (Sattler, Sekulic, Hadzic, Uljevic, & Dervisevic, 2012). On offense, spikes are most effective when the ball is hit at higher velocity, and previous studies have shown a correlation between height of impact of a player performing a spike jump and the velocity at which the ball travels after impact (Forthomme, Croisier, Ciccarone, Crieland, & Cloes, 2005; Wagner, Tilp, von Duvillard, & Mueller, 2009; Sattler, Sekulic, Hadzic, Uljevic, & Dervisevic, 2012). This higher jump allows players to make contact with the ball overhead at a higher point, to strike down the ball on the opponent's side of the court more effectively (Forthomme, Croisier, Ciccarone, Crieland, & Cloes, 2005). Spike jumps are very similar to CMJs: after an approach of 2-3 steps, the player performs a vertical jump, and spikes the ball downward at a high speed. The technique of the approach before a spike jump has been shown to be key to reaching the maximal height (Wagner, Tilp, von Duvillard, & Mueller, 2009), but CMJs don't allow for this technique to be incorporated. This might mean that volleyball players have a higher performance in their sport-specific jump rather than the CMJs used in this study.

The greatest heights in volleyball players was achieved in attack jumps (e.g. spikes), then defensive block jumps, then CMJs (Sattler, Sekulic, Hadzic, Uljevic, & Dervisevic, 2012). This is likely due to the nature of the three jumps: in spike jumps, the player takes a few steps in their approach, and uses arm momentum and the momentum from this approach to propel themselves high into the air to hit the ball. Defensive block shots are often done by the front row players, and don't usually have an approach, rather just a vertical jump upwards, using the arms to gain momentum and then extending them high above the head to block the ball. CMJs consist of a vertical jump similar to the other two types of jumps, but is more controlled and does not allow for arms to be used for momentum upwards, meaning that a lower jump height would be expected.

The volleyball jump height, with both the TIA and IMM calculation method, was the largest compared to the other sports but only significantly larger than swim. It’s not surprising that the volleyball jump height was the largest out of the groups compared, due to the nature of the sport. Additionally, since the volleyball group had the largest impulse (both absolute and normalized), this was transferred successfully into the highest takeoff velocity and the highest jump height. Based on the performance criteria of jump height, the volleyball players were the best jumpers in this study.

Higher division volleyball players were able to achieve a greater jump capacity (in terms of jump height and jump time) and thus a higher spike velocity compared to lower division players (Forthomme, Croisier, Ciccarone, Crieland, & Cloes, 2005). Players of teams that perform better have also been shown to have higher vertical jump values (Ziv & Lidor, 2010). Jump height during a CMJ has also shown to vary depending on the volleyball position played. In CMJs, receivers have been shown to have higher average jumps than setter. In attack jumps, receivers jumped higher than liberos, and liberos performed better than setters in block jumps (Sattler, Sekulic, Hadzic, Uljevic, & Dervisevic, 2012).

The displacement in the eccentric phase was significantly greater (in the negative direction) in the volleyball group compared to the soccer group, meaning that the volleyball players were squatting down deeper during the CMJs compared to soccer. In one study of volleyball and basketball players, the relationship between the best jump performance (by jump height) and the depth of the CMJ squat (the jumper’s preferred depth, a depth of more than 90 degrees of knee flexion, or a depth of less than 90 degrees of knee flexion) was explored (Gheller, et al., 2015). The best CMJ jump height was achieved in the higher squat depth (less than 90 degrees knee joint flexion) or at the preferred athlete-chosen depth (Gheller, et al., 2015). However, a higher peak force was observed when the athletes performed the CMJ with the degree of flexion greater than 90 (a lesser squat depth) (Gheller, et al., 2015). The difference in jump height may explained by a larger range of countermovement being generated in deeper squat depths compared to smaller squat depths, or by an increased time to apply force (Gheller, et al., 2015). The study determined that performing CMJs with knees in a deeper flexed position during the squat portion is a better strategy (Gheller, et al., 2015). The stretch-shortening cycle is very important in volleyball, and the capacity to tolerate high stretch loads (as in a deeper squatted position during a CMJ) is important to the jumps performed in volleyball (Sheppard, et al., 2008). The tradeoff in a deeper squat during a CMJ is that it may take longer to perform than a shallower squat, which might explain why volleyball took longer to jump than soccer. However, this was effective for volleyball players, since they were able to jump higher than the other sports. There was no difference between volleyball and the other sport groups for concentric displacement during the CMJs.

The RSImod calculated with the TIA jump height and IMM jump height was greater for the volleyball group compared to the swim group, but the volleyball group didn't have significant differences with the other two groups. Since the volleyball had significantly higher jump height but lower time to takeoff than swim, this again shows that volleyball had a more explosive jump than swim while still achieving greater jump height.

There was no difference in eccentric or concentric asymmetry index between the volleyball group and the other sport groups in this study. However, the volleyball group also did not have an asymmetry index equal to zero (eccentric asymmetry index of 13.4, and a concentric asymmetry index of 20.1), indicating that jumpers are applying impulse differently between legs during CMJs. Although spike jumps are performed with both legs, one study found that there were significant differences between the distance of the left and right foot with respect to the volleyball player's center of mass, suggesting that spike jumps are not symmetric movements (Wagner, Tilp, von Duvillard, & Mueller, 2009). This is likely due to the approach before a spike jump, or the adjustments that have to be made in the spike jump approach to achieve proper alignment to hit the ball. This, along with the preference of one lower limb side over the other, may account for the asymmetry index in the volleyball group.

Plyometric training has also been shown to improve volleyball players' vertical jump height by 5-10%, even if players are elite and already have high jumping performance (Ziv & Lidor, 2010). Plyometric training emphasizes explosive leg power and general jumping ability, which is key to improving vertical jumps (Ziv & Lidor, 2010; Wagner, Tilp, von Duvillard, & Mueller, 2009). Additionally, CMJs are an inexpensive, easy and quick way to help quantify improvements or decline in volleyball jump performance (Ziv & Lidor, 2010).

A limitation to the volleyball group used in this study is that it was a relatively small sample size (12 players), and all the players were male. While there was no interaction effect found between gender and sport type using ANOVA, this still may be a factor in the outcomes of this study. Future studies should include both genders in all the sport groups.

2.3.5 Gender

Since men typically have a larger skeletal mass and more muscle mass compared to women, it is logical that the male group had a significantly greater mass than the female group. Due to the larger muscle mass in men, it makes sense that the male group was able to exert a larger peak force during CMJs compared to the female group. While there were statistically significant differences found between male and female athletes for peak force, once the peak force was normalized to body mass, there was no significant difference between male and female. Dividing the peak force by body mass accounts for the wide range of body masses in participants and the advantage (in terms of force exerted) that participants with a larger mass would have. The lack of a significant difference between male and females for normalized peak force supports this: if males have a greater muscle mass and thus can produce a larger peak force, if we normalize to this larger mass, they should have similar normalized peak force compared to females. This agrees with previous studies looking at the gender differences in CMJs (McMahon, Rej, & Comfort, 2017).

For 71.7% of females in the study, their dominant leg was the one that exerted the larger peak force. However, for males, only 58.3% did. For all of the groups, the majority of participants exhibited the larger peak force with their dominant leg, with male swim being the only exception (only 40% did).

Males spent a significantly longer amount of time in the eccentric phase, concentric phase, unloading phase, and loading phase compared to females. This meant that males also had a significantly longer total time to takeoff than females. However, there was no significant difference in final phase time between genders. Males also spent a significantly longer time in the air than females, and had a larger jump height than females using both the TIA and IMM method.

Male athletes had significantly larger normalized and absolute eccentric and concentric impulses compared to the female athletes. Males also had a significantly greater takeoff velocity than females. Jump height should correlate with net concentric impulse (since a large concentric impulse should lead to a greater takeoff velocity based on the impulse-momentum relationship, and greater takeoff velocity leads to a greater jump height) (McMahon, Rej, & Comfort, 2017) which agrees with the findings in the current study. Males had a larger concentric impulse, a greater takeoff velocity, and a higher jump height.

There were no significant differences between males and females for asymmetry index in the eccentric or concentric phases. Females had an asymmetry index of 13.6 in the eccentric phase, and 22.2 in the concentric phase. Males had an asymmetry index of 14.1 in the eccentric phase, and 19.6 in the concentric phase.

Males had a larger RSImod (calculated with both TIA and IMM jump height) than females did. A previous study found that men had higher RSImod values, but likely only due to men jumping higher, not necessarily over a shorter jump time (McMahon, Rej, & Comfort, 2017). This agrees with the findings in the current study, since males did jump higher, but also had a longer time to takeoff compared to females. A high RSImod would ideally come from both a high force and a short time it's applied over, since this indicates a high reactive strength capacity and the ability to perform explosive movements.

When comparing male and female performing CMJs, McMahon et al. (2017) found that men had greater jump height, RSImod, normalized peak concentric power, eccentric and concentric displacement, velocity, and normalized impulse than women (McMahon, Rej, & Comfort, 2017). Many studies have confirmed that males jump higher than females (McMahon, Rej & Comfort, 2017; Laffaye, Wagner, & Tombleson, 2014). Men jumped 24% higher than women in the study by McMahon et al., likely due to a larger concentric impulse found in the men than the women in this study (McMahon, Rej, & Comfort, 2017). This also reinforces the fact that normalized peak force may not be a good way to quantify jump performance, as there was no significant difference between males and females in the current study for normalized peak force, but it is clear that the two genders are jumping differently in terms of jump height, jump time, impulse, etc.

The males had a significantly greater eccentric displacement (in the direction of a deeper squat) than the female, and a significantly greater concentric displacement. Since the female athletes had lesser displacement in both the eccentric and concentric phases, this may be because the females were using a "stiffer" lower limb strategy compared to the males. One study found that leg stiffness was greater in women than in men during CMJs, which indicated that the women were using a stiffer lower limb strategy during jumps to increase peak force (McMahon, Rej, & Comfort, 2017). The stiffer leg strategy results in short movement times (not as much movement), but larger forces, which creates force-time curves that have tall and thin impulse (McMahon, Rej, & Comfort, 2017). This is actually preferable in sports, as this indicates a more explosive and would be useful for something like sprinting (McMahon, Rej,

& Comfort, 2017). The current study agrees with this stiff leg finding, as females had shorter jumps, more shallow jumps, but the normalized peak force was not significantly different between males and females.

In a study that strength-matched female basketball players to male basketball players (based on their back squat strength), the CMJ jump height was still significantly higher in males (Rice, et al., 2017). The absolute force was found to be greater in males than females of the same strength, but when normalized to mass, there was no significant difference (Rice, et al., 2017) (similar to what was found in this current study). Males also had significantly greater impulse during the eccentric phase, indicating that the eccentric phase impulse is influenced by sex differences that are independent of matched strength (Rice, et al., 2017). There were no differences in rate of force development between the sexes. A significant difference in absolute concentric impulse did exist, with males having larger impulses than females in the study, but when normalized to mass, there was no significant difference (Rice, et al., 2017). This indicates that there are differences other than just strength between males and females for CMJ performance.

2.3.6 Mass and Force

Depending on the sport, a larger or smaller mass may be more beneficial. For example, in rugby, some positions greatly benefit from a larger muscle and body mass since full body contact is permitted in the sport. In contrast, soccer has much less intentional and forceful body contact, but involves a high amount of cardiovascular fitness, and so smaller body masses may be more typical or beneficial. The results reflect this, with rugby players having a significantly larger mass than soccer players. However, volleyball players were also found to have a significantly higher mass compared to soccer players in this study. This may be due to the tendency for volleyball players to be quite tall, whereas in soccer, being tall is not as much of an advantage in the sport. Additionally, volleyball involves a lower level of cardiovascular fitness compared to soccer, and so a higher mass may not be as detrimental to speed and endurance in the sport.

It is interesting, however, that the swim group had a significantly lower peak force compared to the other three sports. While the CMJ is being used to help measure lower body muscle function, it might also be important to consider whether or not the movement during a CMJ is familiar or foreign to these athletes in their sport. For example, volleyball and soccer frequently involve movements and muscle groups very similar to those used during a CMJ.

However, swimmers don’t often use jumping motions in their sport—perhaps the closest thing to a jump in swimming would be leaving the starting blocks (more of a horizontal projection into the water) or when completing a flip turn.

Soccer exhibited the highest peak force normalized to mass, and was significantly greater than the swim, rugby and volleyball groups. Normalized peak force can be used as a simple way of quantifying how well a participant is jumping, or how efficiently their body mass can produce a force. Soccer participants in this study were able to produce the largest amount of force per kilogram of their body mass out of all of the sport groups. A high normalized peak force is likely to correlate with a higher percentage of muscle mass, as this would be necessary to produce a higher force but still keep mass low. These results would suggest that soccer players in this study have a higher percentage of lean body mass compared to the other three groups of athletes.

The peak force difference between legs was calculated as a simple comparison to see if athletes were exerting similar peak force between their two legs during the CMJs. The lack of any statistically significant difference in the peak force difference between legs for any of the groups compared could mean that it is a poor metric to quantify asymmetry, or simply that none of the sport or gender groups compared exhibited any difference in their leg usage asymmetries.

Intuitively, one would expect that a sport such as swimming would require fairly symmetrical lower limb muscle usage, due to the continuous bilateral movements or continuously alternating unilateral movements involved in swimming. Similarly, one would expect that a sport such as soccer would exhibit lower limb asymmetry. Although high-level soccer athletes need to be proficient in technical skills with both feet, they often still prefer one over the other and will use this leg more often.

Although there were no differences between groups for the peak force difference between legs, these values were also not equal to zero. Theoretically, in a perfectly symmetrical jumper, their left and right leg would exhibit identical peak forces during a CMJ and the peak force difference between legs would be equal to zero. However, we know that perfect muscle symmetry does not exist in humans, both physically and in function. On average, participants in this study exhibited a 6.77±4.75% difference between their two legs. The largest percent difference was found to be 20.2% (in a female soccer participant) while the smallest was found to be 0.63% (in a male soccer participant). It is difficult to say at what point a percent difference in force exerted by each leg would make a tangible difference in jump performance, but a difference of over 20% is a substantial amount.

To maximize overall muscle usage during sport, both sides of the body should be contributing optimally. Sanders et al. (2011) suggested that causes of asymmetries in swimmers include laterality, genetic/early environmental factors, development due to lateral preference and motor dominance, disease factors, effects of injuries on asymmetry and muscle imbalance, overtraining and fatigue, and technical habits (Sanders, Thow, & Fairweather, 2011).

To help understand if laterality and individuals' preferred leg dominance was a main cause of asymmetries while jumping, the leg exerting the greater force while performing the CMJ was compared to the leg the individual declared as their dominant one. Overall, 67.1% of participants exerted the larger force with their dominant leg. This would suggest that for 32.9% of participants, it is not the leg dominance that causes the asymmetry. Although no participants had current or recent lower limb injuries, for high level athletes, it is likely that they have had injuries at some point in their sport careers, which could impact the symmetry of their muscle usage

While peak force is a quick way to help quantify how a participant is jumping, since it is a measurement taken at only one point in time during the CMJ, it may leave out other valuable characteristics. As such, other ways to help quantify jump performance were explored. Using peak force by itself may be a flawed method to quantify jump performance. For example, subjects may exhibit a low jump height despite a high peak force (Dowling & Vamos, 1993). It may be more valuable to quantify the patterns of force being applied, rather than relying just on the peak force (Dowling & Vamos, 1993; Kirby, McBride, Haines, & Dayne, 2011).

2.3.7 Jump Phases

Plyometric exercises (including CMJs) have three phases: the eccentric phase, the amortization phase, and the concentric phase. In the eccentric phase (also known as the loading, deceleration, yielding, or countermovement phase), the muscle spindle activity is increased by pre-stretching muscles and storing potential energy in elastic muscle components (Clark & Lucett, 2010). The amortization phase (also known as the transition phase), is the time between the eccentric contraction and the initiation of the concentric contraction; this is when the muscle switches from overcoming force (generating countermovement) to exerting force in the desired direction (Clark & Lucett, 2010). A longer amortization phase means less neuromuscular efficiency (elastic potential is lost), while a short amortization phase is a powerful, explosive response (Clark & Lucett, 2010). The concentric phase (also known as the unloading phase) occurs immediately after the amortization and uses concentric contraction to produce motion in the desired direction (Clark & Lucett, 2010). Some studies have defined or calculated the phases in the countermovement jumps slightly differently, and so they are not to be confused with one another.

The start of the CMJ unloading phase (also known as the unweighting phase) is indicated by the onset of movement, and continues until the instant at which the ground reaction force measured returns to body weight (McMahon, Suchomel, Lake, & Comfort, 2018). This point at which the force equals the body weight also coincides with the instant at which the peak negative center of mass velocity is achieved (McMahon, Suchomel, Lake, & Comfort, 2018). The unloading phase of the jump is the portion at which the measured normal force is below body weight (McMahon, Suchomel, Lake, & Comfort, 2018). The unloading phase is followed immediately by the eccentric phase.

The eccentric phase begins where the unloading phase ends. Some studies refer to this phase as the loading phase or countermovement phase (Chmielewski, Myer, Kauffman, & Tillman, 2006), but for the purposes of this study, the two are different. During this phase, the athlete is decelerating ("braking") their center of mass (McMahon, Suchomel, Lake, & Comfort, 2018). The eccentric phase starts at the instant of peak negative center of mass velocity and ends where the center of mass velocity returns back to zero (McMahon, Suchomel, Lake, & Comfort, 2018). The eccentric phase is when the muscle-tendon units begin to perform work (work in the opposite direction of the vertical jump) (Chmielewski, Myer, Kauffman, & Tillman, 2006). This phase may also be referred to as the braking phase or the stretching phase (McMahon, Suchomel, Lake, & Comfort, 2018). The eccentric phase is followed immediately by the concentric phase.

The concentric phase, also known as the propulsion phase, begins where the eccentric phase takes off—at the point at which COM velocity is equal to zero—and continues until takeoff (McMahon, Suchomel, Lake, & Comfort, 2018). The performance in the eccentric phase should, in theory, enhance performance in the concentric phase of a jump (Bartlett, 2007), since the eccentric phase is when the jumper is storing energy.

In plyometric exercise, the amortization phase (also known as the transition phase or coupling phase) represents the segment of the exercise between the eccentric and the concentric phases. Alternatively, some literature terms it as the segment between the loading phase and the unloading phase (Chmielewski, Myer, Kauffman, & Tillman, 2006). In this current study, since countermovement jumps are performed by squatting down and immediately propelling the body upwards, the amortization phase should theoretically be zero. An example of when an amortization phase would exist, would be in a non-counter movement jump: when an athlete squats down (as in a CMJ), but hold the squat for a moment to eliminate any momentum. The portion of the non-counter movement where the jumper is holding still in the squat would be the amortization phase.

During the analysis of this study, it was noted that t_3 and t_{max} closely corresponded to one another but were usually not exactly the same. This meant that the eccentric phase length (measured from t_2 to t_3) and the loading phase length (measured from t_2 to t_{max}) also very closely corresponded. Since the eccentric phase is often referred to as the phase in which the muscles are being "loaded," this begs the question if these two phases should theoretically be the same thing. In literature, it is possible that researchers who used the two different phase separation schemes were attempting to quantify the same thing but used different naming systems and criteria. The average length of the eccentric phase was 0.17±0.06 s, and the average length of the loading phase was 0.22±0.12 s. Alternatively, since t_3 is defined as the point at zero center of mass velocity and t_{max} is defined as the time at peak force, the difference between the two time points could perhaps be used as another time phase to help evaluate jumps. The difference in these two could also arise in the fact that when approximating velocity of the jumper, we are using the jumper's center of mass in calculations. However, the human body is largely dynamic and the extension and flexion of joints during jumps may cause error in assuming it is just a point mass.

Using the loading, unloading and takeoff scheme of defining jump phases, one additional significant difference was found between the sport groups. If this means that these phase definitions are more sensitive than the eccentric and concentric scheme, it might be more beneficial to use them over eccentric and concentric. However, the eccentric and concentric scheme is more commonly found in current literature, so it would be easier to compare new study outcomes with those previously done.

The time spent in the final phase (time from the instant of peak force to the time of takeoff) was one of the few variables for which there was no difference between the male and female group. Additionally, the only significant differences between the sport groups were between soccer and swim, with the swim group having a larger final phase time. These are the same sport groups that showed a significant difference for the loading phase time.

While participants weren't explicitly instructed to "jump quickly" or "jump slowly", the time it took them from the initiation of the jump to takeoff of the jump could provide valuable information. The soccer group took the least time to perform the jump, while the swim group took the longest time. Swim taking a significantly longer time to perform the CMJ compared to the other groups again may indicate that this is more of a foreign movement to them, or at least they are not used to performing such movements outside of a water environment. Some studies have found that phase duration did not significantly differ based on jumping ability, which may suggest that phase timing does not provide valuable insight into jump performance (Sole, Mizuguchi, Sato, Moir, & Stone, 2017). However, due to the nature of sport and the benefit that a quick yet high jump has, CMJ phase timing may indeed be useful to evaluate performance.

2.3.8 Force-Time Curve Shape

Soccer was the only sport that exhibited Class 1 CMJs, which were seen in 42.9% of participants in both the male and female groups. The exception was the only non-soccer athlete exhibiting a Class 1 CMJ, one female rugby athlete, which could potentially be an outlier. The female rugby athlete that was grouped into Class 1 jumps had two Class 2 jumps and three Class 1 jumps, and so if this participant completed more trials, they could potentially be grouped into Class 2.

The significance of the different classes observed is not yet well understood. It would appear that a Class 1 jump is more efficient, since the force is being applied in a smoother fashion compared to Class 2. Class 1 jumpers appear to also complete the entire jump quicker—indicating a more explosive jump than Class 2 jumpers. However, it was also noted that in the Class 1 jumpers, their squat depth was an average of 0.17±0.03 m, while in the Class 2 jumpers, the average squat depth was 0.30±0.08 m. This indicates that Class 1 jumpers may be jumping quicker largely due to the fact that their squat depth was much lower than that of the Class 2 jumpers.

While most participants were clearly exhibiting the same class they were assigned, there are cases where jumps were labeled as one class using the Mathematica tool, but visually, closely resembled the other class. For example, Figure 28 shows the CMJ of one participant with the left and right leg separated into two force-time functions. The right leg was classified as a Class 1 using the Mathematica tool, but the left leg was classified as a Class 2. Visually, the two legs do not appear to differ greatly

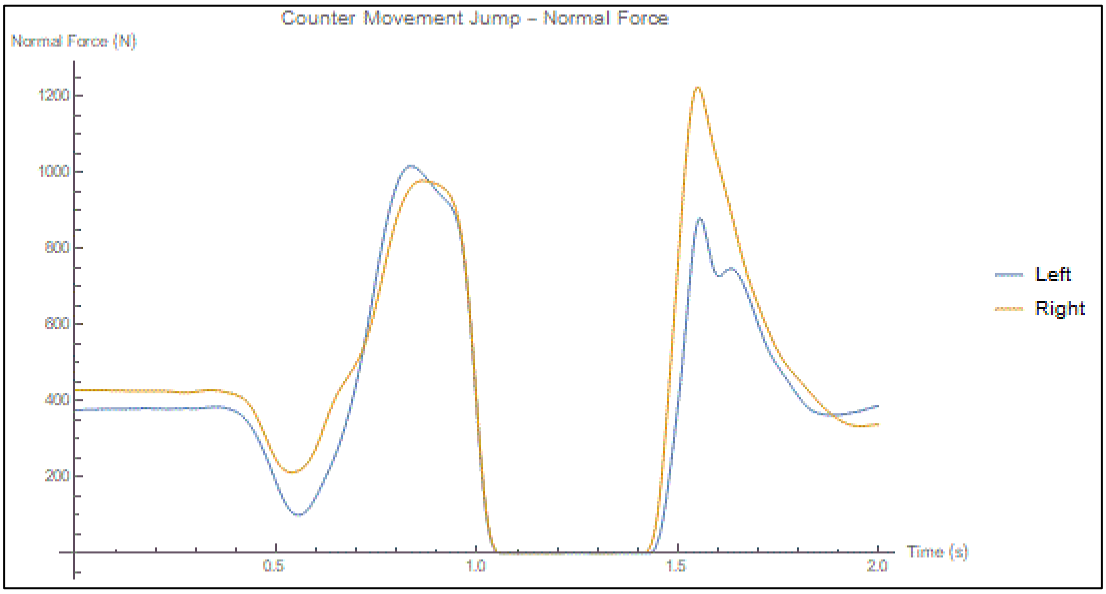


Figure 28: Example of a Participant with a Class 1 (in Yellow, Right Leg) and Class 2 (in Blue, Left Leg) Jump

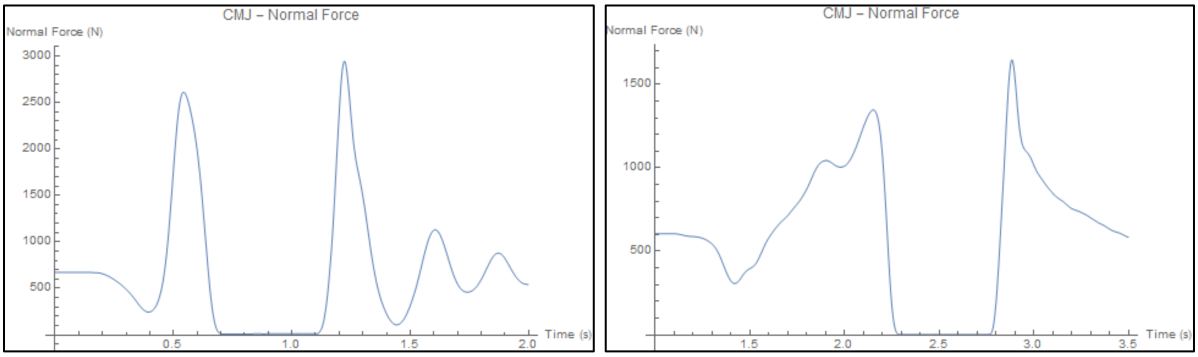
While many studies look at numerical values obtained from using CMJ force-time data (peak force, jump height, impulse, etc.), few studies look at the actual shape of the force-time curve. While quantifiable CMJ variables can tell us the magnitude of improvement with training is, they don’t provide information on what sort of adaptations the jumper has incorporated into their CMJs (Cormie, McBride, & McCaulley, 2009). One study, looking at rugby players performing CMJs on force plates, categorized the athletes’ force-time curves into unimodal (one peak), bimodal high-low (two peaks; one larger) or bimodal similar (two peaks; similar sizes) (McMaster, 2016). The study found that the unimodal peak group produced moderately larger mean concentric force, had a slightly larger eccentric rate of development, and much shorter contraction time comparted to both of the bimodal peak groups (McMaster, 2016). The bimodal peak group produced moderately larger net impulses, and slightly larger jump heights than the unimodal peak group (McMaster, 2016). In this current study, the soccer group was the only group that had a significant portion of athletes with a Class 1 jump, and some of the current study’s findings agree with what was previously found: soccer had the highest normalized peak force, spent the shortest amount of time in the eccentric and concentric phases, had the lowest eccentric impulse, and the Class 1 jumps tended to have a steeper upslope than the Class 2 jumps, which should relate to rate of force development. However, in the current study, soccer did not have the lowest jump height.

Lake and McMahon (2018) assessed the within-subject reliability of force-time curve shape during CMJ trials. They plotted time-normalized data (force as a function of percent of contact phase time) and categorized jumps into single peak or double peak using visual inspection. They were not able to find acceptable within-subject reliability for jump shape category over two, three, five or ten CMJ trials. However, participants were sorted in a subjective manner (using experimenter’s judgement), and used time-normalized plots to visually sort into groups. The method used in this study is objective and uses a mathematical approach to sort force-time curves into categories, thus it is repeatable. However, the within subject reliability of the participants used specifically in this study was not assessed, and so it must be acknowledged that the participant’s jump class could change over the course of more CMJ trials.

In male rugby players, there was no difference found between unimodal (one peak) and bimodal (two peak) groups in terms of jump height and RSImod (Kennedy & Drake, 2018). However, a bimodal force-time curve may indicate inefficiencies in the stretch-shortening cycle, and that the optimal pattern of performance is not being applied (Kennedy & Drake, 2018). While these definitions of unimodal and bimodal do not perfectly match the Class 1 and Class 2 definitions in this paper, the same principles still apply. A Class 2 jump indicates that the jumper has to apply a force over a much larger period of time, which slows the jump down and reduces explosiveness. Class 2 force-time curves are typically characterized by flatter, broader curves through the concentric phase, and are sometimes uneven or bumpy. Class 1 curves are taller, narrower, and smoother than the Class 2 peaks. This suggests that Class 1 jumpers apply force more smoothly and efficiently, and are able to transfer force during the countermovement in a more optimal manner. Thus, athlete should strive to exhibit a Class 1 jump, if possible, or training programs should include plyometric exercises to help athletes achieve quicker, efficient jumps.

2.3.9 Impulse

Kinetic vertical impulse was initially thought to be an advantageous measure of CMJ performance over single-time point measures such as maximum peak force, as the impulse incorporates a wide range of time points into it. Impulse has also been a metric used in literature. However, it was observed that two different athletes could have similar kinetic impulse, but exhibit very different jumps. For example, an athlete that performs a counter movement jump at a low peak force but over a long time interval may have a similar impulse as an athlete that performs the jump at a high peak force, but over a very short time interval (Figure 29). Figure 29a shows a male soccer player performing a CMJ with a peak force of 2609.6 N, a eccentric phase length of 0.065 s and an eccentric phase impulse of 59.105 N·s. Figure 29b is a male swimmer performing a CMJ with a peak force of 1347.1 N, an eccentric phase length of 0.280 s, and an eccentric phase impulse 58.239 N·s. The eccentric phase impulses between these two examples are within 1-2% of each other, yet the peak forces, eccentric phase length, and visual appearance of the jumps are vastly different.



(a) *(b)*

Figure 29: Example of two counter movement jumps with the same eccentric phase impulse, but vastly different jumps

The impulse from t_1 to t_2 is equal to (but opposite sign from) the impulse from t_2 to t_3 (the eccentric phase impulse). This means that the total impulse over the whole jump from t_1 to t_4 is equal to the impulse from t_3 to t_4 (the concentric phase impulse), since the other phases' impulses cancel each other out.

The change in the vertical velocity during a CMJ from the moment of standstill to the moment of takeoff depends on the force exerted by athletes. Takeoff impulse depends on the mean force and the time the force is applied over. A large vertical velocity is desired in CMJs, as quick movements are typically beneficial in sports. However, a compromise must be made in how much time is spent achieving the required impulse, as this time adds to the time from when an athlete recognizes they want to perform a motion (e.g. a jump) to when it is actually completed (Bartlett, 2007). Impulse is equal to the change in momentum, which is equal to the average force multiplied by the time the force acts. A large impulse is needed to produce a large change in momentum, and so a large change in momentum requires a large average force or a long time of action (Bartlett, 2007). Since explosive movements are typically used in sports, athletes use a large average force to achieve this change in momentum, rather than a long time of action (Bartlett, 2007).

One study found that the primary differences between high performing athletes of a variety of sports and low performing ones was a greater magnitude of the force-time curve and the impulse (area under the force-time curve) (Sole, Mizuguchi, Sato, Moir, & Stone, 2017). Based on these findings, training programs should aim to increase height and area of the force-time curve to result in a higher jump height. Based on the impulse momentum relationship, a larger concentric impulse will displace body mass at a higher velocity, and a higher velocity should mean a greater jump height (McMahon, Rej, & Comfort, 2017).

Eccentric phase impulse may also provide useful insight into athlete's CMJ performance. The size and shape of the initial rise in force in this phase has been found to be correlated to jump height, and might help to quantify an athlete's explosiveness (Sole, Mizuguchi, Sato, Moir, & Stone, 2017). Since the eccentric phase represents the stretch-shortening in the CMJ, if an athlete is fatigued, this may present itself during the eccentric phase (Sole, Mizuguchi, Sato, Moir, & Stone, 2017).

A high RSImod in rugby players has been shown to correlate with a taller but thinner active impulse on the force-time curve (McMahon, Jones, Suchomel, Lake, & Comfort, 2018). In studies looking at more senior rugby players, they were able to achieve a larger concentric impulse compared to less experienced players, which resulted in a greater velocity at takeoff (McMahon, Rej, & Comfort, 2017; McMahon, Murphy, Rej, & Comfort, 2017). The concentric phase is important to produce velocity to takeoff during the jump, which in turn, impacts the jump height produced.

2.3.10 Jump Height

Using the IMM method to calculate jump height, there was one additional significant difference between groups found compared to the TIA method: between swim and soccer.

Similar to the jump height calculated with the TIA method, males had a greater jump height than females, and volleyball had a greater jump height compared to swim.

In the study comparing different methods of calculating jump height by Moir (2008), men were found to produce higher jump heights than women in both the IMM method and the TIA method (Moir, 2008). Additionally, Moir found that the TIA method produced greater jump heights than the IMM method (Moir, 2008). The results of this current study agree with what Moir found: both that the jump height calculated for men was greater than that of women, and that the jump heights calculated with the TIA method are greater than those calculated with the IMM method. Moir predicts this height difference between these two methods is due to the necessary assumption for the TIA method: that the center of mass for the subjects is the same at landing as it is at takeoff. If jumpers "tuck" their legs at all during the CMJ, this assumption is violated (Moir, 2008). In the IMM method, however, the calculated is unaffected by this assumption. It is difficult to say which method is more accurate for calculating jump height, as both methods require assumptions and may have confounding factors. The TIA method is the simplest method—however it will often overestimate due to the discrepancy in jumper's center of mass at takeoff versus at landing (Moir, 2008; Linthorne, 2001). Both Moir (2008) and Linthorne (2001) agree that the IMM method gives the most accurate calculation of jump height.

2.3.11 RSImod

The ability to develop force is a requirement in sport, and so a means to measure this is needed. Reactive strength index modified (RSImod) has been validated in previous studies (McMahon, Jones, Suchomel, Lake, & Comfort, 2018; Ebben & Petushek, 2010) and is a useful way to measure explosive strength and power in any plyometric exercise that includes jumping, such as counter movement jumps (Ebben & Petushek, 2010).

Males had a higher RSImod than females when the RSImod was calculated with both the TIA jump height and the IMM jump height. Using both TIA and IMM jump height calculation methods, males jumped higher than females. However, males also had a larger time to takeoff compared to females. RSImod combines the jump height with the time to takeoff, and significant correlations between RSImod, jump height and time to takeoff have been shown to exist (Suchomel, Sole, Bailey, Grazer, & Beckham, 2015).

RSImod remains a useful method to help quantify performance CMJs and can help identify an athlete's readiness to return to sport after injury (Suchomel, Sole, Bailey, Grazer, & Beckham, 2015; Ebben & Petushek, 2010). Benefits of using RSImod as a means to quantify jump performance are that it can be used in any jumping exercise, it's highly reliable, and men and women both respond similarly to RSImod (Ebben & Petushek, 2010). The ratio of the time spent in air to the time spent in muscle contraction may be another useful parameter to help quantify and compare CMJ performance, but this ratio has been shown to have an almost perfect correlation with RSImod (McMahon, Lake, & Comfort, 2018).

Being aware of differences between RSImod in specific sports is important, so that training programs and monitoring can be tailored to that sport. In a study that included men and women's soccer, and women's volleyball, men's soccer was found to have the highest RSImod, followed by women's volleyball, then women's soccer (Suchomel, Sole, Bailey, Grazer, & Beckham, 2015). The highest jump height also correlated to this same ranking, with men's soccer jumping the highest, women's volleyball jumping the next highest, and women's soccer jumping the lowest out of these three groups. Additionally, men's soccer had a shorter time to takeoff compared to women's soccer (Suchomel, Sole, Bailey, Grazer, & Beckham, 2015).

In elite male rugby players, a high RSImod has been shown to correlated with a taller but thinner active impulse, represented by the area under the force-time curve (McMahon, Jones, Suchomel, Lake, & Comfort, 2018). Additionally, it can be assumed that a higher RSImod is attributed to an increase in kinetic and kinematic parameters (McMahon, Jones, Suchomel, Lake, & Comfort, 2018), which is beneficial since RSImod is relatively simple to measure and calculate.

2.3.12 Asymmetry

The participants weren't explicitly told to jump using their legs equally in this study or to modify their jump in any way, such that the natural asymmetries during the jump could be evaluated. However, on the informed consent form completed by participants, the name of the study made it clear that asymmetry was a key parameter of interest. Being aware that this was an important part of the study, participants may have consciously or unconsciously changed their jumping behavior. For example, a study evaluating the asymmetry during the sit to stand movement in healthy participants proposed that asymmetries may exist, but are masked by the researcher asking the participant to perform the task symmetrically (Schofield, et al., 2013). If the jumpers were under the impression that jumping symmetrically was more desirable to the researcher in this study, they may have modified their jump to use legs more equally. This may have been a factor in the asymmetry parameters explored, including the lack of a significant difference between groups for peak force differences between legs.

It's unclear if lower limb asymmetry puts athletes at a higher risk of injury or re-injury, but having asymmetries between sides of the body may mean that muscles are not performing to their full potential. After injury, lower limbs may never return to a symmetrical state, despite return to sport. In a study of ski racers with ACL injuries, the skiers still have significant neuromuscular deficits despite returning to their regular competition (Jordan, Aagaard, & Herzog, 2017). Asymmetries in these ski racers were quantified in vertical jump takeoff and landing, as well as between-limb hamstring and quadriceps strength ratios (Jordan, Aagaard, & Herzog, 2017). ACL re-injury has also been shown to be common in alpine skiers (Jordan, Aagaard, & Herzog, 2015), and those with ACL reconstruction had increased asymmetry in terms of muscle mass, and asymmetry index in the concentric phase (Jordan, Aagaard, & Herzog, 2015).

The asymmetry index of all the participants in this study was 13.8 in the eccentric phase, and 21.3 in the concentric phase. Additionally, the percent difference between limbs in terms of peak force was 6.8%, indicating that participants in this study were not using both legs equally to jump. It is difficult to determine what the cause of this asymmetry is, and if it is something that can be improved upon. It is also difficult to attribute a lower jump performance solely on a lower limb asymmetry.

One study performed computer simulations on two 3D human lower limb neuromusculoskeletal models—one with asymmetry (10% difference), and one without (Yoshioka, Nagano, Hay, & Fukashiro, 2010). The study found that the models performing CMJs were generated successfully, and found that bilateral asymmetry by itself did not have a significant effect on jumping performance (in terms of jump height) (Yoshioka, Nagano, Hay, & Fukashiro, 2010). In the asymmetry model, the strong leg compensated for the weak leg by lateral movement of the body's center of mass to distribute the body's load more proportionately over each leg (Yoshioka, Nagano, Hay, & Fukashiro, 2010). This redistribution of body mass over the weaker leg may also be what is happening in this current study, since the asymmetry index and peak force difference between legs were not equal to zero, as they would be if jumpers were using legs perfectly equally. The redistribution of the body mass slightly toward the stronger leg may be due to leg dominance/preference, or perhaps an injury that has occurred in the past and changed the lower limb muscle usage pattern.

While a relationship between running speed and jump performance exists, a study looking at how lower limb asymmetries running in sport found that in males of team sports, asymmetries between legs in vertical jumps did not significantly correlate to multidirectional sprint speed (Lockie, et al., 2014). So although the athletes in this study did not use their legs symmetrically in the CMJs, perhaps this has no effect on running or sprinting during competition, and the stronger leg can compensate for the weaker leg.

More research into how asymmetries affect performance in competition, injury risk, and injury rehabilitation needs to be done to better understand these relationships. Different methods of quantifying asymmetries may need to be explored to find correlations, as it appears that the human body can often compensate for one side being weaker than the other.

2.3.13 Depth of CMJ

The portion of the CMJ where the jumper squats down (before propelling themselves upward) is the portion in which the countermovement is being generated. In a study on how CMJ jump height varies with the depth of the squat down, it was found that jump height and normalized impulse were greatest when using a deep position compared to a self-selected one (Sanchez-Sixto, Harrison, & Floria, 2018). This might suggest that the deeper countermovement jumps generated more vertical impulse, which allowed jumpers to reach a higher jump height.

In male basketball players, it was found that there was only a small change in jump height with a ride range in the squat depth being used during the CMJs (Mandic, Jakovljevic, & Jaric, 2015). The basketball players consistently preferred a CMJ depth more shallow than the one determined to be the optimal one (Mandic, Jakovljevic, & Jaric, 2015). This suggests that an optimal countermovement depth does exist, but it has a small role in maximizing jump performance relative to other jump characteristics (Mandic, Jakovljevic, & Jaric, 2015). To help athletes jump higher, training programs could encourage athletes to squat to the optimal depth and become accustomed to that depth.

Male athletes with jumping experience varied the depth of the squat during CMJs, and found that a deeper squat corresponded to a lesser peak force, and an increase in jump height and normalized impulse (Kirby, McBride, Haines, & Dayne, 2011). This meant that the primary factor correlating to the CMJ jump height was not peak force, but rather impulse (Kirby, McBride, Haines, & Dayne, 2011). This could also mean that impulse is a good means to assess jump performance, and might be better than peak force.

One study looked at how athletes who had good ankle dorsiflexion performed in CMJs compared to those with inflexible ankle dorsiflexion (Papaiakovou, 2013). The group with good dorsiflexion jumped higher—likely achieved by larger range of motion through all leg joints, higher angular velocities, lower trunk inclination during the squat down, and better overall joint coordinator (Papaiakovou, 2013). The inflexible ankle joint dorsiflexion group raised their heels off the ground to help them during the CMJ, and had a greater horizontal distance between their trunk center of mass and the center of their hip joint (Papaiakovou, 2013). This shows how something easily overlooked, such as ankle flexibility, can have significant impacts on the body's movement of the countermovement jump, and the performance of the jump.

Since Class 1 jumpers were found to squat to a lower depth (0.17±0.03 m) compared to Class 2 jumpers (0.30±0.08 m), this may indicate that the force-time curve shape and the squat depth are also related. It is possible that Class 2 jumpers have to squat to a greater depth to achieve similar heights to the Class 1 jumpers, which would also explain the slower jumps found in Class 2 jumpers. This would support the idea that Class 1 jumps are more ideal—since they are faster than Class 2, yet can jump higher and don't require as deep of a squat during the jump.

2.3.14 Plyometric Training

The stretch-shortening cycle is the term used to describe the muscle contractions in dynamic movements, commonly used in sports (Bartlett, 2007). The mechanism behind the stretch-shortening cycle is the preload, elastic energy storage and release, and reflex potentiation (Bartlett, 2007). While the stretch-shortening cycle is difficult to measure, and there is still much to learn about it, it is known to be important for power and strength training in athletes. The stretch-shortening cycle is a "common sequence of joint actions in which an eccentric (lengthening) muscle contraction, or pre-stretch, precedes a concentric (shortening) muscle contraction" (Bartlett, 2007). This means the stretch-shortening cycle is essential to plyometric training, including CMJs. In the stretch-shortening cycle, a body segment often moves in the opposite direction from the desired one, but uses this initial countermovement to allow the desired movement to occur. This is essential for movements that need force and speed, or to minimize energy use in movements. Any exercise where a muscle tendon unit is lengthened and directly followed by shortening is utilizing the stretch-shortening cycle, and can be considered plyometric (Chmielewski, Myer, Kauffman, & Tillman, 2006). Additionally, increasing strength capacity through resistance training can improve the stretch-shortening cycle, therefore improve CMJ performance and height (McMahon, Rej, & Comfort, 2017). One of the best ways to improve jump height in a variety of athletes is to increase the predictive factors--average force and eccentric rate of force development (Laffaye, Wagner, & Tombleson, 2014).

Plyometric training was initially used only for sport performance purposes, but now is being integrated into rehabilitation of injured athletes for their return to sport (Chmielewski, Myer, Kauffman, & Tillman, 2006). After injury, the plyometric training can begin at a low intensity and slowly increase intensity as rehabilitation progresses (Chmielewski, Myer, Kauffman, & Tillman, 2006). High intensity plyometric exercises are believed to improve neuromuscular impairments after injury and to prepare the musculoskeletal body system for the high forces and explosive movements found in sport (Chmielewski, Myer, Kauffman, & Tillman, 2006). This can help athletes undergoing rehabilitation return to full function for their sport. It's unclear at this point if plyometric training can be used to help prevent re-injury in athletes.

One study found that trained jumpers who frequently complete plyometric exercises as part of their sport (e.g. high level track and field athletes in horizontal or vertical jumping and running events) jump higher than untrained jumpers (McBride & Snyder, 2012). This is likely due to the higher mechanical efficiency and increased force production in the athletes that had undergone frequent plyometric training (McBride & Snyder, 2012). The trained jumpers were able to perform more negative work (in the eccentric phase) to generate energy transferred into performing higher jumps (McBride & Snyder, 2012). This study's results parallel another, that looked at the differences between jumpers and non-jumpers performing CMJs (Cormie, McBride, & McCaulley, 2009). Plyometric training status was found to not only impact the force, velocity, jump height and time points during the jump, but also the shape of the force-time curve (Cormie, McBride, & McCaulley, 2009).

While it's unclear if the time it takes to complete the jump correlates with the jump height, many sports also benefit quick jump times. Using both a short jump time and a high jump height as criteria, a study found that the relationship between jump height and jump time was more complicated than initially anticipated through simulations (Domire & Challis, 2015). Placing a small emphasis on the short time criteria resulted in a small decrease in jump height, but time savings, while a larger emphasis on the time criteria resulted in a much larger decrease in jump height. More research needs to be done to better understand the relationship between the two (Domire & Challis, 2015).

2.3.15 Limitations

Limitations for this study include a small sample size for some sports, and not consistently having both male and female participants from sport teams. There were almost twice as many female participants (n=46) compared to male participants (n=24). However, using Two-Way ANOVA to check for interaction effects between gender and sport should minimize the impact of this on outcomes.

When performing an analysis on left and right leg separately, some participants would slightly shift weight between legs—not so much that the movement was seen visually when recording the jump, but it was evident that movement was occurring when the graphs were plotted. This was made obvious in the quiet standing mass portion of the jump, where the graph for normal force in this stance for both legs was a smooth steady line, but the graphs for left and right leg separately had fluctuations. This may have affected the quiet standing mass for left and right leg separately, but likely not the total standing mass for both legs together. Observing how participants use their legs differently when performing the jump movement may be a useful way to quantify asymmetry between legs in the future.

There was no clear pattern for individual participants for which jump out of the five trials was their strongest, and so fatigue over the course of five jumps is unlikely a factor. The averages of variables of interest over the five jump trials was used rather than just using the best jump, as this may provide a better representation of an athlete's performance and capability compared to one single jump.

The custom template created in Mathematica was first built on data from a few counter movement jumps, and then was iteratively improved upon and further developed to ensure its success over a larger variety of jumps. However, very atypical jumps that have not yet been encountered may still not fit well with the template, due to the large variation in jumps between individuals. Due to the nature of creating a single program to automatically analyze a wide variety of jumps, certain criteria had to be established to obtain the variables of interest. The criteria were selected based on iteratively writing the program, running different CMJs through, and then modifying the program based on new issues encountered. The program works sufficiently well for the athletes in this study (and others that were not included), but there is a possibility that certain CMJs may not work well with the program.

Additionally, the criteria set in the Mathematica program was established using a trial and error method, where CMJs were run through the program and criteria was adjusted until it worked sufficiently well for all the jumps. However, this criteria may introduce additional sources of error. For example, one study found that selecting a takeoff point slightly too early could overestimate jump height by 1.5% (Street, McMillan, Board, Ramussen, & Heneghan, 2001). Another source of error could be introduced during the body weight averaging if participants were not standing perfectly still, which could impact all of the subsequent analysis and calculations. It is not expected that these errors had large effects on the outcome of the study.

The velocity of subjects' center of mass and their center of mass displacement at t_1 was assumed to be zero as this was defined as the onset of jump movement. This assumption was necessary to calculate the velocity and displacement functions from the force function, but it may not necessarily be true.

As with any signals collected, noise may be present in the force-time data recorded. If there were large amounts of noise in the force plate signal, this may impact the timing points selected automatically using the program, or the integrations performed during analysis. However, a Gaussian filter was applied to the data before performing analysis, and so data noise is not expected to have impacted the results of this study.

One limitation with the use of the PASCO PASPORT 2-Axis Force Platform is that there currently is no literature on the reliability of normal force measurements with these force plates over quick, dynamic movements, such as jumps. Force plates use transducers to convert mechanical energy input applied to the plate into an electrical output signal. If loads are being applied and removed and applied again very quickly, the transducer may not respond properly to the second load applied if it did not have sufficient time to recover from the first load applied. In the case of this study, the jumper taking off and landing again after the jump may simulate a similar situation. However, the manufacturer has not described these limitations as being significant, and recommends these force plates to be used for vertical jumps. Additionally, variation of the force in the CMJ and NCMJ with respect to time may be slow enough such that the force plate transducers can respond appropriately. Future work could look at the effect of applying and removing a load quickly to mimic a vertical jump, to quantify how quickly the sensors in these force plates can react to a change in force.

Chapter 3: Use of the Non-Counter Movement Jump in Quantifying Lower Limb Mechanical Muscle Function

3.1 Methods

The Non-Counter Movement Jump (NCMJ) closely resembles the Counter Movement Jump (CMJ) but includes a brief pause at the bottom of the squat to eliminate any momentum. Participants begin in the upright standing position with their hands on their hips, and squat down by flexing at the knees and hips. In a CMJ, the jumper immediately propels themselves vertically into the air. However, in a NCMJ, the jumper holds this squatted position for a short moment. After holding the squat, the jumper then extends knees and hips to jump vertically into the air. While the NCMJ is less frequently used by strength and conditioning coaches and in literature compared to the CMJ, NCMJs can be used to help quantify and understand lower limb strength with a different perspective. The NCMJ can provide different insight into how athletes use their lower limbs when they are not permitted to use their momentum (countermovement) gained by the squat down during a jump.

The NCMJ study was primarily used as follow up to the CMJ study previously completed, to gain a deeper understanding of lower limb mechanical muscle function as it varies between types of jumps. The Mathematica program used for the CMJs was modified and tailored to fit NCMJs such that both types of jumps could be analyzed and evaluated instantaneously. Many of the same variables from the CMJ study apply to the analysis of the NCMJ, but some were omitted for reasons that will be described.

The participants for this study consisted of athletes from the University of Alberta's varsity sports teams. Participants were the same as in the previous study on the counter movement jump, with two additional male volleyball players, and four additional male swimmers. The inclusion of these additional players is due to the nature of the data collection, which collected each jump on separate days, and some athletes were lost to follow-up to complete both jump types. Table 1 outlines the participants from the teams that participated, and included athletes from the swimming, volleyball, rugby and soccer teams.

Table 7: Participants' Sport and Gender

Sport Team	**Gender**	**Number of Participants**	
Swim	Male	9	Σ = 14
	Female	5	
Volleyball	Male	14	Σ = 14
Rugby	Female	20	Σ = 20
Soccer	Male	7	Σ = 28
	Female	21	
Total	Male	30	**Σ = 76**
	Female	46	

The experimental procedures for the non-counter movement jump study were again the same as the counter movement jump, with only a variation during the jump itself. Participants again squatted down upon verbal command (flexing at the knees and hips), but rather than immediately extending knees and hips to propel themselves into the air (as they did with the CMJ), participants held this squatted position for approximately two seconds, as timed by the data collector. The purpose of holding this squatted position was to eliminate any countermovement that the participants could gain during the jump. After the data collector timed this approximate two seconds, the participants were verbally instructed to jump upwards, vertically into the air. Upon landing back on the force plates, participants held the squatted position they landed in until the force exerted on the force plates stabilized and they were instructed to relax.

The Mathematica analysis for the NCMJ was largely the same as the CMJ analysis, with a few exceptions. For the NCMJ Mathematica code, t_1 was set to the onset of movement when the participant was leaving the squatted position. The jump phases were separated into only eccentric/concentric for the NCMJ code, and not unloading/loading/final phase as well as in the CMJ study.

The mass was still calculated from the jumper's quiet stance mass before the jump, and the program checked to ensure this quiet stance was relatively steady, otherwise it would output an error message. One additional parameter was added to the code for the NCMJ study, termed "squat deviation," which is equal to the standard deviation of the average force exerted by the participant while they were in the squatted phase. Squat deviation was calculated with intentions of quantifying how steady participants were holding the squat during the NCMJ.

For the jump height, only the Time in Air (TIA) method was used. Both the TIA method and Impulse Momentum Method (IMM) were both logistically valid in the CMJ, but the TIA method is much simpler, and integrating over the long course over the NCMJ may introduce more error in the IMM method.

The time phases in the NCMJ are somewhat redundant, due to the selection of the starting point for the NCMJ code. The code was written such that the jump “began” (t_1) just after jumpers held steady in the squat, at the first instance of movement out of this squat. Since if jumpers truly eliminated all of their countermovement by pausing in the squat, t_1, t_2 and t_3 should theoretically be the same. However, many participants exhibited a very small countermovement when leaving the squat (i.e. squatted deeper initially) before propelling themselves upward into the air—in a sense “cheating” the NCMJ, as they did not eliminate all countermovement. This meant that eccentric-specific parameters (such as impulse in the eccentric phase) was not calculated in the program as they should theoretically be equal to zero. However, eccentric phase time was still generated in the code to determine how long this small squat took before propelling into the air.

The time points used in the NCMJ code are described in Table 2, and correspond to the force-time curve in Figure 30.

Table 8: Non-Counter Movement Jump Time Point Definitions

Time Point	Definition
t_1	Time at onset of movement (after squatted position)
t_2	Time at peak negative center of mass velocity; where acceleration=0 (onset of braking)
t_3	Time where center of mass velocity=0 (onset of propulsion)
t_4	Time at takeoff
t_5	Time at landing
t_{max}	Time at overall maximum normal force

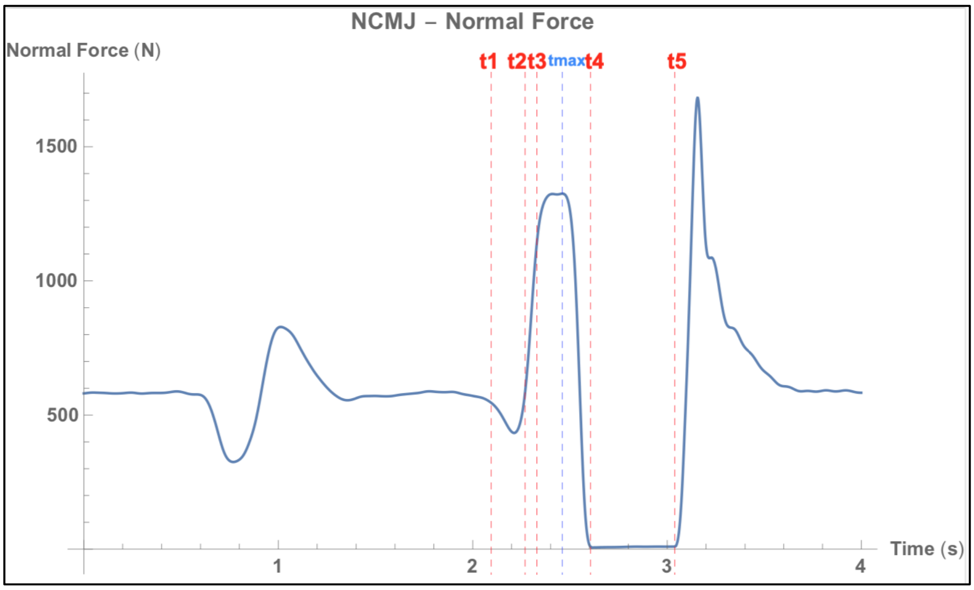


Figure 30: Non-Counter Movement Jump Time Points

When compared to the CMJ, the force-time of the NCMJ is similar, but it is clear that there is a phase in the jump where the force is relatively steady during which the subject is holding the squatted position. The NMCJ Mathematica code was run for all five trials for all jumpers, and results were outputted for statistical analysis. The Mathematica code used is described in Appendix E.

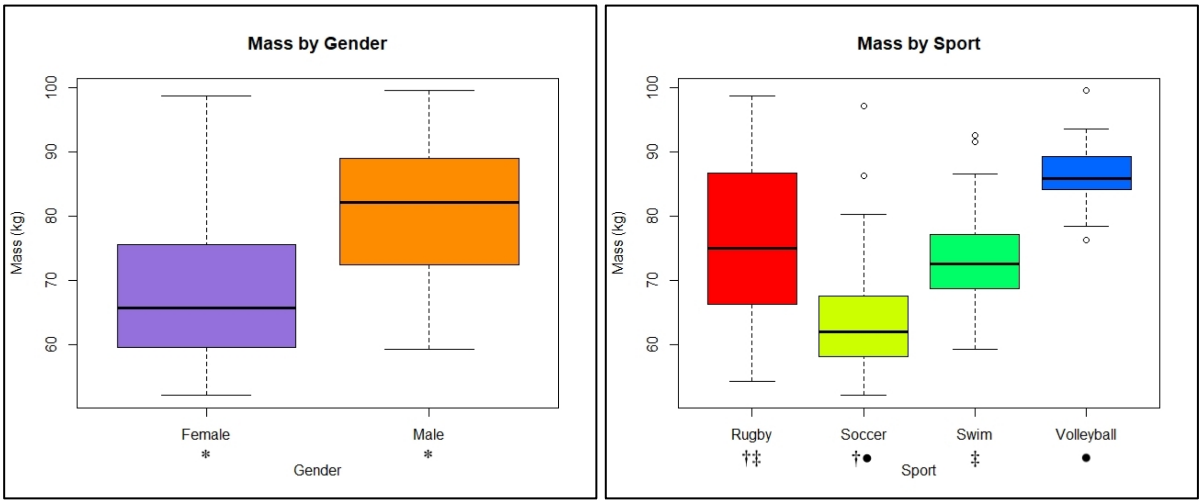
3.2 Results

The variables obtained from the Mathematica analysis for the NCMJs were averaged over the five jump trials and are described in this section. Two-way ANOVA was performed in R to determine if there was a significant interaction effect between gender and sport at the $\alpha < 0.05$ level. While there were no significant interaction effects found in the CMJ study, some variables in the NCMJ study were found to have an interaction effect. Variables that were found to have a significant interaction effect did not undergo subsequent Tukey tests.

For the variables with no interaction effect between gender and sport type, Tukey honest significance difference (HSD) post hoc tests were then performed in R. Appendix A-E include further details on the outcomes of the Two-way ANOVA and the Tukey HSD tests. Results are reported in the form of (average±standard deviation [unit], n=number of participants in group). A significance level of $\alpha=0.05$ is used for all results described in this section.

3.2.1 Mass and Force

Figure 31 contains the results for the mass measured for subjects while in the quiet standing position on the force plates. There was a significant difference between male (80.7±10.6 kg, n=30) and female (68.9±12.2 kg, n=46) for mass in the quiet stance. Significant differences were also found between rugby (76.0±13.0 kg, n=20) and soccer (60.2±6.8 kg, n=28), between rugby and swim (74.5±10.0 kg, n=14), and between soccer and volleyball (86.5±6.1 kg, n=14).

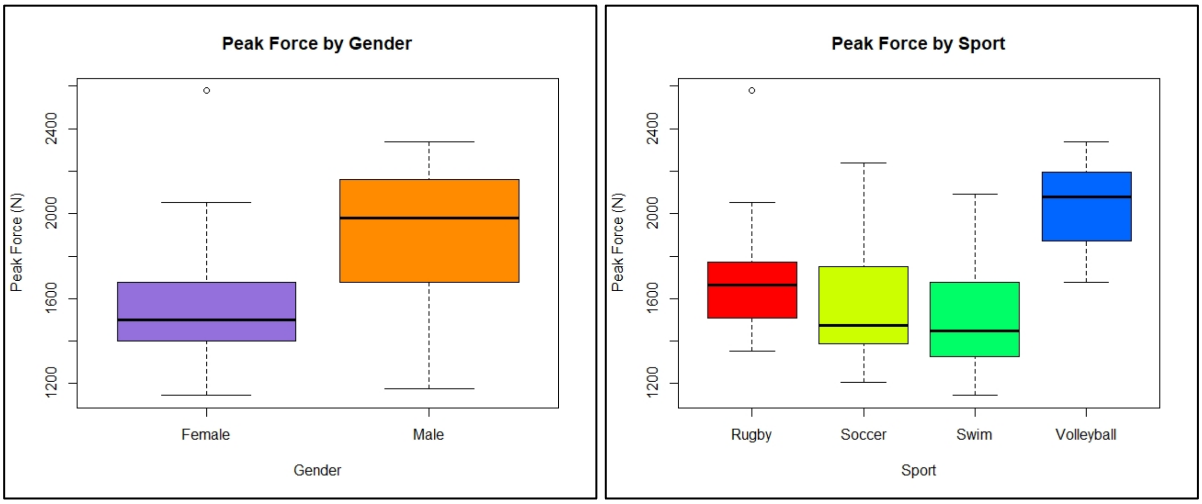


(a) *(b)*

Figure 31: Mass by a) gender and b) sport.

Note: matching symbols under group title indicates a significant difference at the 0.05 level.

Figure 32 displays the peak force measured for both legs combined for all participants. When conducting the Two-way ANOVA, peak force was found to have significant interaction effects (at the 0.05 level) between gender and sport, so further statistical analysis (i.e. Tukey HSD tests) was not done. Overall, the average peak force exerted by all participants was 1673.7±325.4 N, with males' average peak force at 1894.0±309.1 N, and females' average peak force at 1541.0±256.8 N. Swim had an average peak force of 1511.0±274.6 N, volleyball was 2053.3±187.8 N, rugby was 1691.4±274.8 N, and soccer was at 1431.6±68.4 N.

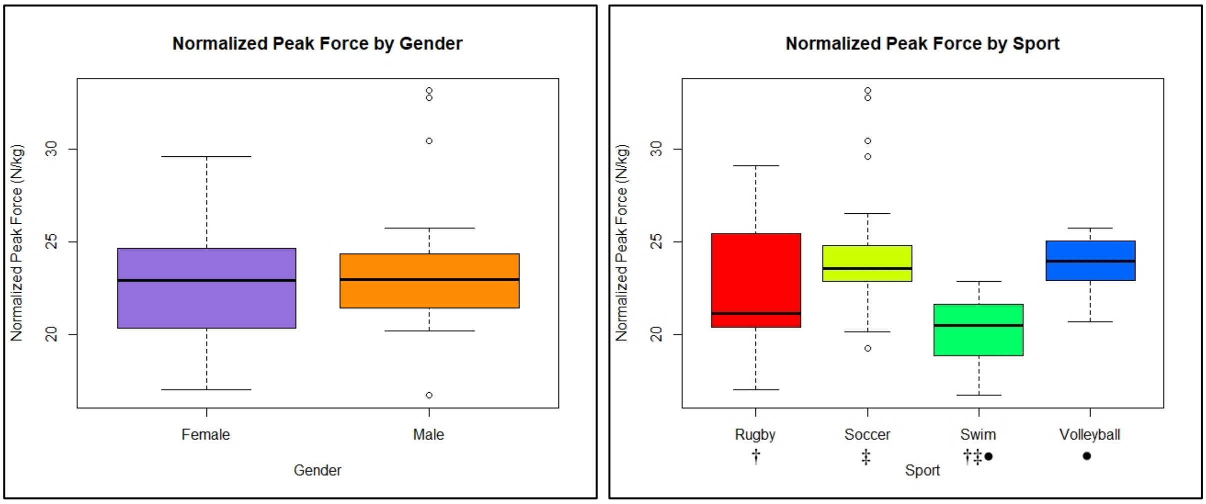


(a) *(b)*

Figure 32: Peak Force by a) gender and b) sport.

Note: matching symbols under group title indicates a significant difference at the 0.05 level.

Normalized peak force (peak force divided by body mass) was also compared (Figure 33). There was no statistical difference between peak force normalized to body mass between males and females. There were, however, significant differences for normalized peak force between swim (20.2±1.7 N, n=14) and the other three groups: rugby (22.6±3.4 N, n=20), soccer (23.8±2.0 N, n=28) and volleyball (23.8±1.6 N, n=14).

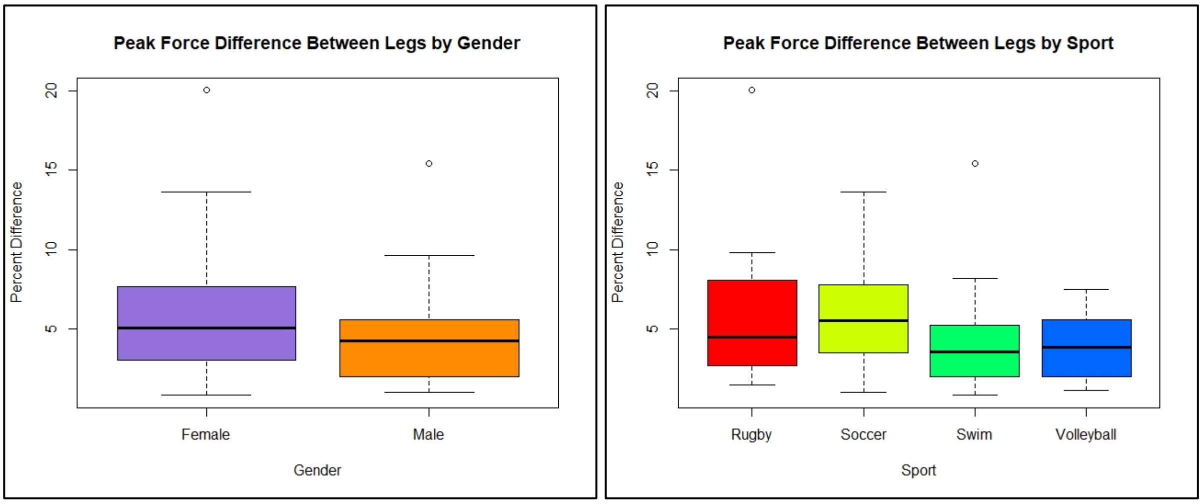


(a) *(b)*

Figure 33: Normalized Peak Force by a) gender and b) sport.

Note: matching symbols under group title indicates a significant difference at the 0.05 level.

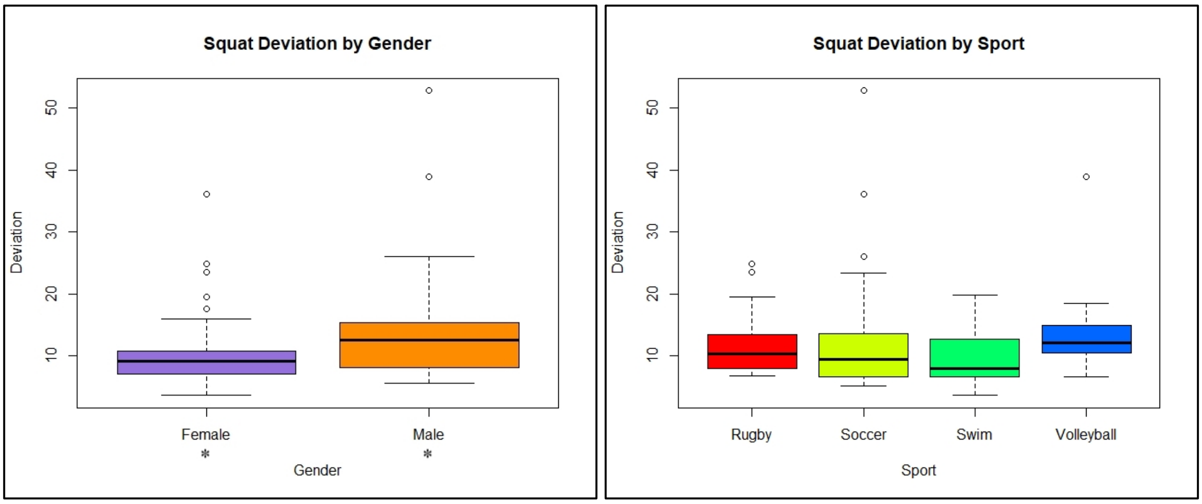
The difference in the peak force exerted by the left and right legs was calculated as a percentage (Figure 34), but no statistically significant differences between gender or sport groups were found. Overall, participants had a 5.2±3.6% difference between the peak force exerted between their left and right legs. Females had a difference of 5.6±3.7% while males had a difference of 4.5±3.1%. Swimmers had a peak force difference of 4.4±3.8%, volleyball had 4.0±2.1%, rugby had 5.8±4.3%, and soccer had 6.3±3.1% difference.



(a) *(b)*

Figure 34: Peak Force Difference Between Legs by a) gender and b) sport.
Note: matching symbols under group title indicates a significant difference at the 0.05 level.

The squat deviation (equal to the standard deviation in the squat phase) is compared in Figure 35. While there were no significant differences between the sports for this measurement, males (14.7±10.0, n=30) had a significantly larger squat deviation than females (10.5±5.9, n=46).



(a) *(b)*

Figure 35: Squat Deviation by a) gender and b) sport.
Note: matching symbols under group title indicates a significant difference at the 0.05 level.

3.2.2 Force-Time Curve Shape

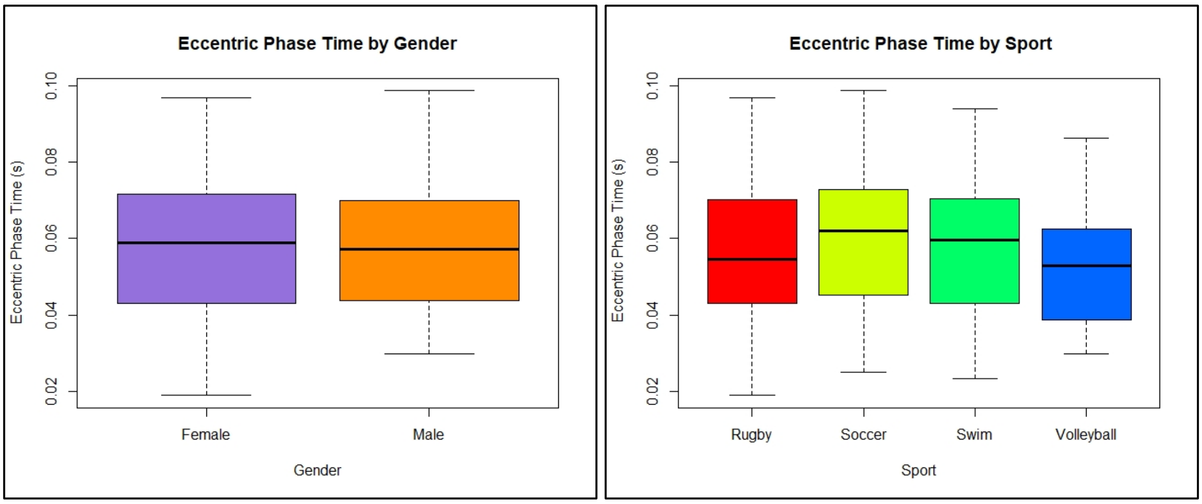
The jump classification of the majority (i.e. 3 or more) of each trial of NCMJ is shown in Table 9. Overall, only 12 subjects (15.8%) exhibited a Class 1 jump, while the remaining 64 subjects (84.2%) were classified as Class 2. Males had 2 (6.7%) Class 1 jumps, while females had 10 (21.7%) Class 1 jumps.

Table 9: Jump Classification of Majority of Jump Trials

Gender	**Sport**	**Classification of Majority of NCMJs**				**N**
		Class 1		**Class 2**		
		#	**(% of N)**	**#**	**(% of N)**	
Male	Swim	0	(0.0)	9	(100.0)	9
	Volleyball	1	(7.1)	13	(92.9)	14
	Soccer	1	(14.3)	6	(85.7)	7
	Total	2	(6.7)	28	(93.3)	30
Female	Swim	0	(0.0)	5	(100.0)	5
	Rugby	7	(35.0)	13	(65.0)	20
	Soccer	3	(14.3)	18	(85.7)	21
	Total	10	(21.7)	36	(78.3)	46
Total	Swim	0	(0.0)	14	(100.0)	14
	Volleyball	1	(7.1)	13	(92.9)	14
	Rugby	7	(35.0)	13	(65.0)	20
	Soccer	4	(14.3)	24	(85.7)	28
	Total	12	(15.8)	64	(84.2)	76

3.2.3 Phase Lengths

Figure 36 compares the time participants spent in the eccentric jump phase. There were no statistically significant differences found when participants were compared by sport or gender. It is important to note here though, that eccentric phase time in a NCMJ should theoretically be equal to zero. The average eccentric phase time across all participants was 0.057±0.02 s, with males at 0.058±0.02 s and females at 0.057±0.02 s. Additionally, swim's eccentric phase time was 0.058±0.02 s, volleyball's was 0.053±0.02 s, rugby was 0.057±0.02, and soccer was 0.059±0.02. It is clear from these numbers that these are all very small values (hundredths of a second) and all the groups are were found to have very similar length of their eccentric phase.

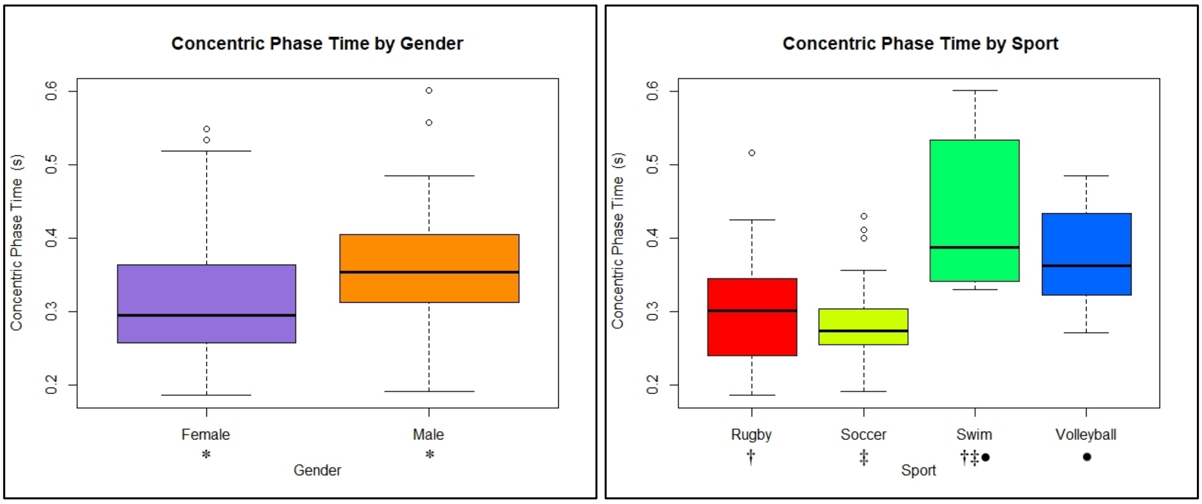


(a) *(b)*

Figure 36: Eccentric Phase Time by a) gender and b) sport.

Note: matching symbols under group title indicates a significant difference at the 0.05 level.

Figure 37 shows the time that participants spent in the concentric jump phase. There was a statistical difference between male (0.362±0.09 s, n=30) and female (0.317±0.09 s, n=46) for concentric phase jump length. Additionally, swim (0.436±0.100 s, n=14) significantly had the largest concentric phase time compared to rugby (0.303±0.082 s, n=20), soccer (0.294±0.058 s, n=28) and volleyball (0.374±0.063 s, n=14).

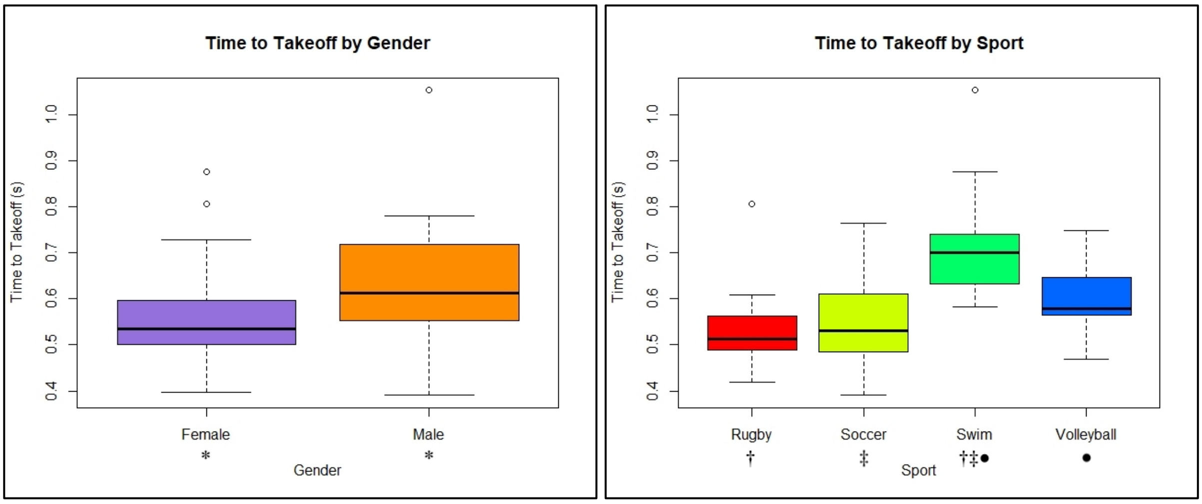


(a) *(b)*

Figure 37: Concentric Phase Time by a) gender and b) sport.

Note: matching symbols under group title indicates a significant difference at the 0.05 level.

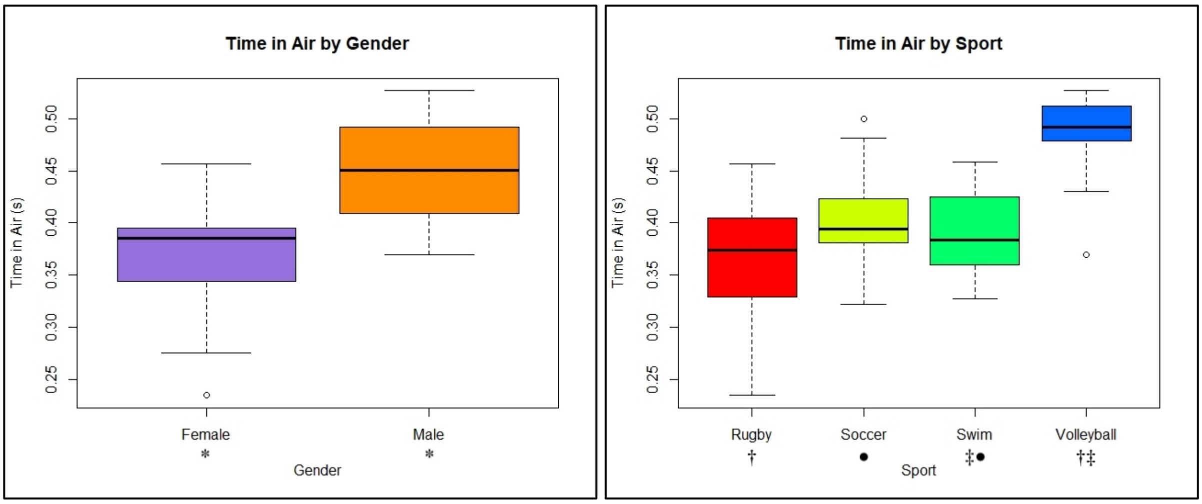
Time to takeoff is compared by group in Figure 38. Males (0.623±0.132 s, n=30) took significantly longer to takeoff compared to females (0.556±0.098 s, n=46), and the swim group (0.714±0.127 s, n=14) took significantly longer than the rugby (0.530±0.083 s, n=20), soccer (0.564±0.085 s, n=28) and volleyball (0.603±0.074 s, n=14) groups.



(a) *(b)*

Figure 38: Time to Takeoff by a) gender and b) sport.
Note: matching symbols under group title indicates a significant difference at the 0.05 level.

Figure 39 shows the results of the time spent in air by the participants. Males (0.452±0.050 s, n=30) were in the air for a significantly longer time than females (0.373±0.046 s, n=46). Additionally, volleyball players (0.485±0.043 s, n=14) were in the air significantly longer than rugby players (0.362±0.057 s, n=20) and swimmers (0.391±0.041 s, n=14). Soccer players (0.395±0.033 s, n=28) also spent significantly longer in the air compared to swimmers.



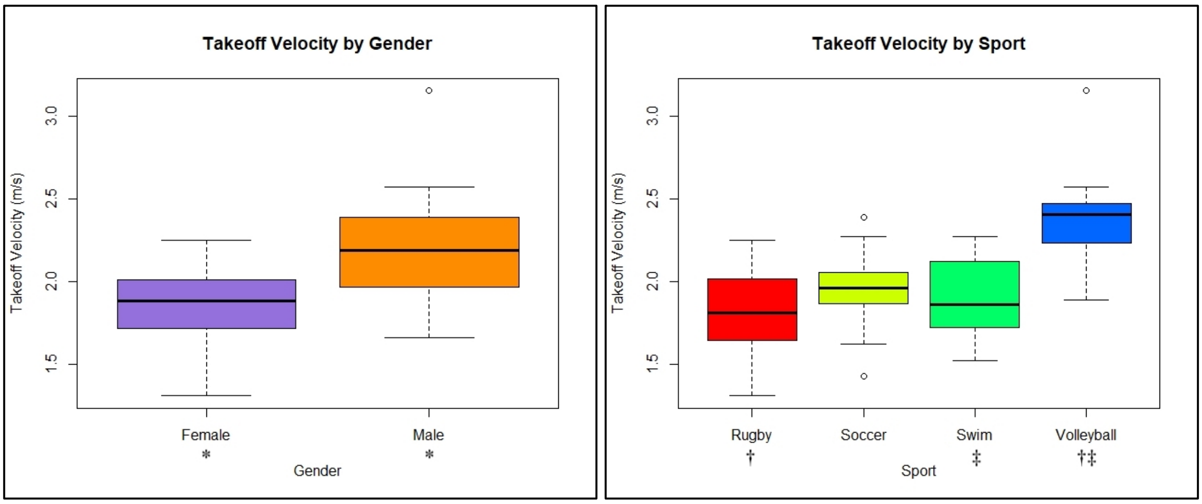
(a) *(b)*

Figure 39: Time in Air by a) Gender and b) sport.

Note: matching symbols under group title indicates a significant difference at the 0.05 level.

3.2.4 Displacement and Velocity

The takeoff velocity (the velocity at t_4) is compared in Figure 40. Males (2.205±0.295 m/s, n=30) had a significantly greater takeoff velocity compared to females (1.847±0.231 m/s, n=46), and the volleyball group (2.382±0.296 m/s, n=14) had a significantly greater takeoff velocity compared to the rugby group (1.805±0.271 m/s, n=20) and swim group (1.908±0.229 m/s, n=14). The soccer group (1.957±0.193 m/s, n=28) did not have significant differences for takeoff velocity compared to the other groups.

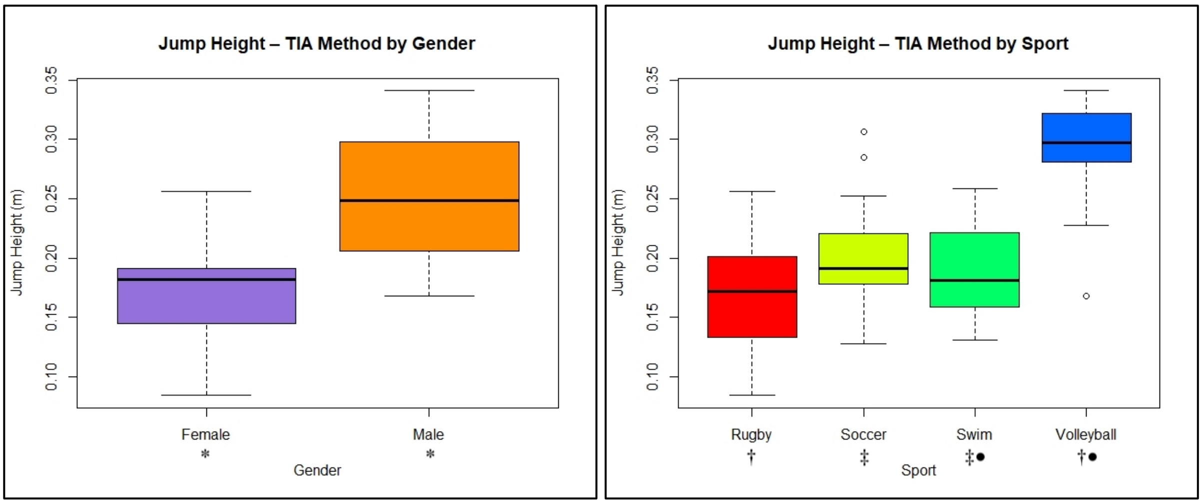


(a) *(b)*

Figure 40: Takeoff Velocity by a) gender and b) sport.

Note: matching symbols under group title indicates a significant difference at the 0.05 level.

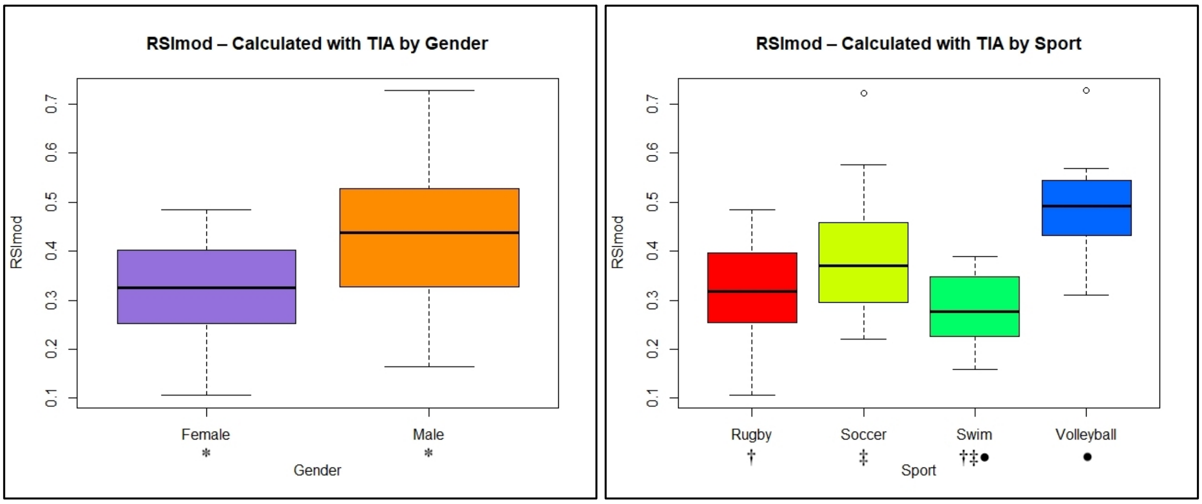
The jump height was calculated using the Time in Air (TIA) method (Figure 41). Males (0.254±0.55 m, n=30) jumped significantly higher than females (0.174±0.040 m, n=46). For the sport groups compared, volleyball (0.291±0.047 m, n=14) jumped significantly higher than swim (0.189±0.039 m, n=14) and rugby (0.165±0.047 m, n=20), and soccer (0.193±0.032 m, n=28) jumped significantly higher than swim.



(a) *(b)*

Figure 41: Jump Height, Calculated with the TIA Method by a) gender and b) sport.
Note: matching symbols under group title indicates a significant difference at the 0.05 level.

RSImod calculated and compared using the jump height calculated with the TIA method (Figure 42). The RSImod was significantly greater in males (0.434±0.133, n=30) than females (0.327±0.093, n=46), and out of all the sport groups, swim (0.280±0.073, n=14) significantly had the lowest RSImod compared to rugby (0.322±0.101, n=28), soccer (0.356±0.084, n=0.084) and volleyball (0.493±0.097, n=14).



(a) *(b)*

Figure 42: RSImod, Calculated with the TIA Jump Height by a) gender and b) sport.
Note: matching symbols under group title indicates a significant difference at the 0.05 level.

3.3 Discussion

The interaction effect found for the absolute force and the percent difference between limbs means that for those two variables, the sport with the higher or lower value depends on if you were looking at males or females in that sport. For these two variables, the Tukey HSD test was not performed.

Applying the Mathematica code to the NCMJ proved to be much more difficult than for the CMJ. With the NCMJs, there was a much larger variety in jumps, and participants often struggled to hold the squat steady for 2 seconds. This made it difficult to define criteria for when the time point at which the jump started, as well as the time point when they left the squat to jump upward. The Mathematica Code was initially written to start at the point that the jumper went from standing position and dropped down into their squat. By starting at the onset of movement from the standing position, the squatted phase (the amortization phase) could have been included in analysis. However, it was very difficult to find specific criteria that would fit the wide variety of force-time curve shapes of participants dropping down into the squat. Some would be very smooth, and immediately hold the squat steady. Others would be uneven and bumpy and would be hard to differentiate the movement of dropping down into the squat from instability in the squat. So instead of starting analysis from the actual start of the NCMJ, it was started from the point of when the jumper left the squat position. In general, the NCMJs were much less smooth than the CMJs, and much harder to write a code that would work for all of them due to a larger variety of jump shapes.

The small countermovement that most participants performed before propelling themselves upwards (as can be seen in Figure 43) indicates the difficulty in performing a "true" non-counter movement jump. Nearly all participants displayed this small dip at the end of the squat phase, right before they propelled themselves upward. There was no verbal instruction specifically to remove this, and so it shows a jumper's natural tendency to use countermovement to their advantage to jump higher. The eccentric phase of the NCMJ should theoretically be zero based on how it was calculated in this study. However, as most of the participants had at least a small countermovement downwards out of their squat before jumping upwards, these participants also had a small eccentric phase. The length of this small eccentric phase was very similar among all the sport groups and between genders. Future studies could perhaps include verbal instructions to refrain from this small countermovement downwards out of the squat and could evaluate how well participants were able to comply.

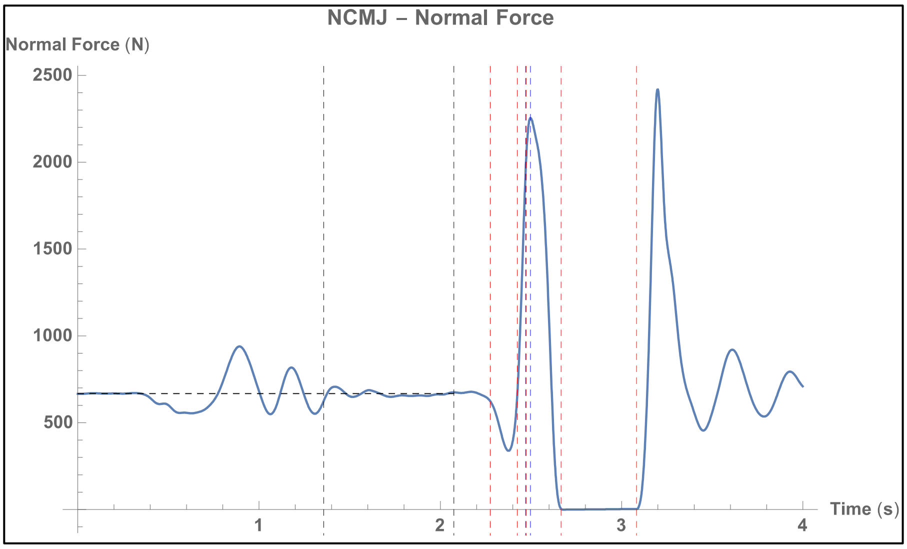


Figure 43: Example of a NCMJ with Unsteady Force during the Squat Portion, and a Small Countermovement Dip when Leaving the Squat Phase.

Many participants struggled to hold a steady squat during the squatted phase, which can also be seen in Figure 43. While it's unclear if this has implications on their performance in their sport, it would be reasonable to assume that athletes with stronger lower limbs and good proprioception would be able to hold the squatted position more steadily. Since lower limb strength and proprioception are important in all the sports examined in this study, athletes who were unable to hold the steady squat could perhaps benefit from a targeted training program to strengthen their lower limbs. The squat deviation variable was generated in the code in an attempt to help quantify how well athletes were holding the squat steadily during the NCMJ.

There were no differences between sports for squat deviation, but males were found to have a significantly larger squat deviation than females. This is interesting, because in the NCMJ study, males had larger peak forces, spent longer in the air, and had higher RSImod than females, and if a researcher were to use these three variables as a way to evaluate jump performance, males performed better than females. This means that perhaps the small movements that males made while in the squatted position during the NCMJ may have given them an advantage. It is possible that by not using extra muscle power to hold a squat perfectly steady, more muscle energy could be put into the actual jump upwards rather than tiring out muscles by disallowing weight shift between them.

The total percentages of Class 1 jumps (15.8%) and Class 2 jumps (84.2%) remained similar to what was found with the CMJs. However, in the CMJs, only soccer players exhibited Class 1 jumps (with the exception of one rugby player). In the NCMJ analysis, the Class 1 jumps are more spread out, and consist mostly of rugby players, with 35.0% of female rugby players falling under Class 1. The difference between the CMJ and NCMJ findings in terms of jump class may be due to the ability of different athletes to store and hold energy in their lower limbs during the squat. For example, soccer players have been found in both CMJs and NCMJs to be jumping relatively quickly but had less Class 1 CMJs than Class 1 NCMJs. This might mean that soccer players jump efficiently and quickly in a CMJ, but when they eliminate momentum, they don't jump as efficiently and smoothly. In the CMJ study, rugby athletes all fell under Class 1 (except one subject), but in the NCMJ study, there was a much greater proportion that fell under Class 2. This could mean that rugby players are able to more efficiently jump upwards after a pause where they eliminate their countermovement.

In both the CMJ and NCMJ, male body mass was significantly greater than that of female. The sport groups that had statistically significant results between each other varied slightly between the CMJ study and the NCMJ study, due to the inclusion of six additional participants in the NCMJ study. This also demonstrates the importance of sample size, and how addition of a few individuals can change which results are significant. In both studies, soccer had a significantly lower mass than both rugby and volleyball. In the NCMJ study, there was also a significant difference in body mass between rugby and swim, with the rugby players having the greater mass.

The absolute peak force in the NCMJ was not compared between groups, due to the interaction effect found. In the CMJ study, absolute peak force was not found to have an interaction effect, and so the between-group comparisons were completed. In both jumps, male had a greater absolute peak force compared to females. In the CMJ, the ranking from highest to lowest absolute peak force was volleyball, soccer, rugby and then swim. However, in the NCMJ, volleyball still had the highest absolute peak force, but rugby was next largest, followed by swim and then soccer. This is an interesting finding, and again may reflect on the type of jump that soccer players seem to excel at—ones where they can be quick and explosive, like a CMJ, and not ones where they can't use countermovement to their advantage.

When the peak force normalized to body mass was calculated and compared, in both the NCMJ and CMJ study, there were no differences between males and females. This is as expected, as males tend to have a greater muscle mass than females which helps them to generate a larger absolute peak force, but also means they have a larger overall body mass. By normalizing to the body mass, this makes for a better comparison between the two genders. The CMJ study found that soccer had a significantly greater normalized peak force compared to all the other sports; however, in the NCMJ study, soccer, volleyball and rugby had no significant differences between each other. The swim group had the lowest normalized peak force compared to the other three groups in the NCMJ study, while the swimmers were only significantly lower than the soccer group in the CMJ study. This again highlights that although a sport group can excel at one type of jump, it may be beneficial to help broaden our understanding of sport-specific characteristics during jumping by looking at more than just one type of jump. When comparing the normalized peak force between the CMJ and the NCMJ, the average across all participants was similar: the CMJ had an average normalized peak force of 23.7 N/kg, while the NCMJ had an average normalized peak force of 22.6 N/kg. The male participants in the CMJ had an average normalized peak force of 24.1 N/kg, and an average of 23.6 N/kg in the NCMJ. The females in the CMJ study had an average normalized peak force of 23.5 N/kg, and in the NCMJ study had an average of 22.6 N/kg. None of these are drastic differences between the CMJ and NCMJ (although the normalized peak force in the NCMJ was slightly lower than the CMJ for these cases), suggesting that the inclusion or exclusion of countermovement in a jump does not have a large impact on how much force per kilogram of body weight a jumper can exert.

The percent difference in the peak force exerted by the left and right leg was not found to have any significant differences between groups in both the CMJ and NCMJ study. However, in both of the jumps, the percent difference was not equal to zero—indicating that athletes favor one side over the other in both types of jumps (if measured by peak force exerted by each side).

In both the CMJ and NCMJ study, males spent statistically longer in the concentric jump phase compared to females, and swimmers spent the longest in this phase compared to these other three sports. However, in the CMJ study, the soccer group also spent statistically the shortest time in the concentric phase, which was not a finding of the NCMJ study. The total time to takeoff for the CMJ and NCMJ studies were also similar: males took longer than females, and the swim group took the longest out of all the sports. In the CMJ study, the soccer group also took significantly less time to takeoff compared to all the other groups, which again was not a finding in the NCMJ study. The soccer players in this study may have modified their usage of their lower limbs in the NCMJ compared to the CMJ, or perhaps their lower limb muscles were not as well-equipped to incorporate the squat of the NCMJ study and thus, they were no longer the quickest jumpers.

Since only the TIA method of calculating jump height relies only upon the subject’s time in the air, the time spent in the air and the jump height had the same significant differences between groups: males jumped higher/longer than females, volleyball players jumped higher/longer than rugby players and swimmers, and soccer players jumped higher/longer than swimmers. In the CMJ study, males also jumped higher/longer compared to females. However, in the CMJ study, the only significant difference for the jump height and time in air between the sport groups was between volleyball and swim, with volleyball jumping higher and were in the air for longer. Since additional differences were found to be significant with the NCMJ, this may be an important jump to use as a tool to help identify more differences in characteristics between sports.

Takeoff velocity was significantly greater in males compared to females for both the CMJ and NCMJ study. In the CMJ study, swim had a significantly lower takeoff velocity compared to soccer and volleyball. However, in the NCMJ study, it was found that swim still had a lower takeoff velocity compared to volleyball (but no difference with soccer), and additionally, volleyball had a higher takeoff velocity compared to rugby.

Literature often defines “good” jump performance as a jump where a large vertical height is reached by the athlete. This is a fair definition, as many sports benefit from their athletes being able to jump higher than their opponents. The average jump height for the CMJ jump was 0.212 m, while it was 0.205 m in the NCMJ. The males jumped an average of 0.285 m in the CMJ and 0.254 m in the NCMJ, indicating a slight decrease in jump height when countermovement was not allowed to be used in a jump. The females jumped an average of 0.174 m in both the CMJ and the NCMJ, which is interesting, as it was expected that the NCMJ jump height would be lower

RSImod is a way to quantify jump "explosiveness" in a CMJ and was used to quantify explosiveness in the same way in the NCMJ in this study. Both the CMJ and NCMJ found that RSImod was higher in males than females, indicating that both CMJs and NCMJs were more explosive for males. Additionally, in both the CMJ and NCMJ, swim had significantly the lowest RSImod compared to all the other sport groups, indicating they were the sport group with the least explosive jump in both types of jumps. The CMJ study also found that soccer was more explosive than rugby, which was not found in the NCMJ study. This is logical, as there were several other jump characteristics of the soccer players that slightly decreased relative to the other sport groups in the NCMJ compared to the CMJ, suggesting soccer players' performance in the NCMJ decreased slightly.

Chapter 4: Applications and Conclusion

4.1 Applications to Injury Rehabilitation

While this study included only subjects with no current lower limb injuries, the Mathematica program can be used as a rehabilitation tool to instantaneously analyze jumps performed on dual force plates for athletes with current or recent injuries. Tracking progress of an athlete's rehabilitation over time using this tool can help determine when athletes are fit to return to their sport. Intuitively, it is assumed that with a current lower limb injury, a jumper would rely more heavily on the uninjured leg, but variations from what is expected may be an indication that there is another underlying issue.

For example, (Figure 44) is a plot of the peak force (for the left and right legs separately) for a female soccer player rehabilitating after left ACL tear, followed by ACL reconstruction surgery. The right leg (the uninjured leg) applies a greater peak force than the left throughout all four dates. It appears that the rehabilitation progress is actually decreasing based on the peak force exerted by each leg. However, Figure 45 shows that the jump height (calculated using the TIA method) has actually been trending upward, despite the peak force being exerted trending down. This is an example of how the Mathematica program is beneficial to practitioners—the decreasing peak force may be misinterpreted as negative or a sign of poor rehabilitation, but if practitioners can instantaneously receive information about other jump parameters (such as jump height), this can provide more insight into rehabilitation progress immediately.

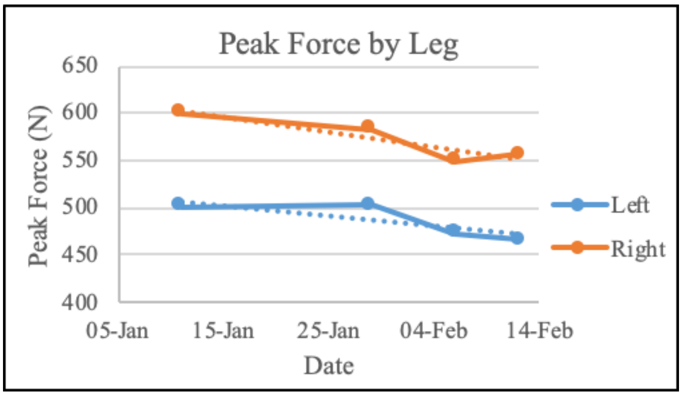


Figure 44: CMJ Peak Force by Leg over Four Dates for a Soccer Player Post-Left ACL Surgery

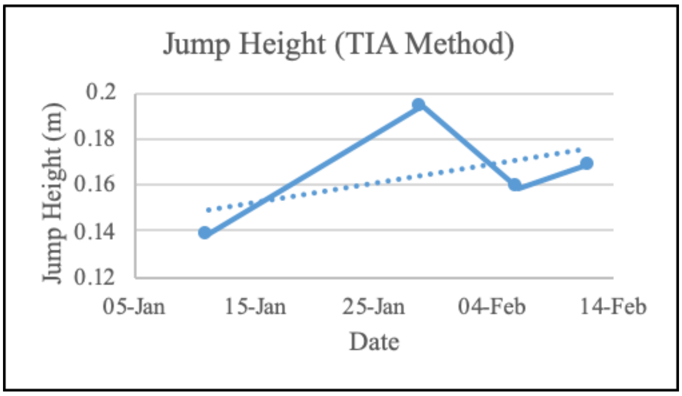


Figure 45: CMJ Jump Height over Four Dates for a Soccer Player Post-ACL Surgery

In a study looking at ski racers who had previous ACL injuries, it was found that the skiers still showed significant neuromuscular deficits, despite returning to sport (Jordan, Aagaard, & Herzog, 2017). These deficits included greater functional asymmetry between legs in vertical jumps during takeoff and landing. This would suggest that even after return to sport, monitoring using CMJs and adequate analysis should be continued, to ensure that any deficits are not masked. Analysis with this tool has already helped to identify a volleyball athlete that had previous lower limb injury that was presumed to be resolved, but was still affecting their CMJ performance. By obtaining baseline expectations for specific levels and types of athletes, comparisons between "healthy" and injured athletes can be made. Additionally, if CMJ analysis and monitoring is included in routine fitness testing, if an athlete becomes injured at some point after this, there is a reliable reference point for CMJ parameters to strive for in rehabilitation of the injury.

4.2 Applications to Performance Improvements

There is a need for a means to instantaneously quantify jump performance and lower limb muscle function for both performance and rehabilitation purposes. Counter movement jumps and non-counter movement jumps are simple, quick and require relatively inexpensive equipment to monitor lower limb function, yet are effective and reliable to do so. The Mathematica program can be improved and tailored for instant analysis in clinical or sport conditioning settings to track lower limb functional muscle performance over time. For example, in elite volleyball players, it is recommended that CMJs are used to monitor performance throughout the whole season (Ziv & Lidor, 2010). Continual athlete monitoring using CMJs and NCMJs can ensure adequate training, such that athletes are not being over-trained (leading to fatigue, higher injury risk, or suboptimal performance), but are training at a high enough intensity or frequency to maximize their potential. Athletes with a weaker CMJ or NCMJ performance can receive extra attention or modified training to raise their performance to that which is expected of someone of their level. A better understanding of sport-specific trends for CMJ and NCMJ parameters helps to set a baseline for what to expect and provides a better understanding for how to tailor athletes' training program to their sport.

4.3 Conclusion

At high levels, athletes undergo strength and conditioning training frequently to perform optimally in their sport, and are often at high risk of lower limb injuries during competition or training. The Mathematica program used in this study has proved to be successful in analyzing and quantifying lower limb body function and asymmetry in athletes during counter movement jumps and non-counter movement jumps, and can be used for performance or injury rehabilitation purposes. Additionally, this study has helped establish normative expectations of specific sports (rugby, soccer, swim and volleyball) relative to one another, such that future work has something to compare to.

In both CMJs and NCMJs, male participants were found to jump higher, take longer to jump, exert a larger absolute peak force, maintain higher impulse (in CMJs), and obtain a higher RSImod (measure of explosiveness) than females, but did not have significant differences from females when peak force was normalized to body mass. Additionally, no differences were found for the asymmetry indexes generated between the genders.

In CMJs, soccer players exerted the highest peak force per kilogram of body mass. They also had the shortest jump times and the smallest impulses, yet had the highest RSImod. Soccer players were the only sport group to display Class 1 (unimodal) force-time curves with the exception of one rugby athlete. Swimmers had the lowest absolute peak force, took the longest to jump, and had the lowest RSImod—suggesting they had the least explosive jumps. Volleyball players had the highest absolute peak force, the largest impulses, spent the longest time in the air, had the highest takeoff velocity and had the highest jump height. The rugby group did not have any statistically significant parameters compared to the other sports, and so it is difficult to determine what normative data for rugby players during CMJs looks like.

In NCMJs, swimmers had the lowest peak force normalized to body mass in NCMJs, longest concentric phase time, and longest time to takeoff. Volleyball players weighed the most, spent the longest time in the air, had the highest takeoff velocity, the highest jump height, and the highest RSImod (although these were not significant with all other sport groups).

Soccer players had the lowest body mass (although only significant with volleyball and rugby). Soccer and volleyball players were tied for their peak force normalized to body mass in the NCMJ study. Rugby players lacked statistically significant results from the all other sport groups for any one specific variable for many of the NCMJ variables, and many of their results averaged somewhere in the middle of the other sport groups'.

This additional study has verified that many of the same sport- and gender-specific jump characteristics found in counter movement jumps can also be found in non-counter movement jumps. The NCMJ Mathematica code was much more difficult to create compared to the CMJ code, due to a larger variation in jumps seen, less literature on NCMJs, and time points that have some ambiguity as to how they can be defined. Future work could continue to refine the NCMJ code to include analysis on the squat phase and look at different parameters such as quantifying how well jumpers are able to hold the squat phase steadily.

Further work should be done to create a more robust dataset for each sport to better understand what "optimal" and "normal" CMJs and NCMJs look like for each type of jump. The addition of another jump, such as a drop jump (stepping off of an elevated box onto the force plates and then immediately performing a jump), could further enhance our understanding of sport-specific jump and lower limb usage. Analyzing a drop jump is possible with small adjustments to the CMJ Mathematica program

Work on this project will continue as it grows and evolves with subsequent findings. This may include involving more athletes in the sports already studied, including varsity athletes from sport groups not yet studied (e.g. hockey), or including both male and female athletes in the sports in this current study (e.g. adding female volleyball). Other means of quantifying lower limb functional muscle strength may lead to an even stronger correlation between the experimental jumps performed and in-sport athlete performance. This may include "time to stabilization": having participants perform jumps on force plates as they did in this study, but upon landing, participants hold a squatted position until the fluctuating force becomes reasonably stable. Alternate methods of quantifying asymmetry may also be explored.

Additional work may also look at incorporating a wider variety of athlete levels. In this study, only varsity athletes were involved. However, future work could look at broadening the scope to look at athletes from the recreational level up to professional and elite athletes, and analyzing how the level of sport influences variables of interest.

Bibliography

American Orthopaedic Society for Sports Medicine. (2019, August). Stop Sport Injuries. Retrieved from www.stopsportsinjuries.org

Baker, L. (2018). Biomechanical analysis of jumping: the influence of external load and countermovement depth on deceleration strategies and performance. Doctoral dissertation, University of Nevada.

Barker, L., Harry, J., & Mercer, J. (2017). Relationships between countermovement jump, ground reaction forces and jump height, reactive strength index, and jump time. Journal of Strength and Conditioning Research, 32(1), 248-254.

Bartlett, R. (2007). Introduction to Sports Biomechanics: Analyzing Human Movement Patterns. Routledge.

Benjanuvatra, N., Edmunds, K., & Blanksby, B. (2007). Jumping ability and swimming grab-start performance in elite and recreational swimmers. International Journal of Aquatic Research and Education, 1, 231-242.

Breed, R., & Young, W. (2003). The effect of a resistance training programme on the grab, track and swing starts in swimming. Journal of Sports Sciences, 21, 213-220.

Brown, S. (2016). Does lower-extremity symmetry matter for anterior cruciate ligament injury risk in male rugby union athletes. Doctoral dissertation, Auckland University of Technology.

Chmielewski, T., Myer, G., Kauffman, D., & Tillman, S. (2006). Plyometric exercise in the rehabilitation of athletes: physiological responses and clinical application. Journal of Orthopaedic & Sports Physical Therapy, 36(5), 308-319.

Clark, M., & Lucett, S. (2010). Plyometric Training Concepts for Performance Enhancement. In NASM's Essentials of Sports Performance Training (pp. 207-226).

Claudino, J., Cronin, J., Mezencio, B., McMaster, D., McGuigan, M., Tricoli, V., . . . Serrao, J. (2017). The countermovement jump to monitor neuromuscular status: a meta-analysis. Journal of Science and Medicine in Sport, 397-402.

Cormie, P., McBride, J., & McCaulley, G. (2009). Power-time, force-time and velocity-time curve analysis of the countermovment jump: impact of training. Journal of Strength and Conditioning Research, 23(1), 177-186.

Crewther, B., Kilduff, L., Cook, C., Cunningham, D., Bunce, P., Bracken, R., & Gaviglio, C. (2012). Scaling strength and power for body mass differences in rugby union players. Journal of Sports Medicine and Physical Fitness, 52, 27-32.

De La Fuente, B., Garcia, F., & Arellano, R. (2003). Are the forces applied during vertical countermovement jump related to the forces applied during the swimming start. In Biomechanics and Medicine in Swimming (pp. 207-212). Publications de l'Universite Sant-Etienne.

Domire, Z., & Challis, J. (2015). Maximum height and minimum time vertical jumping. Journal of Biomechanics, 48, 2865-2870.

Dowling, J., & Vamos, L. (1993). Identification of kinetic and temporal factors related to vertical jump performance. Journal of Applied Biomechanics, 9, 95-110.

Duthie, G., Pyne, D., & Hooper, S. (2003). Applied physiology and game analysis of rugby union. Sports Medicine, 33(13), 973-991.

Ebben, W., & Petushek, E. (2010). Using the reactive strength index modified to evaluate plyometric performance. Journal of Strength and Conditioning Research, 24(8), 1983-1987.

Floria, P., Gomez-Landero, L., Suarez-Arrones, L., & Harrison, A. (2014). Kinetic and kinematic analysis for assessing the differences in countermovement jump performance in rugby players. Journal of Strength and Conditioning Research, 30(9), 2533-2539.

Forthomme, B., Croisier, J., Ciccarone, G., Crieland, J., & Cloes, M. (2005). Factors correlated with volleyball spike velocity. The American Journal of Sports Medicine, 33(10), 1513-1519.

Fried, T., & Lloyd, G. (1992). An overview of common soccer injuries: management and prevention. Sports Medicine, 269-275.

Garcia-Ramos, A., Tomazin, K., Feriche, B., Stronjik, V., De La Fuente, B., Arguelles-Cienfuegos, J., & Strumbelj, B. S. (2016). The relationship between the lower-body muscular profile and swimming start performance. Journal of Human Kinetics, 157-165.

Gheller, R., Pupo, J., Ache-Dias, J., Detanico, D., Padulo, J., & Santos, S. (2015). Effecct of different knee starting angles on intersegmental coordination and performance in vertical jumps. Human Movement Science, 42, 71-80.

Jezdimirovic, M., Joksimovic, A., Stankovic, R., & Bubanj, S. (2013). Differences in the vertical jump in soccer players according to their position on the team. Facta Universitatis: Series Physical Education & Sport, 221-226.

Jordan, M., Aagaard, P., & Herzog, W. (2015). Lower limb asymmetry in mechanical muscle function: A comparison between ski racers with and without ACL reconstruction. Scandanavian Journal of Medicine & Science in Sports, e301-e309.

Jordan, M., Aagaard, P., & Herzog, W. (2017). Anterior cruciate ligament injury/reinjury in alpine ski racing: a narrative review. Open Access Journal of Sports Medicine, 8, 71-83.

Kennedy, R., & Drake, D. (2018). Is a bimodal force-time curve related to countermovement jump performance. Sports, 6(36), 1-13.

Kirby, T., McBride, J., Haines, T., & Dayne, A. (2011). Relative net vertical impulse determines jumping performance. Journal of Applied Biomechanics, 27, 207-214.

Krizaj, J., Rauter, S., Vodicar, J., Hadzic, V., & Simenko, J. (2019). Predictors of vertical jumping capacity in soccer players. Isokinetics and Exercise Science, 27, 9-14.

Laffaye, G., Wagner, P., & Tombleson, T. (2014). Countermovement jump height: gender and sport-specific differences in the force-time variables. Journal of Strength and Conditioning Research, 28(4), 1096-1105.

Lake, J., & McMahon, J. (2018). Within-subject consistency of unimodal and bimodal force application during the countermovement jump. Sports, 6(143), 1-8.

Lee, C., Huang, C., Wang, L., & Lin, D. (2001). Comparison of the dynamics of the swimming grab start, squat jump, and countermovement jump of lower extremity. Proceedings of the Biomechanics Symposia.

Linthorne, N. P. (2001). Analysis of standing vertical jumps using a force platform. American Association of Physics Teachers, 1198-1204.

Lockie, R., Callaghan, S., Berry, S., Cooke, E., Jordan, C., Luczo, T., & Jeffriess, M. (2014). Relationship between unilateral jumping ability and asymmetry on multidirectional speed in team-sport athletes. Journal of Strength and Conditioning Research, 28(12), 3557-3566.

Mandic, R., Jakovljevic, S., & Jaric, S. (2015). Effects of countermovement depth on kinematic and kinetic patterns of maximum vertical jumps. Journal of Electromyography and Kinesiology, 25(2), 265-272.

Marshall, B., Franklyn-Miller, A., Moran, K., King, E., Richter, C., Gore, S., . . . Falvey, E. (2015). Biomechanical symmetry in elite rugby union players during dynamic tasks: an investigation using discrete and continuous data analysis techniques. International Journal of Sports Physiology and Performance, 12(6), 803-811.

McBride, J., & Snyder, J. (2012). Mechanical efficiency and force-time curve variation during repetitive jumping in trained and untrained jumpers. European Journal of Applied Physiology, 112, 3469-3477.

McMahon, J., Jones, P., Suchomel, T., Lake, J., & Comfort, P. (2018). Influence of the reactive strength index modified on force- and power-time curves. International Journal of Sports Physiology and Performance, 220-227.

McMahon, J., Lake, J., & Comfort, P. (2018). Reliability of and relationship between flight time to contraction time ration and reactive strength index modified. Sports, 6(81), 1-10.

McMahon, J., Murphy, S., Rej, S., & Comfort, P. (2017). Countermovement jump phase characteristics of senior and academy rugby league players. International Journal of Sports Physiology and Performance, 12(6), 803-811.

McMahon, J., Rej, S., & Comfort, P. (2017). Sex differences in countermovement jump phase characteristics. Sports, 5-8.

McMahon, J., Suchomel, T., Lake, J., & Comfort, P. (2018). Understanding the key phases of the countermovement jump force-time curve. Strength and Conditioning Journal, 96-106.

McMaster, T. (2016). A comparison of unimodal and bimodal countermovment jump force-time curves. Proceedings of Sport and Exercise Science New Zealand Annual Conference.

Mizuguchi, S. (2012). Net impulse and net impulse characteristics in vertical jumping. Doctoral dissertation, East Tennessee State University.

Moir, G. L. (2008). Three different methods of calculating vertical jump height from force platform data in men and women. Measurement in Physical Education and Exercise Science, 207-218.

Nordin, A., Kivi, D., Zerpa, C., & Newhouse, I. (2014). Comparison of methods for assessing vertical jump height performance. 681-684.

Nordin, A., Kivi, D., Zerpa, C., & Newhouse, I. (2014). Comparison of methods for assessing vertical jump height performance. Proceedings of the International Society of Biomechanics in Sports, (pp. 681-684).

Papaiakovou, G. (2013). Kinematic and kinetic differences in the execution of vertical jumps between people with good and poor ankle joint dorsiflexion. Journal of Sports Sciences, 1-8.

PASCO. (2013). PASCO Capstone Software Version 1.1.1. Roseville, California, USA.

R Core Team. (2019). R: A language and environment for statistical computing. Vienna, Austria: R Foundation for Statistical Computing. Retrieved from https://www.R-project.org/

R Core Team. (2019). R: A language and environment for statistical computing, Version 3.5.3. Vienna, Austria: R Foundation for Statistical Computing. Retrieved from https://www.R-project.org/

Rice, P., Goodman, C., Capps, C., Triplett, N., Erickson, T., & McBride, J. (2017). Force- and power-time curve comparison during jumping between strength-matched male and female basketball players. European Journal of Sport Science, 17(3), 286-293.

Robinson, P., & Mills, S. (2000). Relationship between scrummaging strength and standard field tests for power in rugby. Proceedings of the International Symposium on Biomechanics in Sports.

Rojano Ortega, D., Rodriguez Bies, E., & Berral de la Rosa, F. (2010). Analysis of the vertical ground reaction forces and temporal factors in the landing phase of a countermovement jump. Journal of Sports Science and Medicine, 282-287.

Sanchez-Sixto, A., Harrison, A., & Floria, P. (2018). Larger countermovement increases the jump height of countermovement jump. Sports, 6(131), 1-8.

Sanders, R. H., Thow, J., & Fairweather, M. (2011). Asymmetries in Swimming: Where do they come from? Journal of Swimming Research, (18)1-11.

Sattler, T., Sekulic, D., Hadzic, V., Uljevic, O., & Dervisevic, E. (2012). Vertical jumping tests in volleyball: reliability, validity, and playing-position specifics. Journal of Strength and Conditioning Research, 26(6), 1532-1538.

Schofield, J., Parent, E., Lewicke, J., Carey, J., El-Rich, M., & Adeeb, S. (2013). Characterizing asymmetry across the whole sit to stand movement in healthy participants. Journal of Biomechanics, 46(15), 2730-2735.

Sheppard, J., Cronin, J., Gabbett, T., McGuigan, M., Etxebarria, N., & Newton, R. (2008). Relative importance of strength, power, and anthropometric measures to jump performance of elite volleyball players. Journal of Strength and Conditioning Research, 22(3), 758-765.

Sole, C., Mizuguchi, S., Sato, K., Moir, G., & Stone, M. (2017). Phase characteristics of the countermovement jump force-time curve: a comparisoin of athletes by jumping ability. Journal of Strength and Conditioning Research, 32(4), 1155-1165.

Street, G., McMillan, S., Board, W., Ramussen, M., & Heneghan, J. (2001). Sources of error in determining countermovement jump height with the impulse method. Journal of Applied Biomechanics, 17, 23-54.

Suchomel, T., Sole, C., Bailey, C., Grazer, J., & Beckham, G. (2015). A comparison of reactive strength index-modified between six U.S. collegiate athletic teams. Journal of Strength and Conditioning Research, 29(5), 1310-1316.

Tillin, N., Pain, M., & Folland, J. (2013). Explosive force production during isometric squats correlates with athletic performance in rugby union players. Journal of Sports Sciences, 31(1), 66-76.

Turner, A., Brazier, J., Bishop, C., Chavda, S., Cree, J., & Read, P. (2015). Data analysis for strength and conditioning coaches: using excel to analyze reliability, differences and relationships. Strength and Conditioning Journal, 76-83.

Turner, A., Brazier, J., Bishop, C., Chavda, S., Cree, J., & Read, P. (2015). Data anlysis for strength and conditioning coaches: using excel to analyze reliability, differences and relationships. Strength and Conditioning Journal, 76-83.

Wagner, H., Tilp, M., von Duvillard, S., & Mueller, E. (2009). Kinematic analysis of volleyball spike jump. International Jounral of Sports Medicine, 30(10), 760-765.

Wang, Y., & Zhang, N. (2016). Effects of plyometric training on soccer players (review). Experimental and Therapeutic Medicine, 12, 550-554.

West, D., Owen, N., Cunningham, D., Cook, C., & Kilduff, L. (2011). Strength and power predictors of swimming starts in international sprint swimmers. Journal of Strength and Conditioning Research, 950-955.

Winkelman, N. (2011). The difference between the countermovement and non-countermovement jump: implications on performance. UK Strength and Conditioning Association, 4-8.

Wolfram Research, Inc. (2015). Wolfram Mathematica, Version 10.3.1.0. Champaign, Illinois, USA.

Yoshioka, S., Nagano, A., Hay, D., & Fukashiro, S. (2010). The effect of bilateral asymmetry of muscle strength on jumping height of the countermovement jump: a computer simulation study. Journal of Sports Sciences, 28(2), 209-218.

Zakas, A. (2006). Bilateral isokinetic peak torque of quadriceps and hamstring muscles in professional soccer players with dominance on one or both two sides. Journal of Sports Medicine and Physical Fitness, 46, 28-35.

Ziv, G., & Lidor, R. (2010). Vertical jump in female and male volleyball players: a review of observational and experimental studies. Scandinavian Journal of Medicine and Science in Sports, 556-567.

Appendices

Appendix A. CMJ Mass and Force Results

CMJ Mass

Two-Way ANOVA Results		
	Yes	No
Statistical Evidence for Interaction Effect Between Gender and Sport?		X
Statistical Significance Found between Gender groups?	X	
Statistical Significance Found between Sport groups?	X	

Tukey HSD Test
For groups with significant differences, which group has the larger value?
Rugby>Soccer Volleyball>Soccer Male>Female

CMJ Mass (kg)

Gender	Sport	Mean	Standard Deviation	N
Male	Swim	73.921	10.775	5
	Volleyball	87.953	6.642	12
	Soccer	75.772	12.713	7
	Total	81.477	11.298	24
Female	Swim	73.684	9.891	5
	Rugby	76.240	13.379	20
	Soccer	61.939	6.423	21
	Total	69.433	12.305	46
Total	Swim	73.803	9.752	10
	Volleyball	87.953	6.642	12
	Rugby	76.240	13.379	20
	Soccer	65.397	10.183	28
	Total	73.563	13.208	70

CMJ Mass (kg)

			95% Confidence Level			
Group A	Group B	Mean Difference (A-B)	Lower Bound	Upper Bound	P	
Soccer	Rugby	-13.854	-21.658	-6.049	0.000	*
Swim	Rugby	-8.459	-18.784	1.865	0.145	
Volleyball	Rugby	-0.331	-10.065	9.403	1.000	
Swim	Soccer	5.394	-4.426	15.215	0.474	
Volleyball	Soccer	13.523	4.325	22.721	0.001	*
Volleyball	Swim	8.128	-3.286	19.543	0.247	
Male	Female	12.043	6.960	17.127	0.000	*

*Mean difference is significant at α=0.05 level

CMJ Peak Force

Two-Way ANOVA Results		
	Yes	No
Statistical Evidence for Interaction Effect Between Gender and Sport?		X
Statistical Significance Found between Gender groups?	X	
Statistical Significance Found between Sport groups?	X	

Tukey HSD Test
For groups with significant differences, which group has the larger value?
Rugby>Swim Soccer>Swim Volleyball>Swim Male>Female

CMJ Peak Force (N)

Gender	Sport	Mean	Standard Deviation	N
Male	Swim	1600.271	236.216	5
	Volleyball	2029.355	174.313	12

	Soccer	2049.428	369.878	7
	Total	1945.817	304.615	24
Female	Swim	1339.469	133.007	5
	Rugby	1661.018	262.803	20
	Soccer	1621.885	220.909	21
	Total	1608.202	248.544	46
Total	Swim	1469.870	227.058	10
	Volleyball	2029.355	174.313	12
	Rugby	1661.018	262.803	20
	Soccer	1728.771	319.521	28
	Total	1723.956	311.883	70

CMJ Peak Force (N)

			95% Confidence Level			
Group A	**Group B**	**Mean Difference (A-B)**	**Lower Bound**	**Upper Bound**	**P**	
Soccer	Rugby	-16.651	-203.274	169.971	0.995	
Swim	Rugby	-359.955	-606.834	-113.077	0.002	*
Volleyball	Rugby	30.722	-202.038	263.481	0.985	
Swim	Soccer	-343.304	-578.133	-108.476	0.002	*
Volleyball	Soccer	47.373	-172.564	267.310	0.941	
Volleyball	Swim	390.677	117.743	663.612	0.002	*
Male	Female	337.615	216.055	459.175	0.000	*

*Mean difference is significant at α=0.05 level

CMJ Normalized Peak Force

Two-Way ANOVA Results		
	Yes	No
Statistical Evidence for Interaction Effect Between Gender and Sport?		X
Statistical Significance Found between Gender groups?		X
Statistical Significance Found between Sport groups?	X	

Tukey HSD Test
For groups with significant differences, which group has the larger value?
Soccer>Rugby Soccer>Swim Soccer>Volleyball

CMJ Normalized Peak Force (N/kg)

Gender	**Sport**	**Mean**	**Standard Deviation**	**N**
Male	Swim	21.717	2.083	5
	Volleyball	23.083	1.224	12
	Soccer	27.547	6.390	7
	Total	24.101	4.185	24
Female	Swim	18.253	0.822	5
	Rugby	21.974	2.355	20
	Soccer	26.279	3.218	21
	Total	23.535	3.834	46
Total	Swim	19.985	2.358	10
	Volleyball	23.083	1.224	12
	Rugby	21.974	2.355	20
	Soccer	26.596	4.130	28
	Total	23.729	3.937	70

CMJ Normalized Peak Force (N/kg)

			95% Confidence Level			
Group A	**Group B**	**Mean Difference (A-B)**	**Lower Bound**	**Upper Bound**	**P**	
Soccer	Rugby	4.480	2.126	6.833	0.000	*
Swim	Rugby	-2.274	-5.387	0.840	0.227	
Volleyball	Rugby	0.543	-2.392	3.478	0.962	

Swim	Soccer	-6.753	-9.715	-3.792	0.000	*
Volleyball	Soccer	-3.937	-6.710	-1.163	0.002	*
Volleyball	Swim	2.817	-0.625	6.258	0.146	
Male	Female	0.565	-0.968	2.098	0.464	

*Mean difference is significant at α=0.05 level

CMJ Peak Force Difference between Legs

Two-Way ANOVA Results	Yes	No
Statistical Evidence for Interaction Effect Between Gender and Sport?		X
Statistical Significance Found between Gender groups?		X
Statistical Significance Found between Sport groups?		X

Tukey HSD Test
For groups with significant differences, which group has the larger value?
None

CMJ Peak Force Difference between Legs (%)

Gender	Sport	Mean	Standard Deviation	N
Male	Swim	5.403	2.494	5
	Volleyball	5.760	4.030	12
	Soccer	4.540	3.091	7
	Total	5.330	3.410	24
Female	Swim	5.135	2.439	5
	Rugby	8.405	5.027	20
	Soccer	7.261	5.760	21
	Total	7.527	5.192	46
Total	Swim	5.269	2.330	10
	Volleyball	5.760	4.030	12
	Rugby	8.405	5.027	20
	Soccer	6.580	5.305	28
	Total	6.774	4.750	70

CMJ Peak Force Difference between Legs (%)			95% Confidence Level		
Group A	Group B	Mean Difference (A-B)	Lower Bound	Upper Bound	P
Soccer	Rugby	-1.275	-4.924	2.3732	0.7932
Swim	Rugby	-2.037	-6.863	2.7893	0.6827
Volleyball	Rugby	-0.448	-4.998	4.1028	0.9938
Swim	Soccer	-0.762	-5.353	3.8289	0.9717
Volleyball	Soccer	0.8276	-3.472	5.1273	0.957
Volleyball	Swim	1.5895	-3.746	6.9252	0.8606
Male	Female	-2.197	-4.574	0.1791	0.0693

*Mean difference is significant at α=0.05 level

Appendix B. CMJ Jump Class Results

		Classification of Majority of CMJs				
Gender	Sport	Class 1		Class 2		N
		#	(% of N)	#	(% of N)	
Male	Swim	0	(0.0)	5	(100.0)	5
	Volleyball	0	(0.0)	12	(100.0)	12
	Soccer	3	(42.9)	4	(57.1)	7
	Total	3	(12.5)	21	(87.5)	24
Female	Swim	0	(0.0)	5	(100.0)	5
	Rugby	1	(5.0)	19	(95.0)	20
	Soccer	9	(42.9)	12	(57.1)	21

	Total	10	(21.7)	36	(78.3)	46
Total	Swim	0	(0.0)	10	(100.0)	10
	Volleyball	0	(0.0)	12	(100.0)	12
	Rugby	1	(5.0)	19	(95.0)	20
	Soccer	12	(42.9)	16	(57.1)	28
	Total	13	(18.6)	57	(81.4)	70

Appendix C. CMJ Phase Lengths Results

CMJ Eccentric Phase Time

Two-Way ANOVA Results		
	Yes	No
Statistical Evidence for Interaction Effect Between Gender and Sport?		X
Statistical Significance Found between Gender groups?	X	
Statistical Significance Found between Sport groups?	X	

Tukey HSD Test
For groups with significant differences, which group has the larger value?
Rugby>Soccer Swim>Rugby Swim>Soccer Swim>Volleyball Male>Female

CMJ Eccentric Phase Time (s)

Gender	Sport	Mean	Standard Deviation	N
Male	Swim	0.239	0.031	5
	Volleyball	0.197	0.034	12
	Soccer	0.167	0.108	7
	Total	0.197	0.066	24
Female	Swim	0.224	0.024	5
	Rugby	0.161	0.030	20
	Soccer	0.121	0.027	21
	Total	0.150	0.042	46
Total	Swim	0.232	0.027	10
	Volleyball	0.197	0.034	12
	Rugby	0.161	0.030	20
	Soccer	0.133	0.060	28
	Total	0.166	0.056	70

CMJ Eccentric Phase Time (s)

			95% Confidence Level			
Group A	Group B	Mean Difference (A-B)	Lower Bound	Upper Bound	P	
Soccer	Rugby	-0.040	-0.073	-0.006	0.014	*
Swim	Rugby	0.047	0.003	0.092	0.033	*
Volleyball	Rugby	-0.011	-0.053	0.030	0.889	
Swim	Soccer	0.087	0.045	0.129	0.000	*
Volleyball	Soccer	0.028	-0.011	0.068	0.244	
Volleyball	Swim	-0.059	-0.108	-0.009	0.013	*
Male	Female	0.047	0.026	0.069	0.000	*

*Mean difference is significant at α=0.05 level

CMJ Concentric Phase Time

Two-Way ANOVA Results		
	Yes	No
Statistical Evidence for Interaction Effect Between Gender and Sport?		X
Statistical Significance Found between Gender groups?	X	

Tukey HSD Test
For groups with significant differences, which group has the larger value?
Rugby>Soccer Swim>Rugby

Statistical Significance Found between Sport groups?	X		Swim>Soccer Volleyball>Soccer Swim>Volleyball Male>Female

CMJ Concentric Phase Time (s)

Gender	Sport	Mean	Standard Deviation	N
Male	Swim	0.381	0.040	5
	Volleyball	0.335	0.028	12
	Soccer	0.261	0.060	7
	Total	0.323	0.060	24
Female	Swim	0.382	0.036	5
	Rugby	0.297	0.036	20
	Soccer	0.238	0.037	21
	Total	0.279	0.058	46
Total	Swim	0.382	0.036	10
	Volleyball	0.335	0.028	12
	Rugby	0.297	0.036	20
	Soccer	0.244	0.044	28
	Total	0.294	0.062	70

CMJ Concentric Phase Time (s)

			95% Confidence Level			
Group A	Group B	Mean Difference (A-B)	Lower Bound	Upper Bound	P	
Soccer	Rugby	-0.064	-0.094	-0.035	0.000	*
Swim	Rugby	0.063	0.024	0.102	0.000	*
Volleyball	Rugby	-0.006	-0.043	0.031	0.975	
Swim	Soccer	0.127	0.090	0.164	0.000	*
Volleyball	Soccer	0.058	0.024	0.093	0.000	*
Volleyball	Swim	-0.069	-0.112	-0.025	0.001	*
Male	Female	0.043	0.024	0.063	0.000	*

*Mean difference is significant at α=0.05 level

CMJ Unloading Phase Time

Two-Way ANOVA Results			Tukey HSD Test
	Yes	No	For groups with significant differences, which group has the larger value?
Statistical Evidence for Interaction Effect Between Gender and Sport?		X	
Statistical Significance Found between Gender groups?	X		Rugby>Soccer Swim>Rugby Swim>Soccer Swim>Volleyball Male>Female
Statistical Significance Found between Sport groups?	X		

CMJ Unloading Phase Time (s)

Gender	Sport	Mean	Standard Deviation	N
Male	Swim	0.429	0.021	5
	Volleyball	0.367	0.064	12
	Soccer	0.320	0.068	7
	Total	0.366	0.069	24
Female	Swim	0.382	0.057	5
	Rugby	0.311	0.039	20
	Soccer	0.270	0.044	21
	Total	0.300	0.055	46
Total	Swim	0.406	0.047	10

	Volleyball	0.367	0.064	12
	Rugby	0.311	0.039	20
	Soccer	0.282	0.054	28
	Total	0.323	0.068	70

CMJ Unloading Phase Time (s)

			95% Confidence Level			
Group A	**Group B**	**Mean Difference (A-B)**	**Lower Bound**	**Upper Bound**	**P**	
Soccer	Rugby	-0.045	-0.083	-0.007	0.013	*
Swim	Rugby	0.062	0.011	0.112	0.010	*
Volleyball	Rugby	-0.010	-0.058	0.037	0.940	
Swim	Soccer	0.107	0.059	0.155	0.000	*
Volleyball	Soccer	0.035	-0.010	0.080	0.173	
Volleyball	Swim	-0.072	-0.127	-0.016	0.006	*
Male	Female	0.067	0.042	0.091	0.000	*

*Mean difference is significant at α=0.05 level

CMJ Loading Phase Time

Two-Way ANOVA Results		
	Yes	No
Statistical Evidence for Interaction Effect Between Gender and Sport?		X
Statistical Significance Found between Gender groups?	X	
Statistical Significance Found between Sport groups?	X	

Tukey HSD Test
For groups with significant differences, which group has the larger value?
Swim>Soccer Male>Female

CMJ Loading Phase Time (s)

Gender	Sport	Mean	Standard Deviation	N
Male	Swim	0.373	0.125	5
	Volleyball	0.300	0.115	12
	Soccer	0.223	0.171	7
	Total	0.293	0.140	24
Female	Swim	0.298	0.147	5
	Rugby	0.197	0.077	20
	Soccer	0.152	0.052	21
	Total	0.188	0.087	46
Total	Swim	0.336	0.135	10
	Volleyball	0.300	0.115	12
	Rugby	0.197	0.077	20
	Soccer	0.170	0.098	28
	Total	0.224	0.118	70

CMJ Loading Phase Time (s)

			95% Confidence Level			
Group A	**Group B**	**Mean Difference (A-B)**	**Lower Bound**	**Upper Bound**	**P**	
Soccer	Rugby	-0.053	-0.130	0.024	0.274	
Swim	Rugby	0.086	-0.016	0.188	0.128	
Volleyball	Rugby	-0.002	-0.099	0.094	1.000	
Swim	Soccer	0.139	0.042	0.236	0.002	*
Volleyball	Soccer	0.051	-0.040	0.142	0.458	
Volleyball	Swim	-0.088	-0.201	0.024	0.175	
Male	Female	0.105	0.055	0.155	0.000	*

*Mean difference is significant at α=0.05 level

CMJ Final Phase Time

Two-Way ANOVA Results		
	Yes	No
Statistical Evidence for Interaction Effect Between Gender and Sport?		X
Statistical Significance Found between Gender groups?		X
Statistical Significance Found between Sport groups?	X	

Tukey HSD Test
For groups with significant differences, which group has the larger value?
Swim>Soccer

CMJ Final Phase Time (s)

Gender	Sport	Mean	Standard Deviation	N
Male	Swim	0.248	0.138	5
	Volleyball	0.232	0.087	12
	Soccer	0.205	0.060	7
	Total	0.227	0.090	24
Female	Swim	0.308	0.102	5
	Rugby	0.261	0.062	20
	Soccer	0.207	0.042	21
	Total	0.241	0.067	46
Total	Swim	0.278	0.118	10
	Volleyball	0.232	0.087	12
	Rugby	0.261	0.062	20
	Soccer	0.206	0.046	28
	Total	0.236	0.076	70

CMJ Final Phase Time (s)

			95% Confidence Level			
Group A	Group B	Mean Difference (A-B)	Lower Bound	Upper Bound	P	
Soccer	Rugby	-0.051	-0.106	0.005	0.084	
Swim	Rugby	0.024	-0.049	0.097	0.825	
Volleyball	Rugby	-0.015	-0.084	0.054	0.941	
Swim	Soccer	0.075	0.005	0.145	0.031	*
Volleyball	Soccer	0.036	-0.029	0.101	0.474	
Volleyball	Swim	-0.039	-0.120	0.042	0.590	
Male	Female	-0.014	-0.050	0.022	0.436	

*Mean difference is significant at α=0.05 level

CMJ Time to Takeoff

Two-Way ANOVA Results		
	Yes	No
Statistical Evidence for Interaction Effect Between Gender and Sport?		X
Statistical Significance Found between Gender groups?	X	
Statistical Significance Found between Sport groups?	X	

Tukey HSD Test
For groups with significant differences, which group has the larger value?
Rugby>Soccer Swim>Rugby Swim>Soccer Volleyball>Soccer Swim>Volleyball Male>Female

CMJ Time to Takeoff (s)

Gender	Sport	Mean	Standard Deviation	N
Male	Swim	1.050	0.076	5
	Volleyball	0.899	0.096	12
	Soccer	0.748	0.217	7
	Total	0.886	0.171	24

Female	Swim	0.988	0.098	5
	Rugby	0.769	0.090	20
	Soccer	0.629	0.096	21
	Total	0.729	0.146	46
Total	Swim	1.019	0.089	10
	Volleyball	0.899	0.096	12
	Rugby	0.769	0.090	20
	Soccer	0.659	0.142	28
	Total	0.783	0.171	70

CMJ Time to Takeoff (s)

			95% Confidence Level			
Group A	**Group B**	**Mean Difference (A-B)**	**Lower Bound**	**Upper Bound**	**P**	
Soccer	Rugby	-0.149	-0.235	-0.064	0.000	*
Swim	Rugby	0.171	0.058	0.284	0.001	*
Volleyball	Rugby	-0.028	-0.134	0.079	0.904	
Swim	Soccer	0.321	0.213	0.428	0.000	*
Volleyball	Soccer	0.122	0.021	0.223	0.011	*
Volleyball	Swim	-0.199	-0.324	-0.074	0.000	*
Male	Female	0.157	0.102	0.213	0.000	*

*Mean difference is significant at α=0.05 level

CMJ Time in Air

Two-Way ANOVA Results		
	Yes	No
Statistical Evidence for Interaction Effect Between Gender and Sport?		X
Statistical Significance Found between Gender groups?	X	
Statistical Significance Found between Sport groups?	X	

Tukey HSD Test
For groups with significant differences, which group has the larger value?
Volleyball>Swim Male>Female

CMJ Time in Air (s)

Gender	**Sport**	**Mean**	**Standard Deviation**	**N**
Male	Swim	0.446	0.047	5
	Volleyball	0.506	0.026	12
	Soccer	0.459	0.049	7
	Total	0.480	0.046	24
Female	Swim	0.349	0.024	5
	Rugby	0.368	0.054	20
	Soccer	0.386	0.032	21
	Total	0.374	0.043	46
Total	Swim	0.397	0.062	10
	Volleyball	0.506	0.026	12
	Rugby	0.368	0.054	20
	Soccer	0.404	0.048	28
	Total	0.410	0.067	70

CMJ Time in Air (s)

			95% Confidence Level		
Group A	**Group B**	**Mean Difference (A-B)**	**Lower Bound**	**Upper Bound**	**P**
Soccer	Rugby	0.009	-0.023	0.041	0.873
Swim	Rugby	-0.024	-0.066	0.018	0.446
Volleyball	Rugby	0.032	-0.008	0.072	0.153
Swim	Soccer	-0.033	-0.073	0.007	0.141
Volleyball	Soccer	0.023	-0.015	0.060	0.377

Volleyball	Swim	0.056	0.009	0.102	0.012	*
Male	Female	0.106	0.085	0.126	0.000	*

*Mean difference is significant at α=0.05 level

Appendix D. CMJ Impulse Results

CMJ Eccentric Phase Impulse

Two-Way ANOVA Results	Yes	No
Statistical Evidence for Interaction Effect Between Gender and Sport?		X
Statistical Significance Found between Gender groups?	X	
Statistical Significance Found between Sport groups?	X	

Tukey HSD Test
For groups with significant differences, which group has the larger value?
Rugby>Soccer Volleyball>Soccer Male>Female

CMJ Eccentric Phase Impulse (N·s)

Gender	Sport	Mean	Standard Deviation	N
Male	Swim	86.832	19.185	5
	Volleyball	115.780	17.895	12
	Soccer	80.450	30.112	7
	Total	99.445	27.154	24
Female	Swim	79.983	11.160	5
	Rugby	83.387	18.465	20
	Soccer	65.689	12.369	21
	Total	74.938	17.248	46
Total	Swim	83.407	15.231	10
	Volleyball	115.780	17.895	12
	Rugby	83.387	18.465	20
	Soccer	69.380	18.900	28
	Total	83.340	24.023	70

CMJ Eccentric Phase Impulse (N·s)

			95% Confidence Level			
Group A	Group B	Mean Difference (A-B)	Lower Bound	Upper Bound	P	
Soccer	Rugby	-20.134	-33.946	-6.322	0.002	*
Swim	Rugby	-12.233	-30.505	6.039	0.299	
Volleyball	Rugby	7.886	-9.341	25.113	0.624	
Swim	Soccer	7.901	-9.479	25.281	0.630	
Volleyball	Soccer	28.020	11.742	44.298	0.000	*
Volleyball	Swim	20.120	-0.081	40.320	0.051	
Male	Female	24.507	15.510	33.504	0.000	*

*Mean difference is significant at α=0.05 level

CMJ Concentric Phase Impulse

Two-Way ANOVA Results	Yes	No
Statistical Evidence for Interaction Effect Between Gender and Sport?		X
Statistical Significance Found between Gender groups?	X	
Statistical Significance Found between Sport groups?	X	

Tukey HSD Test
For groups with significant differences, which group has the larger value?
Rugby>Soccer Rugby>Swim Volleyball>Soccer Volleyball>Swim Male>Female

CMJ Concentric Phase Impulse (N·s)

Gender	Sport	Mean	Standard Deviation	N
Male	Swim	156.925	18.854	5
	Volleyball	212.536	18.717	12
	Soccer	165.474	25.518	7
	Total	187.224	32.819	24
Female	Swim	114.295	14.273	5
	Rugby	133.639	20.379	20
	Soccer	114.561	13.430	21
	Total	122.827	19.119	46
Total	Swim	135.610	27.447	10
	Volleyball	212.536	18.717	12
	Rugby	133.639	20.379	20
	Soccer	127.289	27.970	28
	Total	144.906	39.310	70

CMJ Concentric Phase Impulse (N·s)

			95% Confidence Level			
Group A	Group B	Mean Difference (A-B)	Lower Bound	Upper Bound	P	
Soccer	Rugby	-22.449	-36.603	-8.296	0.001	*
Swim	Rugby	-30.228	-48.951	-11.505	0.000	*
Volleyball	Rugby	14.500	-3.153	32.152	0.144	
Swim	Soccer	-7.779	-25.588	10.030	0.659	
Volleyball	Soccer	36.949	20.269	53.629	0.000	*
Volleyball	Swim	44.728	24.029	65.427	0.000	*
Male	Female	64.397	55.178	73.616	0.000	*

*Mean difference is significant at α=0.05 level

CMJ Normalized Eccentric Phase Impulse

Two-Way ANOVA Results		
	Yes	No
Statistical Evidence for Interaction Effect Between Gender and Sport?		X
Statistical Significance Found between Gender groups?	X	
Statistical Significance Found between Sport groups?	X	

Tukey HSD Test
For groups with significant differences, which group has the larger value?
Volleyball>Soccer Male>Female

CMJ Normalized Eccentric Phase Impulse (N·s/kg)

Gender	Sport	Mean	Standard Deviation	N
Male	Swim	1.183	0.265	5
	Volleyball	1.314	0.165	12
	Soccer	1.036	0.262	7
	Total	1.206	0.241	24
Female	Swim	1.090	0.124	5
	Rugby	1.099	0.182	20
	Soccer	1.058	0.144	21
	Total	1.079	0.158	46
Total	Swim	1.137	0.201	10
	Volleyball	1.314	0.165	12
	Rugby	1.099	0.182	20
	Soccer	1.053	0.175	28
	Total	1.123	0.198	70

CMJ Normalized Eccentric Phase Impulse (N·s/kg)

95% Confidence Level

Group A	Group B	Mean Difference (A-B)	Lower Bound	Upper Bound	P	
Soccer	Rugby	-0.078	-0.218	0.062	0.458	
Swim	Rugby	-0.026	-0.211	0.159	0.983	
Volleyball	Rugby	0.088	-0.086	0.263	0.545	
Swim	Soccer	0.052	-0.124	0.228	0.861	
Volleyball	Soccer	0.166	0.002	0.331	0.047	*
Volleyball	Swim	0.114	-0.090	0.318	0.461	
Male	Female	0.126	0.035	0.217	0.007	*

*Mean difference is significant at α=0.05 level

CMJ Normalized Concentric Phase Impulse

Two-Way ANOVA Results		
	Yes	No
Statistical Evidence for Interaction Effect Between Gender and Sport?		X
Statistical Significance Found between Gender groups?	X	
Statistical Significance Found between Sport groups?	X	

Tukey HSD Test
For groups with significant differences, which group has the larger value?
Soccer>Swim Volleyball>Swim Male>Female

CMJ Normalized Concentric Phase Impulse (N·s/kg)

Gender	Sport	Mean	Standard Deviation	N
Male	Swim	2.135	0.184	5
	Volleyball	2.420	0.155	12
	Soccer	2.195	0.192	7
	Total	2.295	0.209	24
Female	Swim	1.559	0.147	5
	Rugby	1.784	0.293	20
	Soccer	1.852	0.134	21
	Total	1.791	0.232	46
Total	Swim	1.847	0.342	10
	Volleyball	2.420	0.155	12
	Rugby	1.784	0.293	20
	Soccer	1.937	0.211	28
	Total	1.963	0.328	70

CMJ Normalized Concentric Phase Impulse (N·s/kg)

			95% Confidence Level			
Group A	Group B	Mean Difference (A-B)	Lower Bound	Upper Bound	P	
Soccer	Rugby	0.028	-0.131	0.186	0.968	
Swim	Rugby	-0.189	-0.399	0.021	0.091	
Volleyball	Rugby	0.131	-0.066	0.329	0.306	
Swim	Soccer	-0.217	-0.417	-0.017	0.028	*
Volleyball	Soccer	0.104	-0.083	0.291	0.463	
Volleyball	Swim	0.321	0.089	0.553	0.003	*
Male	Female	0.504	0.401	0.607	0.000	*

*Mean difference is significant at α=0.05 level

CMJ Asymmetry Index – Eccentric Phase

Two-Way ANOVA Results		
	Yes	No
Statistical Evidence for Interaction Effect Between Gender and Sport?		X

Tukey HSD Test
For groups with significant differences, which group has the larger value?

Statistical Significance Found between Gender groups?		X	None
Statistical Significance Found between Sport groups?		X	

CMJ Asymmetry Index – Eccentric Phase

Gender	Sport	Mean	Standard Deviation	N
Male	Swim	17.853	7.387	5
	Volleyball	13.385	6.892	12
	Soccer	12.585	3.850	7
	Total	14.083	6.333	24
Female	Swim	11.584	4.895	5
	Rugby	13.582	6.184	20
	Soccer	14.145	7.558	21
	Total	13.622	6.652	46
Total	Swim	14.718	6.769	10
	Volleyball	13.385	6.892	12
	Rugby	13.582	6.184	20
	Soccer	13.755	6.788	28
	Total	13.780	6.502	70

CMJ Asymmetry Index – Eccentric Phase

			95% Confidence Level		
Group A	Group B	Mean Difference (A-B)	Lower Bound	Upper Bound	P
Soccer	Rugby	0.058	-5.046	5.161	1.000
Swim	Rugby	0.906	-5.845	7.656	0.985
Volleyball	Rugby	-0.658	-7.023	5.707	0.993
Swim	Soccer	0.848	-5.573	7.269	0.985
Volleyball	Soccer	-0.715	-6.729	5.299	0.989
Volleyball	Swim	-1.563	-9.027	5.900	0.946
Male	Female	0.461	-2.863	3.785	0.783

*Mean difference is significant at α=0.05 level

CMJ Asymmetry Index – Concentric Phase

Two-Way ANOVA Results			Tukey HSD Test
	Yes	No	For groups with significant differences, which group has the larger value?
Statistical Evidence for Interaction Effect Between Gender and Sport?		X	
Statistical Significance Found between Gender groups?		X	Rugby>Soccer Swim>Soccer
Statistical Significance Found between Sport groups?	X		

CMJ Asymmetry Index – Concentric Phase

Gender	Sport	Mean	Standard Deviation	N
Male	Swim	24.100	8.700	5
	Volleyball	20.093	9.203	12
	Soccer	15.595	6.191	7
	Total	19.616	8.549	24
Female	Swim	32.970	16.110	5
	Rugby	26.253	17.514	20
	Soccer	15.661	6.228	21
	Total	22.148	14.490	46
Total	Swim	28.535	13.071	10
	Volleyball	20.093	9.203	12

	Rugby	26.253	17.514	20
	Soccer	15.645	6.103	28
	Total	21.280	12.758	70

CMJ Asymmetry Index – Concentric Phase			95% Confidence Level			
Group A	Group B	Mean Difference (A-B)	Lower Bound	Upper Bound	P	
Soccer	Rugby	-9.975	-19.187	-0.763	0.029	*
Swim	Rugby	3.548	-8.638	15.735	0.869	
Volleyball	Rugby	-3.628	-15.118	7.861	0.839	
Swim	Soccer	13.523	1.932	25.115	0.016	*
Volleyball	Soccer	6.347	-4.510	17.204	0.419	
Volleyball	Swim	-7.176	-20.649	6.296	0.501	
Male	Female	-2.532	-8.533	3.468	0.402	

*Mean difference is significant at α=0.05 level

Appendix E. CMJ Displacement and Velocity Results

CMJ Takeoff Velocity

Two-Way ANOVA Results		
	Yes	No
Statistical Evidence for Interaction Effect Between Gender and Sport?		X
Statistical Significance Found between Gender groups?	X	
Statistical Significance Found between Sport groups?	X	

Tukey HSD Test
For groups with significant differences, which group has the larger value?
Soccer>Swim Volleyball>Swim Male>Female

CMJ Takeoff Velocity (m/s)

Gender	Sport	Mean	Standard Deviation	N
Male	Swim	2.135	0.184	5
	Volleyball	2.420	0.155	12
	Soccer	2.195	0.192	7
	Total	2.295	0.209	24
Female	Swim	1.559	0.147	5
	Rugby	1.784	0.293	20
	Soccer	1.852	0.134	21
	Total	1.791	0.232	46
Total	Swim	1.847	0.342	10
	Volleyball	2.420	0.155	12
	Rugby	1.784	0.293	20
	Soccer	1.937	0.211	28
	Total	1.963	0.328	70

CMJ Takeoff Velocity (m/s)			95% Confidence Level			
Group A	Group B	Mean Difference (A-B)	Lower Bound	Upper Bound	P	
Soccer	Rugby	0.027	-0.131	0.186	0.969	
Swim	Rugby	-0.190	-0.399	0.020	0.091	
Volleyball	Rugby	0.131	-0.066	0.329	0.306	
Swim	Soccer	-0.217	-0.416	-0.017	0.028	*
Volleyball	Soccer	0.104	-0.083	0.291	0.462	
Volleyball	Swim	0.321	0.089	0.553	0.003	*
Male	Female	0.504	0.401	0.608	0.000	*

*Mean difference is significant at α=0.05 level

CMJ Jump Height – TIA Method

Two-Way ANOVA Results		
	Yes	No
Statistical Evidence for Interaction Effect Between Gender and Sport?		X
Statistical Significance Found between Gender groups?	X	
Statistical Significance Found between Sport groups?	X	

Tukey HSD Test
For groups with significant differences, which group has the larger value?
Volleyball>Swim Male>Female

CMJ Jump Height – TIA Method (m)

Gender	Sport	Mean	Standard Deviation	N
Male	Swim	0.246	0.050	5
	Volleyball	0.315	0.031	12
	Soccer	0.261	0.057	7
	Total	0.285	0.052	24
Female	Swim	0.150	0.021	5
	Rugby	0.170	0.049	20
	Soccer	0.184	0.030	21
	Total	0.174	0.040	46
Total	Swim	0.198	0.062	10
	Volleyball	0.315	0.031	12
	Rugby	0.170	0.049	20
	Soccer	0.203	0.050	28
	Total	0.212	0.069	70

CMJ Jump Height – TIA Method (m)

			95% Confidence Level			
Group A	Group B	Mean Difference (A-B)	Lower Bound	Upper Bound	P	
Soccer	Rugby	0.005	-0.026	0.037	0.968	
Swim	Rugby	-0.027	-0.069	0.014	0.317	
Volleyball	Rugby	0.034	-0.005	0.074	0.106	
Swim	Soccer	-0.033	-0.072	0.007	0.140	
Volleyball	Soccer	0.029	-0.008	0.066	0.178	
Volleyball	Swim	0.062	0.016	0.108	0.004	*
Male	Female	0.111	0.090	0.131	0.000	*

*Mean difference is significant at α=0.05 level

CMJ Jump Height – IMM Method

Two-Way ANOVA Results		
	Yes	No
Statistical Evidence for Interaction Effect Between Gender and Sport?		X
Statistical Significance Found between Gender groups?	X	
Statistical Significance Found between Sport groups?	X	

Tukey HSD Test
For groups with significant differences, which group has the larger value?
Soccer>Swim Volleyball>Swim Male>Female

CMJ Jump Height – IMM Method (m)

Gender	Sport	Mean	Standard Deviation	N
Male	Swim	0.234	0.039	5
	Volleyball	0.300	0.037	12
	Soccer	0.248	0.044	7
	Total	0.271	0.048	24
Female	Swim	0.125	0.024	5
	Rugby	0.167	0.056	20

	Soccer	0.176	0.025	21
	Total	0.167	0.043	46
Total	Swim	0.180	0.065	10
	Volleyball	0.300	0.037	12
	Rugby	0.167	0.056	20
	Soccer	0.194	0.043	28
	Total	0.202	0.067	70

CMJ Jump Height – IMM Method (m)

			95% Confidence Level			
Group A	Group B	Mean Difference (A-B)	Lower Bound	Upper Bound	P	
Soccer	Rugby	0.001	-0.031	0.033	1.000	
Swim	Rugby	-0.039	-0.081	0.002	0.071	
Volleyball	Rugby	0.029	-0.011	0.068	0.229	
Swim	Soccer	-0.041	-0.080	-0.001	0.044	*
Volleyball	Soccer	0.028	-0.010	0.065	0.214	
Volleyball	Swim	0.068	0.022	0.115	0.001	*
Male	Female	0.104	0.084	0.125	0.000	*

*Mean difference is significant at α=0.05 level

CMJ Peak (Negative) Eccentric Displacement

Two-Way ANOVA Results		
	Yes	No
Statistical Evidence for Interaction Effect Between Gender and Sport?		X
Statistical Significance Found between Gender groups?	X	
Statistical Significance Found between Sport groups?	X	

Tukey HSD Test
For groups with significant differences, which group has the larger value?
Soccer>Rugby Soccer>Swim Soccer>Volleyball Female>Male

CMJ Peak (Negative) Eccentric Displacement (m)

Gender	Sport	Mean	Standard Deviation	N
Male	Swim	-0.407	0.110	5
	Volleyball	-0.377	0.046	12
	Soccer	-0.257	0.087	7
	Total	-0.348	0.094	24
Female	Swim	-0.328	0.026	5
	Rugby	-0.263	0.052	20
	Soccer	-0.204	0.045	21
	Total	-0.243	0.062	46
Total	Swim	-0.368	0.086	10
	Volleyball	-0.377	0.046	12
	Rugby	-0.263	0.052	20
	Soccer	-0.217	0.061	28
	Total	-0.279	0.089	70

CMJ Peak (Negative) Eccentric Displacement (m)

			95% Confidence Level			
Group A	Group B	Mean Difference (A-B)	Lower Bound	Upper Bound	P	
Soccer	Rugby	0.073	0.028	0.117	0.000	*
Swim	Rugby	-0.052	-0.110	0.007	0.105	
Volleyball	Rugby	-0.009	-0.064	0.047	0.977	
Swim	Soccer	-0.124	-0.180	-0.068	0.000	*
Volleyball	Soccer	-0.081	-0.134	-0.029	0.001	*
Volleyball	Swim	0.043	-0.022	0.108	0.309	

Male	Female	-0.105	-0.134	-0.076	0.000	*

*Mean difference is significant at α=0.05 level

CMJ Peak (Positive) Concentric Displacement

Two-Way ANOVA Results		
	Yes	No
Statistical Evidence for Interaction Effect Between Gender and Sport?		X
Statistical Significance Found between Gender groups?	X	
Statistical Significance Found between Sport groups?		X

Tukey HSD Test
For groups with significant differences, which group has the larger value?
Male>Female

CMJ Peak (Positive) Concentric Displacement (m)

Gender	Sport	Mean	Standard Deviation	N
Male	Swim	0.174	0.031	5
	Volleyball	0.185	0.031	12
	Soccer	0.170	0.013	7
	Total	0.179	0.027	24
Female	Swim	0.121	0.024	5
	Rugby	0.146	0.025	20
	Soccer	0.148	0.026	21
	Total	0.144	0.026	46
Total	Swim	0.147	0.038	10
	Volleyball	0.185	0.031	12
	Rugby	0.146	0.025	20
	Soccer	0.153	0.026	28
	Total	0.156	0.031	70

CMJ Peak (Positive) Concentric Displacement (m)

			95% Confidence Level			
Group A	Group B	Mean Difference (A-B)	Lower Bound	Upper Bound	P	
Soccer	Rugby	-0.002	-0.022	0.019	0.997	
Swim	Rugby	-0.016	-0.043	0.010	0.384	
Volleyball	Rugby	0.005	-0.020	0.030	0.961	
Swim	Soccer	-0.015	-0.040	0.011	0.433	
Volleyball	Soccer	0.006	-0.017	0.030	0.898	
Volleyball	Swim	0.021	-0.009	0.050	0.252	
Male	Female	0.034	0.021	0.048	0.000	*

*Mean difference is significant at α=0.05 level

CMJ RSImod – Calculated with TIA

Two-Way ANOVA Results		
	Yes	No
Statistical Evidence for Interaction Effect Between Gender and Sport?		X
Statistical Significance Found between Gender groups?	X	
Statistical Significance Found between Sport groups?	X	

Tukey HSD Test
For groups with significant differences, which group has the larger value?
Soccer>Rugby Rugby>Swim Soccer>Swim Volleyball>Swim Male>Female

CMJ RSImod – Calculated with TIA

Gender	Sport	Mean	Standard Deviation	N

Male	Swim	0.236	0.046	5
	Volleyball	0.355	0.055	12
	Soccer	0.366	0.094	7
	Total	0.333	0.082	24
Female	Swim	0.156	0.040	5
	Rugby	0.225	0.071	20
	Soccer	0.300	0.064	21
	Total	0.252	0.081	46
Total	Swim	0.196	0.058	10
	Volleyball	0.355	0.055	12
	Rugby	0.225	0.071	20
	Soccer	0.316	0.077	28
	Total	0.280	0.090	70

CMJ RSImod – Calculated with TIA

			95% Confidence Level			
Group A	**Group B**	**Mean Difference (A-B)**	**Lower Bound**	**Upper Bound**	**P**	
Soccer	Rugby	0.070	0.019	0.121	0.003	*
Swim	Rugby	-0.071	-0.138	-0.003	0.037	*
Volleyball	Rugby	0.048	-0.016	0.112	0.204	
Swim	Soccer	-0.141	-0.205	-0.077	0.000	*
Volleyball	Soccer	-0.023	-0.083	0.038	0.756	
Volleyball	Swim	0.118	0.044	0.193	0.000	*
Male	Female	0.082	0.049	0.115	0.000	*

*Mean difference is significant at α=0.05 level

CMJ RSImod – Calculated with IMM

Two-Way ANOVA Results		
	Yes	No
Statistical Evidence for Interaction Effect Between Gender and Sport?		X
Statistical Significance Found between Gender groups?	X	
Statistical Significance Found between Sport groups?	X	

Tukey HSD Test
For groups with significant differences, which group has the larger value?
Soccer>Rugby Rugby>Swim Soccer>Swim Volleyball>Swim Male>Female

CMJ RSImod – Calculated with IMM

Gender	**Sport**	**Mean**	**Standard Deviation**	**N**
Male	Swim	0.224	0.033	5
	Volleyball	0.338	0.057	12
	Soccer	0.349	0.090	7
	Total	0.318	0.079	24
Female	Swim	0.131	0.039	5
	Rugby	0.221	0.079	20
	Soccer	0.289	0.063	21
	Total	0.242	0.085	46
Total	Swim	0.177	0.060	10
	Volleyball	0.338	0.057	12
	Rugby	0.221	0.079	20
	Soccer	0.304	0.074	28
	Total	0.268	0.090	70

CMJ RSImod – Calculated with IMM

95% Confidence Level

Group A	Group B	Mean Difference (A-B)	Lower Bound	Upper Bound	P	
Soccer	Rugby	0.064	0.012	0.116	0.011	*
Swim	Rugby	-0.082	-0.151	-0.013	0.014	*
Volleyball	Rugby	0.042	-0.024	0.107	0.343	
Swim	Soccer	-0.146	-0.211	-0.080	0.000	*
Volleyball	Soccer	-0.022	-0.084	0.039	0.775	
Volleyball	Swim	0.123	0.047	0.200	0.000	*
Male	Female	0.075	0.041	0.109	0.000	*

*Mean difference is significant at α=0.05 level

Appendix F. Visual Basic Code for Formatting Data (for CMJ and NCMJ)

A code was written in visual basic to convert the data collected by the force plates in .csv format to a matrix format in Microsoft Excel to be inserted into the Mathematica code. This was used for both the CMJ study and the NCMJ study. The code is as follows:

```
Sub datafiltering()

Sheet1.Name = "Sheet1"

Dim ws As Worksheet

Set ws = ThisWorkbook.Sheets.Add(After:= _
        ThisWorkbook.Sheets(ThisWorkbook.Sheets.Count))
   ws.Name = "Sheet2"


Handler:
Sheets("Sheet1").Range("A:A,F:F,L:L,N:N,S:S,Y:Y,AA:AA,AF:AF,AL:AL,AN:AN,AS:AS,AY:AY,
BA:BA,BF:BF,BL:BL").Copy Destination:=Sheets("Sheet2").Columns()

Sheets("Sheet2").Activate

Cells(1, 17) = "R1L"
Cells(1, 18) = "R1R"
Cells(1, 20) = "R2L"
Cells(1, 21) = "R2R"
Cells(1, 23) = "R3L"
Cells(1, 24) = "R3R"
Cells(1, 26) = "R4L"
Cells(1, 27) = "R4R"
Cells(1, 29) = "R5L"
Cells(1, 30) = "R5R"


a = ""
Dim sht As Worksheet
Dim LastRow As Long
Set sht = ThisWorkbook.Worksheets("Sheet2")
LastRow = sht.Cells(sht.Rows.Count, "A").End(xlUp).Row

RunIndex = 1
For j = 1 To 15
```

```
    'If we're at time
    If RunIndex = 1 Then
        TimeIndex = j
    End If

    'If we're at left or right
    If (RunIndex = 2) Or (RunIndex = 3) Then
        Cells(2, j) = "Left"
        For i = 3 To LastRow
            If i = 3 Then
                a = a & "{" & Cells(i, TimeIndex) & "," & Cells(i, j) & "}"
            Else
                If i < 2003 Then
                    a = a & "," & "{" & Cells(i, TimeIndex) & "," & Cells(i, j) & "}"
                End If
                If i > 2003 Then
                    a = a & "," & "{" & Cells(i, TimeIndex) & "," & Cells(i, j) & "}"
                End If
            End If

            If i = 2003 Then
                a = a & "," & "{" & Cells(i, TimeIndex) & "," & Cells(i, j) & "}"
                Cells(3, j + 15) = "{" & a
                a = ""
            End If

        Next i

        Cells(4, j + 15) = a & "}"

        a = ""
        End If
    If (RunIndex = 3) Then
        Cells(2, j) = "Right"
        RunIndex = 0
        End If
    RunIndex = RunIndex + 1
Next j


'Making plots
Range("A:A,B:B,C:C").Select
    Range("C1").Activate
    ActiveSheet.Shapes.AddChart2(240, xlXYScatterSmoothNoMarkers).Select
    ActiveChart.SetSourceData Source:=Range( _
        "Sheet2!$A:$A,Sheet2!$B:$B,Sheet2!$C:$C")
    ActiveChart.ChartTitle.Select
    ActiveChart.ChartTitle.Text = "Run #1"
    Selection.Format.TextFrame2.TextRange.Characters.Text = "Run #1"
    With Selection.Format.TextFrame2.TextRange.Characters(1, 6).ParagraphFormat
        .TextDirection = msoTextDirectionLeftToRight
        .Alignment = msoAlignCenter
    End With
    With Selection.Format.TextFrame2.TextRange.Characters(1, 6).Font
        .BaselineOffset = 0
```

```
    .Bold = msoFalse
    .NameComplexScript = "+mn-cs"
    .NameFarEast = "+mn-ea"
    .Fill.Visible = msoTrue
    .Fill.ForeColor.RGB = RGB(89, 89, 89)
    .Fill.Transparency = 0
    .Fill.Solid
    .Size = 14
    .Italic = msoFalse
    .Kerning = 12
    .Name = "+mn-lt"
    .UnderlineStyle = msoNoUnderline
    .Spacing = 0
    .Strike = msoNoStrike
End With


ActiveChart.ChartArea.Select
Range("D:D,E:E,F:F").Select
Range("F1").Activate
ActiveSheet.Shapes.AddChart2(240, xlXYScatterSmoothNoMarkers).Select
ActiveChart.SetSourceData Source:=Range( _
    "Sheet2!$D:$D,Sheet2!$E:$E,Sheet2!$F:$F")
ActiveChart.ChartTitle.Select
ActiveChart.ChartTitle.Text = "Run #2"
Selection.Format.TextFrame2.TextRange.Characters.Text = "Run #2"
With Selection.Format.TextFrame2.TextRange.Characters(1, 6).ParagraphFormat
    .TextDirection = msoTextDirectionLeftToRight
    .Alignment = msoAlignCenter
End With
With Selection.Format.TextFrame2.TextRange.Characters(1, 6).Font
    .BaselineOffset = 0
    .Bold = msoFalse
    .NameComplexScript = "+mn-cs"
    .NameFarEast = "+mn-ea"
    .Fill.Visible = msoTrue
    .Fill.ForeColor.RGB = RGB(89, 89, 89)
    .Fill.Transparency = 0
    .Fill.Solid
    .Size = 14
    .Italic = msoFalse
    .Kerning = 12
    .Name = "+mn-lt"
    .UnderlineStyle = msoNoUnderline
    .Spacing = 0
    .Strike = msoNoStrike
End With



Range("G:G,H:H,I:I").Select
Range("I1").Activate
ActiveSheet.Shapes.AddChart2(240, xlXYScatterSmoothNoMarkers).Select
ActiveChart.SetSourceData Source:=Range( _
    "Sheet2!$G:$G,Sheet2!$H:$H,Sheet2!$I:$I")
```

```
ActiveWindow.SmallScroll Down:=3
ActiveChart.ChartTitle.Select
ActiveChart.ChartTitle.Text = "Run #3"
Selection.Format.TextFrame2.TextRange.Characters.Text = "Run #3"
With Selection.Format.TextFrame2.TextRange.Characters(1, 6).ParagraphFormat
   .TextDirection = msoTextDirectionLeftToRight
   .Alignment = msoAlignCenter
End With
With Selection.Format.TextFrame2.TextRange.Characters(1, 6).Font
   .BaselineOffset = 0
   .Bold = msoFalse
   .NameComplexScript = "+mn-cs"
   .NameFarEast = "+mn-ea"
   .Fill.Visible = msoTrue
   .Fill.ForeColor.RGB = RGB(89, 89, 89)
   .Fill.Transparency = 0
   .Fill.Solid
   .Size = 14
   .Italic = msoFalse
   .Kerning = 12
   .Name = "+mn-lt"
   .UnderlineStyle = msoNoUnderline
   .Spacing = 0
   .Strike = msoNoStrike
End With


Range("J:J,K:K,L:L").Select
Range("L1").Activate
ActiveSheet.Shapes.AddChart2(240, xlXYScatterSmoothNoMarkers).Select
ActiveChart.SetSourceData Source:=Range( _
   "Sheet2!$J:$J,Sheet2!$K:$K,Sheet2!$L:$L")
ActiveChart.ChartTitle.Select
ActiveChart.ChartTitle.Text = "Run #4"
Selection.Format.TextFrame2.TextRange.Characters.Text = "Run #4"
With Selection.Format.TextFrame2.TextRange.Characters(1, 6).ParagraphFormat
   .TextDirection = msoTextDirectionLeftToRight
   .Alignment = msoAlignCenter
End With
With Selection.Format.TextFrame2.TextRange.Characters(1, 6).Font
   .BaselineOffset = 0
   .Bold = msoFalse
   .NameComplexScript = "+mn-cs"
   .NameFarEast = "+mn-ea"
   .Fill.Visible = msoTrue
   .Fill.ForeColor.RGB = RGB(89, 89, 89)
   .Fill.Transparency = 0
   .Fill.Solid
   .Size = 14
   .Italic = msoFalse
   .Kerning = 12
   .Name = "+mn-lt"
   .UnderlineStyle = msoNoUnderline
   .Spacing = 0
   .Strike = msoNoStrike
```

```
    End With


    Range("M:M,N:N,O:O").Select
    Range("O1").Activate
    ActiveSheet.Shapes.AddChart2(240, xlXYScatterSmoothNoMarkers).Select
    ActiveChart.SetSourceData Source:=Range( _
        "Sheet2!$M:$M,Sheet2!$N:$N,Sheet2!$O:$O")
    ActiveChart.ChartTitle.Select
    ActiveChart.ChartTitle.Text = "Run #5"
    Selection.Format.TextFrame2.TextRange.Characters.Text = "Run #5"
    With Selection.Format.TextFrame2.TextRange.Characters(1, 6).ParagraphFormat
        .TextDirection = msoTextDirectionLeftToRight
        .Alignment = msoAlignCenter
    End With
    With Selection.Format.TextFrame2.TextRange.Characters(1, 6).Font
        .BaselineOffset = 0
        .Bold = msoFalse
        .NameComplexScript = "+mn-cs"
        .NameFarEast = "+mn-ea"
        .Fill.Visible = msoTrue
        .Fill.ForeColor.RGB = RGB(89, 89, 89)
        .Fill.Transparency = 0
        .Fill.Solid
        .Size = 14
        .Italic = msoFalse
        .Kerning = 12
        .Name = "+mn-lt"
        .UnderlineStyle = msoNoUnderline
        .Spacing = 0
        .Strike = msoNoStrike
    End With


'moveplots


    ActiveChart.ChartArea.Select
    ActiveSheet.Shapes("Chart 5").IncrementLeft 441
    ActiveSheet.Shapes("Chart 5").IncrementTop -6.75
    ActiveChart.ChartArea.Select
    ActiveSheet.Shapes("Chart 4").IncrementLeft 355.5
    ActiveSheet.Shapes("Chart 4").IncrementTop 182.25
    ActiveChart.ChartArea.Select
    ActiveSheet.Shapes("Chart 3").IncrementLeft -324
    ActiveSheet.Shapes("Chart 3").IncrementTop 13.5
    ActiveChart.ChartArea.Select
    ActiveSheet.Shapes("Chart 2").IncrementLeft -8.25
    ActiveSheet.Shapes("Chart 2").IncrementTop 219
    ActiveSheet.ChartObjects("Chart 3").Activate
    ActiveSheet.Shapes("Chart 3").IncrementLeft 314.25
    ActiveSheet.Shapes("Chart 3").IncrementTop 426
    ActiveSheet.ChartObjects("Chart 4").Activate
    ActiveSheet.Shapes("Chart 4").IncrementLeft -365.25
    ActiveSheet.Shapes("Chart 4").IncrementTop 482.25
```

```
  ActiveWindow.LargeScroll Down:=-1
  ActiveSheet.ChartObjects("Chart 5").Activate
  ActiveSheet.Shapes("Chart 5").IncrementLeft -453
  ActiveSheet.Shapes("Chart 5").IncrementTop 889.5

End Sub
```

Appendix G. CMJ Mathematica Code

The Mathematica code used to analyze the counter movement jump trials in this study is described here. The force-time data must be entered into the code as a matrix (obtained using the visual basic code described in Appendix F) into the Mathematica code line as, for example, “DataRun1Li=” for the matrix corresponding to Run 1 Left Leg. The horizontal lines indicate a separation of cells in Mathematica, such that each sectioned cell can be ran separately and will store the desired variables, and the last cell pulls the desired variables from each individual cell to output it in an organized manner. The cells are organized in order of Run 1 Left Leg (“Run1L”), Run 1 Right Leg (“Run1R”), Run 1 Both Legs (“Run1Both”), Run 2 Left Leg (“Run2L”), Run 2 Right Leg (“Run2R”)… etc. through to Run 5 Both Legs (“Run5Both”). These cells are identical, except for the jump trial upon which it is running, and the run number prefix for the variables. The last cell combines and outputs the desired variables for all of these combinations., and the code for the final cell that compiles all the variables, is as follows:

```
(*time point definitions:
t1 = onset of movement;
t2 = peak negative COM velocity (also force = force from body weight only;
t3 = zero COM velocity;
t4 = takeoff; t5 = landing; tmax = time at peak force;
eccentric phase = t2-t3; concentric phase = t3-t4;
unloading phase = t1-t2; loading phase = t2-tmax; final/takeoff phase = tmax-t4;*)
```

```
(* Run1L *)
  DataRun1Li =};
DataRun1Li = DeleteCases[DataRun1Li, y_ /; y[[1]] > 2];
Clear[x, a, b, c, d]
DataRun1L = Table[DataRun1Li[[4 i]], {i, 1, Length[DataRun1Li]/4}];
f = DataRun1L[[All, 2]];
f = GaussianFilter[f, 10];
Datarefined =
  Table[{DataRun1L[[i, 1]], f[[i]]}, {i, 1, Length[DataRun1L]}];
funcRun1L =
  Interpolation[Datarefined, Method -> "Spline",
   InterpolationOrder -> 3];
funcRun1Lp = D[funcRun1L[x], x];
LL = Length[DataRun1L];
lastTimeRun1L = DataRun1L[[LL]][[1]];

Spline3[DataRun1L_] := (k = Length[DataRun1L];
  atable = Table[Subscript[a, i], {i, 1, k - 3}];
  btable = Table[Subscript[b, i], {i, 1, k - 3}];
  ctable = Table[Subscript[c, i], {i, 1, k - 3}];
  dtable = Table[Subscript[d, i], {i, 1, k - 3}];
  Poly = Table[
    atable[[i]] + btable[[i]]*x + ctable[[i]]*x^2 +
     dtable[[i]]*x^3, {i, 1, k - 3}];
  Polyd =
   Table[btable[[i]] + 2*ctable[[i]]*x + 3*dtable[[i]]*x^2, {i, 1,
     k - 3}];
  intermediate = Table[DataRun1L[[i + 2]], {i, 1, k - 4}];
  Equations = Table[0, {i, 1, 4 (k - 3)}];
  Do[Equations[[i]] = (Poly[[i]] /. x -> DataRun1L[[i + 2, 1]]) ==
     DataRun1L[[i + 2, 2]], {i, 1, k - 4}];
  Do[Equations[[
    i + k - 4]] = (Poly[[i + 1]] /. x -> DataRun1L[[i + 2, 1]]) ==
     DataRun1L[[i + 2, 2]], {i, 1, k - 4}];
  Equations[[
```

```
    2 (k - 4) + 1]] = (Poly[[1]] /. x -> DataRun1L[[1, 1]]) ==
    DataRun1L[[1, 2]];
  Equations[[
    2 (k - 4) + 2]] = (Poly[[1]] /. x -> DataRun1L[[2, 1]]) ==
    DataRun1L[[2, 2]];
  Equations[[
    2 (k - 4) + 3]] = (Poly[[k - 3]] /. x -> DataRun1L[[k - 1, 1]]) ==
     DataRun1L[[k - 1, 2]];
  Equations[[
    2 (k - 4) + 4]] = (Poly[[k - 3]] /. x -> DataRun1L[[k, 1]]) ==
    DataRun1L[[k, 2]];
  Do[Equations[[
     2 (k - 3) + 2 +
      i]] = (D[Poly[[i]], x] /.
       x -> intermediate[[i, 1]]) == (D[Poly[[i + 1]], x] /.
       x -> intermediate[[i, 1]]), {i, 1, k - 4}];
  Do[Equations[[
     2 (k - 3) + 2 + k - 4 +
      i]] = (D[Poly[[i]], {x, 2}] /.
       x -> intermediate[[i, 1]]) == (D[Poly[[i + 1]], {x, 2}] /.
       x -> intermediate[[i, 1]]), {i, 1, k - 4}];
  un = Flatten[{atable, btable, ctable, dtable}];
  f = Solve[Equations];
  Poly = Poly /. f[[1]];
  Polyd = Polyd /. f[[1]];
  xs = intermediate;
  xs = Prepend[Append[xs, DataRun1L[[k]]], DataRun1L[[1]]];
  pf = Table[{Poly[[i]], xs[[i, 1]] <= x <= xs[[i + 1, 1]]}, {i, 1,
     k - 3}];
  pdf = Table[{Polyd[[i]], xs[[i, 1]] <= x <= xs[[i + 1, 1]]}, {i, 1,
     k - 3}];
  sol = {Piecewise[pf], Piecewise[pdf]})
Timing[y = Spline3[Datarefined]];

(*Max force and tmax it occurs*)

globalmintimeRun1L = ArgMin[funcRun1L[x], 0 < x < lastTimeRun1L, x];
globalminRun1L = MinValue[funcRun1L[x], 0 < x < lastTimeRun1L, x];
peakMaxRun1L = MaxValue[funcRun1L[x], 0 < x < globalmintimeRun1L, x];
tmaxRun1L = ArgMax[funcRun1L[x], 0 < x < globalmintimeRun1L, x];
(* t4Run1L *)

t4Run1L = ArgMin[funcRun1L[x], tmaxRun1L < x < lastTimeRun1L, x];
While[funcRun1L[t4Run1L] <= Max[0, (1 + globalminRun1L)],
  t4Run1L = ArgMin[funcRun1L[x], tmaxRun1L < x < (t4Run1L - 0.01), x]];
(* t5Run1L *)

LocalMaxx5 =
  ArgMax[{funcRun1L[x], t4Run1L + 0.2 <= x <= lastTimeRun1L - 0.1}, x];
LocalMax5 = funcRun1L[LocalMaxx5];
mlist = Solve[
   y[[2]] == funcRun1L[0]*0.05 && x > t4Run1L + 0.2 &&
    x < LocalMaxx5 && funcRun1L[x] < LocalMax5*0.1, x];
minlist = x /. mlist;
t5Run1L = Last[minlist];

noforceRun1L =
  NIntegrate[
    funcRun1L[
     x], {x, (t4Run1L + 0.05), (t5Run1L - 0.05)}]/((t5Run1L -
       0.05) - (t4Run1L + 0.05));

(* t1Run1L *)

earlyGlobalMin = ArgMin[funcRun1L[x], 0.1 < x < tmaxRun1L, x];
slope = (funcRun1L[0] - funcRun1L[earlyGlobalMin])/(0 -
      earlyGlobalMin) + 25;
```

```
(* t1list is list of times where derivative = defined threshold*)

t1list = Solve[y[[2]] == slope, x];
v = t1list[[1]][[1]];
t1Run1L = (x /. v);

(*find quiet stance mass by taking average from t=0 to just before \
the jump "starts", but only if it appears to be relatively steady*)

BWforceRun1L =
  NIntegrate[
    funcRun1L[x], {x, 0.05, (t1Run1L - 0.1)}]/((t1Run1L - 0.1) - 0.05);
If[(funcRun1L[0] - funcRun1L[(t1Run1L - 0.1)])/(0 - (t1Run1L - 0.1)) >
    40, (Print["->Issue with quiet stance mass"])];
If[t1Run1L <= 0.15, Print["Check mass calc."]];
massRun1L = BWforceRun1L/9.81;
If[Abs[noforceRun1L] > 0.05*BWforceRun1L,
  Print["Possible error: force while subject is in air 5% body \
weight"]];

(*acceleration*)

accelRun1L[x_] := (funcRun1L[x] - BWforceRun1L)/massRun1L;
(*velocity curve*)

velocityRun1L =
  NDSolveValue[{accelRun1L[x] == f2'[x], f2[t1Run1L] == 0},
   f2[x], {x, 0, lastTimeRun1L}];
(* t2Run1L *)

t2Run1L = ArgMin[velocityRun1L, t1Run1L < x < tmaxRun1L, x];
(*displacement curve*)

displacementRun1L =
  NDSolveValue[{accelRun1L[x] == g2"[x],
    g2[t1Run1L] == g2'[t1Run1L] == 0}, g2[x], {x, 0, lastTimeRun1L}];
(* t3Run1L *)

t3Run1L = ArgMin[displacementRun1L, t2Run1L < x < t4Run1L, x];
(*power*)
powerRun1L = funcRun1L[x]*velocityRun1L;

(*outputs*)
force1Run1L = funcRun1L[t1Run1L];
velocity1Run1L = velocityRun1L /. x -> t1Run1L;
displacement1Run1L = displacementRun1L /. x -> t1Run1L;
power1Run1L = powerRun1L /. x -> t1Run1L;
force2Run1L = funcRun1L[t2Run1L];
velocity2Run1L = velocityRun1L /. x -> t2Run1L;
displacement2Run1L = displacementRun1L /. x -> t2Run1L;
power2Run1L = powerRun1L /. x -> t2Run1L;
force3Run1L = funcRun1L[t3Run1L];
velocity3Run1L = velocityRun1L /. x -> t3Run1L;
displacement3Run1L = displacementRun1L /. x -> t3Run1L;
power3Run1L = powerRun1L /. x -> t3Run1L;
force4Run1L = funcRun1L[t4Run1L];
velocity4Run1L = velocityRun1L /. x -> t4Run1L;
displacement4Run1L = displacementRun1L /. x -> t4Run1L;
power4Run1L = powerRun1L /. x -> t4Run1L;
force5Run1L = funcRun1L[t5Run1L];
velocity5Run1L = velocityRun1L /. x -> t5Run1L;
displacement5Run1L = displacementRun1L /. x -> t5Run1L;
power5Run1L = powerRun1L /. x -> t5Run1L;
forcemaxRun1L = funcRun1L[tmaxRun1L];
velocitymaxRun1L = velocityRun1L /. x -> tmaxRun1L;
displacementmaxRun1L = displacementRun1L /. x -> tmaxRun1L;
powermaxRun1L = powerRun1L /. x -> tmaxRun1L;
```

```
impulse12Run1L = NIntegrate[funcRun1L[x], {x, t1Run1L, t2Run1L}];
impulse23Run1L = NIntegrate[funcRun1L[x], {x, t2Run1L, t3Run1L}];
impulse34Run1L = NIntegrate[funcRun1L[x], {x, t3Run1L, t4Run1L}];
impulse45Run1L = NIntegrate[funcRun1L[x], {x, t4Run1L, t5Run1L}];
impulse2maxRun1L = NIntegrate[funcRun1L[x], {x, t2Run1L, tmaxRun1L}];

maxforce12Run1L = MaxValue[{funcRun1L[x], t1Run1L <= x <= t2Run1L}, x];
minforce12Run1L = MinValue[{funcRun1L[x], t1Run1L <= x <= t2Run1L}, x];
maxvelocity12Run1L =
  MaxValue[{velocityRun1L, t1Run1L <= x <= t2Run1L}, x];
minvelocity12Run1L =
  MinValue[{velocityRun1L, t1Run1L <= x <= t2Run1L}, x];
maxdisplacement12Run1L =
  MaxValue[{displacementRun1L, t1Run1L <= x <= t2Run1L}, x];
mindisplacement12Run1L =
  MinValue[{displacementRun1L, t1Run1L <= x <= t2Run1L}, x];
maxpower12Run1L = MaxValue[{powerRun1L, t1Run1L <= x <= t2Run1L}, x];
minpower12Run1L = MinValue[{powerRun1L, t1Run1L <= x <= t2Run1L}, x];

maxforce23Run1L = MaxValue[{funcRun1L[x], t2Run1L <= x <= t3Run1L}, x];
minforce23Run1L = MinValue[{funcRun1L[x], t2Run1L <= x <= t3Run1L}, x];
maxvelocity23Run1L =
  MaxValue[{velocityRun1L, t2Run1L <= x <= t3Run1L}, x];
minvelocity23Run1L =
  MinValue[{velocityRun1L, t2Run1L <= x <= t3Run1L}, x];
maxdisplacement23Run1L =
  MaxValue[{displacementRun1L, t2Run1L <= x <= t3Run1L}, x];
mindisplacement23Run1L =
  MinValue[{displacementRun1L, t2Run1L <= x <= t3Run1L}, x];
maxpower23Run1L = MaxValue[{powerRun1L, t2Run1L <= x <= t3Run1L}, x];
minpower23Run1L = MinValue[{powerRun1L, t2Run1L <= x <= t3Run1L}, x];

maxforce34Run1L = MaxValue[{funcRun1L[x], t3Run1L <= x <= t4Run1L}, x];
minforce34Run1L = MinValue[{funcRun1L[x], t3Run1L <= x <= t4Run1L}, x];
maxvelocity34Run1L =
  MaxValue[{velocityRun1L, t3Run1L <= x <= t4Run1L}, x];
minvelocity34Run1L =
  MinValue[{velocityRun1L, t3Run1L <= x <= t4Run1L}, x];
maxdisplacement34Run1L =
  MaxValue[{displacementRun1L, t3Run1L <= x <= t4Run1L}, x];
mindisplacement34Run1L =
  MinValue[{displacementRun1L, t3Run1L <= x <= t4Run1L}, x];
maxpower34Run1L = MaxValue[{powerRun1L, t3Run1L <= x <= t4Run1L}, x];
minpower34Run1L = MinValue[{powerRun1L, t3Run1L <= x <= t4Run1L}, x];

maxforce45Run1L = MaxValue[{funcRun1L[x], t4Run1L <= x <= t5Run1L}, x];
minforce45Run1L = MinValue[{funcRun1L[x], t4Run1L <= x <= t5Run1L}, x];
maxvelocity45Run1L =
  MaxValue[{velocityRun1L, t4Run1L <= x <= t5Run1L}, x];
minvelocity45Run1L =
  MinValue[{velocityRun1L, t4Run1L <= x <= t5Run1L}, x];
maxdisplacement45Run1L =
  MaxValue[{displacementRun1L, t4Run1L <= x <= t5Run1L}, x];
mindisplacement45Run1L =
  MinValue[{displacementRun1L, t4Run1L <= x <= t5Run1L}, x];
maxpower45Run1L = MaxValue[{powerRun1L, t4Run1L <= x <= t5Run1L}, x];
minpower45Run1L = MinValue[{powerRun1L, t4Run1L <= x <= t5Run1L}, x];

maxforce2maxRun1L =
  MaxValue[{funcRun1L[x], t2Run1L <= x <= tmaxRun1L}, x];
minforce2maxRun1L =
  MinValue[{funcRun1L[x], t2Run1L <= x <= tmaxRun1L}, x];
maxvelocity2maxRun1L =
  MaxValue[{velocityRun1L, t2Run1L <= x <= tmaxRun1L}, x];
minvelocity2maxRun1L =
  MinValue[{velocityRun1L, t2Run1L <= x <= tmaxRun1L}, x];
```

```
maxdisplacement2maxRun1L =
  MaxValue[{displacementRun1L, t2Run1L <= x <= tmaxRun1L}, x];
mindisplacement2maxRun1L =
  MinValue[{displacementRun1L, t2Run1L <= x <= tmaxRun1L}, x];
maxpower2maxRun1L =
  MaxValue[{powerRun1L, t2Run1L <= x <= tmaxRun1L}, x];
minpower2maxRun1L =
  MinValue[{powerRun1L, t2Run1L <= x <= tmaxRun1L}, x];

(*TableRun1L={{"mass= ",massRun1L},{},
{"t1= ",t1Run1L,"force @ t1= ",force1Run1L,"veloc. @ t1= \
",velocity1Run1L,"displace. @ t1= ",displacement1Run1L,"power @ t1= \
",power1Run1L},
{"t2= ",t2Run1L,"force @ t2= ",force2Run1L,"veloc. @ t2= \
",velocity2Run1L,"displace. @ t2= ",displacement2Run1L,"power @ t2= \
",power2Run1L},
{"t3= ",t3Run1L,"force @ t3= ",force3Run1L,"veloc. @ t3= \
",velocity3Run1L,"displace. @ t3= ",displacement3Run1L,"power @ t3= \
",power3Run1L},
{"t4= ",t4Run1L,"force @ t4= ",force4Run1L,"veloc. @ t4= \
",velocity4Run1L,"displace. @ t4= ",displacement4Run1L,"power @ t4= \
",power4Run1L},
{"t5= ",t5Run1L,"force @ t5= ",force5Run1L,"veloc. @ t5= \
",velocity5Run1L,"displace. @ t5= ",displacement5Run1L,"power @ t5= \
",power5Run1L},
{"tmax= ",tmaxRun1L,"force @ tmax= ",forcemaxRun1L,"veloc. @ tmax= \
",velocitymaxRun1L,"displace. @ tmax= ",displacementmaxRun1L,"power @ \
tmax= ",powermaxRun1L},{},
{"impulse t1-t2= ",impulse12Run1L,"impulse t2-t3= \
",impulse23Run1L,"impulse t3-t4= ",impulse34Run1L,"impulse t4-t5= \
",impulse45Run1L,"impulse t2-tmax= ",impulse2maxRun1L},
{"max force t1-t2= ",maxforce12Run1L,"max force t2-t3= \
",maxforce23Run1L,"max force t3-t4= ",maxforce34Run1L,"max force \
t4-t5= ",maxforce45Run1L,"max force t2-tmax= ",maxforce2maxRun1L},
{"min force t1-t2= ",minforce12Run1L,"min force t2-t3= \
",minforce23Run1L,"min force t3-t4= ",minforce34Run1L,"min force \
t4-t5= ",minforce45Run1L,"min force t2-tmax= \
",minforce2maxRun1L},{"max veloc. t1-t2= ",maxvelocity12Run1L,"max \
veloc. t2-t3= ",maxvelocity23Run1L,"max veloc. t3-t4= \
",maxvelocity34Run1L,"max veloc. t4-t5= ",maxvelocity45Run1L,"max \
veloc. t2-tmax= ",maxvelocity2maxRun1L},
{"min veloc. t1-t2= ",minvelocity12Run1L,"min veloc. t2-t3= \
",minvelocity23Run1L,"min veloc. t3-t4= ",minvelocity34Run1L,"min \
veloc. t4-t5= ",minvelocity45Run1L,"min veloc. t2-tmax= \
",minvelocity2maxRun1L},{"max displace. t1-t2= \
",maxdisplacement12Run1L,"max displace. t2-t3= \
",maxdisplacement23Run1L,"max displace. t3-t4= \
",maxdisplacement34Run1L,"max displace. t4-t5= \
",maxdisplacement45Run1L,"max displace. t2-tmax= \
",maxdisplacement2maxRun1L},
{"min displace. t1-t2= ",mindisplacement12Run1L,"min displace. t2-t3= \
",mindisplacement23Run1L,"min displace. t3-t4= \
",mindisplacement34Run1L,"min displace. t4-t5= \
",mindisplacement45Run1L,"min displace. t2-tmax= \
",mindisplacement2maxRun1L},{"max power t1-t2= ",maxpower12Run1L,"max \
power t2-t3= ",maxpower23Run1L,"max power t3-t4= \
",maxpower34Run1L,"max power t4-t5= ",maxpower45Run1L,"max power \
t2-tmax= ",maxpower2maxRun1L},
{"min power t1-t2= ",minpower12Run1L,"min power t2-t3= \
",minpower23Run1L,"min power t3-t4= ",minpower34Run1L,"min power \
t4-t5= ",minpower45Run1L,"min power t2-tmax= ",minpower2maxRun1L}} ;
Grid[TableRun1L,Frame\[Rule]All]*)

(*classify the type of CMJ based on number of peaks--look at the \
number of maxima/minima of the first derivative*)
Clear[findAllRoots]
SyntaxInformation[
```

```
   findAllRoots] = {"LocalVariables" -> {"Plot", {2, 2}},
   "ArgumentsPattern" -> {_, _, OptionsPattern[]}};
SetAttributes[findAllRoots, HoldAll];

Options[findAllRoots] =
  Join[{"ShowPlot" -> False, PlotRange -> All},
   FilterRules[Options[Plot], Except[PlotRange]]];

findAllRoots[fn_, {l_, lmin_, lmax_}, opts : OptionsPattern[]] :=
 Module[{pl, p, x, localFunction, brackets},
  localFunction = ReleaseHold[Hold[fn] /. HoldPattern[l] :> x];
  If[lmin != lmax,
   pl = Plot[localFunction, {x, lmin, lmax},
     Evaluate@
      FilterRules[Join[{opts}, Options[findAllRoots]], Options[Plot]]];
   p = Cases[pl, Line[{x__}] :> x, Infinity];
   If[OptionValue["ShowPlot"],
    Print[Show[pl, PlotLabel -> "Finding roots for this function",
      ImageSize -> 200, BaseStyle -> {FontSize -> 8}]]], p = {}];
  brackets =
   Map[First,
    Select[(*This Split trick pretends that two points on the curve \
are "equal" if the function values have _opposite _ sign.Pairs of \
such sign-changes form the brackets for the subsequent FindRoot*)
     Split[p, Sign[Last[#2]] == -Sign[Last[#1]] &],
     Length[#1] == 2 &], {2}];
  x /. Apply[FindRoot[localFunction == 0, {x, ##1}] &,
    brackets, {1}] /. x -> {}]
rootfuncRun1L[t_] := funcRun1L"[x] /. x -> t;
rootlist =
  findAllRoots[rootfuncRun1L[x], {x, t3Run1L, t4Run1L},
   "ShowPlot" -> False];
localrootsRun1L = Length[rootlist];
If[localrootsRun1L == 1, {runtypeRun1L = 1,
   Print["CMJ Type 1 (one peak)"]}, {runtypeRun1L = 2,
   Print["CMJ Type 2 (two or more peaks)"]}];

line1Run1L = Line[{{t1Run1L, -5000}, {t1Run1L, 5000}}];
line2Run1L = Line[{{t2Run1L, -5000}, {t2Run1L, 5000}}];
line3Run1L = Line[{{t3Run1L, -5000}, {t3Run1L, 5000}}];
line4Run1L = Line[{{t4Run1L, -5000}, {t4Run1L, 5000}}];
line5Run1L = Line[{{t5Run1L, -5000}, {t5Run1L, 5000}}];
linemaxRun1L = Line[{{tmaxRun1L, -5000}, {tmaxRun1L, 5000}}];
lineStyle = {Thin, Red, Dashed};
bodyweightlineRun1L =
  Line[{{0, BWforceRun1L}, {t1Run1L, BWforceRun1L}}];
Print["Body Weight= ", BWforceRun1L]
Plot[{funcRun1L[x]}, {x, 0, lastTimeRun1L}, ImageSize -> 600,
 PlotStyle -> {Automatic},
 Epilog -> {Directive[lineStyle], line1Run1L, line2Run1L, line3Run1L,
   line4Run1L, line5Run1L, Directive[Thin, Blue, Dashed],
   linemaxRun1L, Directive[Thin, Black, Dashed],
   bodyweightlineRun1L}, PlotRange -> All,
 BaseStyle -> { FontSize -> 12}, PlotLabel -> "CMJ - Normal Force",
 AxesLabel -> {"Time (s)", "Normal Force (N)"} ]
Plot[{velocityRun1L}, {x, 0, lastTimeRun1L}, ImageSize -> 600,
 PlotStyle -> {Automatic},
 Epilog -> {Directive[lineStyle], line1Run1L, line2Run1L, line3Run1L,
   line4Run1L, line5Run1L, Directive[Thin, Blue, Dashed],
   linemaxRun1L}, PlotRange -> All, BaseStyle -> { FontSize -> 12},
 PlotLabel -> "CMJ - COM Velocity",
 AxesLabel -> {"Time (s)", "Normal Force (N)"} ]
(*Plot[{displacementRun1L},{x,0,lastTimeRun1L},ImageSize\[Rule]600,\
PlotStyle \[Rule] \
{Automatic},Epilog\[Rule]{Directive[lineStyle],line1Run1L,line2Run1L,\
line3Run1L,line4Run1L,line5Run1L, \
Directive[Thin,Blue,Dashed],linemaxRun1L},PlotRange\[Rule]All, \
```

```
BaseStyle\[Rule]{ FontSize\[Rule] 12},PlotLabel\[Rule] "CMJ - COM \
Displacement", AxesLabel\[Rule]{"Time (s)", "Normal Force (N)"} ]
Plot[{powerRun1L},{x,0,lastTimeRun1L},ImageSize\[Rule]600,PlotStyle \
\[Rule] {Automatic},Epilog\[Rule]{Directive[lineStyle],line1Run1L,\
line2Run1L,line3Run1L,line4Run1L,line5Run1L, \
Directive[Thin,Blue,Dashed],linemaxRun1L},PlotRange\[Rule]All, \
BaseStyle\[Rule]{ FontSize\[Rule] 12},PlotLabel\[Rule] "CMJ - Power", \
AxesLabel\[Rule]{"Time (s)", "Normal Force (N)"} ]*)
```

```
(* Run1R *)
  DataRun1Ri =};
DataRun1Ri = DeleteCases[DataRun1Ri, y_ /; y[[1]] > 2];
Clear[x, a, b, c, d]
DataRun1R = Table[DataRun1Ri[[4 i]], {i, 1, Length[DataRun1Ri]/4}];
f = DataRun1R[[All, 2]];
f = GaussianFilter[f, 10];
Datarefined =
  Table[{DataRun1R[[i, 1]], f[[i]]}, {i, 1, Length[DataRun1R]}];
funcRun1R =
  Interpolation[Datarefined, Method -> "Spline",
   InterpolationOrder -> 3];
funcRun1Rp = D[funcRun1R[x], x];
LL = Length[DataRun1R];
lastTimeRun1R = DataRun1R[[LL]][[1]];

Spline3[DataRun1R_] := (k = Length[DataRun1R];
  atable = Table[Subscript[a, i], {i, 1, k - 3}];
  btable = Table[Subscript[b, i], {i, 1, k - 3}];
  ctable = Table[Subscript[c, i], {i, 1, k - 3}];
  dtable = Table[Subscript[d, i], {i, 1, k - 3}];
  Poly = Table[
    atable[[i]] + btable[[i]]*x + ctable[[i]]*x^2 +
     dtable[[i]]*x^3, {i, 1, k - 3}];
  Polyd =
   Table[btable[[i]] + 2*ctable[[i]]*x + 3*dtable[[i]]*x^2, {i, 1,
     k - 3}];
  intermediate = Table[DataRun1R[[i + 2]], {i, 1, k - 4}];
  Equations = Table[0, {i, 1, 4 (k - 3)}];
  Do[Equations[[i]] = (Poly[[i]] /. x -> DataRun1R[[i + 2, 1]]) ==
     DataRun1R[[i + 2, 2]], {i, 1, k - 4}];
  Do[Equations[[
     i + k - 4]] = (Poly[[i + 1]] /. x -> DataRun1R[[i + 2, 1]]) ==
     DataRun1R[[i + 2, 2]], {i, 1, k - 4}];
  Equations[[
    2 (k - 4) + 1]] = (Poly[[1]] /. x -> DataRun1R[[1, 1]]) ==
    DataRun1R[[1, 2]];
  Equations[[
    2 (k - 4) + 2]] = (Poly[[1]] /. x -> DataRun1R[[2, 1]]) ==
    DataRun1R[[2, 2]];
  Equations[[
    2 (k - 4) + 3]] = (Poly[[k - 3]] /. x -> DataRun1R[[k - 1, 1]]) ==
     DataRun1R[[k - 1, 2]];
  Equations[[
    2 (k - 4) + 4]] = (Poly[[k - 3]] /. x -> DataRun1R[[k, 1]]) ==
    DataRun1R[[k, 2]];
  Do[Equations[[
     2 (k - 3) + 2 +
      i]] = (D[Poly[[i]], x] /.
       x -> intermediate[[i, 1]]) == (D[Poly[[i + 1]], x] /.
       x -> intermediate[[i, 1]]), {i, 1, k - 4}];
  Do[Equations[[
     2 (k - 3) + 2 + k - 4 +
      i]] = (D[Poly[[i]], {x, 2}] /.
       x -> intermediate[[i, 1]]) == (D[Poly[[i + 1]], {x, 2}] /.
       x -> intermediate[[i, 1]]), {i, 1, k - 4}];
  un = Flatten[{atable, btable, ctable, dtable}];
  f = Solve[Equations];
  Poly = Poly /. f[[1]];
```

```
  Polyd = Polyd /. f[[1]];
  xs = intermediate;
  xs = Prepend[Append[xs, DataRun1R[[k]]], DataRun1R[[1]]];
  pf = Table[{Poly[[i]], xs[[i, 1]] <= x <= xs[[i + 1, 1]]}, {i, 1,
     k - 3}];
  pdf = Table[{Polyd[[i]], xs[[i, 1]] <= x <= xs[[i + 1, 1]]}, {i, 1,
     k - 3}];
  sol = {Piecewise[pf], Piecewise[pdf]})
Timing[y = Spline3[Datarefined]];

(*Max force and tmax it occurs*)

globalmintimeRun1R = ArgMin[funcRun1R[x], 0 < x < lastTimeRun1R, x];
globalminRun1R = MinValue[funcRun1R[x], 0 < x < lastTimeRun1R, x];
peakMaxRun1R = MaxValue[funcRun1R[x], 0 < x < globalmintimeRun1R, x];
tmaxRun1R = ArgMax[funcRun1R[x], 0 < x < globalmintimeRun1R, x];
(* t4Run1R *)

t4Run1R = ArgMin[funcRun1R[x], tmaxRun1R < x < lastTimeRun1R, x];
While[funcRun1R[t4Run1R] <= Max[0, (1 + globalminRun1R)],
  t4Run1R = ArgMin[funcRun1R[x], tmaxRun1R < x < (t4Run1R - 0.01), x]];
(* t5Run1R *)

LocalMaxx5 =
  ArgMax[{funcRun1R[x], t4Run1R + 0.2 <= x <= lastTimeRun1R - 0.1}, x];
LocalMax5 = funcRun1R[LocalMaxx5];
mlist = Solve[
   y[[2]] == funcRun1R[0]*0.05 && x > t4Run1R + 0.2 &&
    x < LocalMaxx5 && funcRun1R[x] < LocalMax5*0.1, x];
minlist = x /. mlist;
t5Run1R = Last[minlist];

noforceRun1R =
  NIntegrate[
    funcRun1R[
     x], {x, (t4Run1R + 0.05), (t5Run1R - 0.05)}]/((t5Run1R -
      0.05) - (t4Run1R + 0.05));

(* t1Run1R *)

earlyGlobalMin = ArgMin[funcRun1R[x], 0.1 < x < tmaxRun1R, x];
slope = (funcRun1R[0] - funcRun1R[earlyGlobalMin])/(0 -
      earlyGlobalMin) + 25;
(* t1list is list of times where derivative = defined threshold*)

t1list = Solve[y[[2]] == slope, x];
v = t1list[[1]][[1]];
t1Run1R = (x /. v);

(*find quiet stance mass by taking average from t=0 to just before \
the jump "starts", but only if it appears to be relatively steady*)

BWforceRun1R =
  NIntegrate[
    funcRun1R[x], {x, 0.05, (t1Run1R - 0.1)}]/((t1Run1R - 0.1) - 0.05);
If[(funcRun1R[0] - funcRun1R[(t1Run1R - 0.1)])/(0 - (t1Run1R - 0.1)) >
    40, (Print["->Issue with quiet stance mass"])];
If[t1Run1R <= 0.15, Print["Check mass calc."]];
massRun1R = BWforceRun1R/9.81;
If[Abs[noforceRun1R] > 0.05*BWforceRun1R,
  Print["Possible error: force while subject is in air >5% body \
weight"]];

(*acceleration*)

accelRun1R[x_] := (funcRun1R[x] - BWforceRun1R)/massRun1R;
(*velocity curve*)
```

```
velocityRun1R =
  NDSolveValue[{accelRun1R[x] == f2'[x], f2[t1Run1R] == 0},
   f2[x], {x, 0, lastTimeRun1R}];
(* t2Run1R *)

t2Run1R = ArgMin[velocityRun1R, t1Run1R < x < tmaxRun1R, x];
(*displacement curve*)

displacementRun1R =
  NDSolveValue[{accelRun1R[x] == g2''[x],
    g2[t1Run1R] == g2'[t1Run1R] == 0}, g2[x], {x, 0, lastTimeRun1R}];
(* t3Run1R *)

t3Run1R = ArgMin[displacementRun1R, t2Run1R < x < t4Run1R, x];
(*power*)
powerRun1R = funcRun1R[x]*velocityRun1R;

(*outputs*)
force1Run1R = funcRun1R[t1Run1R];
velocity1Run1R = velocityRun1R /. x -> t1Run1R;
displacement1Run1R = displacementRun1R /. x -> t1Run1R;
power1Run1R = powerRun1R /. x -> t1Run1R;
force2Run1R = funcRun1R[t2Run1R];
velocity2Run1R = velocityRun1R /. x -> t2Run1R;
displacement2Run1R = displacementRun1R /. x -> t2Run1R;
power2Run1R = powerRun1R /. x -> t2Run1R;
force3Run1R = funcRun1R[t3Run1R];
velocity3Run1R = velocityRun1R /. x -> t3Run1R;
displacement3Run1R = displacementRun1R /. x -> t3Run1R;
power3Run1R = powerRun1R /. x -> t3Run1R;
force4Run1R = funcRun1R[t4Run1R];
velocity4Run1R = velocityRun1R /. x -> t4Run1R;
displacement4Run1R = displacementRun1R /. x -> t4Run1R;
power4Run1R = powerRun1R /. x -> t4Run1R;
force5Run1R = funcRun1R[t5Run1R];
velocity5Run1R = velocityRun1R /. x -> t5Run1R;
displacement5Run1R = displacementRun1R /. x -> t5Run1R;
power5Run1R = powerRun1R /. x -> t5Run1R;
forcemaxRun1R = funcRun1R[tmaxRun1R];
velocitymaxRun1R = velocityRun1R /. x -> tmaxRun1R;
displacementmaxRun1R = displacementRun1R /. x -> tmaxRun1R;
powermaxRun1R = powerRun1R /. x -> tmaxRun1R;

impulse12Run1R = NIntegrate[funcRun1R[x], {x, t1Run1R, t2Run1R}];
impulse23Run1R = NIntegrate[funcRun1R[x], {x, t2Run1R, t3Run1R}];
impulse34Run1R = NIntegrate[funcRun1R[x], {x, t3Run1R, t4Run1R}];
impulse45Run1R = NIntegrate[funcRun1R[x], {x, t4Run1R, t5Run1R}];
impulse2maxRun1R = NIntegrate[funcRun1R[x], {x, t2Run1R, tmaxRun1R}];

maxforce12Run1R = MaxValue[{funcRun1R[x], t1Run1R <= x <= t2Run1R}, x];
minforce12Run1R = MinValue[{funcRun1R[x], t1Run1R <= x <= t2Run1R}, x];
maxvelocity12Run1R =
  MaxValue[{velocityRun1R, t1Run1R <= x <= t2Run1R}, x];
minvelocity12Run1R =
  MinValue[{velocityRun1R, t1Run1R <= x <= t2Run1R}, x];
maxdisplacement12Run1R =
  MaxValue[{displacementRun1R, t1Run1R <= x <= t2Run1R}, x];
mindisplacement12Run1R =
  MinValue[{displacementRun1R, t1Run1R <= x <= t2Run1R}, x];
maxpower12Run1R = MaxValue[{powerRun1R, t1Run1R <= x <= t2Run1R}, x];
minpower12Run1R = MinValue[{powerRun1R, t1Run1R <= x <= t2Run1R}, x];

maxforce23Run1R = MaxValue[{funcRun1R[x], t2Run1R <= x <= t3Run1R}, x];
minforce23Run1R = MinValue[{funcRun1R[x], t2Run1R <= x <= t3Run1R}, x];
maxvelocity23Run1R =
  MaxValue[{velocityRun1R, t2Run1R <= x <= t3Run1R}, x];
```

```
minvelocity23Run1R =
  MinValue[{velocityRun1R, t2Run1R <= x <= t3Run1R}, x];
maxdisplacement23Run1R =
  MaxValue[{displacementRun1R, t2Run1R <= x <= t3Run1R}, x];
mindisplacement23Run1R =
  MinValue[{displacementRun1R, t2Run1R <= x <= t3Run1R}, x];
maxpower23Run1R = MaxValue[{powerRun1R, t2Run1R <= x <= t3Run1R}, x];
minpower23Run1R = MinValue[{powerRun1R, t2Run1R <= x <= t3Run1R}, x];

maxforce34Run1R = MaxValue[{funcRun1R[x], t3Run1R <= x <= t4Run1R}, x];
minforce34Run1R = MinValue[{funcRun1R[x], t3Run1R <= x <= t4Run1R}, x];
maxvelocity34Run1R =
  MaxValue[{velocityRun1R, t3Run1R <= x <= t4Run1R}, x];
minvelocity34Run1R =
  MinValue[{velocityRun1R, t3Run1R <= x <= t4Run1R}, x];
maxdisplacement34Run1R =
  MaxValue[{displacementRun1R, t3Run1R <= x <= t4Run1R}, x];
mindisplacement34Run1R =
  MinValue[{displacementRun1R, t3Run1R <= x <= t4Run1R}, x];
maxpower34Run1R = MaxValue[{powerRun1R, t3Run1R <= x <= t4Run1R}, x];
minpower34Run1R = MinValue[{powerRun1R, t3Run1R <= x <= t4Run1R}, x];

maxforce45Run1R = MaxValue[{funcRun1R[x], t4Run1R <= x <= t5Run1R}, x];
minforce45Run1R = MinValue[{funcRun1R[x], t4Run1R <= x <= t5Run1R}, x];
maxvelocity45Run1R =
  MaxValue[{velocityRun1R, t4Run1R <= x <= t5Run1R}, x];
minvelocity45Run1R =
  MinValue[{velocityRun1R, t4Run1R <= x <= t5Run1R}, x];
maxdisplacement45Run1R =
  MaxValue[{displacementRun1R, t4Run1R <= x <= t5Run1R}, x];
mindisplacement45Run1R =
  MinValue[{displacementRun1R, t4Run1R <= x <= t5Run1R}, x];
maxpower45Run1R = MaxValue[{powerRun1R, t4Run1R <= x <= t5Run1R}, x];
minpower45Run1R = MinValue[{powerRun1R, t4Run1R <= x <= t5Run1R}, x];

maxforce2maxRun1R =
  MaxValue[{funcRun1R[x], t2Run1R <= x <= tmaxRun1R}, x];
minforce2maxRun1R =
  MinValue[{funcRun1R[x], t2Run1R <= x <= tmaxRun1R}, x];
maxvelocity2maxRun1R =
  MaxValue[{velocityRun1R, t2Run1R <= x <= tmaxRun1R}, x];
minvelocity2maxRun1R =
  MinValue[{velocityRun1R, t2Run1R <= x <= tmaxRun1R}, x];
maxdisplacement2maxRun1R =
  MaxValue[{displacementRun1R, t2Run1R <= x <= tmaxRun1R}, x];
mindisplacement2maxRun1R =
  MinValue[{displacementRun1R, t2Run1R <= x <= tmaxRun1R}, x];
maxpower2maxRun1R =
  MaxValue[{powerRun1R, t2Run1R <= x <= tmaxRun1R}, x];
minpower2maxRun1R =
  MinValue[{powerRun1R, t2Run1R <= x <= tmaxRun1R}, x];

(*TableRun1R={{"mass= ",massRun1R},{},
{"t1= ",t1Run1R,"force @ t1= ",force1Run1R,"veloc. @ t1= \
",velocity1Run1R,"displace. @ t1= ",displacement1Run1R,"power @ t1= \
",power1Run1R},
{"t2= ",t2Run1R,"force @ t2= ",force2Run1R,"veloc. @ t2= \
",velocity2Run1R,"displace. @ t2= ",displacement2Run1R,"power @ t2= \
",power2Run1R},
{"t3= ",t3Run1R,"force @ t3= ",force3Run1R,"veloc. @ t3= \
",velocity3Run1R,"displace. @ t3= ",displacement3Run1R,"power @ t3= \
",power3Run1R},
{"t4= ",t4Run1R,"force @ t4= ",force4Run1R,"veloc. @ t4= \
",velocity4Run1R,"displace. @ t4= ",displacement4Run1R,"power @ t4= \
",power4Run1R},
{"t5= ",t5Run1R,"force @ t5= ",force5Run1R,"veloc. @ t5= \
",velocity5Run1R,"displace. @ t5= ",displacement5Run1R,"power @ t5= \
```

```
",power5Run1R},
{"tmax= ",tmaxRun1R,"force @ tmax= ",forcemaxRun1R,"veloc. @ tmax= \
",velocitymaxRun1R,"displace. @ tmax= ",displacementmaxRun1R,"power @ \
tmax= ",powermaxRun1R},{},
{"impulse t1-t2= ",impulse12Run1R,"impulse t2-t3= \
",impulse23Run1R,"impulse t3-t4= ",impulse34Run1R,"impulse t4-t5= \
",impulse45Run1R,"impulse t2-tmax= ",impulse2maxRun1R},
{"max force t1-t2= ",maxforce12Run1R,"max force t2-t3= \
",maxforce23Run1R,"max force t3-t4= ",maxforce34Run1R,"max force \
t4-t5= ",maxforce45Run1R,"max force t2-tmax= ",maxforce2maxRun1R},
{"min force t1-t2= ",minforce12Run1R,"min force t2-t3= \
",minforce23Run1R,"min force t3-t4= ",minforce34Run1R,"min force \
t4-t5= ",minforce45Run1R,"min force t2-tmax= \
",minforce2maxRun1R},{"max veloc. t1-t2= ",maxvelocity12Run1R,"max \
veloc. t2-t3= ",maxvelocity23Run1R,"max veloc. t3-t4= \
",maxvelocity34Run1R,"max veloc. t4-t5= ",maxvelocity45Run1R,"max \
veloc. t2-tmax= ",maxvelocity2maxRun1R},
{"min veloc. t1-t2= ",minvelocity12Run1R,"min veloc. t2-t3= \
",minvelocity23Run1R,"min veloc. t3-t4= ",minvelocity34Run1R,"min \
veloc. t4-t5= ",minvelocity45Run1R,"min veloc. t2-tmax= \
",minvelocity2maxRun1R},{"max displace. t1-t2= \
",maxdisplacement12Run1R,"max displace. t2-t3= \
",maxdisplacement23Run1R,"max displace. t3-t4= \
",maxdisplacement34Run1R,"max displace. t4-t5= \
",maxdisplacement45Run1R,"max displace. t2-tmax= \
",maxdisplacement2maxRun1R},
{"min displace. t1-t2= ",mindisplacement12Run1R,"min displace. t2-t3= \
",mindisplacement23Run1R,"min displace. t3-t4= \
",mindisplacement34Run1R,"min displace. t4-t5= \
",mindisplacement45Run1R,"min displace. t2-tmax= \
",mindisplacement2maxRun1R},{"max power t1-t2= ",maxpower12Run1R,"max \
power t2-t3= ",maxpower23Run1R,"max power t3-t4= \
",maxpower34Run1R,"max power t4-t5= ",maxpower45Run1R,"max power \
t2-tmax= ",maxpower2maxRun1R},
{"min power t1-t2= ",minpower12Run1R,"min power t2-t3= \
",minpower23Run1R,"min power t3-t4= ",minpower34Run1R,"min power \
t4-t5= ",minpower45Run1R,"min power t2-tmax= ",minpower2maxRun1R}} ;
Grid[TableRun1R,Frame\[Rule]All]*)

(*classify the type of CMJ based on number of peaks--look at the \
number of maxima/minima of the first derivative*)
Clear[findAllRoots]
SyntaxInformation[
   findAllRoots] = {"LocalVariables" -> {"Plot", {2, 2}},
   "ArgumentsPattern" -> {_, _, OptionsPattern[]}};
SetAttributes[findAllRoots, HoldAll];

Options[findAllRoots] =
  Join[{"ShowPlot" -> False, PlotRange -> All},
   FilterRules[Options[Plot], Except[PlotRange]]];

findAllRoots[fn_, {l_, lmin_, lmax_}, opts : OptionsPattern[]] :=
 Module[{pl, p, x, localFunction, brackets},
  localFunction = ReleaseHold[Hold[fn] /. HoldPattern[l] :> x];
  If[lmin != lmax,
   pl = Plot[localFunction, {x, lmin, lmax},
     Evaluate@
      FilterRules[Join[{opts}, Options[findAllRoots]], Options[Plot]]];
   p = Cases[pl, Line[{x__}] :> x, Infinity];
   If[OptionValue["ShowPlot"],
    Print[Show[pl, PlotLabel -> "Finding roots for this function",
      ImageSize -> 200, BaseStyle -> {FontSize -> 8}]]], p = {}];
  brackets =
   Map[First,
    Select[(*This Split trick pretends that two points on the curve \
are "equal" if the function values have _opposite _ sign.Pairs of \
such sign-changes form the brackets for the subsequent FindRoot*)
```

```
      Split[p, Sign[Last[#2]] == -Sign[Last[#1]] &],
      Length[#1] == 2 &], {2}];
  x /. Apply[FindRoot[localFunction == 0, {x, ##1}] &,
      brackets, {1}] /. x -> {}]
rootfuncRun1R[t_] := funcRun1R''[x] /. x -> t;
rootlist =
  findAllRoots[rootfuncRun1R[x], {x, t3Run1R, t4Run1R},
   "ShowPlot" -> False];
localrootsRun1R = Length[rootlist];
If[localrootsRun1R == 1, {runtypeRun1R = 1,
   Print["CMJ Type 1 (one peak)"]}, {runtypeRun1R = 2,
   Print["CMJ Type 2 (two or more peaks)"]}];

line1Run1R = Line[{{t1Run1R, -5000}, {t1Run1R, 5000}}];
line2Run1R = Line[{{t2Run1R, -5000}, {t2Run1R, 5000}}];
line3Run1R = Line[{{t3Run1R, -5000}, {t3Run1R, 5000}}];
line4Run1R = Line[{{t4Run1R, -5000}, {t4Run1R, 5000}}];
line5Run1R = Line[{{t5Run1R, -5000}, {t5Run1R, 5000}}];
linemaxRun1R = Line[{{tmaxRun1R, -5000}, {tmaxRun1R, 5000}}];
lineStyle = {Thin, Red, Dashed};
bodyweightlineRun1R =
  Line[{{0, BWforceRun1R}, {t1Run1R, BWforceRun1R}}];
Print["Body Weight= ", BWforceRun1R]
Plot[{funcRun1R[x]}, {x, 0, lastTimeRun1R}, ImageSize -> 600,
 PlotStyle -> {Automatic},
 Epilog -> {Directive[lineStyle], line1Run1R, line2Run1R, line3Run1R,
   line4Run1R, line5Run1R, Directive[Thin, Blue, Dashed],
   linemaxRun1R , Directive[Thin, Black, Dashed],
   bodyweightlineRun1R}, PlotRange -> All,
 BaseStyle -> { FontSize -> 12}, PlotLabel -> "CMJ - Normal Force",
 AxesLabel -> {"Time (s)", "Normal Force (N)"} ]
Plot[{velocityRun1R}, {x, 0, lastTimeRun1R}, ImageSize -> 600,
 PlotStyle -> {Automatic},
 Epilog -> {Directive[lineStyle], line1Run1R, line2Run1R, line3Run1R,
   line4Run1R, line5Run1R, Directive[Thin, Blue, Dashed],
   linemaxRun1R}, PlotRange -> All, BaseStyle -> { FontSize -> 12},
 PlotLabel -> "CMJ - COM Velocity",
 AxesLabel -> {"Time (s)", "Normal Force (N)"} ]
(*Plot[{displacementRun1R},{x,0,lastTimeRun1R},ImageSize\[Rule]600,\
PlotStyle \[Rule] \
{Automatic},Epilog\[Rule]{Directive[lineStyle],line1Run1R,line2Run1R,\
line3Run1R,line4Run1R,line5Run1R, \
Directive[Thin,Blue,Dashed],linemaxRun1R},PlotRange\[Rule]All, \
BaseStyle\[Rule]{ FontSize\[Rule] 12},PlotLabel\[Rule] "CMJ - COM \
Displacement", AxesLabel\[Rule]{"Time (s)", "Normal Force (N)"} ]
Plot[{powerRun1R},{x,0,lastTimeRun1R},ImageSize\[Rule]600,PlotStyle \
\[Rule] {Automatic},Epilog\[Rule]{Directive[lineStyle],line1Run1R,\
line2Run1R,line3Run1R,line4Run1R,line5Run1R, \
Directive[Thin,Blue,Dashed],linemaxRun1R},PlotRange\[Rule]All, \
BaseStyle\[Rule]{ FontSize\[Rule] 12},PlotLabel\[Rule] "CMJ - Power", \
AxesLabel\[Rule]{"Time (s)", "Normal Force (N)"} ]*)
```

```
(* Run1Both *)
DataRun1Both = DataRun1Li;
DataRun1Both =
  Insert[DataRun1Both // Transpose, DataRun1Ri[[All, 2]], 3] //
   Transpose;
DataRun1Both[[All, 2]] += DataRun1Both[[All, 3]];
DataRun1Both[[All, ;; 2]];
DataRun1Both =
  Table[DataRun1Both[[4 i]], {i, 1, Length[DataRun1Both]/4}];

f = DataRun1Both[[All, 2]];
f = GaussianFilter[f, 10];
Datarefined =
  Table[{DataRun1Both[[i, 1]], f[[i]]}, {i, 1,
    Length[DataRun1Both]}];
funcRun1Both =
```

```
  Interpolation[Datarefined, Method -> "Spline",
   InterpolationOrder -> 3];
funcRun1Bothp = D[funcRun1Both[x], x];
LL = Length[DataRun1Both];
lastTimeRun1Both = DataRun1Both[[LL]][[1]];

Spline3[DataRun1Both_] := (k = Length[DataRun1Both];
  atable = Table[Subscript[a, i], {i, 1, k - 3}];
  btable = Table[Subscript[b, i], {i, 1, k - 3}];
  ctable = Table[Subscript[c, i], {i, 1, k - 3}];
  dtable = Table[Subscript[d, i], {i, 1, k - 3}];
  Poly = Table[
    atable[[i]] + btable[[i]]*x + ctable[[i]]*x^2 +
     dtable[[i]]*x^3, {i, 1, k - 3}];
  Polyd =
   Table[btable[[i]] + 2*ctable[[i]]*x + 3*dtable[[i]]*x^2, {i, 1,
     k - 3}];
  intermediate = Table[DataRun1Both[[i + 2]], {i, 1, k - 4}];
  Equations = Table[0, {i, 1, 4 (k - 3)}];
  Do[Equations[[i]] = (Poly[[i]] /. x -> DataRun1Both[[i + 2, 1]]) ==
     DataRun1Both[[i + 2, 2]], {i, 1, k - 4}];
  Do[Equations[[
     i + k - 4]] = (Poly[[i + 1]] /. x -> DataRun1Both[[i + 2, 1]]) ==
      DataRun1Both[[i + 2, 2]], {i, 1, k - 4}];
  Equations[[
    2 (k - 4) + 1]] = (Poly[[1]] /. x -> DataRun1Both[[1, 1]]) ==
    DataRun1Both[[1, 2]];
  Equations[[
    2 (k - 4) + 2]] = (Poly[[1]] /. x -> DataRun1Both[[2, 1]]) ==
    DataRun1Both[[2, 2]];
  Equations[[
    2 (k - 4) +
     3]] = (Poly[[k - 3]] /. x -> DataRun1Both[[k - 1, 1]]) ==
    DataRun1Both[[k - 1, 2]];
  Equations[[
    2 (k - 4) + 4]] = (Poly[[k - 3]] /. x -> DataRun1Both[[k, 1]]) ==
    DataRun1Both[[k, 2]];
  Do[Equations[[
     2 (k - 3) + 2 +
      i]] = (D[Poly[[i]], x] /.
       x -> intermediate[[i, 1]]) == (D[Poly[[i + 1]], x] /.
       x -> intermediate[[i, 1]]), {i, 1, k - 4}];
  Do[Equations[[
     2 (k - 3) + 2 + k - 4 +
      i]] = (D[Poly[[i]], {x, 2}] /.
       x -> intermediate[[i, 1]]) == (D[Poly[[i + 1]], {x, 2}] /.
       x -> intermediate[[i, 1]]), {i, 1, k - 4}];
  un = Flatten[{atable, btable, ctable, dtable}];
  f = Solve[Equations];
  Poly = Poly /. f[[1]];
  Polyd = Polyd /. f[[1]];
  xs = intermediate;
  xs = Prepend[Append[xs, DataRun1Both[[k]]], DataRun1Both[[1]]];
  pf = Table[{Poly[[i]], xs[[i, 1]] <= x <= xs[[i + 1, 1]]}, {i, 1,
     k - 3}];
  pdf = Table[{Polyd[[i]], xs[[i, 1]] <= x <= xs[[i + 1, 1]]}, {i, 1,
     k - 3}];
  sol = {Piecewise[pf], Piecewise[pdf]})
Timing[y = Spline3[Datarefined]];

(*Max force and tmax it occurs*)

globalmintimeRun1Both =
  ArgMin[funcRun1Both[x], 0 < x < lastTimeRun1Both, x];
globalminRun1Both =
  MinValue[funcRun1Both[x], 0 < x < lastTimeRun1Both, x];
peakMaxRun1Both =
```

```
  MaxValue[funcRun1Both[x], 0 < x < globalmintimeRun1Both, x];
tmaxRun1Both =
  ArgMax[funcRun1Both[x], 0 < x < globalmintimeRun1Both, x];
(* t4Run1Both *)

t4Run1Both =
  ArgMin[funcRun1Both[x], tmaxRun1Both < x < lastTimeRun1Both, x];
While[funcRun1Both[t4Run1Both] <= Max[0, (1 + globalminRun1Both)],
  t4Run1Both =
   ArgMin[funcRun1Both[x], tmaxRun1Both < x < (t4Run1Both - 0.01), x]];
(* t5Run1Both *)

LocalMaxx5 =
  ArgMax[{funcRun1Both[x],
    t4Run1Both + 0.2 <= x <= lastTimeRun1Both - 0.1}, x];
LocalMax5 = funcRun1Both[LocalMaxx5];
mlist = Solve[
   y[[2]] == funcRun1Both[0]*0.05 && x > t4Run1Both + 0.2 &&
    x < LocalMaxx5 && funcRun1Both[x] < LocalMax5*0.1, x];
minlist = x /. mlist;
t5Run1Both = Last[minlist];

noforceRun1Both =
  NIntegrate[
    funcRun1Both[
     x], {x, (t4Run1Both + 0.05), (t5Run1Both -
       0.05)}]/((t5Run1Both - 0.05) - (t4Run1Both + 0.05));

(* t1Run1Both *)

earlyGlobalMin = ArgMin[funcRun1Both[x], 0.1 < x < tmaxRun1Both, x];
slope = (funcRun1Both[0] - funcRun1Both[earlyGlobalMin])/(0 -
      earlyGlobalMin) + 25;
(* t1list is list of times where derivative = defined threshold*)

t1list = Solve[y[[2]] == slope, x];
v = t1list[[1]][[1]];
t1Run1Both = (x /. v);

(*find quiet stance mass by taking average from t=0 to just before \
the jump "starts", but only if it appears to be relatively steady*)

BWforceRun1Both =
  NIntegrate[
    funcRun1Both[x], {x,
     0.05, (t1Run1Both - 0.1)}]/((t1Run1Both - 0.1) - 0.05);
If[(funcRun1Both[0] -
      funcRun1Both[(t1Run1Both - 0.1)])/(0 - (t1Run1Both - 0.1)) >
   40, (Print["->Issue with quiet stance mass"])];
If[t1Run1Both <= 0.15, Print["Check mass calc."]];
massRun1Both = BWforceRun1Both/9.81;
If[Abs[noforceRun1Both] > 0.05*BWforceRun1Both,
  Print["Possible error: force while subject is in air > 5% body \
weight"]];

(*acceleration*)

accelRun1Both[x_] := (funcRun1Both[x] - BWforceRun1Both)/
   massRun1Both;
(*velocity curve*)

velocityRun1Both =
  NDSolveValue[{accelRun1Both[x] == f2'[x], f2[t1Run1Both] == 0},
   f2[x], {x, 0, lastTimeRun1Both}];
(* t2Run1Both *)

t2Run1Both =
```

```
  ArgMin[velocityRun1Both, t1Run1Both < x < tmaxRun1Both, x];
(*displacement curve*)

displacementRun1Both =
  NDSolveValue[{accelRun1Both[x] == g2''[x],
    g2[t1Run1Both] == g2'[t1Run1Both] == 0},
   g2[x], {x, 0, lastTimeRun1Both}];
(* t3Run1Both *)

t3Run1Both =
  ArgMin[displacementRun1Both, t2Run1Both < x < t4Run1Both, x];
(*power*)
powerRun1Both = funcRun1Both[x]*velocityRun1Both;

(*outputs*)
force1Run1Both = funcRun1Both[t1Run1Both];
velocity1Run1Both = velocityRun1Both /. x -> t1Run1Both;
displacement1Run1Both = displacementRun1Both /. x -> t1Run1Both;
power1Run1Both = powerRun1Both /. x -> t1Run1Both;
force2Run1Both = funcRun1Both[t2Run1Both];
velocity2Run1Both = velocityRun1Both /. x -> t2Run1Both;
displacement2Run1Both = displacementRun1Both /. x -> t2Run1Both;
power2Run1Both = powerRun1Both /. x -> t2Run1Both;
force3Run1Both = funcRun1Both[t3Run1Both];
velocity3Run1Both = velocityRun1Both /. x -> t3Run1Both;
displacement3Run1Both = displacementRun1Both /. x -> t3Run1Both;
power3Run1Both = powerRun1Both /. x -> t3Run1Both;
force4Run1Both = funcRun1Both[t4Run1Both];
velocity4Run1Both = velocityRun1Both /. x -> t4Run1Both;
displacement4Run1Both = displacementRun1Both /. x -> t4Run1Both;
power4Run1Both = powerRun1Both /. x -> t4Run1Both;
force5Run1Both = funcRun1Both[t5Run1Both];
velocity5Run1Both = velocityRun1Both /. x -> t5Run1Both;
displacement5Run1Both = displacementRun1Both /. x -> t5Run1Both;
power5Run1Both = powerRun1Both /. x -> t5Run1Both;
forcemaxRun1Both = funcRun1Both[tmaxRun1Both];
velocitymaxRun1Both = velocityRun1Both /. x -> tmaxRun1Both;
displacementmaxRun1Both = displacementRun1Both /. x -> tmaxRun1Both;
powermaxRun1Both = powerRun1Both /. x -> tmaxRun1Both;

impulse12Run1Both =
  NIntegrate[funcRun1Both[x], {x, t1Run1Both, t2Run1Both}];
impulse23Run1Both =
  NIntegrate[funcRun1Both[x], {x, t2Run1Both, t3Run1Both}];
impulse34Run1Both =
  NIntegrate[funcRun1Both[x], {x, t3Run1Both, t4Run1Both}];
impulse45Run1Both =
  NIntegrate[funcRun1Both[x], {x, t4Run1Both, t5Run1Both}];
impulse2maxRun1Both =
  NIntegrate[funcRun1Both[x], {x, t2Run1Both, tmaxRun1Both}];

maxforce12Run1Both =
  MaxValue[{funcRun1Both[x], t1Run1Both <= x <= t2Run1Both}, x];
minforce12Run1Both =
  MinValue[{funcRun1Both[x], t1Run1Both <= x <= t2Run1Both}, x];
maxvelocity12Run1Both =
  MaxValue[{velocityRun1Both, t1Run1Both <= x <= t2Run1Both}, x];
minvelocity12Run1Both =
  MinValue[{velocityRun1Both, t1Run1Both <= x <= t2Run1Both}, x];
maxdisplacement12Run1Both =
  MaxValue[{displacementRun1Both, t1Run1Both <= x <= t2Run1Both}, x];
mindisplacement12Run1Both =
  MinValue[{displacementRun1Both, t1Run1Both <= x <= t2Run1Both}, x];
maxpower12Run1Both =
  MaxValue[{powerRun1Both, t1Run1Both <= x <= t2Run1Both}, x];
minpower12Run1Both =
  MinValue[{powerRun1Both, t1Run1Both <= x <= t2Run1Both}, x];
```

```
maxforce23Run1Both =
  MaxValue[{funcRun1Both[x], t2Run1Both <= x <= t3Run1Both}, x];
minforce23Run1Both =
  MinValue[{funcRun1Both[x], t2Run1Both <= x <= t3Run1Both}, x];
maxvelocity23Run1Both =
  MaxValue[{velocityRun1Both, t2Run1Both <= x <= t3Run1Both}, x];
minvelocity23Run1Both =
  MinValue[{velocityRun1Both, t2Run1Both <= x <= t3Run1Both}, x];
maxdisplacement23Run1Both =
  MaxValue[{displacementRun1Both, t2Run1Both <= x <= t3Run1Both}, x];
mindisplacement23Run1Both =
  MinValue[{displacementRun1Both, t2Run1Both <= x <= t3Run1Both}, x];
maxpower23Run1Both =
  MaxValue[{powerRun1Both, t2Run1Both <= x <= t3Run1Both}, x];
minpower23Run1Both =
  MinValue[{powerRun1Both, t2Run1Both <= x <= t3Run1Both}, x];

maxforce34Run1Both =
  MaxValue[{funcRun1Both[x], t3Run1Both <= x <= t4Run1Both}, x];
minforce34Run1Both =
  MinValue[{funcRun1Both[x], t3Run1Both <= x <= t4Run1Both}, x];
maxvelocity34Run1Both =
  MaxValue[{velocityRun1Both, t3Run1Both <= x <= t4Run1Both}, x];
minvelocity34Run1Both =
  MinValue[{velocityRun1Both, t3Run1Both <= x <= t4Run1Both}, x];
maxdisplacement34Run1Both =
  MaxValue[{displacementRun1Both, t3Run1Both <= x <= t4Run1Both}, x];
mindisplacement34Run1Both =
  MinValue[{displacementRun1Both, t3Run1Both <= x <= t4Run1Both}, x];
maxpower34Run1Both =
  MaxValue[{powerRun1Both, t3Run1Both <= x <= t4Run1Both}, x];
minpower34Run1Both =
  MinValue[{powerRun1Both, t3Run1Both <= x <= t4Run1Both}, x];

maxforce45Run1Both =
  MaxValue[{funcRun1Both[x], t4Run1Both <= x <= t5Run1Both}, x];
minforce45Run1Both =
  MinValue[{funcRun1Both[x], t4Run1Both <= x <= t5Run1Both}, x];
maxvelocity45Run1Both =
  MaxValue[{velocityRun1Both, t4Run1Both <= x <= t5Run1Both}, x];
minvelocity45Run1Both =
  MinValue[{velocityRun1Both, t4Run1Both <= x <= t5Run1Both}, x];
maxdisplacement45Run1Both =
  MaxValue[{displacementRun1Both, t4Run1Both <= x <= t5Run1Both}, x];
mindisplacement45Run1Both =
  MinValue[{displacementRun1Both, t4Run1Both <= x <= t5Run1Both}, x];
maxpower45Run1Both =
  MaxValue[{powerRun1Both, t4Run1Both <= x <= t5Run1Both}, x];
minpower45Run1Both =
  MinValue[{powerRun1Both, t4Run1Both <= x <= t5Run1Both}, x];

maxforce2maxRun1Both =
  MaxValue[{funcRun1Both[x], t2Run1Both <= x <= tmaxRun1Both}, x];
minforce2maxRun1Both =
  MinValue[{funcRun1Both[x], t2Run1Both <= x <= tmaxRun1Both}, x];
maxvelocity2maxRun1Both =
  MaxValue[{velocityRun1Both, t2Run1Both <= x <= tmaxRun1Both}, x];
minvelocity2maxRun1Both =
  MinValue[{velocityRun1Both, t2Run1Both <= x <= tmaxRun1Both}, x];
maxdisplacement2maxRun1Both =
  MaxValue[{displacementRun1Both, t2Run1Both <= x <= tmaxRun1Both}, x];
mindisplacement2maxRun1Both =
  MinValue[{displacementRun1Both, t2Run1Both <= x <= tmaxRun1Both}, x];
maxpower2maxRun1Both =
  MaxValue[{powerRun1Both, t2Run1Both <= x <= tmaxRun1Both}, x];
minpower2maxRun1Both =
```

```
  MinValue[{powerRun1Both, t2Run1Both <= x <= tmaxRun1Both}, x];

(*TableRun1Both={{"mass= ",massRun1Both},{},
{"t1= ",t1Run1Both,"force @ t1= ",force1Run1Both,"veloc. @ t1= \
",velocity1Run1Both,"displace. @ t1= ",displacement1Run1Both,"power @ \
t1= ",power1Run1Both},
{"t2= ",t2Run1Both,"force @ t2= ",force2Run1Both,"veloc. @ t2= \
",velocity2Run1Both,"displace. @ t2= ",displacement2Run1Both,"power @ \
t2= ",power2Run1Both},
{"t3= ",t3Run1Both,"force @ t3= ",force3Run1Both,"veloc. @ t3= \
",velocity3Run1Both,"displace. @ t3= ",displacement3Run1Both,"power @ \
t3= ",power3Run1Both},
{"t4= ",t4Run1Both,"force @ t4= ",force4Run1Both,"veloc. @ t4= \
",velocity4Run1Both,"displace. @ t4= ",displacement4Run1Both,"power @ \
t4= ",power4Run1Both},
{"t5= ",t5Run1Both,"force @ t5= ",force5Run1Both,"veloc. @ t5= \
",velocity5Run1Both,"displace. @ t5= ",displacement5Run1Both,"power @ \
t5= ",power5Run1Both},
{"tmax= ",tmaxRun1Both,"force @ tmax= ",forcemaxRun1Both,"veloc. @ \
tmax= ",velocitymaxRun1Both,"displace. @ tmax= \
",displacementmaxRun1Both,"power @ tmax= ",powermaxRun1Both},{},
{"impulse t1-t2= ",impulse12Run1Both,"impulse t2-t3= \
",impulse23Run1Both,"impulse t3-t4= ",impulse34Run1Both,"impulse \
t4-t5= ",impulse45Run1Both,"impulse t2-tmax= ",impulse2maxRun1Both},
{"max force t1-t2= ",maxforce12Run1Both,"max force t2-t3= \
",maxforce23Run1Both,"max force t3-t4= ",maxforce34Run1Both,"max \
force t4-t5= ",maxforce45Run1Both,"max force t2-tmax= \
",maxforce2maxRun1Both},
{"min force t1-t2= ",minforce12Run1Both,"min force t2-t3= \
",minforce23Run1Both,"min force t3-t4= ",minforce34Run1Both,"min \
force t4-t5= ",minforce45Run1Both,"min force t2-tmax= \
",minforce2maxRun1Both},{"max veloc. t1-t2= \
",maxvelocity12Run1Both,"max veloc. t2-t3= \
",maxvelocity23Run1Both,"max veloc. t3-t4= \
",maxvelocity34Run1Both,"max veloc. t4-t5= \
",maxvelocity45Run1Both,"max veloc. t2-tmax= \
",maxvelocity2maxRun1Both},
{"min veloc. t1-t2= ",minvelocity12Run1Both,"min veloc. t2-t3= \
",minvelocity23Run1Both,"min veloc. t3-t4= \
",minvelocity34Run1Both,"min veloc. t4-t5= \
",minvelocity45Run1Both,"min veloc. t2-tmax= \
",minvelocity2maxRun1Both},{"max displace. t1-t2= \
",maxdisplacement12Run1Both,"max displace. t2-t3= \
",maxdisplacement23Run1Both,"max displace. t3-t4= \
",maxdisplacement34Run1Both,"max displace. t4-t5= \
",maxdisplacement45Run1Both,"max displace. t2-tmax= \
",maxdisplacement2maxRun1Both},
{"min displace. t1-t2= ",mindisplacement12Run1Both,"min displace. \
t2-t3= ",mindisplacement23Run1Both,"min displace. t3-t4= \
",mindisplacement34Run1Both,"min displace. t4-t5= \
",mindisplacement45Run1Both,"min displace. t2-tmax= \
",mindisplacement2maxRun1Both},{"max power t1-t2= \
",maxpower12Run1Both,"max power t2-t3= ",maxpower23Run1Both,"max \
power t3-t4= ",maxpower34Run1Both,"max power t4-t5= \
",maxpower45Run1Both,"max power t2-tmax= ",maxpower2maxRun1Both},
{"min power t1-t2= ",minpower12Run1Both,"min power t2-t3= \
",minpower23Run1Both,"min power t3-t4= ",minpower34Run1Both,"min \
power t4-t5= ",minpower45Run1Both,"min power t2-tmax= \
",minpower2maxRun1Both}} ;
Grid[TableRun1Both,Frame\[Rule]All]*)

(*classify the type of CMJ based on number of peaks--look at the \
number of maxima/minima of the first derivative*)
Clear[findAllRoots]
SyntaxInformation[
   findAllRoots] = {"LocalVariables" -> {"Plot", {2, 2}},
   "ArgumentsPattern" -> {_, _, OptionsPattern[]}};
```

```
SetAttributes[findAllRoots, HoldAll];

Options[findAllRoots] =
  Join[{"ShowPlot" -> False, PlotRange -> All},
   FilterRules[Options[Plot], Except[PlotRange]]];

findAllRoots[fn_, {l_, lmin_, lmax_}, opts : OptionsPattern[]] :=
 Module[{pl, p, x, localFunction, brackets},
  localFunction = ReleaseHold[Hold[fn] /. HoldPattern[l] :> x];
  If[lmin != lmax,
   pl = Plot[localFunction, {x, lmin, lmax},
     Evaluate@
      FilterRules[Join[{opts}, Options[findAllRoots]], Options[Plot]]];
   p = Cases[pl, Line[{x__}] :> x, Infinity];
   If[OptionValue["ShowPlot"],
    Print[Show[pl, PlotLabel -> "Finding roots for this function",
      ImageSize -> 200, BaseStyle -> {FontSize -> 8}]]], p = {}];
  brackets =
   Map[First,
    Select[(*This Split trick pretends that two points on the curve \
are "equal" if the function values have _opposite _ sign.Pairs of \
such sign-changes form the brackets for the subsequent FindRoot*)
     Split[p, Sign[Last[#2]] == -Sign[Last[#1]] &],
     Length[#1] == 2 &], {2}];
  x /. Apply[FindRoot[localFunction == 0, {x, ##1}] &,
     brackets, {1}] /. x -> {}]
rootfuncRun1Both[t_] := funcRun1Both"[x] /. x -> t;
rootlist =
  findAllRoots[rootfuncRun1Both[x], {x, t3Run1Both, t4Run1Both},
   "ShowPlot" -> False];
localrootsRun1Both = Length[rootlist];
If[localrootsRun1Both == 1, {runtypeRun1Both = 1,
   Print["CMJ Type 1 (one peak)"]}, {runtypeRun1Both = 2,
   Print["CMJ Type 2 (two or more peaks)"]}];

line1Run1Both = Line[{{t1Run1Both, -5000}, {t1Run1Both, 5000}}];
line2Run1Both = Line[{{t2Run1Both, -5000}, {t2Run1Both, 5000}}];
line3Run1Both = Line[{{t3Run1Both, -5000}, {t3Run1Both, 5000}}];
line4Run1Both = Line[{{t4Run1Both, -5000}, {t4Run1Both, 5000}}];
line5Run1Both = Line[{{t5Run1Both, -5000}, {t5Run1Both, 5000}}];
linemaxRun1Both =
  Line[{{tmaxRun1Both, -5000}, {tmaxRun1Both, 5000}}];
bodyweightlineRun1Both =
  Line[{{0, BWforceRun1Both}, {t1Run1Both, BWforceRun1Both}}];
lineStyle = {Thin, Red, Dashed};
Print["Body Weight= ", BWforceRun1Both]
Plot[{funcRun1Both[x]}, {x, 0, lastTimeRun1Both}, ImageSize -> 600,
 PlotStyle -> {Automatic},
 Epilog -> {Directive[lineStyle], line1Run1Both, line2Run1Both,
   line3Run1Both, line4Run1Both, line5Run1Both,
   Directive[Thin, Blue, Dashed], linemaxRun1Both,
   Directive[Thin, Black, Dashed], bodyweightlineRun1Both},
 PlotRange -> All, BaseStyle -> { FontSize -> 12},
 PlotLabel -> "CMJ - Normal Force",
 AxesLabel -> {"Time (s)", "Normal Force (N)"} ]
Plot[{velocityRun1Both}, {x, 0, lastTimeRun1Both}, ImageSize -> 600,
 PlotStyle -> {Automatic},
 Epilog -> {Directive[lineStyle], line1Run1Both, line2Run1Both,
   line3Run1Both, line4Run1Both, line5Run1Both,
   Directive[Thin, Blue, Dashed], linemaxRun1Both}, PlotRange -> All,
 BaseStyle -> { FontSize -> 12}, PlotLabel -> "CMJ - COM Velocity",
 AxesLabel -> {"Time (s)", "Normal Force (N)"} ]
(*Plot[{displacementRun1Both},{x,0,lastTimeRun1Both},ImageSize\[Rule]\
600,PlotStyle \[Rule] \
{Automatic},Epilog\[Rule]{Directive[lineStyle],line1Run1Both,\
line2Run1Both,line3Run1Both,line4Run1Both,line5Run1Both, \
Directive[Thin,Blue,Dashed],linemaxRun1Both},PlotRange\[Rule]All, \
```

```
BaseStyle\[Rule]{ FontSize\[Rule] 12},PlotLabel\[Rule] "CMJ - COM \
Displacement", AxesLabel\[Rule]{"Time (s)", "Normal Force (N)"} ]
Plot[{powerRun1Both},{x,0,lastTimeRun1Both},ImageSize\[Rule]600,\
PlotStyle \[Rule] \
{Automatic},Epilog\[Rule]{Directive[lineStyle],line1Run1Both,\
line2Run1Both,line3Run1Both,line4Run1Both,line5Run1Both, \
Directive[Thin,Blue,Dashed],linemaxRun1Both},PlotRange\[Rule]All, \
BaseStyle\[Rule]{ FontSize\[Rule] 12},PlotLabel\[Rule] "CMJ - Power", \
AxesLabel\[Rule]{"Time (s)", "Normal Force (N)"} ]*)
```

(Cells for Run 2 through Run 5 inserted here, with all code that includes the naming system “Run1” in it replaced with the corresponding run number (i.e. “Run2” for the second jump trial, etc.)

```
 (*ENTER IN NAME AND DATE*)
NameID = "NAME";
DateID = "DATE";

IDRun1 = "" <> NameID <> " CMJ Run 1 " <> DateID <> "";
IDRun2 = "" <> NameID <> " CMJ Run 2 " <> DateID <> "";
IDRun3 = "" <> NameID <> " CMJ Run 3 " <> DateID <> "";
IDRun4 = "" <> NameID <> " CMJ Run 4 " <> DateID <> "";
IDRun5 = "" <> NameID <> " CMJ Run 5 " <> DateID <> "";

(* Run1 *)
solutionRun1 = {{IDRun1, "", ""}, {"", "",
    ""}, {"Variable", "Left", "Right"}, {"", "", ""},
   {"run class.:", runtypeRun1L, runtypeRun1R}, {"mass:", massRun1L,
    massRun1R}, {"t1:", t1Run1L, t1Run1R}, {"t2:", t2Run1L,
    t2Run1R}, {"t3:", t3Run1L, t3Run1R}, {"t4:", t4Run1L,
    t4Run1R}, {"t5:", t5Run1L, t5Run1R}, {"tmax:", tmaxRun1L,
    tmaxRun1R}, {"", "", ""}, {"force @ t1:", force1Run1L,
    force1Run1R}, {"force @ t2:", force2Run1L,
    force2Run1R}, {"force @ t3:", force3Run1L,
    force3Run1R}, {"force @ t4:", force4Run1L,
    force4Run1R}, {"force @ t5:", force5Run1L,
    force5Run1R}, {"force @ tmax:", forcemaxRun1L,
    forcemaxRun1R}, {"veloc. @ t1:", velocity1Run1L,
    velocity1Run1R}, {"veloc. @ t2:", velocity2Run1L,
    velocity2Run1R}, {"veloc. @ t3:", velocity3Run1L,
    velocity3Run1R}, {"veloc. @ t4:", velocity4Run1L,
    velocity4Run1R}, {"veloc. @ t5:", velocity5Run1L,
    velocity5Run1R}, {"veloc. @ tmax:", velocitymaxRun1L,
    velocitymaxRun1R}, {"displace. @ t1:", displacement1Run1L,
    displacement1Run1R}, {"displace. @ t2:", displacement2Run1L,
    displacement2Run1R}, {"displace. @ t3:", displacement3Run1L,
    displacement3Run1R}, {"displace. @ t4:", displacement4Run1L,
    displacement4Run1R}, {"displace. @ t5:", displacement5Run1L,
    displacement5Run1R}, {"displace. @ tmax:", displacementmaxRun1L,
    displacementmaxRun1R}, {"power @ t1:", power1Run1L,
    power1Run1R}, {"power @ t2:", power2Run1L,
    power2Run1R}, {"power @ t3:", power3Run1L,
    power3Run1R}, {"power @ t4:", power4Run1L,
    power4Run1R}, {"power @ t5:", power5Run1L,
    power5Run1R}, {"power @ tmax:", powermaxRun1L,
    powermaxRun1R}, {"", "", ""}, {"impulse t1-t2:", impulse12Run1L,
    impulse12Run1R}, {"impulse t2-t3:", impulse23Run1L,
    impulse23Run1R}, {"impulse t3-t4:", impulse34Run1L,
    impulse34Run1R}, {"impulse t4-t5:", impulse45Run1L,
    impulse45Run1R}, {"impulse t2-tmax:", impulse2maxRun1L,
    impulse2maxRun1R}, {"max force t1-t2:", maxforce12Run1L,
    maxforce12Run1R}, {"max force t2-t3:", maxforce23Run1L,
    maxforce23Run1R}, {"max force t3-t4:", maxforce34Run1L,
    maxforce34Run1R}, {"max force t4-t5:", maxforce45Run1L,
    maxforce45Run1R}, {"max force t2-tmax:", maxforce2maxRun1L,
    maxforce2maxRun1R}, {"min force t1-t2:", minforce12Run1L,
    minforce12Run1R}, {"min force t2-t3:", minforce23Run1L,
```

```
minforce23Run1R}, {"min force t3-t4:", minforce34Run1L,
minforce34Run1R}, {"min force t4-t5:", minforce45Run1L,
minforce45Run1R}, {"min force t2-tmax:", minforce2maxRun1L,
minforce2maxRun1R}, {"max veloc. t1-t2:", maxvelocity12Run1L,
maxvelocity12Run1R}, {"max veloc. t2-t3:", maxvelocity23Run1L,
maxvelocity23Run1R}, {"max veloc. t3-t4:", maxvelocity34Run1L,
maxvelocity34Run1R}, {"max veloc. t4-t5:", maxvelocity45Run1L,
maxvelocity45Run1R}, {"max veloc. t2-tmax:", maxvelocity2maxRun1L,
 maxvelocity2maxRun1R}, {"min veloc. t1-t2:", minvelocity12Run1L,
minvelocity12Run1R}, {"min veloc. t2-t3:", minvelocity23Run1L,
minvelocity23Run1R}, {"min veloc. t3-t4:", minvelocity34Run1L,
minvelocity34Run1R}, {"min veloc. t4-t5:", minvelocity45Run1L,
minvelocity45Run1R}, {"min veloc. t2-tmax:", minvelocity2maxRun1L,
 minvelocity2maxRun1R}, {"max displace. t1-t2:",
maxdisplacement12Run1L,
maxdisplacement12Run1R}, {"max displace. t2-t3:",
maxdisplacement23Run1L,
maxdisplacement23Run1R}, {"max displace. t3-t4:",
maxdisplacement34Run1L,
maxdisplacement34Run1R}, {"max displace. t4-t5:",
maxdisplacement45Run1L,
maxdisplacement45Run1R}, {"max displace. t2-tmax:",
maxdisplacement2maxRun1L,
maxdisplacement2maxRun1R}, {"min displace. t1-t2:",
mindisplacement12Run1L,
mindisplacement12Run1R}, {"min displace. t2-t3:",
mindisplacement23Run1L,
mindisplacement23Run1R}, {"min displace. t3-t4:",
mindisplacement34Run1L,
mindisplacement34Run1R}, {"min displace. t4-t5:",
mindisplacement45Run1L,
mindisplacement45Run1R}, {"min displace. t2-tmax:",
mindisplacement2maxRun1L,
mindisplacement2maxRun1R}, {"max power t1-t2:", maxpower12Run1L,
maxpower12Run1R}, {"max power t2-t3:", maxpower23Run1L,
maxpower23Run1R}, {"max power t3-t4:", maxpower34Run1L,
maxpower34Run1R}, {"max power t4-t5:", maxpower45Run1L,
maxpower45Run1R}, {"max power t2-tmax:", maxpower2maxRun1L,
maxpower2maxRun1R}, {"min power t1-t2:", minpower12Run1L,
minpower12Run1R}, {"min power t2-t3:", minpower23Run1L,
minpower23Run1R}, {"min power t3-t4:", minpower34Run1L,
minpower34Run1R}, {"min power t4-t5:", minpower45Run1L,
minpower45Run1R}, {"min power t2-tmax:", minpower2maxRun1L,
minpower2maxRun1R}, {"avg. force in air:", noforceRun1L,
noforceRun1R}, {"", "", ""}, {"Legs Analyzed Together:", "",
""}, {"", "", ""}, {"Variable", "Legs Combined", ""}, {"", "",
""}, {"run class.:", runtypeRun1Both, ""}, {"mass:", massRun1Both,
 ""}, {"t1:", t1Run1Both, ""}, {"t2:", t2Run1Both, ""}, {"t3:",
t3Run1Both, ""}, {"t4:", t4Run1Both, ""}, {"t5:", t5Run1Both,
""}, {"tmax:", tmaxRun1Both, ""}, {"", "", ""}, {"force @ t1:",
force1Run1Both, ""}, {"force @ t2:", force2Run1Both,
""}, {"force @ t3:", force3Run1Both, ""}, {"force @ t4:",
force4Run1Both, ""}, {"force @ t5:", force5Run1Both,
""}, {"force @ tmax:", forcemaxRun1Both, ""}, {"veloc. @ t1:",
velocity1Run1Both, ""}, {"veloc. @ t2:", velocity2Run1Both,
""}, {"veloc. @ t3:", velocity3Run1Both, ""}, {"veloc. @ t4:",
velocity4Run1Both, ""}, {"veloc. @ t5:", velocity5Run1Both,
""}, {"veloc. @ tmax:", velocitymaxRun1Both,
""}, {"displace. @ t1:", displacement1Run1Both,
""}, {"displace. @ t2:", displacement2Run1Both,
""}, {"displace. @ t3:", displacement3Run1Both,
""}, {"displace. @ t4:", displacement4Run1Both,
""}, {"displace. @ t5:", displacement5Run1Both,
""}, {"displace. @ tmax:", displacementmaxRun1Both,
""}, {"power @ t1:", power1Run1Both, ""}, {"power @ t2:",
power2Run1Both, ""}, {"power @ t3:", power3Run1Both,
""}, {"power @ t4:", power4Run1Both, ""}, {"power @ t5:",
```

```
    power5Run1Both, ""}, {"power @ tmax:", powermaxRun1Both, ""}, {"",
     "", ""}, {"impulse t1-t2:", impulse12Run1Both,
    ""}, {"impulse t2-t3:", impulse23Run1Both, ""}, {"impulse t3-t4:",
     impulse34Run1Both, ""}, {"impulse t4-t5:", impulse45Run1Both,
    ""}, {"impulse t2-tmax:", impulse2maxRun1Both,
    ""}, {"max force t1-t2:", maxforce12Run1Both,
    ""}, {"max force t2-t3:", maxforce23Run1Both,
    ""}, {"max force t3-t4:", maxforce34Run1Both,
    ""}, {"max force t4-t5:", maxforce45Run1Both,
    ""}, {"max force t2-tmax:", maxforce2maxRun1Both,
    ""}, {"min force t1-t2:", minforce12Run1Both,
    ""}, {"min force t2-t3:", minforce23Run1Both,
    ""}, {"min force t3-t4:", minforce34Run1Both,
    ""}, {"min force t4-t5:", minforce45Run1Both,
    ""}, {"min force t2-tmax:", minforce2maxRun1Both,
    ""}, {"max veloc. t1-t2:", maxvelocity12Run1Both,
    ""}, {"max veloc. t2-t3:", maxvelocity23Run1Both,
    ""}, {"max veloc. t3-t4:", maxvelocity34Run1Both,
    ""}, {"max veloc. t4-t5:", maxvelocity45Run1Both,
    ""}, {"max veloc. t2-tmax:", maxvelocity2maxRun1Both,
    ""}, {"min veloc. t1-t2:", minvelocity12Run1Both,
    ""}, {"min veloc. t2-t3:", minvelocity23Run1Both,
    ""}, {"min veloc. t3-t4:", minvelocity34Run1Both,
    ""}, {"min veloc. t4-t5:", minvelocity45Run1Both,
    ""}, {"min veloc. t2-tmax:", minvelocity2maxRun1Both,
    ""}, {"max displace. t1-t2:", maxdisplacement12Run1Both,
    ""}, {"max displace. t2-t3:", maxdisplacement23Run1Both,
    ""}, {"max displace. t3-t4:", maxdisplacement34Run1Both,
    ""}, {"max displace. t4-t5:", maxdisplacement45Run1Both,
    ""}, {"max displace. t2-tmax:", maxdisplacement2maxRun1Both,
    ""}, {"min displace. t1-t2:", mindisplacement12Run1Both,
    ""}, {"min displace. t2-t3:", mindisplacement23Run1Both,
    ""}, {"min displace. t3-t4:", mindisplacement34Run1Both,
    ""}, {"min displace. t4-t5:", mindisplacement45Run1Both,
    ""}, {"min displace. t2-tmax:", mindisplacement2maxRun1Both,
    ""}, {"max power t1-t2:", maxpower12Run1Both,
    ""}, {"max power t2-t3:", maxpower23Run1Both,
    ""}, {"max power t3-t4:", maxpower34Run1Both,
    ""}, {"max power t4-t5:", maxpower45Run1Both,
    ""}, {"max power t2-tmax:", maxpower2maxRun1Both,
    ""}, {"min power t1-t2:", minpower12Run1Both,
    ""}, {"min power t2-t3:", minpower23Run1Both,
    ""}, {"min power t3-t4:", minpower34Run1Both,
    ""}, {"min power t4-t5:", minpower45Run1Both,
    ""}, {"min power t2-tmax:", minpower2maxRun1Both,
    ""}, {"avg. force in air:", noforceRun1Both, ""}};
ForceGraphRun1Both =
 Plot[Tooltip[funcRun1Both[x]], {x, 0, lastTimeRun1Both},
  ImageSize -> 600, PlotStyle -> {Automatic},
  PlotLegends -> {"Legs Combined"},
  Epilog -> {Directive[Blue, Dashed], line1Run1Both, line2Run1Both,
    line3Run1Both, line4Run1Both, line5Run1Both,
    Directive[Blue, Dashing[None]], linemaxRun1Both},
  PlotRange -> All,
  PlotLabel -> "Counter Movement Jump - Normal Force",
  AxesLabel -> {"Time (s)", "Normal Force (N)"} ]
ForceGraphRun1 =
 Plot[Tooltip[{funcRun1L[x], funcRun1R[x]}], {x, 0, lastTimeRun1Both},
   ImageSize -> 600,
  PlotLegends -> (Placed[#, Right] & /@ {{"Left", "Right"},
     LineLegend[{{Red, Dashed}, Dashed, Black}, {"Left", "Right"}]}),
   PlotStyle -> {Automatic},
  Epilog -> {Directive[Red, Dashed], line1Run1L, line2Run1L,
    line3Run1L, line4Run1L, line5Run1L, Directive[Red, Dashing[None]],
     linemaxRun1L, Directive[Black, Dashed], line1Run1R, line2Run1R,
    line3Run1R, line4Run1R, line5Run1R,
    Directive[Black, Dashing[None]], linemaxRun1R}, PlotRange -> All,
```

```
  PlotLabel -> "Counter Movement Jump - Normal Force",
  AxesLabel -> {"Time (s)", "Normal Force (N)"} ]
VelocityGraphRun1Both =
 Plot[Tooltip[velocityRun1Both], {x, 0, lastTimeRun1Both},
  ImageSize -> 600, PlotStyle -> {Automatic},
  PlotLegends -> {"Legs Combined"},
  Epilog -> {Directive[Blue, Dashed], line1Run1Both, line2Run1Both,
    line3Run1Both, line4Run1Both, line5Run1Both,
    Directive[Blue, Dashing[None]], linemaxRun1Both},
  PlotRange -> All, PlotLabel -> "Counter Movement Jump - Velocity",
  AxesLabel -> {"Time (s)", "Velocity (m/s)"} ]
VelocityGraphRun1 =
 Plot[Tooltip[{velocityRun1L, velocityRun1R}], {x, 0,
   lastTimeRun1Both}, ImageSize -> 600,
  PlotLegends -> (Placed[#, Right] & /@ {{"Left", "Right"},
      LineLegend[{{Red, Dashed}, Dashed, Black}, {"Left", "Right"}]}),
   PlotStyle -> {Automatic}, PlotRange -> All,
  Epilog -> {Directive[Red, Dashed], line1Run1L, line2Run1L,
    line3Run1L, line4Run1L, line5Run1L, Directive[Red, Dashing[None]],
     linemaxRun1L, Directive[Black, Dashed], line1Run1R, line2Run1R,
    line3Run1R, line4Run1R, line5Run1R,
    Directive[Black, Dashing[None]], linemaxRun1R},
  PlotLabel -> "Counter Movement Jump - Velocity",
  AxesLabel -> {"Time (s)", "Velocity (m/s)"} ]
DisplacementGraphRun1Both =
 Plot[Tooltip[displacementRun1Both], {x, 0, lastTimeRun1Both},
  ImageSize -> 600, PlotStyle -> {Automatic},
  PlotLegends -> {"Legs Combined"},
  Epilog -> {Directive[Blue, Dashed], line1Run1Both, line2Run1Both,
    line3Run1Both, line4Run1Both, line5Run1Both,
    Directive[Blue, Dashing[None]], linemaxRun1Both},
  PlotRange -> All,
  PlotLabel -> "Counter Movement Jump - Displacement",
  AxesLabel -> {"Time (s)", "Displacement (m)"} ]
DisplacementGraphRun1 =
 Plot[Tooltip[{displacementRun1L, displacementRun1R}], {x, 0,
   lastTimeRun1Both}, ImageSize -> 600,
  PlotLegends -> (Placed[#, Right] & /@ {{"Left", "Right"},
      LineLegend[{{Red, Dashed}, Dashed, Black}, {"Left", "Right"}]}),
   PlotStyle -> {Automatic}, PlotRange -> All,
  Epilog -> {Directive[Red, Dashed], line1Run1L, line2Run1L,
    line3Run1L, line4Run1L, line5Run1L, Directive[Red, Dashing[None]],
     linemaxRun1L, Directive[Black, Dashed], line1Run1R, line2Run1R,
    line3Run1R, line4Run1R, line5Run1R,
    Directive[Black, Dashing[None]], linemaxRun1R},
  PlotLabel -> "Counter Movement Jump - Displacement",
  AxesLabel -> {"Time (s)", "Displacement (m)"} ]
PowerGraphRun1Both =
 Plot[Tooltip[powerRun1Both], {x, 0, lastTimeRun1Both},
  ImageSize -> 600, PlotStyle -> {Automatic},
  Epilog -> {Directive[Blue, Dashed],
    PlotLegends -> {"Legs Combined"}, line1Run1Both, line2Run1Both,
    line3Run1Both, line4Run1Both, line5Run1Both,
    Directive[Blue, Dashing[None]], linemaxRun1Both},
  PlotRange -> All, PlotLabel -> "Counter Movement Jump - Power",
  AxesLabel -> {"Time (s)", "Power (W)"} ]
PowerGraphRun1 =
 Plot[Tooltip[{powerRun1L, powerRun1R}], {x, 0, lastTimeRun1Both},
  ImageSize -> 600,
  PlotLegends -> (Placed[#, Right] & /@ {{"Left", "Right"},
      LineLegend[{{Red, Dashed}, Dashed, Black}, {"Left", "Right"}]}),
   PlotStyle -> {Automatic}, PlotRange -> All,
  Epilog -> {Directive[Red, Dashed], line1Run1L, line2Run1L,
    line3Run1L, line4Run1L, line5Run1L, Directive[Red, Dashing[None]],
     linemaxRun1L, Directive[Black, Dashed], line1Run1R, line2Run1R,
    line3Run1R, line4Run1R, line5Run1R,
    Directive[Black, Dashing[None]], linemaxRun1R},
```

```
  PlotLabel -> "Counter Movement Jump - Power",
  AxesLabel -> {"Time (s)", "Power (W)"} ]


(* Run2 *)
solutionRun2 = {{IDRun2, "", ""}, {"", "",
    ""}, {"Variable", "Left", "Right"}, {"", "", ""},
   {"run class.:", runtypeRun2L, runtypeRun2R}, {"mass:", massRun2L,
    massRun2R}, {"t1:", t1Run2L, t1Run2R}, {"t2:", t2Run2L,
    t2Run2R}, {"t3:", t3Run2L, t3Run2R}, {"t4:", t4Run2L,
    t4Run2R}, {"t5:", t5Run2L, t5Run2R}, {"tmax:", tmaxRun2L,
    tmaxRun2R}, {"", "", ""}, {"force @ t1:", force1Run2L,
    force1Run2R}, {"force @ t2:", force2Run2L,
    force2Run2R}, {"force @ t3:", force3Run2L,
    force3Run2R}, {"force @ t4:", force4Run2L,
    force4Run2R}, {"force @ t5:", force5Run2L,
    force5Run2R}, {"force @ tmax:", forcemaxRun2L,
    forcemaxRun2R}, {"veloc. @ t1:", velocity1Run2L,
    velocity1Run2R}, {"veloc. @ t2:", velocity2Run2L,
    velocity2Run2R}, {"veloc. @ t3:", velocity3Run2L,
    velocity3Run2R}, {"veloc. @ t4:", velocity4Run2L,
    velocity4Run2R}, {"veloc. @ t5:", velocity5Run2L,
    velocity5Run2R}, {"veloc. @ tmax:", velocitymaxRun2L,
    velocitymaxRun2R}, {"displace. @ t1:", displacement1Run2L,
    displacement1Run2R}, {"displace. @ t2:", displacement2Run2L,
    displacement2Run2R}, {"displace. @ t3:", displacement3Run2L,
    displacement3Run2R}, {"displace. @ t4:", displacement4Run2L,
    displacement4Run2R}, {"displace. @ t5:", displacement5Run2L,
    displacement5Run2R}, {"displace. @ tmax:", displacementmaxRun2L,
    displacementmaxRun2R}, {"power @ t1:", power1Run2L,
    power1Run2R}, {"power @ t2:", power2Run2L,
    power2Run2R}, {"power @ t3:", power3Run2L,
    power3Run2R}, {"power @ t4:", power4Run2L,
    power4Run2R}, {"power @ t5:", power5Run2L,
    power5Run2R}, {"power @ tmax:", powermaxRun2L,
    powermaxRun2R}, {"", "", ""}, {"impulse t1-t2:", impulse12Run2L,
    impulse12Run2R}, {"impulse t2-t3:", impulse23Run2L,
    impulse23Run2R}, {"impulse t3-t4:", impulse34Run2L,
    impulse34Run2R}, {"impulse t4-t5:", impulse45Run2L,
    impulse45Run2R}, {"impulse t2-tmax:", impulse2maxRun2L,
    impulse2maxRun2R}, {"max force t1-t2:", maxforce12Run2L,
    maxforce12Run2R}, {"max force t2-t3:", maxforce23Run2L,
    maxforce23Run2R}, {"max force t3-t4:", maxforce34Run2L,
    maxforce34Run2R}, {"max force t4-t5:", maxforce45Run2L,
    maxforce45Run2R}, {"max force t2-tmax:", maxforce2maxRun2L,
    maxforce2maxRun2R}, {"min force t1-t2:", minforce12Run2L,
    minforce12Run2R}, {"min force t2-t3:", minforce23Run2L,
    minforce23Run2R}, {"min force t3-t4:", minforce34Run2L,
    minforce34Run2R}, {"min force t4-t5:", minforce45Run2L,
    minforce45Run2R}, {"min force t2-tmax:", minforce2maxRun2L,
    minforce2maxRun2R}, {"max veloc. t1-t2:", maxvelocity12Run2L,
    maxvelocity12Run2R}, {"max veloc. t2-t3:", maxvelocity23Run2L,
    maxvelocity23Run2R}, {"max veloc. t3-t4:", maxvelocity34Run2L,
    maxvelocity34Run2R}, {"max veloc. t4-t5:", maxvelocity45Run2L,
    maxvelocity45Run2R}, {"max veloc. t2-tmax:", maxvelocity2maxRun2L,
     maxvelocity2maxRun2R}, {"min veloc. t1-t2:", minvelocity12Run2L,
    minvelocity12Run2R}, {"min veloc. t2-t3:", minvelocity23Run2L,
    minvelocity23Run2R}, {"min veloc. t3-t4:", minvelocity34Run2L,
    minvelocity34Run2R}, {"min veloc. t4-t5:", minvelocity45Run2L,
    minvelocity45Run2R}, {"min veloc. t2-tmax:", minvelocity2maxRun2L,
     minvelocity2maxRun2R}, {"max displace. t1-t2:",
    maxdisplacement12Run2L,
    maxdisplacement12Run2R}, {"max displace. t2-t3:",
    maxdisplacement23Run2L,
    maxdisplacement23Run2R}, {"max displace. t3-t4:",
    maxdisplacement34Run2L,
    maxdisplacement34Run2R}, {"max displace. t4-t5:",
```

```
maxdisplacement45Run2L,
maxdisplacement45Run2R}, {"max displace. t2-tmax:",
maxdisplacement2maxRun2L,
maxdisplacement2maxRun2R}, {"min displace. t1-t2:",
mindisplacement12Run2L,
mindisplacement12Run2R}, {"min displace. t2-t3:",
mindisplacement23Run2L,
mindisplacement23Run2R}, {"min displace. t3-t4:",
mindisplacement34Run2L,
mindisplacement34Run2R}, {"min displace. t4-t5:",
mindisplacement45Run2L,
mindisplacement45Run2R}, {"min displace. t2-tmax:",
mindisplacement2maxRun2L,
mindisplacement2maxRun2R}, {"max power t1-t2:", maxpower12Run2L,
maxpower12Run2R}, {"max power t2-t3:", maxpower23Run2L,
maxpower23Run2R}, {"max power t3-t4:", maxpower34Run2L,
maxpower34Run2R}, {"max power t4-t5:", maxpower45Run2L,
maxpower45Run2R}, {"max power t2-tmax:", maxpower2maxRun2L,
maxpower2maxRun2R}, {"min power t1-t2:", minpower12Run2L,
minpower12Run2R}, {"min power t2-t3:", minpower23Run2L,
minpower23Run2R}, {"min power t3-t4:", minpower34Run2L,
minpower34Run2R}, {"min power t4-t5:", minpower45Run2L,
minpower45Run2R}, {"min power t2-tmax:", minpower2maxRun2L,
minpower2maxRun2R}, {"avg. force in air:", noforceRun2L,
noforceRun2R}, {"", "", ""}, {"Legs Analyzed Together:", "",
""}, {"", "", ""}, {"Variable", "Legs Combined", ""}, {"", "",
""}, {"run class.:", runtypeRun2Both, ""}, {"mass:", massRun2Both,
 ""}, {"t1:", t1Run2Both, ""}, {"t2:", t2Run2Both, ""}, {"t3:",
t3Run2Both, ""}, {"t4:", t4Run2Both, ""}, {"t5:", t5Run2Both,
""}, {"tmax:", tmaxRun2Both, ""}, {"", "", ""}, {"force @ t1:",
force1Run2Both, ""}, {"force @ t2:", force2Run2Both,
""}, {"force @ t3:", force3Run2Both, ""}, {"force @ t4:",
force4Run2Both, ""}, {"force @ t5:", force5Run2Both,
""}, {"force @ tmax:", forcemaxRun2Both, ""}, {"veloc. @ t1:",
velocity1Run2Both, ""}, {"veloc. @ t2:", velocity2Run2Both,
""}, {"veloc. @ t3:", velocity3Run2Both, ""}, {"veloc. @ t4:",
velocity4Run2Both, ""}, {"veloc. @ t5:", velocity5Run2Both,
""}, {"veloc. @ tmax:", velocitymaxRun2Both,
""}, {"displace. @ t1:", displacement1Run2Both,
""}, {"displace. @ t2:", displacement2Run2Both,
""}, {"displace. @ t3:", displacement3Run2Both,
""}, {"displace. @ t4:", displacement4Run2Both,
""}, {"displace. @ t5:", displacement5Run2Both,
""}, {"displace. @ tmax:", displacementmaxRun2Both,
""}, {"power @ t1:", power1Run2Both, ""}, {"power @ t2:",
power2Run2Both, ""}, {"power @ t3:", power3Run2Both,
""}, {"power @ t4:", power4Run2Both, ""}, {"power @ t5:",
power5Run2Both, ""}, {"power @ tmax:", powermaxRun2Both, ""}, {"",
 "", ""}, {"impulse t1-t2:", impulse12Run2Both,
""}, {"impulse t2-t3:", impulse23Run2Both, ""}, {"impulse t3-t4:",
 impulse34Run2Both, ""}, {"impulse t4-t5:", impulse45Run2Both,
""}, {"impulse t2-tmax:", impulse2maxRun2Both,
""}, {"max force t1-t2:", maxforce12Run2Both,
""}, {"max force t2-t3:", maxforce23Run2Both,
""}, {"max force t3-t4:", maxforce34Run2Both,
""}, {"max force t4-t5:", maxforce45Run2Both,
""}, {"max force t2-tmax:", maxforce2maxRun2Both,
""}, {"min force t1-t2:", minforce12Run2Both,
""}, {"min force t2-t3:", minforce23Run2Both,
""}, {"min force t3-t4:", minforce34Run2Both,
""}, {"min force t4-t5:", minforce45Run2Both,
""}, {"min force t2-tmax:", minforce2maxRun2Both,
""}, {"max veloc. t1-t2:", maxvelocity12Run2Both,
""}, {"max veloc. t2-t3:", maxvelocity23Run2Both,
""}, {"max veloc. t3-t4:", maxvelocity34Run2Both,
""}, {"max veloc. t4-t5:", maxvelocity45Run2Both,
""}, {"max veloc. t2-tmax:", maxvelocity2maxRun2Both,
```

```
    ""}, {"min veloc. t1-t2:", minvelocity12Run2Both,
    ""}, {"min veloc. t2-t3:", minvelocity23Run2Both,
    ""}, {"min veloc. t3-t4:", minvelocity34Run2Both,
    ""}, {"min veloc. t4-t5:", minvelocity45Run2Both,
    ""}, {"min veloc. t2-tmax:", minvelocity2maxRun2Both,
    ""}, {"max displace. t1-t2:", maxdisplacement12Run2Both,
    ""}, {"max displace. t2-t3:", maxdisplacement23Run2Both,
    ""}, {"max displace. t3-t4:", maxdisplacement34Run2Both,
    ""}, {"max displace. t4-t5:", maxdisplacement45Run2Both,
    ""}, {"max displace. t2-tmax:", maxdisplacement2maxRun2Both,
    ""}, {"min displace. t1-t2:", mindisplacement12Run2Both,
    ""}, {"min displace. t2-t3:", mindisplacement23Run2Both,
    ""}, {"min displace. t3-t4:", mindisplacement34Run2Both,
    ""}, {"min displace. t4-t5:", mindisplacement45Run2Both,
    ""}, {"min displace. t2-tmax:", mindisplacement2maxRun2Both,
    ""}, {"max power t1-t2:", maxpower12Run2Both,
    ""}, {"max power t2-t3:", maxpower23Run2Both,
    ""}, {"max power t3-t4:", maxpower34Run2Both,
    ""}, {"max power t4-t5:", maxpower45Run2Both,
    ""}, {"max power t2-tmax:", maxpower2maxRun2Both,
    ""}, {"min power t1-t2:", minpower12Run2Both,
    ""}, {"min power t2-t3:", minpower23Run2Both,
    ""}, {"min power t3-t4:", minpower34Run2Both,
    ""}, {"min power t4-t5:", minpower45Run2Both,
    ""}, {"min power t2-tmax:", minpower2maxRun2Both,
    ""}, {"avg. force in air:", noforceRun2Both, ""}};
ForceGraphRun2Both =
 Plot[Tooltip[funcRun2Both[x]], {x, 0, lastTimeRun2Both},
  ImageSize -> 600, PlotStyle -> {Automatic},
  PlotLegends -> {"Legs Combined"},
  Epilog -> {Directive[Blue, Dashed], line1Run2Both, line2Run2Both,
    line3Run2Both, line4Run2Both, line5Run2Both,
    Directive[Blue, Dashing[None]], linemaxRun2Both},
  PlotRange -> All,
  PlotLabel -> "Counter Movement Jump - Normal Force",
  AxesLabel -> {"Time (s)", "Normal Force (N)"} ]
ForceGraphRun2 =
 Plot[Tooltip[{funcRun2L[x], funcRun2R[x]}], {x, 0, lastTimeRun2Both},
   ImageSize -> 600,
  PlotLegends -> (Placed[#, Right] & /@ {{"Left", "Right"},
     LineLegend[{{Red, Dashed}, Dashed, Black}, {"Left", "Right"}]}),
   PlotStyle -> {Automatic},
  Epilog -> {Directive[Red, Dashed], line1Run2L, line2Run2L,
    line3Run2L, line4Run2L, line5Run2L, Directive[Red, Dashing[None]],
     linemaxRun2L, Directive[Black, Dashed], line1Run2R, line2Run2R,
    line3Run2R, line4Run2R, line5Run2R,
    Directive[Black, Dashing[None]], linemaxRun2R}, PlotRange -> All,
  PlotLabel -> "Counter Movement Jump - Normal Force",
  AxesLabel -> {"Time (s)", "Normal Force (N)"} ]
VelocityGraphRun2Both =
 Plot[Tooltip[velocityRun2Both], {x, 0, lastTimeRun2Both},
  ImageSize -> 600, PlotStyle -> {Automatic},
  PlotLegends -> {"Legs Combined"},
  Epilog -> {Directive[Blue, Dashed], line1Run2Both, line2Run2Both,
    line3Run2Both, line4Run2Both, line5Run2Both,
    Directive[Blue, Dashing[None]], linemaxRun2Both},
  PlotRange -> All, PlotLabel -> "Counter Movement Jump - Velocity",
  AxesLabel -> {"Time (s)", "Velocity (m/s)"} ]
VelocityGraphRun2 =
 Plot[Tooltip[{velocityRun2L, velocityRun2R}], {x, 0,
   lastTimeRun2Both}, ImageSize -> 600,
  PlotLegends -> (Placed[#, Right] & /@ {{"Left", "Right"},
     LineLegend[{{Red, Dashed}, Dashed, Black}, {"Left", "Right"}]}),
   PlotStyle -> {Automatic}, PlotRange -> All,
  Epilog -> {Directive[Red, Dashed], line1Run2L, line2Run2L,
    line3Run2L, line4Run2L, line5Run2L, Directive[Red, Dashing[None]],
     linemaxRun2L, Directive[Black, Dashed], line1Run2R, line2Run2R,
```

```
   line3Run2R, line4Run2R, line5Run2R,
   Directive[Black, Dashing[None]], linemaxRun2R},
  PlotLabel -> "Counter Movement Jump - Velocity",
  AxesLabel -> {"Time (s)", "Velocity (m/s)"} ]
DisplacementGraphRun2Both =
 Plot[Tooltip[displacementRun2Both], {x, 0, lastTimeRun2Both},
  ImageSize -> 600, PlotStyle -> {Automatic},
  PlotLegends -> {"Legs Combined"},
  Epilog -> {Directive[Blue, Dashed], line1Run2Both, line2Run2Both,
   line3Run2Both, line4Run2Both, line5Run2Both,
   Directive[Blue, Dashing[None]], linemaxRun2Both},
  PlotRange -> All,
  PlotLabel -> "Counter Movement Jump - Displacement",
  AxesLabel -> {"Time (s)", "Displacement (m)"} ]
DisplacementGraphRun2 =
 Plot[Tooltip[{displacementRun2L, displacementRun2R}], {x, 0,
   lastTimeRun2Both}, ImageSize -> 600,
  PlotLegends -> (Placed[#, Right] & /@ {{"Left", "Right"},
     LineLegend[{{Red, Dashed}, Dashed, Black}, {"Left", "Right"}]}),
   PlotStyle -> {Automatic}, PlotRange -> All,
  Epilog -> {Directive[Red, Dashed], line1Run2L, line2Run2L,
   line3Run2L, line4Run2L, line5Run2L, Directive[Red, Dashing[None]],
    linemaxRun2L, Directive[Black, Dashed], line1Run2R, line2Run2R,
   line3Run2R, line4Run2R, line5Run2R,
   Directive[Black, Dashing[None]], linemaxRun2R},
  PlotLabel -> "Counter Movement Jump - Displacement",
  AxesLabel -> {"Time (s)", "Displacement (m)"} ]
PowerGraphRun2Both =
 Plot[Tooltip[powerRun2Both], {x, 0, lastTimeRun2Both},
  ImageSize -> 600, PlotStyle -> {Automatic},
  Epilog -> {Directive[Blue, Dashed],
   PlotLegends -> {"Legs Combined"}, line1Run2Both, line2Run2Both,
   line3Run2Both, line4Run2Both, line5Run2Both,
   Directive[Blue, Dashing[None]], linemaxRun2Both},
  PlotRange -> All, PlotLabel -> "Counter Movement Jump - Power",
  AxesLabel -> {"Time (s)", "Power (W)"} ]
PowerGraphRun2 =
 Plot[Tooltip[{powerRun2L, powerRun2R}], {x, 0, lastTimeRun2Both},
  ImageSize -> 600,
  PlotLegends -> (Placed[#, Right] & /@ {{"Left", "Right"},
     LineLegend[{{Red, Dashed}, Dashed, Black}, {"Left", "Right"}]}),
   PlotStyle -> {Automatic}, PlotRange -> All,
  Epilog -> {Directive[Red, Dashed], line1Run2L, line2Run2L,
   line3Run2L, line4Run2L, line5Run2L, Directive[Red, Dashing[None]],
    linemaxRun2L, Directive[Black, Dashed], line1Run2R, line2Run2R,
   line3Run2R, line4Run2R, line5Run2R,
   Directive[Black, Dashing[None]], linemaxRun2R},
  PlotLabel -> "Counter Movement Jump - Power",
  AxesLabel -> {"Time (s)", "Power (W)"} ]


(* Run3 *)
solutionRun3 = {{IDRun3, "", ""}, {"", "",
    ""}, {"Variable", "Left", "Right"}, {"", "", ""},
   {"run class.:", runtypeRun3L, runtypeRun3R}, {"mass:", massRun3L,
   massRun3R}, {"t1:", t1Run3L, t1Run3R}, {"t2:", t2Run3L,
   t2Run3R}, {"t3:", t3Run3L, t3Run3R}, {"t4:", t4Run3L,
   t4Run3R}, {"t5:", t5Run3L, t5Run3R}, {"tmax:", tmaxRun3L,
   tmaxRun3R}, {"", "", ""}, {"force @ t1:", force1Run3L,
   force1Run3R}, {"force @ t2:", force2Run3L,
   force2Run3R}, {"force @ t3:", force3Run3L,
   force3Run3R}, {"force @ t4:", force4Run3L,
   force4Run3R}, {"force @ t5:", force5Run3L,
   force5Run3R}, {"force @ tmax:", forcemaxRun3L,
   forcemaxRun3R}, {"veloc. @ t1:", velocity1Run3L,
   velocity1Run3R}, {"veloc. @ t2:", velocity2Run3L,
   velocity2Run3R}, {"veloc. @ t3:", velocity3Run3L,
```

```
velocity3Run3R}, {"veloc. @ t4:", velocity4Run3L,
velocity4Run3R}, {"veloc. @ t5:", velocity5Run3L,
velocity5Run3R}, {"veloc. @ tmax:", velocitymaxRun3L,
velocitymaxRun3R}, {"displace. @ t1:", displacement1Run3L,
displacement1Run3R}, {"displace. @ t2:", displacement2Run3L,
displacement2Run3R}, {"displace. @ t3:", displacement3Run3L,
displacement3Run3R}, {"displace. @ t4:", displacement4Run3L,
displacement4Run3R}, {"displace. @ t5:", displacement5Run3L,
displacement5Run3R}, {"displace. @ tmax:", displacementmaxRun3L,
displacementmaxRun3R}, {"power @ t1:", power1Run3L,
power1Run3R}, {"power @ t2:", power2Run3L,
power2Run3R}, {"power @ t3:", power3Run3L,
power3Run3R}, {"power @ t4:", power4Run3L,
power4Run3R}, {"power @ t5:", power5Run3L,
power5Run3R}, {"power @ tmax:", powermaxRun3L,
powermaxRun3R}, {"", "", ""}, {"impulse t1-t2:", impulse12Run3L,
impulse12Run3R}, {"impulse t2-t3:", impulse23Run3L,
impulse23Run3R}, {"impulse t3-t4:", impulse34Run3L,
impulse34Run3R}, {"impulse t4-t5:", impulse45Run3L,
impulse45Run3R}, {"impulse t2-tmax:", impulse2maxRun3L,
impulse2maxRun3R}, {"max force t1-t2:", maxforce12Run3L,
maxforce12Run3R}, {"max force t2-t3:", maxforce23Run3L,
maxforce23Run3R}, {"max force t3-t4:", maxforce34Run3L,
maxforce34Run3R}, {"max force t4-t5:", maxforce45Run3L,
maxforce45Run3R}, {"max force t2-tmax:", maxforce2maxRun3L,
maxforce2maxRun3R}, {"min force t1-t2:", minforce12Run3L,
minforce12Run3R}, {"min force t2-t3:", minforce23Run3L,
minforce23Run3R}, {"min force t3-t4:", minforce34Run3L,
minforce34Run3R}, {"min force t4-t5:", minforce45Run3L,
minforce45Run3R}, {"min force t2-tmax:", minforce2maxRun3L,
minforce2maxRun3R}, {"max veloc. t1-t2:", maxvelocity12Run3L,
maxvelocity12Run3R}, {"max veloc. t2-t3:", maxvelocity23Run3L,
maxvelocity23Run3R}, {"max veloc. t3-t4:", maxvelocity34Run3L,
maxvelocity34Run3R}, {"max veloc. t4-t5:", maxvelocity45Run3L,
maxvelocity45Run3R}, {"max veloc. t2-tmax:", maxvelocity2maxRun3L,
 maxvelocity2maxRun3R}, {"min veloc. t1-t2:", minvelocity12Run3L,
minvelocity12Run3R}, {"min veloc. t2-t3:", minvelocity23Run3L,
minvelocity23Run3R}, {"min veloc. t3-t4:", minvelocity34Run3L,
minvelocity34Run3R}, {"min veloc. t4-t5:", minvelocity45Run3L,
minvelocity45Run3R}, {"min veloc. t2-tmax:", minvelocity2maxRun3L,
 minvelocity2maxRun3R}, {"max displace. t1-t2:",
maxdisplacement12Run3L,
maxdisplacement12Run3R}, {"max displace. t2-t3:",
maxdisplacement23Run3L,
maxdisplacement23Run3R}, {"max displace. t3-t4:",
maxdisplacement34Run3L,
maxdisplacement34Run3R}, {"max displace. t4-t5:",
maxdisplacement45Run3L,
maxdisplacement45Run3R}, {"max displace. t2-tmax:",
maxdisplacement2maxRun3L,
maxdisplacement2maxRun3R}, {"min displace. t1-t2:",
mindisplacement12Run3L,
mindisplacement12Run3R}, {"min displace. t2-t3:",
mindisplacement23Run3L,
mindisplacement23Run3R}, {"min displace. t3-t4:",
mindisplacement34Run3L,
mindisplacement34Run3R}, {"min displace. t4-t5:",
mindisplacement45Run3L,
mindisplacement45Run3R}, {"min displace. t2-tmax:",
mindisplacement2maxRun3L,
mindisplacement2maxRun3R}, {"max power t1-t2:", maxpower12Run3L,
maxpower12Run3R}, {"max power t2-t3:", maxpower23Run3L,
maxpower23Run3R}, {"max power t3-t4:", maxpower34Run3L,
maxpower34Run3R}, {"max power t4-t5:", maxpower45Run3L,
maxpower45Run3R}, {"max power t2-tmax:", maxpower2maxRun3L,
maxpower2maxRun3R}, {"min power t1-t2:", minpower12Run3L,
minpower12Run3R}, {"min power t2-t3:", minpower23Run3L,
```

```
minpower23Run3R}, {"min power t3-t4:", minpower34Run3L,
minpower34Run3R}, {"min power t4-t5:", minpower45Run3L,
minpower45Run3R}, {"min power t2-tmax:", minpower2maxRun3L,
minpower2maxRun3R}, {"avg. force in air:", noforceRun3L,
noforceRun3R}, {"", "", ""}, {"Legs Analyzed Together:", "",
""}, {"", "", ""}, {"Variable", "Legs Combined", ""}, {"", "",
""}, {"run class.:", runtypeRun3Both, ""}, {"mass:", massRun3Both,
 ""}, {"t1:", t1Run3Both, ""}, {"t2:", t2Run3Both, ""}, {"t3:",
t3Run3Both, ""}, {"t4:", t4Run3Both, ""}, {"t5:", t5Run3Both,
""}, {"tmax:", tmaxRun3Both, ""}, {"", "", ""}, {"force @ t1:",
force1Run3Both, ""}, {"force @ t2:", force2Run3Both,
""}, {"force @ t3:", force3Run3Both, ""}, {"force @ t4:",
force4Run3Both, ""}, {"force @ t5:", force5Run3Both,
""}, {"force @ tmax:", forcemaxRun3Both, ""}, {"veloc. @ t1:",
velocity1Run3Both, ""}, {"veloc. @ t2:", velocity2Run3Both, ""},
{"veloc. @ t3:", velocity3Run3Both, ""}, {"veloc. @ t4:",
velocity4Run3Both, ""}, {"veloc. @ t5:", velocity5Run3Both,
""}, {"veloc. @ tmax:", velocitymaxRun3Both,
""}, {"displace. @ t1:", displacement1Run3Both,
""}, {"displace. @ t2:", displacement2Run3Both,
""}, {"displace. @ t3:", displacement3Run3Both,
""}, {"displace. @ t4:", displacement4Run3Both,
""}, {"displace. @ t5:", displacement5Run3Both,
""}, {"displace. @ tmax:", displacementmaxRun3Both,
""}, {"power @ t1:", power1Run3Both, ""}, {"power @ t2:",
power2Run3Both, ""}, {"power @ t3:", power3Run3Both,
""}, {"power @ t4:", power4Run3Both, ""}, {"power @ t5:",
power5Run3Both, ""}, {"power @ tmax:", powermaxRun3Both, ""}, {"",
 "", ""}, {"impulse t1-t2:", impulse12Run3Both,
""}, {"impulse t2-t3:", impulse23Run3Both, ""}, {"impulse t3-t4:",
 impulse34Run3Both, ""}, {"impulse t4-t5:", impulse45Run3Both,
""}, {"impulse t2-tmax:", impulse2maxRun3Both,
""}, {"max force t1-t2:", maxforce12Run3Both,
""}, {"max force t2-t3:", maxforce23Run3Both,
""}, {"max force t3-t4:", maxforce34Run3Both,
""}, {"max force t4-t5:", maxforce45Run3Both,
""}, {"max force t2-tmax:", maxforce2maxRun3Both,
""}, {"min force t1-t2:", minforce12Run3Both,
""}, {"min force t2-t3:", minforce23Run3Both,
""}, {"min force t3-t4:", minforce34Run3Both,
""}, {"min force t4-t5:", minforce45Run3Both,
""}, {"min force t2-tmax:", minforce2maxRun3Both,
""}, {"max veloc. t1-t2:", maxvelocity12Run3Both,
""}, {"max veloc. t2-t3:", maxvelocity23Run3Both,
""}, {"max veloc. t3-t4:", maxvelocity34Run3Both,
""}, {"max veloc. t4-t5:", maxvelocity45Run3Both,
""}, {"max veloc. t2-tmax:", maxvelocity2maxRun3Both,
""}, {"min veloc. t1-t2:", minvelocity12Run3Both,
""}, {"min veloc. t2-t3:", minvelocity23Run3Both,
""}, {"min veloc. t3-t4:", minvelocity34Run3Both,
""}, {"min veloc. t4-t5:", minvelocity45Run3Both,
""}, {"min veloc. t2-tmax:", minvelocity2maxRun3Both,
""}, {"max displace. t1-t2:", maxdisplacement12Run3Both,
""}, {"max displace. t2-t3:", maxdisplacement23Run3Both,
""}, {"max displace. t3-t4:", maxdisplacement34Run3Both,
""}, {"max displace. t4-t5:", maxdisplacement45Run3Both,
""}, {"max displace. t2-tmax:", maxdisplacement2maxRun3Both,
""}, {"min displace. t1-t2:", mindisplacement12Run3Both,
""}, {"min displace. t2-t3:", mindisplacement23Run3Both,
""}, {"min displace. t3-t4:", mindisplacement34Run3Both,
""}, {"min displace. t4-t5:", mindisplacement45Run3Both,
""}, {"min displace. t2-tmax:", mindisplacement2maxRun3Both,
""}, {"max power t1-t2:", maxpower12Run3Both,
""}, {"max power t2-t3:", maxpower23Run3Both,
""}, {"max power t3-t4:", maxpower34Run3Both,
""}, {"max power t4-t5:", maxpower45Run3Both,
""}, {"max power t2-tmax:", maxpower2maxRun3Both,
```

```
    ""}, {"min power t1-t2:", minpower12Run3Both,
    ""}, {"min power t2-t3:", minpower23Run3Both,
    ""}, {"min power t3-t4:", minpower34Run3Both,
    ""}, {"min power t4-t5:", minpower45Run3Both,
    ""}, {"min power t2-tmax:", minpower2maxRun3Both,
    ""}, {"avg. force in air:", noforceRun3Both, ""}};
ForceGraphRun3Both =
 Plot[Tooltip[funcRun3Both[x]], {x, 0, lastTimeRun3Both},
  ImageSize -> 600, PlotStyle -> {Automatic},
  PlotLegends -> {"Legs Combined"},
  Epilog -> {Directive[Blue, Dashed], line1Run3Both, line2Run3Both,
    line3Run3Both, line4Run3Both, line5Run3Both,
    Directive[Blue, Dashing[None]], linemaxRun3Both},
  PlotRange -> All,
  PlotLabel -> "Counter Movement Jump - Normal Force",
  AxesLabel -> {"Time (s)", "Normal Force (N)"} ]
ForceGraphRun3 =
 Plot[Tooltip[{funcRun3L[x], funcRun3R[x]}], {x, 0, lastTimeRun3Both},
   ImageSize -> 600,
  PlotLegends -> (Placed[#, Right] & /@ {{"Left", "Right"},
     LineLegend[{{Red, Dashed}, Dashed, Black}, {"Left", "Right"}]}),
   PlotStyle -> {Automatic},
  Epilog -> {Directive[Red, Dashed], line1Run3L, line2Run3L,
    line3Run3L, line4Run3L, line5Run3L, Directive[Red, Dashing[None]],
     linemaxRun3L, Directive[Black, Dashed], line1Run3R, line2Run3R,
    line3Run3R, line4Run3R, line5Run3R,
    Directive[Black, Dashing[None]], linemaxRun3R}, PlotRange -> All,
  PlotLabel -> "Counter Movement Jump - Normal Force",
  AxesLabel -> {"Time (s)", "Normal Force (N)"} ]
VelocityGraphRun3Both =
 Plot[Tooltip[velocityRun3Both], {x, 0, lastTimeRun3Both},
  ImageSize -> 600, PlotStyle -> {Automatic},
  PlotLegends -> {"Legs Combined"},
  Epilog -> {Directive[Blue, Dashed], line1Run3Both, line2Run3Both,
    line3Run3Both, line4Run3Both, line5Run3Both,
    Directive[Blue, Dashing[None]], linemaxRun3Both},
  PlotRange -> All, PlotLabel -> "Counter Movement Jump - Velocity",
  AxesLabel -> {"Time (s)", "Velocity (m/s)"} ]
VelocityGraphRun3 =
 Plot[Tooltip[{velocityRun3L, velocityRun3R}], {x, 0,
   lastTimeRun3Both}, ImageSize -> 600,
  PlotLegends -> (Placed[#, Right] & /@ {{"Left", "Right"},
     LineLegend[{{Red, Dashed}, Dashed, Black}, {"Left", "Right"}]}),
   PlotStyle -> {Automatic}, PlotRange -> All,
  Epilog -> {Directive[Red, Dashed], line1Run3L, line2Run3L,
    line3Run3L, line4Run3L, line5Run3L, Directive[Red, Dashing[None]],
     linemaxRun3L, Directive[Black, Dashed], line1Run3R, line2Run3R,
    line3Run3R, line4Run3R, line5Run3R,
    Directive[Black, Dashing[None]], linemaxRun3R},
  PlotLabel -> "Counter Movement Jump - Velocity",
  AxesLabel -> {"Time (s)", "Velocity (m/s)"} ]
DisplacementGraphRun3Both =
 Plot[Tooltip[displacementRun3Both], {x, 0, lastTimeRun3Both},
  ImageSize -> 600, PlotStyle -> {Automatic},
  PlotLegends -> {"Legs Combined"},
  Epilog -> {Directive[Blue, Dashed], line1Run3Both, line2Run3Both,
    line3Run3Both, line4Run3Both, line5Run3Both,
    Directive[Blue, Dashing[None]], linemaxRun3Both},
  PlotRange -> All,
  PlotLabel -> "Counter Movement Jump - Displacement",
  AxesLabel -> {"Time (s)", "Displacement (m)"} ]
DisplacementGraphRun3 =
 Plot[Tooltip[{displacementRun3L, displacementRun3R}], {x, 0,
   lastTimeRun3Both}, ImageSize -> 600,
  PlotLegends -> (Placed[#, Right] & /@ {{"Left", "Right"},
     LineLegend[{{Red, Dashed}, Dashed, Black}, {"Left", "Right"}]}),
   PlotStyle -> {Automatic}, PlotRange -> All,
```

```
  Epilog -> {Directive[Red, Dashed], line1Run3L, line2Run3L,
    line3Run3L, line4Run3L, line5Run3L, Directive[Red, Dashing[None]],
     linemaxRun3L, Directive[Black, Dashed], line1Run3R, line2Run3R,
    line3Run3R, line4Run3R, line5Run3R,
    Directive[Black, Dashing[None]], linemaxRun3R},
  PlotLabel -> "Counter Movement Jump - Displacement",
  AxesLabel -> {"Time (s)", "Displacement (m)"} ]
PowerGraphRun3Both =
 Plot[Tooltip[powerRun3Both], {x, 0, lastTimeRun3Both},
  ImageSize -> 600, PlotStyle -> {Automatic},
  Epilog -> {Directive[Blue, Dashed],
    PlotLegends -> {"Legs Combined"}, line1Run3Both, line2Run3Both,
    line3Run3Both, line4Run3Both, line5Run3Both,
    Directive[Blue, Dashing[None]], linemaxRun3Both},
  PlotRange -> All, PlotLabel -> "Counter Movement Jump - Power",
  AxesLabel -> {"Time (s)", "Power (W)"} ]
PowerGraphRun3 =
 Plot[Tooltip[{powerRun3L, powerRun3R}], {x, 0, lastTimeRun3Both},
  ImageSize -> 600,
  PlotLegends -> (Placed[#, Right] & /@ {{"Left", "Right"},
      LineLegend[{{Red, Dashed}, Dashed, Black}, {"Left", "Right"}]}),
   PlotStyle -> {Automatic}, PlotRange -> All,
  Epilog -> {Directive[Red, Dashed], line1Run3L, line2Run3L,
    line3Run3L, line4Run3L, line5Run3L, Directive[Red, Dashing[None]],
     linemaxRun3L, Directive[Black, Dashed], line1Run3R, line2Run3R,
    line3Run3R, line4Run3R, line5Run3R,
    Directive[Black, Dashing[None]], linemaxRun3R},
  PlotLabel -> "Counter Movement Jump - Power",
  AxesLabel -> {"Time (s)", "Power (W)"} ]


(* Run4 *)
solutionRun4 = {{IDRun4, "", ""}, {"", "",
    ""}, {"Variable", "Left", "Right"}, {"", "", ""},
   {"run class.:", runtypeRun4L, runtypeRun4R}, {"mass:", massRun4L,
    massRun4R}, {"t1:", t1Run4L, t1Run4R}, {"t2:", t2Run4L,
    t2Run4R}, {"t3:", t3Run4L, t3Run4R}, {"t4:", t4Run4L,
    t4Run4R}, {"t5:", t5Run4L, t5Run4R}, {"tmax:", tmaxRun4L,
    tmaxRun4R}, {"", "", ""}, {"force @ t1:", force1Run4L,
    force1Run4R}, {"force @ t2:", force2Run4L,
    force2Run4R}, {"force @ t3:", force3Run4L,
    force3Run4R}, {"force @ t4:", force4Run4L,
    force4Run4R}, {"force @ t5:", force5Run4L,
    force5Run4R}, {"force @ tmax:", forcemaxRun4L,
    forcemaxRun4R}, {"veloc. @ t1:", velocity1Run4L,
    velocity1Run4R}, {"veloc. @ t2:", velocity2Run4L,
    velocity2Run4R}, {"veloc. @ t3:", velocity3Run4L,
    velocity3Run4R}, {"veloc. @ t4:", velocity4Run4L,
    velocity4Run4R}, {"veloc. @ t5:", velocity5Run4L,
    velocity5Run4R}, {"veloc. @ tmax:", velocitymaxRun4L,
    velocitymaxRun4R}, {"displace. @ t1:", displacement1Run4L,
    displacement1Run4R}, {"displace. @ t2:", displacement2Run4L,
    displacement2Run4R}, {"displace. @ t3:", displacement3Run4L,
    displacement3Run4R}, {"displace. @ t4:", displacement4Run4L,
    displacement4Run4R}, {"displace. @ t5:", displacement5Run4L,
    displacement5Run4R}, {"displace. @ tmax:", displacementmaxRun4L,
    displacementmaxRun4R}, {"power @ t1:", power1Run4L,
    power1Run4R}, {"power @ t2:", power2Run4L,
    power2Run4R}, {"power @ t3:", power3Run4L,
    power3Run4R}, {"power @ t4:", power4Run4L,
    power4Run4R}, {"power @ t5:", power5Run4L,
    power5Run4R}, {"power @ tmax:", powermaxRun4L,
    powermaxRun4R}, {"", "", ""}, {"impulse t1-t2:", impulse12Run4L,
    impulse12Run4R}, {"impulse t2-t3:", impulse23Run4L,
    impulse23Run4R}, {"impulse t3-t4:", impulse34Run4L,
    impulse34Run4R}, {"impulse t4-t5:", impulse45Run4L,
    impulse45Run4R}, {"impulse t2-tmax:", impulse2maxRun4L,
```

```
impulse2maxRun4R}, {"max force t1-t2:", maxforce12Run4L,
maxforce12Run4R}, {"max force t2-t3:", maxforce23Run4L,
maxforce23Run4R}, {"max force t3-t4:", maxforce34Run4L,
maxforce34Run4R}, {"max force t4-t5:", maxforce45Run4L,
maxforce45Run4R}, {"max force t2-tmax:", maxforce2maxRun4L,
maxforce2maxRun4R}, {"min force t1-t2:", minforce12Run4L,
minforce12Run4R}, {"min force t2-t3:", minforce23Run4L,
minforce23Run4R}, {"min force t3-t4:", minforce34Run4L,
minforce34Run4R}, {"min force t4-t5:", minforce45Run4L,
minforce45Run4R}, {"min force t2-tmax:", minforce2maxRun4L,
minforce2maxRun4R}, {"max veloc. t1-t2:", maxvelocity12Run4L,
maxvelocity12Run4R}, {"max veloc. t2-t3:", maxvelocity23Run4L,
maxvelocity23Run4R}, {"max veloc. t3-t4:", maxvelocity34Run4L,
maxvelocity34Run4R}, {"max veloc. t4-t5:", maxvelocity45Run4L,
maxvelocity45Run4R}, {"max veloc. t2-tmax:", maxvelocity2maxRun4L,
 maxvelocity2maxRun4R}, {"min veloc. t1-t2:", minvelocity12Run4L,
minvelocity12Run4R}, {"min veloc. t2-t3:", minvelocity23Run4L,
minvelocity23Run4R}, {"min veloc. t3-t4:", minvelocity34Run4L,
minvelocity34Run4R}, {"min veloc. t4-t5:", minvelocity45Run4L,
minvelocity45Run4R}, {"min veloc. t2-tmax:", minvelocity2maxRun4L,
 minvelocity2maxRun4R}, {"max displace. t1-t2:",
maxdisplacement12Run4L,
maxdisplacement12Run4R}, {"max displace. t2-t3:",
maxdisplacement23Run4L,
maxdisplacement23Run4R}, {"max displace. t3-t4:",
maxdisplacement34Run4L,
maxdisplacement34Run4R}, {"max displace. t4-t5:",
maxdisplacement45Run4L,
maxdisplacement45Run4R}, {"max displace. t2-tmax:",
maxdisplacement2maxRun4L,
maxdisplacement2maxRun4R}, {"min displace. t1-t2:",
mindisplacement12Run4L,
mindisplacement12Run4R}, {"min displace. t2-t3:",
mindisplacement23Run4L,
mindisplacement23Run4R}, {"min displace. t3-t4:",
mindisplacement34Run4L,
mindisplacement34Run4R}, {"min displace. t4-t5:",
mindisplacement45Run4L,
mindisplacement45Run4R}, {"min displace. t2-tmax:",
mindisplacement2maxRun4L,
mindisplacement2maxRun4R}, {"max power t1-t2:", maxpower12Run4L,
maxpower12Run4R}, {"max power t2-t3:", maxpower23Run4L,
maxpower23Run4R}, {"max power t3-t4:", maxpower34Run4L,
maxpower34Run4R}, {"max power t4-t5:", maxpower45Run4L,
maxpower45Run4R}, {"max power t2-tmax:", maxpower2maxRun4L,
maxpower2maxRun4R}, {"min power t1-t2:", minpower12Run4L,
minpower12Run4R}, {"min power t2-t3:", minpower23Run4L,
minpower23Run4R}, {"min power t3-t4:", minpower34Run4L,
minpower34Run4R}, {"min power t4-t5:", minpower45Run4L,
minpower45Run4R}, {"min power t2-tmax:", minpower2maxRun4L,
minpower2maxRun4R}, {"avg. force in air:", noforceRun4L,
noforceRun4R}, {"", "", ""}, {"Legs Analyzed Together:", "",
""}, {"", "", ""}, {"Variable", "Legs Combined", ""}, {"", "",
""}, {"run class.:", runtypeRun4Both, ""}, {"mass:", massRun4Both,
 ""}, {"t1:", t1Run4Both, ""}, {"t2:", t2Run4Both, ""}, {"t3:",
t3Run4Both, ""}, {"t4:", t4Run4Both, ""}, {"t5:", t5Run4Both,
""}, {"tmax:", tmaxRun4Both, ""}, {"", "", ""}, {"force @ t1:",
force1Run4Both, ""}, {"force @ t2:", force2Run4Both,
""}, {"force @ t3:", force3Run4Both, ""}, {"force @ t4:",
force4Run4Both, ""}, {"force @ t5:", force5Run4Both,
""}, {"force @ tmax:", forcemaxRun4Both, ""}, {"veloc. @ t1:",
velocity1Run4Both, ""}, {"veloc. @ t2:", velocity2Run4Both,
""}, {"veloc. @ t3:", velocity3Run4Both, ""}, {"veloc. @ t4:",
velocity4Run4Both, ""}, {"veloc. @ t5:", velocity5Run4Both,
""}, {"veloc. @ tmax:", velocitymaxRun4Both,
""}, {"displace. @ t1:", displacement1Run4Both,
""}, {"displace. @ t2:", displacement2Run4Both,
```

```
    ""}, {"displace. @ t3:", displacement3Run4Both,
    ""}, {"displace. @ t4:", displacement4Run4Both,
    ""}, {"displace. @ t5:", displacement5Run4Both,
    ""}, {"displace. @ tmax:", displacementmaxRun4Both,
    ""}, {"power @ t1:", power1Run4Both, ""}, {"power @ t2:",
    power2Run4Both, ""}, {"power @ t3:", power3Run4Both,
    ""}, {"power @ t4:", power4Run4Both, ""}, {"power @ t5:",
    power5Run4Both, ""}, {"power @ tmax:", powermaxRun4Both, ""}, {"",
     "", ""}, {"impulse t1-t2:", impulse12Run4Both,
    ""}, {"impulse t2-t3:", impulse23Run4Both, ""}, {"impulse t3-t4:",
     impulse34Run4Both, ""}, {"impulse t4-t5:", impulse45Run4Both,
    ""}, {"impulse t2-tmax:", impulse2maxRun4Both,
    ""}, {"max force t1-t2:", maxforce12Run4Both,
    ""}, {"max force t2-t3:", maxforce23Run4Both,
    ""}, {"max force t3-t4:", maxforce34Run4Both,
    ""}, {"max force t4-t5:", maxforce45Run4Both,
    ""}, {"max force t2-tmax:", maxforce2maxRun4Both,
    ""}, {"min force t1-t2:", minforce12Run4Both,
    ""}, {"min force t2-t3:", minforce23Run4Both,
    ""}, {"min force t3-t4:", minforce34Run4Both,
    ""}, {"min force t4-t5:", minforce45Run4Both,
    ""}, {"min force t2-tmax:", minforce2maxRun4Both,
    ""}, {"max veloc. t1-t2:", maxvelocity12Run4Both,
    ""}, {"max veloc. t2-t3:", maxvelocity23Run4Both,
    ""}, {"max veloc. t3-t4:", maxvelocity34Run4Both,
    ""}, {"max veloc. t4-t5:", maxvelocity45Run4Both,
    ""}, {"max veloc. t2-tmax:", maxvelocity2maxRun4Both,
    ""}, {"min veloc. t1-t2:", minvelocity12Run4Both,
    ""}, {"min veloc. t2-t3:", minvelocity23Run4Both,
    ""}, {"min veloc. t3-t4:", minvelocity34Run4Both,
    ""}, {"min veloc. t4-t5:", minvelocity45Run4Both,
    ""}, {"min veloc. t2-tmax:", minvelocity2maxRun4Both,
    ""}, {"max displace. t1-t2:", maxdisplacement12Run4Both,
    ""}, {"max displace. t2-t3:", maxdisplacement23Run4Both,
    ""}, {"max displace. t3-t4:", maxdisplacement34Run4Both,
    ""}, {"max displace. t4-t5:", maxdisplacement45Run4Both,
    ""}, {"max displace. t2-tmax:", maxdisplacement2maxRun4Both,
    ""}, {"min displace. t1-t2:", mindisplacement12Run4Both,
    ""}, {"min displace. t2-t3:", mindisplacement23Run4Both,
    ""}, {"min displace. t3-t4:", mindisplacement34Run4Both,
    ""}, {"min displace. t4-t5:", mindisplacement45Run4Both,
    ""}, {"min displace. t2-tmax:", mindisplacement2maxRun4Both,
    ""}, {"max power t1-t2:", maxpower12Run4Both,
    ""}, {"max power t2-t3:", maxpower23Run4Both,
    ""}, {"max power t3-t4:", maxpower34Run4Both,
    ""}, {"max power t4-t5:", maxpower45Run4Both,
    ""}, {"max power t2-tmax:", maxpower2maxRun4Both,
    ""}, {"min power t1-t2:", minpower12Run4Both,
    ""}, {"min power t2-t3:", minpower23Run4Both,
    ""}, {"min power t3-t4:", minpower34Run4Both,
    ""}, {"min power t4-t5:", minpower45Run4Both,
    ""}, {"min power t2-tmax:", minpower2maxRun4Both,
    ""}, {"avg. force in air:", noforceRun4Both, ""}};
ForceGraphRun4Both =
 Plot[Tooltip[funcRun4Both[x]], {x, 0, lastTimeRun4Both},
  ImageSize -> 600, PlotStyle -> {Automatic},
  PlotLegends -> {"Legs Combined"},
  Epilog -> {Directive[Blue, Dashed], line1Run4Both, line2Run4Both,
    line3Run4Both, line4Run4Both, line5Run4Both,
    Directive[Blue, Dashing[None]], linemaxRun4Both},
  PlotRange -> All,
  PlotLabel -> "Counter Movement Jump - Normal Force",
  AxesLabel -> {"Time (s)", "Normal Force (N)"} ]
ForceGraphRun4 =
 Plot[Tooltip[{funcRun4L[x], funcRun4R[x]}], {x, 0, lastTimeRun4Both},
   ImageSize -> 600,
  PlotLegends -> (Placed[#, Right] & /@ {{"Left", "Right"},
```

```
     LineLegend[{{Red, Dashed}, Dashed, Black}, {"Left", "Right"}]}),
   PlotStyle -> {Automatic},
  Epilog -> {Directive[Red, Dashed], line1Run4L, line2Run4L,
    line3Run4L, line4Run4L, line5Run4L, Directive[Red, Dashing[None]],
     linemaxRun4L, Directive[Black, Dashed], line1Run4R, line2Run4R,
    line3Run4R, line4Run4R, line5Run4R,
    Directive[Black, Dashing[None]], linemaxRun4R}, PlotRange -> All,
  PlotLabel -> "Counter Movement Jump - Normal Force",
  AxesLabel -> {"Time (s)", "Normal Force (N)"} ]
VelocityGraphRun4Both =
 Plot[Tooltip[velocityRun4Both], {x, 0, lastTimeRun4Both},
  ImageSize -> 600, PlotStyle -> {Automatic},
  PlotLegends -> {"Legs Combined"},
  Epilog -> {Directive[Blue, Dashed], line1Run4Both, line2Run4Both,
    line3Run4Both, line4Run4Both, line5Run4Both,
    Directive[Blue, Dashing[None]], linemaxRun4Both},
  PlotRange -> All, PlotLabel -> "Counter Movement Jump - Velocity",
  AxesLabel -> {"Time (s)", "Velocity (m/s)"} ]
VelocityGraphRun4 =
 Plot[Tooltip[{velocityRun4L, velocityRun4R}], {x, 0,
   lastTimeRun4Both}, ImageSize -> 600,
  PlotLegends -> (Placed[#, Right] & /@ {{"Left", "Right"},
     LineLegend[{{Red, Dashed}, Dashed, Black}, {"Left", "Right"}]}),
   PlotStyle -> {Automatic}, PlotRange -> All,
  Epilog -> {Directive[Red, Dashed], line1Run4L, line2Run4L,
    line3Run4L, line4Run4L, line5Run4L, Directive[Red, Dashing[None]],
     linemaxRun4L, Directive[Black, Dashed], line1Run4R, line2Run4R,
    line3Run4R, line4Run4R, line5Run4R,
    Directive[Black, Dashing[None]], linemaxRun4R},
  PlotLabel -> "Counter Movement Jump - Velocity",
  AxesLabel -> {"Time (s)", "Velocity (m/s)"} ]
DisplacementGraphRun4Both =
 Plot[Tooltip[displacementRun4Both], {x, 0, lastTimeRun4Both},
  ImageSize -> 600, PlotStyle -> {Automatic},
  PlotLegends -> {"Legs Combined"},
  Epilog -> {Directive[Blue, Dashed], line1Run4Both, line2Run4Both,
    line3Run4Both, line4Run4Both, line5Run4Both,
    Directive[Blue, Dashing[None]], linemaxRun4Both},
  PlotRange -> All,
  PlotLabel -> "Counter Movement Jump - Displacement",
  AxesLabel -> {"Time (s)", "Displacement (m)"} ]
DisplacementGraphRun4 =
 Plot[Tooltip[{displacementRun4L, displacementRun4R}], {x, 0,
   lastTimeRun4Both}, ImageSize -> 600,
  PlotLegends -> (Placed[#, Right] & /@ {{"Left", "Right"},
     LineLegend[{{Red, Dashed}, Dashed, Black}, {"Left", "Right"}]}),
   PlotStyle -> {Automatic}, PlotRange -> All,
  Epilog -> {Directive[Red, Dashed], line1Run4L, line2Run4L,
    line3Run4L, line4Run4L, line5Run4L, Directive[Red, Dashing[None]],
     linemaxRun4L, Directive[Black, Dashed], line1Run4R, line2Run4R,
    line3Run4R, line4Run4R, line5Run4R,
    Directive[Black, Dashing[None]], linemaxRun4R},
  PlotLabel -> "Counter Movement Jump - Displacement",
  AxesLabel -> {"Time (s)", "Displacement (m)"} ]
PowerGraphRun4Both =
 Plot[Tooltip[powerRun4Both], {x, 0, lastTimeRun4Both},
  ImageSize -> 600, PlotStyle -> {Automatic},
  Epilog -> {Directive[Blue, Dashed],
    PlotLegends -> {"Legs Combined"}, line1Run4Both, line2Run4Both,
    line3Run4Both, line4Run4Both, line5Run4Both,
    Directive[Blue, Dashing[None]], linemaxRun4Both},
  PlotRange -> All, PlotLabel -> "Counter Movement Jump - Power",
  AxesLabel -> {"Time (s)", "Power (W)"} ]
PowerGraphRun4 =
 Plot[Tooltip[{powerRun4L, powerRun4R}], {x, 0, lastTimeRun4Both},
  ImageSize -> 600,
  PlotLegends -> (Placed[#, Right] & /@ {{"Left", "Right"},
```

```
     LineLegend[{{Red, Dashed}, Dashed, Black}, {"Left", "Right"}]}),
   PlotStyle -> {Automatic}, PlotRange -> All,
  Epilog -> {Directive[Red, Dashed], line1Run4L, line2Run4L,
    line3Run4L, line4Run4L, line5Run4L, Directive[Red, Dashing[None]],
     linemaxRun4L, Directive[Black, Dashed], line1Run4R, line2Run4R,
    line3Run4R, line4Run4R, line5Run4R,
    Directive[Black, Dashing[None]], linemaxRun4R},
  PlotLabel -> "Counter Movement Jump - Power",
  AxesLabel -> {"Time (s)", "Power (W)"} ]


(* Run5 *)
solutionRun5 = {{IDRun5, "", ""}, {"", "",
    ""}, {"Variable", "Left", "Right"}, {"", "", ""},
   {"run class.:", runtypeRun5L, runtypeRun5R}, {"mass:", massRun5L,
    massRun5R}, {"t1:", t1Run5L, t1Run5R}, {"t2:", t2Run5L,
    t2Run5R}, {"t3:", t3Run5L, t3Run5R}, {"t4:", t4Run5L,
    t4Run5R}, {"t5:", t5Run5L, t5Run5R}, {"tmax:", tmaxRun5L,
    tmaxRun5R}, {"", "", ""}, {"force @ t1:", force1Run5L,
    force1Run5R}, {"force @ t2:", force2Run5L,
    force2Run5R}, {"force @ t3:", force3Run5L,
    force3Run5R}, {"force @ t4:", force4Run5L,
    force4Run5R}, {"force @ t5:", force5Run5L,
    force5Run5R}, {"force @ tmax:", forcemaxRun5L,
    forcemaxRun5R}, {"veloc. @ t1:", velocity1Run5L,
    velocity1Run5R}, {"veloc. @ t2:", velocity2Run5L,
    velocity2Run5R}, {"veloc. @ t3:", velocity3Run5L,
    velocity3Run5R}, {"veloc. @ t4:", velocity4Run5L,
    velocity4Run5R}, {"veloc. @ t5:", velocity5Run5L,
    velocity5Run5R}, {"veloc. @ tmax:", velocitymaxRun5L,
    velocitymaxRun5R}, {"displace. @ t1:", displacement1Run5L,
    displacement1Run5R}, {"displace. @ t2:", displacement2Run5L,
    displacement2Run5R}, {"displace. @ t3:", displacement3Run5L,
    displacement3Run5R}, {"displace. @ t4:", displacement4Run5L,
    displacement4Run5R}, {"displace. @ t5:", displacement5Run5L,
    displacement5Run5R}, {"displace. @ tmax:", displacementmaxRun5L,
    displacementmaxRun5R}, {"power @ t1:", power1Run5L,
    power1Run5R}, {"power @ t2:", power2Run5L,
    power2Run5R}, {"power @ t3:", power3Run5L,
    power3Run5R}, {"power @ t4:", power4Run5L,
    power4Run5R}, {"power @ t5:", power5Run5L,
    power5Run5R}, {"power @ tmax:", powermaxRun5L,
    powermaxRun5R}, {"", "", ""}, {"impulse t1-t2:", impulse12Run5L,
    impulse12Run5R}, {"impulse t2-t3:", impulse23Run5L,
    impulse23Run5R}, {"impulse t3-t4:", impulse34Run5L,
    impulse34Run5R}, {"impulse t4-t5:", impulse45Run5L,
    impulse45Run5R}, {"impulse t2-tmax:", impulse2maxRun5L,
    impulse2maxRun5R}, {"max force t1-t2:", maxforce12Run5L,
    maxforce12Run5R}, {"max force t2-t3:", maxforce23Run5L,
    maxforce23Run5R}, {"max force t3-t4:", maxforce34Run5L,
    maxforce34Run5R}, {"max force t4-t5:", maxforce45Run5L,
    maxforce45Run5R}, {"max force t2-tmax:", maxforce2maxRun5L,
    maxforce2maxRun5R}, {"min force t1-t2:", minforce12Run5L,
    minforce12Run5R}, {"min force t2-t3:", minforce23Run5L,
    minforce23Run5R}, {"min force t3-t4:", minforce34Run5L,
    minforce34Run5R}, {"min force t4-t5:", minforce45Run5L,
    minforce45Run5R}, {"min force t2-tmax:", minforce2maxRun5L,
    minforce2maxRun5R}, {"max veloc. t1-t2:", maxvelocity12Run5L,
    maxvelocity12Run5R}, {"max veloc. t2-t3:", maxvelocity23Run5L,
    maxvelocity23Run5R}, {"max veloc. t3-t4:", maxvelocity34Run5L,
    maxvelocity34Run5R}, {"max veloc. t4-t5:", maxvelocity45Run5L,
    maxvelocity45Run5R}, {"max veloc. t2-tmax:", maxvelocity2maxRun5L,
     maxvelocity2maxRun5R}, {"min veloc. t1-t2:", minvelocity12Run5L,
    minvelocity12Run5R}, {"min veloc. t2-t3:", minvelocity23Run5L,
    minvelocity23Run5R}, {"min veloc. t3-t4:", minvelocity34Run5L,
    minvelocity34Run5R}, {"min veloc. t4-t5:", minvelocity45Run5L,
    minvelocity45Run5R}, {"min veloc. t2-tmax:", minvelocity2maxRun5L,
```

```
 minvelocity2maxRun5R}, {"max displace. t1-t2:",
maxdisplacement12Run5L,
maxdisplacement12Run5R}, {"max displace. t2-t3:",
maxdisplacement23Run5L,
maxdisplacement23Run5R}, {"max displace. t3-t4:",
maxdisplacement34Run5L,
maxdisplacement34Run5R}, {"max displace. t4-t5:",
maxdisplacement45Run5L,
maxdisplacement45Run5R}, {"max displace. t2-tmax:",
maxdisplacement2maxRun5L,
maxdisplacement2maxRun5R}, {"min displace. t1-t2:",
mindisplacement12Run5L,
mindisplacement12Run5R}, {"min displace. t2-t3:",
mindisplacement23Run5L,
mindisplacement23Run5R}, {"min displace. t3-t4:",
mindisplacement34Run5L,
mindisplacement34Run5R}, {"min displace. t4-t5:",
mindisplacement45Run5L,
mindisplacement45Run5R}, {"min displace. t2-tmax:",
mindisplacement2maxRun5L,
mindisplacement2maxRun5R}, {"max power t1-t2:", maxpower12Run5L,
maxpower12Run5R}, {"max power t2-t3:", maxpower23Run5L,
maxpower23Run5R}, {"max power t3-t4:", maxpower34Run5L,
maxpower34Run5R}, {"max power t4-t5:", maxpower45Run5L,
maxpower45Run5R}, {"max power t2-tmax:", maxpower2maxRun5L,
maxpower2maxRun5R}, {"min power t1-t2:", minpower12Run5L,
minpower12Run5R}, {"min power t2-t3:", minpower23Run5L,
minpower23Run5R}, {"min power t3-t4:", minpower34Run5L,
minpower34Run5R}, {"min power t4-t5:", minpower45Run5L,
minpower45Run5R}, {"min power t2-tmax:", minpower2maxRun5L,
minpower2maxRun5R}, {"avg. force in air:", noforceRun5L,
noforceRun5R}, {"", "", ""}, {"Legs Analyzed Together:", "",
""}, {"", "", ""}, {"Variable", "Legs Combined", ""}, {"", "",
""}, {"run class.:", runtypeRun5Both, ""},
{"mass:", massRun5Both, ""}, {"t1:", t1Run5Both, ""}, {"t2:",
t2Run5Both, ""}, {"t3:", t3Run5Both, ""}, {"t4:", t4Run5Both,
""}, {"t5:", t5Run5Both, ""}, {"tmax:", tmaxRun5Both, ""}, {"",
"", ""}, {"force @ t1:", force1Run5Both, ""}, {"force @ t2:",
force2Run5Both, ""}, {"force @ t3:", force3Run5Both,
""}, {"force @ t4:", force4Run5Both, ""}, {"force @ t5:",
force5Run5Both, ""}, {"force @ tmax:", forcemaxRun5Both,
""}, {"veloc. @ t1:", velocity1Run5Both, ""}, {"veloc. @ t2:",
velocity2Run5Both, ""}, {"veloc. @ t3:", velocity3Run5Both,
""}, {"veloc. @ t4:", velocity4Run5Both, ""}, {"veloc. @ t5:",
velocity5Run5Both, ""}, {"veloc. @ tmax:", velocitymaxRun5Both,
""}, {"displace. @ t1:", displacement1Run5Both,
""}, {"displace. @ t2:", displacement2Run5Both,
""}, {"displace. @ t3:", displacement3Run5Both,
""}, {"displace. @ t4:", displacement4Run5Both,
""}, {"displace. @ t5:", displacement5Run5Both,
""}, {"displace. @ tmax:", displacementmaxRun5Both,
""}, {"power @ t1:", power1Run5Both, ""}, {"power @ t2:",
power2Run5Both, ""}, {"power @ t3:", power3Run5Both,
""}, {"power @ t4:", power4Run5Both, ""}, {"power @ t5:",
power5Run5Both, ""}, {"power @ tmax:", powermaxRun5Both, ""}, {"",
 "", ""}, {"impulse t1-t2:", impulse12Run5Both,
""}, {"impulse t2-t3:", impulse23Run5Both, ""}, {"impulse t3-t4:",
 impulse34Run5Both, ""}, {"impulse t4-t5:", impulse45Run5Both,
""}, {"impulse t2-tmax:", impulse2maxRun5Both,
""}, {"max force t1-t2:", maxforce12Run5Both,
""}, {"max force t2-t3:", maxforce23Run5Both,
""}, {"max force t3-t4:", maxforce34Run5Both,
""}, {"max force t4-t5:", maxforce45Run5Both,
""}, {"max force t2-tmax:", maxforce2maxRun5Both,
""}, {"min force t1-t2:", minforce12Run5Both,
""}, {"min force t2-t3:", minforce23Run5Both,
""}, {"min force t3-t4:", minforce34Run5Both,
```

```
    ""}, {"min force t4-t5:", minforce45Run5Both,
    ""}, {"min force t2-tmax:", minforce2maxRun5Both,
    ""}, {"max veloc. t1-t2:", maxvelocity12Run5Both,
    ""}, {"max veloc. t2-t3:", maxvelocity23Run5Both,
    ""}, {"max veloc. t3-t4:", maxvelocity34Run5Both,
    ""}, {"max veloc. t4-t5:", maxvelocity45Run5Both,
    ""}, {"max veloc. t2-tmax:", maxvelocity2maxRun5Both,
    ""}, {"min veloc. t1-t2:", minvelocity12Run5Both,
    ""}, {"min veloc. t2-t3:", minvelocity23Run5Both,
    ""}, {"min veloc. t3-t4:", minvelocity34Run5Both,
    ""}, {"min veloc. t4-t5:", minvelocity45Run5Both,
    ""}, {"min veloc. t2-tmax:", minvelocity2maxRun5Both,
    ""}, {"max displace. t1-t2:", maxdisplacement12Run5Both,
    ""}, {"max displace. t2-t3:", maxdisplacement23Run5Both,
    ""}, {"max displace. t3-t4:", maxdisplacement34Run5Both,
    ""}, {"max displace. t4-t5:", maxdisplacement45Run5Both,
    ""}, {"max displace. t2-tmax:", maxdisplacement2maxRun5Both,
    ""}, {"min displace. t1-t2:", mindisplacement12Run5Both,
    ""}, {"min displace. t2-t3:", mindisplacement23Run5Both,
    ""}, {"min displace. t3-t4:", mindisplacement34Run5Both,
    ""}, {"min displace. t4-t5:", mindisplacement45Run5Both,
    ""}, {"min displace. t2-tmax:", mindisplacement2maxRun5Both,
    ""}, {"max power t1-t2:", maxpower12Run5Both,
    ""}, {"max power t2-t3:", maxpower23Run5Both,
    ""}, {"max power t3-t4:", maxpower34Run5Both,
    ""}, {"max power t4-t5:", maxpower45Run5Both,
    ""}, {"max power t2-tmax:", maxpower2maxRun5Both,
    ""}, {"min power t1-t2:", minpower12Run5Both,
    ""}, {"min power t2-t3:", minpower23Run5Both,
    ""}, {"min power t3-t4:", minpower34Run5Both,
    ""}, {"min power t4-t5:", minpower45Run5Both,
    ""}, {"min power t2-tmax:", minpower2maxRun5Both,
    ""}, {"avg. force in air:", noforceRun5Both, ""}};
ForceGraphRun5Both =
 Plot[Tooltip[funcRun5Both[x]], {x, 0, lastTimeRun5Both},
  ImageSize -> 600, PlotStyle -> {Automatic},
  PlotLegends -> {"Legs Combined"},
  Epilog -> {Directive[Blue, Dashed], line1Run5Both, line2Run5Both,
    line3Run5Both, line4Run5Both, line5Run5Both,
    Directive[Blue, Dashing[None]], linemaxRun5Both},
  PlotRange -> All,
  PlotLabel -> "Counter Movement Jump - Normal Force",
  AxesLabel -> {"Time (s)", "Normal Force (N)"} ]
ForceGraphRun5 =
 Plot[Tooltip[{funcRun5L[x], funcRun5R[x]}], {x, 0, lastTimeRun5Both},
   ImageSize -> 600,
  PlotLegends -> (Placed[#, Right] & /@ {{"Left", "Right"},
     LineLegend[{{Red, Dashed}, Dashed, Black}, {"Left", "Right"}]}),
   PlotStyle -> {Automatic},
  Epilog -> {Directive[Red, Dashed], line1Run5L, line2Run5L,
    line3Run5L, line4Run5L, line5Run5L, Directive[Red, Dashing[None]],
     linemaxRun5L, Directive[Black, Dashed], line1Run5R, line2Run5R,
    line3Run5R, line4Run5R, line5Run5R,
    Directive[Black, Dashing[None]], linemaxRun5R}, PlotRange -> All,
  PlotLabel -> "Counter Movement Jump - Normal Force",
  AxesLabel -> {"Time (s)", "Normal Force (N)"} ]
VelocityGraphRun5Both =
 Plot[Tooltip[velocityRun5Both], {x, 0, lastTimeRun5Both},
  ImageSize -> 600, PlotStyle -> {Automatic},
  PlotLegends -> {"Legs Combined"},
  Epilog -> {Directive[Blue, Dashed], line1Run5Both, line2Run5Both,
    line3Run5Both, line4Run5Both, line5Run5Both,
    Directive[Blue, Dashing[None]], linemaxRun5Both},
  PlotRange -> All, PlotLabel -> "Counter Movement Jump - Velocity",
  AxesLabel -> {"Time (s)", "Velocity (m/s)"} ]
VelocityGraphRun5 =
 Plot[Tooltip[{velocityRun5L, velocityRun5R}], {x, 0,
```

```
   lastTimeRun5Both}, ImageSize -> 600,
  PlotLegends -> (Placed[#, Right] & /@ {{"Left", "Right"},
     LineLegend[{{Red, Dashed}, Dashed, Black}, {"Left", "Right"}]}),
   PlotStyle -> {Automatic}, PlotRange -> All,
  Epilog -> {Directive[Red, Dashed], line1Run5L, line2Run5L,
    line3Run5L, line4Run5L, line5Run5L, Directive[Red, Dashing[None]],
     linemaxRun5L, Directive[Black, Dashed], line1Run5R, line2Run5R,
    line3Run5R, line4Run5R, line5Run5R,
    Directive[Black, Dashing[None]], linemaxRun5R},
  PlotLabel -> "Counter Movement Jump - Velocity",
  AxesLabel -> {"Time (s)", "Velocity (m/s)"} ]
DisplacementGraphRun5Both =
 Plot[Tooltip[displacementRun5Both], {x, 0, lastTimeRun5Both},
  ImageSize -> 600, PlotStyle -> {Automatic},
  PlotLegends -> {"Legs Combined"},
  Epilog -> {Directive[Blue, Dashed], line1Run5Both, line2Run5Both,
    line3Run5Both, line4Run5Both, line5Run5Both,
    Directive[Blue, Dashing[None]], linemaxRun5Both},
  PlotRange -> All,
  PlotLabel -> "Counter Movement Jump - Displacement",
  AxesLabel -> {"Time (s)", "Displacement (m)"} ]
DisplacementGraphRun5 =
 Plot[Tooltip[{displacementRun5L, displacementRun5R}], {x, 0,
   lastTimeRun5Both}, ImageSize -> 600,
  PlotLegends -> (Placed[#, Right] & /@ {{"Left", "Right"},
     LineLegend[{{Red, Dashed}, Dashed, Black}, {"Left", "Right"}]}),
   PlotStyle -> {Automatic}, PlotRange -> All,
  Epilog -> {Directive[Red, Dashed], line1Run5L, line2Run5L,
    line3Run5L, line4Run5L, line5Run5L, Directive[Red, Dashing[None]],
     linemaxRun5L, Directive[Black, Dashed], line1Run5R, line2Run5R,
    line3Run5R, line4Run5R, line5Run5R,
    Directive[Black, Dashing[None]], linemaxRun5R},
  PlotLabel -> "Counter Movement Jump - Displacement",
  AxesLabel -> {"Time (s)", "Displacement (m)"} ]
PowerGraphRun5Both =
 Plot[Tooltip[powerRun5Both], {x, 0, lastTimeRun5Both},
  ImageSize -> 600, PlotStyle -> {Automatic},
  Epilog -> {Directive[Blue, Dashed],
    PlotLegends -> {"Legs Combined"}, line1Run5Both, line2Run5Both,
    line3Run5Both, line4Run5Both, line5Run5Both,
    Directive[Blue, Dashing[None]], linemaxRun5Both},
  PlotRange -> All, PlotLabel -> "Counter Movement Jump - Power",
  AxesLabel -> {"Time (s)", "Power (W)"} ]
PowerGraphRun5 =
 Plot[Tooltip[{powerRun5L, powerRun5R}], {x, 0, lastTimeRun5Both},
  ImageSize -> 600,
  PlotLegends -> (Placed[#, Right] & /@ {{"Left", "Right"},
     LineLegend[{{Red, Dashed}, Dashed, Black}, {"Left", "Right"}]}),
   PlotStyle -> {Automatic}, PlotRange -> All,
  Epilog -> {Directive[Red, Dashed], line1Run5L, line2Run5L,
    line3Run5L, line4Run5L, line5Run5L, Directive[Red, Dashing[None]],
     linemaxRun5L, Directive[Black, Dashed], line1Run5R, line2Run5R,
    line3Run5R, line4Run5R, line5Run5R,
    Directive[Black, Dashing[None]], linemaxRun5R},
  PlotLabel -> "Counter Movement Jump - Power",
  AxesLabel -> {"Time (s)", "Power (W)"} ]

(*EXPORT ALL RUNS*)
Export[
 "C:\\Users\\Admin\\Desktop\\Mathematica Analysis Exports\\" <>
  NameID <> " " <> DateID <> ".xlsx", {solutionRun1, solutionRun2,
  solutionRun3, solutionRun4, solutionRun5}]
Export["C:\\Users\\Admin\\Desktop\\Mathematica Analysis Exports\\" <>
  IDRun1 <> ".png", {{"
    " ForceGraphRun1Both}, {"
    " ForceGraphRun1}, {"
    " VelocityGraphRun1Both}, {"
```

```
    " VelocityGraphRun1}, {"
    " DisplacementGraphRun1Both}, {"
    " DisplacementGraphRun1}, {"
    " PowerGraphRun1Both}, {"
    " PowerGraphRun1}} // MatrixForm]
Export["C:\\Users\\Admin\\Desktop\\Mathematica Analysis Exports\\" <>
  IDRun2 <> ".png", {{"
    " ForceGraphRun2Both}, {"
    " ForceGraphRun2}, {"
    " VelocityGraphRun2Both}, {"
    " VelocityGraphRun2}, {"
    " DisplacementGraphRun2Both}, {"
    " DisplacementGraphRun2}, {"
    " PowerGraphRun2Both}, {"
    " PowerGraphRun2}} // MatrixForm]
Export["C:\\Users\\Admin\\Desktop\\Mathematica Analysis Exports\\" <>
  IDRun3 <> ".png", {{"
    " ForceGraphRun3Both}, {"
    " ForceGraphRun3}, {"
    " VelocityGraphRun3Both}, {"
    " VelocityGraphRun3}, {"
    " DisplacementGraphRun3Both}, {"
    " DisplacementGraphRun3}, {"
    " PowerGraphRun3Both}, {"
    " PowerGraphRun3}} // MatrixForm]
Export["C:\\Users\\Admin\\Desktop\\Mathematica Analysis Exports\\" <>
  IDRun4 <> ".png", {{"
    " ForceGraphRun4Both}, {"
    " ForceGraphRun4}, {"
    " VelocityGraphRun4Both}, {"
    " VelocityGraphRun4}, {"
    " DisplacementGraphRun4Both}, {"
    " DisplacementGraphRun4}, {"
    " PowerGraphRun4Both}, {"
    " PowerGraphRun4}} // MatrixForm]
Export["C:\\Users\\Admin\\Desktop\\Mathematica Analysis Exports\\" <>
  IDRun5 <> ".png", {{"
    " ForceGraphRun5Both}, {"
    " ForceGraphRun5}, {"
    " VelocityGraphRun5Both}, {"
    " VelocityGraphRun5}, {"
    " DisplacementGraphRun5Both}, {"
    " DisplacementGraphRun5}, {"
    " PowerGraphRun5Both}, {"
    " PowerGraphRun5}} // MatrixForm]
```

Appendix H. NCMJ Mass and Force Results

NCMJ Mass

Two-Way ANOVA Results		
	Yes	No
Statistical Evidence for Interaction Effect Between Gender and Sport?		X
Statistical Significance Found between Gender groups?	X	
Statistical Significance Found between Sport groups?	X	

Tukey HSD Test
For groups with significant differences, which group has the larger value?
Male>Female Rugby>Soccer Rugby>Swim Volleyball>Soccer

NCMJ Mass (kg)

Gender	Sport	Mean	Standard Deviation	N
Male	Swim	75.628	10.611	9
	Volleyball	86.518	6.094	14

	Soccer	75.478	12.684	7
	Total	80.675	10.578	30
Female	Swim	72.481	9.463	5
	Rugby	76.030	12.986	20
	Soccer	61.305	6.415	21
	Total	68.922	12.182	46
Total	Swim	74.504	9.964	14
	Volleyball	86.518	6.094	14
	Rugby	76.030	12.986	20
	Soccer	60.221	6.829	28
	Total	74.484	13.124	76

NCMJ Mass (kg)

			95% Confidence Level			
Group A	**Group B**	**Mean Difference (A-B)**	**Lower Bound**	**Upper Bound**	**P**	
Soccer	Rugby	-14.120	-21.678	-6.563	0.000	*
Swim	Rugby	-9.082	-18.077	-0.086	0.047	*
Volleyball	Rugby	-1.265	-10.260	7.730	0.983	
Swim	Soccer	5.038	-3.411	13.488	0.403	
Volleyball	Soccer	12.855	4.406	21.305	0.001	*
Volleyball	Swim	7.817	-1.940	17.574	0.160	
Male	Female	11.753	7.163	16.344	0.000	*

*Mean difference is significant at α=0.05 level

NCMJ Peak Force

Two-Way ANOVA Results		
	Yes	No
Statistical Evidence for Interaction Effect Between Gender and Sport?	X	
Statistical Significance Found between Gender groups?	X	
Statistical Significance Found between Sport groups?	X	

Interaction effect exists between gender and sport for peak force, so a Tukey HSD test was not completed after the Two-Way ANOVA.

NCMJ Peak Force (N)

Gender	Sport	Mean	Standard Deviation	N
Male	Swim	1584.238	290.669	9
	Volleyball	2053.329	187.789	14
	Soccer	1973.521	249.385	7
	Total	1893.980	309.100	30
Female	Swim	1379.279	205.932	5
	Rugby	1691.402	274.835	20
	Soccer	1436.194	165.797	21
	Total	1540.968	256.836	46
Total	Swim	1511.038	274.642	14
	Volleyball	2053.329	187.789	14
	Rugby	1691.402	274.835	20
	Soccer	1431.642	168.371	28
	Total	1673.745	325.358	76

NCMJ Normalized Peak Force

Two-Way ANOVA Results	Tukey HSD Test

	Yes	No
Statistical Evidence for Interaction Effect Between Gender and Sport?		X
Statistical Significance Found between Gender groups?		X
Statistical Significance Found between Sport groups?	X	

For groups with significant differences, which group has the larger value?
Rugby>Swim Soccer>Swim Volleyball>Swim

NCMJ Normalized Peak Force (N/kg)

Gender	Sport	Mean	Standard Deviation	N
Male	Swim	20.898	1.784	9
	Volleyball	23.773	1.595	14
	Soccer	26.693	5.315	7
	Total	23.592	3.529	30
Female	Swim	18.999	0.480	5
	Rugby	22.567	3.389	20
	Soccer	23.500	2.243	21
	Total	22.605	2.987	46
Total	Swim	20.220	1.709	14
	Volleyball	23.773	1.595	14
	Rugby	22.567	3.389	20
	Soccer	23.835	1.991	28
	Total	22.596	2.748	76

NCMJ Normalized Peak Force (N/kg)

			95% Confidence Level			
Group A	Group B	Mean Difference (A-B)	Lower Bound	Upper Bound	P	
Soccer	Rugby	1.502	-0.652	3.655	0.266	
Swim	Rugby	-2.962	-5.525	-0.398	0.017	*
Volleyball	Rugby	0.210	-2.353	2.774	0.996	
Swim	Soccer	-4.463	-6.871	-2.056	0.000	*
Volleyball	Soccer	-1.291	-3.699	1.116	0.496	
Volleyball	Swim	3.172	0.392	5.952	0.019	*
Male	Female	0.979	-0.329	2.287	0.140	

*Mean difference is significant at α=0.05 level

NCMJ Peak Force Difference

Two-Way ANOVA Results		
	Yes	No
Statistical Evidence for Interaction Effect Between Gender and Sport?	X	
Statistical Significance Found between Gender groups?		X
Statistical Significance Found between Sport groups?		X

Interaction effect exists between gender and sport for peak force difference, so a Tukey HSD test was not completed after the Two-Way ANOVA.

NCMJ Peak Force Difference (%)

Gender	Sport	Mean	Standard Deviation	N
Male	Swim	5.740	4.087	9
	Volleyball	4.028	2.091	14
	Soccer	3.913	3.576	7
	Total	4.515	3.144	30

Female	Swim	1.925	1.323	5
	Rugby	5.823	4.310	20
	Soccer	6.355	2.981	21
	Total	5.642	3.706	46
Total	Swim	4.378	3.797	14
	Volleyball	4.028	2.091	14
	Rugby	5.823	4.310	20
	Soccer	6.289	3.100	28
	Total	5.197	3.565	76

NCMJ Squat Deviation

Two-Way ANOVA Results		
	Yes	No
Statistical Evidence for Interaction Effect Between Gender and Sport?		X
Statistical Significance Found between Gender groups?	X	
Statistical Significance Found between Sport groups?		X

Tukey HSD Test
For groups with significant differences, which group has the larger value?
Male>Female

NCMJ Squat Deviation

Gender	Sport	Mean	Standard Deviation	N
Male	Swim	11.213	4.618	9
	Volleyball	13.739	8.005	14
	Soccer	21.281	15.510	7
	Total	14.741	9.953	30
Female	Swim	6.540	2.177	5
	Rugby	11.921	5.238	20
	Soccer	10.019	6.740	21
	Total	10.468	5.911	46
Total	Swim	9.544	4.470	14
	Volleyball	13.739	8.005	14
	Rugby	11.921	5.238	20
	Soccer	8.381	2.711	28
	Total	10.995	5.665	76

NCMJ Squat Deviation

			95% Confidence Level			
Group A	Group B	Mean Difference (A-B)	Lower Bound	Upper Bound	P	
Soccer	Rugby	-0.155	-5.903	5.594	1.000	
Swim	Rugby	-5.124	-11.965	1.718	0.209	
Volleyball	Rugby	-2.455	-9.297	4.387	0.781	
Swim	Soccer	-4.969	-11.396	1.457	0.185	
Volleyball	Soccer	-2.300	-8.727	4.126	0.782	
Volleyball	Swim	2.669	-4.752	10.090	0.780	
Male	Female	4.273	0.782	7.765	0.017	*

*Mean difference is significant at α=0.05 level

Appendix I. NCMJ Jump Class Results

		Classification of Majority of NCMJs				
Gender	Sport	Class 1		Class 2		N
		#	(% of N)	#	(% of N)	
Male	Swim	0	(0.0)	9	(100.0)	9
	Volleyball	1	(7.1)	13	(92.9)	14
	Soccer	1	(14.3)	6	(85.7)	7

	Total	2	(6.7)	28	(93.3)	30
Female	Swim	0	(0.0)	5	(100.0)	5
	Rugby	7	(35.0)	13	(65.0)	20
	Soccer	3	(14.3)	18	(85.7)	21
	Total	10	(21.7)	36	(78.3)	46
Total	Swim	0	(0.0)	14	(100.0)	14
	Volleyball	1	(7.1)	13	(92.9)	14
	Rugby	7	(35.0)	13	(65.0)	20
	Soccer	4	(14.3)	24	(85.7)	28
	Total	12	(15.8)	64	(84.2)	76

Appendix J. NCMJ Phase Length Results

NCMJ Eccentric Phase Time

Two-Way ANOVA Results		
	Yes	No
Statistical Evidence for Interaction Effect Between Gender and Sport?		X
Statistical Significance Found between Gender groups?		X
Statistical Significance Found between Sport groups?		X

Tukey HSD Test
For groups with significant differences, which group has the larger value?
N/A

NCMJ Eccentric Phase Time (s)

Gender	Sport	Mean	Standard Deviation	N
Male	Swim	0.065	0.016	9
	Volleyball	0.053	0.017	14
	Soccer	0.058	0.022	7
	Total	0.058	0.018	30
Female	Swim	0.046	0.018	5
	Rugby	0.057	0.022	20
	Soccer	0.060	0.018	21
	Total	0.057	0.020	46
Total	Swim	0.058	0.019	14
	Volleyball	0.053	0.017	14
	Rugby	0.057	0.022	20
	Soccer	0.059	0.019	28
	Total	0.057	0.019	76

NCMJ Eccentric Phase Time (s)

			95% Confidence Level		
Group A	Group B	Mean Difference (A-B)	Lower Bound	Upper Bound	P
Soccer	Rugby	0.003	-0.012	0.018	0.964
Swim	Rugby	0.001	-0.016	0.019	0.998
Volleyball	Rugby	-0.005	-0.022	0.013	0.906
Swim	Soccer	-0.002	-0.018	0.015	0.995
Volleyball	Soccer	-0.007	-0.024	0.009	0.661
Volleyball	Swim	-0.006	-0.025	0.013	0.861
Male	Female	0.001	-0.008	0.010	0.854

*Mean difference is significant at α=0.05 level

NCMJ Concentric Phase Time

Two-Way ANOVA Results		
	Yes	No
Statistical Evidence for Interaction Effect Between Gender and Sport?		X

Tukey HSD Test
For groups with significant differences, which group has the larger value?

Statistical Significance Found between Gender groups?	X		Male>Female Swim>Rugby Swim>Soccer Swim>Volleyball
Statistical Significance Found between Sport groups?	X		

NCMJ Concentric Phase Time (s)

Gender	Sport	Mean	Standard Deviation	N
Male	Swim	0.404	0.102	9
	Volleyball	0.374	0.063	14
	Soccer	0.282	0.088	7
	Total	0.362	0.092	30
Female	Swim	0.492	0.075	5
	Rugby	0.303	0.082	20
	Soccer	0.290	0.057	21
	Total	0.317	0.093	46
Total	Swim	0.436	0.100	14
	Volleyball	0.374	0.063	14
	Rugby	0.303	0.082	20
	Soccer	0.294	0.058	28
	Total	0.347	0.095	76

NCMJ Concentric Phase Time (s)

			95% Confidence Level			
Group A	**Group B**	**Mean Difference (A-B)**	**Lower Bound**	**Upper Bound**	**P**	
Soccer	Rugby	-0.026	-0.084	0.032	0.638	
Swim	Rugby	0.104	0.035	0.173	0.001	*
Volleyball	Rugby	0.027	-0.042	0.096	0.737	
Swim	Soccer	0.130	0.066	0.195	0.000	*
Volleyball	Soccer	0.053	-0.012	0.118	0.148	
Volleyball	Swim	-0.077	-0.152	-0.003	0.040	*
Male	Female	0.044	0.009	0.079	0.015	*

*Mean difference is significant at α=0.05 level

NCMJ Time to Takeoff

Two-Way ANOVA Results		
	Yes	No
Statistical Evidence for Interaction Effect Between Gender and Sport?		X
Statistical Significance Found between Gender groups?	X	
Statistical Significance Found between Sport groups?	X	

Tukey HSD Test
For groups with significant differences, which group has the larger value?
Male>Female Swim>Rugby Swim>Soccer Swim>Volleyball

NCMJ Time to Takeoff (s)

Gender	Sport	Mean	Standard Deviation	N
Male	Swim	0.727	0.137	9
	Volleyball	0.603	0.074	14
	Soccer	0.527	0.141	7
	Total	0.623	0.132	30
Female	Swim	0.692	0.119	5
	Rugby	0.530	0.083	20
	Soccer	0.548	0.084	21
	Total	0.556	0.098	46
Total	Swim	0.714	0.127	14
	Volleyball	0.603	0.074	14

	Rugby	0.530	0.083	20
	Soccer	0.564	0.085	28
	Total	0.596	0.115	76

NCMJ Time to Takeoff (s)

			95% Confidence Level			
Group A	**Group B**	**Mean Difference (A-B)**	**Lower Bound**	**Upper Bound**	**P**	
Swim	Rugby	0.141	0.052	0.231	0.001	*
Volleyball	Rugby	0.006	-0.083	0.096	0.998	
Swim	Soccer	0.145	0.061	0.230	0.000	*
Volleyball	Soccer	0.010	-0.074	0.095	0.988	
Volleyball	Swim	-0.135	-0.232	-0.038	0.003	*
Male	Female	0.067	0.021	0.113	0.005	*
Swim	Rugby	0.141	0.052	0.231	0.001	*

*Mean difference is significant at α=0.05 level

NCMJ Time in Air

Two-Way ANOVA Results		
	Yes	No
Statistical Evidence for Interaction Effect Between Gender and Sport?		X
Statistical Significance Found between Gender groups?	X	
Statistical Significance Found between Sport groups?	X	

Tukey HSD Test
For groups with significant differences, which group has the larger value?
Male>Female Volleyball>Rugby Soccer>Swim Volleyball>Swim

NCMJ Time in Air (s)

Gender	Sport	Mean	Standard Deviation	N
Male	Swim	0.413	0.031	9
	Volleyball	0.485	0.043	14
	Soccer	0.435	0.042	7
	Total	0.452	0.050	30
Female	Swim	0.351	0.018	5
	Rugby	0.362	0.057	20
	Soccer	0.390	0.034	21
	Total	0.373	0.046	46
Total	Swim	0.391	0.041	14
	Volleyball	0.485	0.043	14
	Rugby	0.362	0.057	20
	Soccer	0.395	0.033	28
	Total	0.404	0.064	76

NCMJ Time in Air (s)

			95% Confidence Level			
Group A	**Group B**	**Mean Difference (A-B)**	**Lower Bound**	**Upper Bound**	**P**	
Swim	Rugby	-0.022	-0.061	0.018	0.473	
Volleyball	Rugby	0.045	0.006	0.084	0.019	*
Swim	Soccer	-0.041	-0.078	-0.004	0.023	*
Volleyball	Soccer	0.025	-0.012	0.062	0.281	
Volleyball	Swim	0.066	0.024	0.109	0.001	*
Male	Female	0.078	0.058	0.098	0.000	*
Swim	Rugby	-0.022	-0.061	0.018	0.473	

*Mean difference is significant at α=0.05 level

Appendix K. NCMJ Displacement and Velocity Results

NCMJ Takeoff Velocity

Two-Way ANOVA Results		
	Yes	No
Statistical Evidence for Interaction Effect Between Gender and Sport?		X
Statistical Significance Found between Gender groups?	X	
Statistical Significance Found between Sport groups?	X	

Tukey HSD Test
For groups with significant differences, which group has the larger value?
Male>Female Volleyball>Rugby Volleyball>Swim

NCMJ Takeoff Velocity (m/s)

Gender	Sport	Mean	Standard Deviation	N
Male	Swim	2.021	0.199	9
	Volleyball	2.382	0.296	14
	Soccer	2.089	0.192	7
	Total	2.205	0.295	30
Female	Swim	1.705	0.109	5
	Rugby	1.805	0.271	20
	Soccer	1.921	0.187	21
	Total	1.847	0.231	46
Total	Swim	1.908	0.229	14
	Volleyball	2.382	0.296	14
	Rugby	1.805	0.271	20
	Soccer	1.957	0.193	28
	Total	1.993	0.330	76

NCMJ Takeoff Velocity (m/s)

			95% Confidence Level			
Group A	Group B	Mean Difference (A-B)	Lower Bound	Upper Bound	P	
Soccer	Rugby	0.068	-0.112	0.248	0.751	
Swim	Rugby	-0.128	-0.342	0.087	0.404	
Volleyball	Rugby	0.219	0.004	0.433	0.044	*
Swim	Soccer	-0.196	-0.397	0.005	0.059	
Volleyball	Soccer	0.150	-0.051	0.352	0.211	
Volleyball	Swim	0.346	0.114	0.579	0.001	*
Male	Female	0.358	0.249	0.468	0.000	*

*Mean difference is significant at α=0.05 level

NCMJ Jump Height – TIA Method

Two-Way ANOVA Results		
	Yes	No
Statistical Evidence for Interaction Effect Between Gender and Sport?		X
Statistical Significance Found between Gender groups?	X	
Statistical Significance Found between Sport groups?	X	

Tukey HSD Test
For groups with significant differences, which group has the larger value?
Male>Female Volleyball>Rugby Soccer>Swim Volleyball>Swim

NCMJ Jump Height – TIA Method (m)

Gender	Sport	Mean	Standard Deviation	N
Male	Swim	0.210	0.032	9
	Volleyball	0.291	0.047	14
	Soccer	0.234	0.046	7
	Total	0.254	0.055	30
Female	Swim	0.151	0.015	5

Gender	Sport	Mean	Standard Deviation	N
	Rugby	0.165	0.047	20
	Soccer	0.188	0.032	21
	Total	0.174	0.040	46
Total	Swim	0.189	0.039	14
	Volleyball	0.291	0.047	14
	Rugby	0.165	0.047	20
	Soccer	0.193	0.032	28
	Total	0.205	0.063	76

NCMJ Jump Height – TIA Method (m)

			95% Confidence Level			
Group A	**Group B**	**Mean Difference (A-B)**	**Lower Bound**	**Upper Bound**	**P**	
Soccer	Rugby	0.014	-0.017	0.045	0.628	
Swim	Rugby	-0.027	-0.064	0.010	0.225	
Volleyball	Rugby	0.046	0.009	0.083	0.009	*
Swim	Soccer	-0.041	-0.076	-0.007	0.013	*
Volleyball	Soccer	0.032	-0.003	0.067	0.084	
Volleyball	Swim	0.073	0.033	0.113	0.000	*
Male	Female	0.079	0.061	0.098	0.000	*

*Mean difference is significant at α=0.05 level

NCMJ RSImod – Calculated with TIA

Two-Way ANOVA Results		
	Yes	No
Statistical Evidence for Interaction Effect Between Gender and Sport?		X
Statistical Significance Found between Gender groups?	X	
Statistical Significance Found between Sport groups?	X	

Tukey HSD Test
For groups with significant differences, which group has the larger value?
Male>Female Rugby>Swim Soccer>Swim Volleyball>Swim

NCMJ RSImod – Calculated with TIA

Gender	Sport	Mean	Standard Deviation	N
Male	Swim	0.306	0.073	9
	Volleyball	0.493	0.097	14
	Soccer	0.480	0.153	7
	Total	0.434	0.133	30
Female	Swim	0.231	0.046	5
	Rugby	0.322	0.101	20
	Soccer	0.355	0.079	21
	Total	0.327	0.093	46
Total	Swim	0.280	0.073	14
	Volleyball	0.493	0.097	14
	Rugby	0.322	0.101	20
	Soccer	0.356	0.084	28
	Total	0.359	0.118	76

NCMJ RSImod – Calculated with TIA

			95% Confidence Level			
Group A	**Group B**	**Mean Difference (A-B)**	**Lower Bound**	**Upper Bound**	**P**	
Soccer	Rugby	0.037	-0.036	0.110	0.544	
Swim	Rugby	-0.111	-0.199	-0.024	0.007	*
Volleyball	Rugby	0.064	-0.023	0.151	0.222	
Swim	Soccer	-0.149	-0.230	-0.067	0.000	*
Volleyball	Soccer	0.027	-0.055	0.109	0.822	
Volleyball	Swim	0.175	0.081	0.270	0.000	*

Male	Female	0.107	0.063	0.151	0.000	*

*Mean difference is significant at α=0.05 level

Appendix L. NCMJ Mathematica Code

The Mathematica code used to analyze the counter movement jump trials in this study is described here. The force-time data must be entered into the code as a matrix (obtained using the visual basic code described in Appendix F) into the Mathematica code line as, for example, “DataRun1Li=” for the matrix corresponding to Run 1 Left Leg. The horizontal lines indicate a separation of cells in Mathematica, such that each sectioned cell can be ran separately and will store the desired variables, and the last cell pulls the desired variables from each individual cell to output it in an organized manner. The cells are organized in order of Run 1 Left Leg (“Run1L”), Run 1 Right Leg (“Run1R”), Run 1 Both Legs (“Run1Both”), Run 2 Left Leg (“Run2L”), Run 2 Right Leg (“Run2R”)… etc. through to Run 5 Both Legs (“Run5Both”). These cells are identical, except for the jump trial upon which it is running, and the run number prefix for the variables. The last cell combines and outputs the desired variables for all of these combinations, and the code for the final cell that compiles all the variables, is as follows:

```
(*time point definitions;
t1 = onset of movement;
t2 = peak negative COM velocity (also force = body weight force;
t3 = zero COM velocity; t4 = takeoff; t5 = landing; tmax = time at \
peak force;
thalf=halfway between t1 and t4*)
```

```
(* Run1L *)
 DataRun1Li =};
DataRun1Li = DeleteCases[DataRun1Li, y_ /; y[[1]] > 4];
Clear[x, a, b, c, d]
DataRun1L = Table[DataRun1Li[[4 i]], {i, 1, Length[DataRun1Li]/4}];
f = DataRun1L[[All, 2]];
f = GaussianFilter[f, 10];
Datarefined =
 Table[{DataRun1L[[i, 1]], f[[i]]}, {i, 1, Length[DataRun1L]}];
funcRun1L =
 Interpolation[Datarefined, Method -> "Spline",
  InterpolationOrder -> 3];
funcRun1Lp = D[funcRun1L[x], x];
LL = Length[DataRun1L];
lastTimeRun1L = DataRun1L[[LL]][[1]];

Spline3[DataRun1L_] := (k = Length[DataRun1L];
 atable = Table[Subscript[a, i], {i, 1, k - 3}];
 btable = Table[Subscript[b, i], {i, 1, k - 3}];
 ctable = Table[Subscript[c, i], {i, 1, k - 3}];
 dtable = Table[Subscript[d, i], {i, 1, k - 3}];
 Poly = Table[
   atable[[i]] + btable[[i]]*x + ctable[[i]]*x^2 +
    dtable[[i]]*x^3, {i, 1, k - 3}];
 Polyd =
  Table[btable[[i]] + 2*ctable[[i]]*x + 3*dtable[[i]]*x^2, {i, 1,
    k - 3}];
 intermediate = Table[DataRun1L[[i + 2]], {i, 1, k - 4}];
 Equations = Table[0, {i, 1, 4 (k - 3)}];
 Do[Equations[[i]] = (Poly[[i]] /. x -> DataRun1L[[i + 2, 1]]) ==
    DataRun1L[[i + 2, 2]], {i, 1, k - 4}];
 Do[Equations[[
    i + k - 4]] = (Poly[[i + 1]] /. x -> DataRun1L[[i + 2, 1]]) ==
    DataRun1L[[i + 2, 2]], {i, 1, k - 4}];
 Equations[[
   2 (k - 4) + 1]] = (Poly[[1]] /. x -> DataRun1L[[1, 1]]) ==
   DataRun1L[[1, 2]];
 Equations[[
   2 (k - 4) + 2]] = (Poly[[1]] /. x -> DataRun1L[[2, 1]]) ==
   DataRun1L[[2, 2]];
 Equations[[
```

```
   2 (k - 4) + 3]] = (Poly[[k - 3]] /. x -> DataRun1L[[k - 1, 1]]) ==
    DataRun1L[[k - 1, 2]];
  Equations[[
   2 (k - 4) + 4]] = (Poly[[k - 3]] /. x -> DataRun1L[[k, 1]]) ==
   DataRun1L[[k, 2]];
  Do[Equations[[
    2 (k - 3) + 2 +
     i]] = (D[Poly[[i]], x] /.
      x -> intermediate[[i, 1]]) == (D[Poly[[i + 1]], x] /.
      x -> intermediate[[i, 1]]), {i, 1, k - 4}];
  Do[Equations[[
    2 (k - 3) + 2 + k - 4 +
     i]] = (D[Poly[[i]], {x, 2}] /.
      x -> intermediate[[i, 1]]) == (D[Poly[[i + 1]], {x, 2}] /.
      x -> intermediate[[i, 1]]), {i, 1, k - 4}];
  un = Flatten[{atable, btable, ctable, dtable}];
  f = Solve[Equations];
  Poly = Poly /. f[[1]];
  Polyd = Polyd /. f[[1]];
  xs = intermediate;
  xs = Prepend[Append[xs, DataRun1L[[k]]], DataRun1L[[1]]];
  pf = Table[{Poly[[i]], xs[[i, 1]] <= x <= xs[[i + 1, 1]]}, {i, 1,
     k - 3}];
  pdf = Table[{Polyd[[i]], xs[[i, 1]] <= x <= xs[[i + 1, 1]]}, {i, 1,
     k - 3}];
  sol = {Piecewise[pf], Piecewise[pdf]})
Timing[y = Spline3[Datarefined]];

(*Max force and tmax it occurs*)

globalmintimeRun1L = ArgMin[funcRun1L[x], 2 < x < lastTimeRun1L, x];
globalminRun1L = MinValue[funcRun1L[x], 2 < x < lastTimeRun1L, x];
peakMaxRun1L = MaxValue[funcRun1L[x], 2 < x < globalmintimeRun1L, x];
tmaxRun1L = ArgMax[funcRun1L[x], 2 < x < globalmintimeRun1L, x];
(* t4Run1L *)

t4Run1L = ArgMin[funcRun1L[x], tmaxRun1L < x < lastTimeRun1L, x];
While[funcRun1L[t4Run1L] <= Max[0, (1 + globalminRun1L)],
  t4Run1L = ArgMin[funcRun1L[x], tmaxRun1L < x < (t4Run1L - 0.01), x]];
(* t5Run1L *)

LocalMaxx5 =
  ArgMax[{funcRun1L[x], t4Run1L + 0.2 <= x <= lastTimeRun1L - 0.1}, x];
LocalMax5 = funcRun1L[LocalMaxx5];
mlist = Solve[
   y[[2]] == funcRun1L[0]*0.05 && x > t4Run1L + 0.2 &&
    x < LocalMaxx5 && funcRun1L[x] < LocalMax5*0.1, x];
minlist = x /. mlist;
t5Run1L = Last[minlist];
(*force while in air*)

noforceRun1L =
  NIntegrate[
    funcRun1L[
     x], {x, (t4Run1L + 0.05), (t5Run1L - 0.05)}]/((t5Run1L -
       0.05) - (t4Run1L + 0.05));
(* t0Run1L - to find quiet stance mass*)

earlyGlobalMin = ArgMin[funcRun1L[x], 0.1 < x < tmaxRun1L, x];
slope = (funcRun1L[0] - funcRun1L[earlyGlobalMin])/(0 -
      earlyGlobalMin) + 25;
(* t0list is list of times where derivative = defined threshold*)

t0list = Solve[y[[2]] == slope, x];
v = t0list[[1]][[1]];
t0Run1L = (x /. v);
```

```
(*find quiet stance mass by taking average from t=0 to just before \
the jump "starts", but only if it appears to be relatively steady*)

BWforceRun1L =
  NIntegrate[
    funcRun1L[x], {x, 0.05, (t0Run1L - 0.1)}]/((t0Run1L - 0.1) - 0.05);
If[(funcRun1L[0] - funcRun1L[(t0Run1L - 0.1)])/(0 - (t0Run1L - 0.1)) >
    40, (Print["->Issue with quiet stance mass"])];
If[t0Run1L <= 0.15, Print["Check mass calc."]];
massRun1L = BWforceRun1L/9.81;
If[Abs[noforceRun1L] > 0.05*BWforceRun1L,
  Print["Possible error: force while subject is in air 5% body \
weight"]];

(* t1Run1L *)

t1localmin =
  ArgMin[funcRun1L[x], (tmaxRun1L - 0.3) < x < tmaxRun1L, x];
Clear[findAllRoots]
SyntaxInformation[
   findAllRoots] = {"LocalVariables" -> {"Plot", {2, 2}},
   "ArgumentsPattern" -> {_, _, OptionsPattern[]}};
SetAttributes[findAllRoots, HoldAll];
Options[findAllRoots] =
  Join[{"ShowPlot" -> False, PlotRange -> All},
   FilterRules[Options[Plot], Except[PlotRange]]];
findAllRoots[fn_, {l_, lmin_, lmax_}, opts : OptionsPattern[]] :=
 Module[{pl, p, x, localFunction, brackets},
  localFunction = ReleaseHold[Hold[fn] /. HoldPattern[l] :> x];
  If[lmin != lmax,
   pl = Plot[localFunction, {x, lmin, lmax},
     Evaluate@
      FilterRules[Join[{opts}, Options[findAllRoots]], Options[Plot]]];
   p = Cases[pl, Line[{x__}] :> x, Infinity];
   If[OptionValue["ShowPlot"],
    Print[Show[pl, PlotLabel -> "Finding roots for this function",
      ImageSize -> 200, BaseStyle -> {FontSize -> 8}]]], p = {}];
  brackets =
   Map[First,
    Select[(*This Split trick pretends that two points on the curve \
are "equal" if the function values have _opposite _ sign.Pairs of \
such sign-changes form the brackets for the subsequent FindRoot*)
     Split[p, Sign[Last[#2]] == -Sign[Last[#1]] &],
     Length[#1] == 2 &], {2}];
  x /. Apply[FindRoot[localFunction == 0, {x, ##1}] &,
     brackets, {1}] /. x -> {}]
rootfuncRun1L[t_] := funcRun1L"[x] /. x -> t;
t1rootlist =
  findAllRoots[rootfuncRun1L[x], {x, (tmaxRun1L - 0.5), t1localmin},
   "ShowPlot" -> False];
t1rootlistlength = Length[t1rootlist];
While[funcRun1L[t1rootlist[[t1rootlistlength]]] <
   funcRun1L[t1rootlist[[t1rootlistlength - 1]]],
  t1rootlistlength = t1rootlistlength - 1];
t1bound = t1rootlist[[t1rootlistlength]];

t1localmin2 = ArgMin[funcRun1L[x], t1bound < x < tmaxRun1L, x];
slope2 = (funcRun1L[t1bound] - funcRun1L[t1localmin2])/(t1bound -
      t1localmin2) + 25;
t1list = Solve[
   y[[2]] == slope2 &&
    x > (t1localmin - (tmaxRun1L - t1localmin) - 0.05) &&
    x < t1localmin, x];
v = t1list[[1]][[1]];
t1Run1L = (x /. v);

cutoffRun1L = t1Run1L - 0.2;
```

```
thalfRun1L = (0.5*(t4Run1L - t1Run1L)) + t1Run1L;

(*acceleration, velocity, displacement, power*)

accelRun1L[x_] := (funcRun1L[x] - BWforceRun1L)/massRun1L;
velocityRun1L =
  NDSolveValue[{accelRun1L[x] == f2'[x], f2[t1Run1L] == 0},
   f2[x], {x, 0, lastTimeRun1L}];
displacementRun1L =
  NDSolveValue[{accelRun1L[x] == g2''[x],
    g2[t1Run1L] == g2'[t1Run1L] == 0}, g2[x], {x, 0, lastTimeRun1L}];
powerRun1L = funcRun1L[x]*velocityRun1L;

(* t2Run1L *)

t2Run1L = ArgMin[velocityRun1L, t1Run1L < x < tmaxRun1L, x];
(* t3Run1L *)

t3Run1L = ArgMin[displacementRun1L, t2Run1L < x < t4Run1L, x];

(*tsquat \[Rule] average and standard deviation of force during squat*)

tearlymaxRun1L = ArgMax[funcRun1L[x], t0Run1L < x < t1Run1L, x];
tsquatlist =
  findAllRoots[rootfuncRun1L[x], {x, tearlymaxRun1L, cutoffRun1L},
   "ShowPlot" -> False];
tpos = 1;
While[funcRun1L[tsquatlist[[tpos]]] >
   funcRun1L[tsquatlist[[tpos + 1]]], tpos = tpos + 1];
tsquatRun1L = tsquatlist[[tpos]];
squatlistRun1L = DeleteCases[Datarefined, y_ /; y[[1]] > t1Run1L];
squatlistRun1L =
  DeleteCases[squatlistRun1L, y_ /; y[[1]] < tsquatRun1L];
squatlistRun1L = squatlistRun1L[[All, 2]];
squatavgRun1L = Mean[squatlistRun1L];
squatstdevRun1L = StandardDeviation[squatlistRun1L];

(*outputs*)
force1Run1L = funcRun1L[t1Run1L];
velocity1Run1L = velocityRun1L /. x -> t1Run1L;
displacement1Run1L = displacementRun1L /. x -> t1Run1L;
power1Run1L = powerRun1L /. x -> t1Run1L;
force2Run1L = funcRun1L[t2Run1L];
velocity2Run1L = velocityRun1L /. x -> t2Run1L;
displacement2Run1L = displacementRun1L /. x -> t2Run1L;
power2Run1L = powerRun1L /. x -> t2Run1L;
force3Run1L = funcRun1L[t3Run1L];
velocity3Run1L = velocityRun1L /. x -> t3Run1L;
displacement3Run1L = displacementRun1L /. x -> t3Run1L;
power3Run1L = powerRun1L /. x -> t3Run1L;
force4Run1L = funcRun1L[t4Run1L];
velocity4Run1L = velocityRun1L /. x -> t4Run1L;
displacement4Run1L = displacementRun1L /. x -> t4Run1L;
power4Run1L = powerRun1L /. x -> t4Run1L;
force5Run1L = funcRun1L[t5Run1L];
velocity5Run1L = velocityRun1L /. x -> t5Run1L;
displacement5Run1L = displacementRun1L /. x -> t5Run1L;
power5Run1L = powerRun1L /. x -> t5Run1L;
forcemaxRun1L = funcRun1L[tmaxRun1L];
velocitymaxRun1L = velocityRun1L /. x -> tmaxRun1L;
displacementmaxRun1L = displacementRun1L /. x -> tmaxRun1L;
powermaxRun1L = powerRun1L /. x -> tmaxRun1L;
forcehalfRun1L = funcRun1L[thalfRun1L];
velocityhalfRun1L = velocityRun1L /. x -> thalfRun1L;
displacementhalfRun1L = displacementRun1L /. x -> thalfRun1L;
powerhalfRun1L = powerRun1L /. x -> thalfRun1L;
```

```
impulse12Run1L = NIntegrate[funcRun1L[x], {x, t1Run1L, t2Run1L}];
impulse23Run1L = NIntegrate[funcRun1L[x], {x, t2Run1L, t3Run1L}];
impulse34Run1L = NIntegrate[funcRun1L[x], {x, t3Run1L, t4Run1L}];
impulse45Run1L = NIntegrate[funcRun1L[x], {x, t4Run1L, t5Run1L}];
impulse2maxRun1L = NIntegrate[funcRun1L[x], {x, t2Run1L, tmaxRun1L}];
impulse1halfRun1L = NIntegrate[funcRun1L[x], {x, t1Run1L, thalfRun1L}];

maxforce12Run1L = MaxValue[{funcRun1L[x], t1Run1L <= x <= t2Run1L}, x];
minforce12Run1L = MinValue[{funcRun1L[x], t1Run1L <= x <= t2Run1L}, x];
maxvelocity12Run1L =
  MaxValue[{velocityRun1L, t1Run1L <= x <= t2Run1L}, x];
minvelocity12Run1L =
  MinValue[{velocityRun1L, t1Run1L <= x <= t2Run1L}, x];
maxdisplacement12Run1L =
  MaxValue[{displacementRun1L, t1Run1L <= x <= t2Run1L}, x];
mindisplacement12Run1L =
  MinValue[{displacementRun1L, t1Run1L <= x <= t2Run1L}, x];
maxpower12Run1L = MaxValue[{powerRun1L, t1Run1L <= x <= t2Run1L}, x];
minpower12Run1L = MinValue[{powerRun1L, t1Run1L <= x <= t2Run1L}, x];

maxforce23Run1L = MaxValue[{funcRun1L[x], t2Run1L <= x <= t3Run1L}, x];
minforce23Run1L = MinValue[{funcRun1L[x], t2Run1L <= x <= t3Run1L}, x];
maxvelocity23Run1L =
  MaxValue[{velocityRun1L, t2Run1L <= x <= t3Run1L}, x];
minvelocity23Run1L =
  MinValue[{velocityRun1L, t2Run1L <= x <= t3Run1L}, x];
maxdisplacement23Run1L =
  MaxValue[{displacementRun1L, t2Run1L <= x <= t3Run1L}, x];
mindisplacement23Run1L =
  MinValue[{displacementRun1L, t2Run1L <= x <= t3Run1L}, x];
maxpower23Run1L = MaxValue[{powerRun1L, t2Run1L <= x <= t3Run1L}, x];
minpower23Run1L = MinValue[{powerRun1L, t2Run1L <= x <= t3Run1L}, x];

maxforce34Run1L = MaxValue[{funcRun1L[x], t3Run1L <= x <= t4Run1L}, x];
minforce34Run1L = MinValue[{funcRun1L[x], t3Run1L <= x <= t4Run1L}, x];
maxvelocity34Run1L =
  MaxValue[{velocityRun1L, t3Run1L <= x <= t4Run1L}, x];
minvelocity34Run1L =
  MinValue[{velocityRun1L, t3Run1L <= x <= t4Run1L}, x];
maxdisplacement34Run1L =
  MaxValue[{displacementRun1L, t3Run1L <= x <= t4Run1L}, x];
mindisplacement34Run1L =
  MinValue[{displacementRun1L, t3Run1L <= x <= t4Run1L}, x];
maxpower34Run1L = MaxValue[{powerRun1L, t3Run1L <= x <= t4Run1L}, x];
minpower34Run1L = MinValue[{powerRun1L, t3Run1L <= x <= t4Run1L}, x];

maxforce45Run1L = MaxValue[{funcRun1L[x], t4Run1L <= x <= t5Run1L}, x];
minforce45Run1L = MinValue[{funcRun1L[x], t4Run1L <= x <= t5Run1L}, x];
maxvelocity45Run1L =
  MaxValue[{velocityRun1L, t4Run1L <= x <= t5Run1L}, x];
minvelocity45Run1L =
  MinValue[{velocityRun1L, t4Run1L <= x <= t5Run1L}, x];
maxdisplacement45Run1L =
  MaxValue[{displacementRun1L, t4Run1L <= x <= t5Run1L}, x];
mindisplacement45Run1L =
  MinValue[{displacementRun1L, t4Run1L <= x <= t5Run1L}, x];
maxpower45Run1L = MaxValue[{powerRun1L, t4Run1L <= x <= t5Run1L}, x];
minpower45Run1L = MinValue[{powerRun1L, t4Run1L <= x <= t5Run1L}, x];

maxforce2maxRun1L =
  MaxValue[{funcRun1L[x], t2Run1L <= x <= tmaxRun1L}, x];
minforce2maxRun1L =
  MinValue[{funcRun1L[x], t2Run1L <= x <= tmaxRun1L}, x];
maxvelocity2maxRun1L =
  MaxValue[{velocityRun1L, t2Run1L <= x <= tmaxRun1L}, x];
minvelocity2maxRun1L =
  MinValue[{velocityRun1L, t2Run1L <= x <= tmaxRun1L}, x];
```

```
maxdisplacement2maxRun1L =
  MaxValue[{displacementRun1L, t2Run1L <= x <= tmaxRun1L}, x];
mindisplacement2maxRun1L =
  MinValue[{displacementRun1L, t2Run1L <= x <= tmaxRun1L}, x];
maxpower2maxRun1L =
  MaxValue[{powerRun1L, t2Run1L <= x <= tmaxRun1L}, x];
minpower2maxRun1L =
  MinValue[{powerRun1L, t2Run1L <= x <= tmaxRun1L}, x];

maxforce1halfRun1L =
  MaxValue[{funcRun1L[x], t1Run1L <= x <= thalfRun1L}, x];
minforce1halfRun1L =
  MinValue[{funcRun1L[x], t1Run1L <= x <= thalfRun1L}, x];
maxvelocity1halfRun1L =
  MaxValue[{velocityRun1L, t1Run1L <= x <= thalfRun1L}, x];
minvelocity1halfRun1L =
  MinValue[{velocityRun1L, t1Run1L <= x <= thalfRun1L}, x];
maxdisplacement1halfRun1L =
  MaxValue[{displacementRun1L, t1Run1L <= x <= thalfRun1L}, x];
mindisplacement1halfRun1L =
  MinValue[{displacementRun1L, t1Run1L <= x <= thalfRun1L}, x];
maxpower1halfRun1L =
  MaxValue[{powerRun1L, t1Run1L <= x <= thalfRun1L}, x];
minpower1halfRun1L =
  MinValue[{powerRun1L, t1Run1L <= x <= thalfRun1L}, x];

(*TableRun1L={{"mass= ",massRun1L},{},
{"t1= ",t1Run1L,"force @ t1= ",force1Run1L,"veloc. @ t1= \
",velocity1Run1L,"displace. @ t1= ",displacement1Run1L,"power @ t1= \
",power1Run1L},{"t2= ",t2Run1L,"force @ t2= ",force2Run1L,"veloc. @ \
t2= ",velocity2Run1L,"displace. @ t2= ",displacement2Run1L,"power @ \
t2= ",power2Run1L},{"t3= ",t3Run1L,"force @ t3= ",force3Run1L,"veloc. \
@ t3= ",velocity3Run1L,"displace. @ t3= ",displacement3Run1L,"power @ \
t3= ",power3Run1L},{"t4= ",t4Run1L,"force @ t4= ",force4Run1L,"veloc. \
@ t4= ",velocity4Run1L,"displace. @ t4= ",displacement4Run1L,"power @ \
t4= ",power4Run1L},{"t5= ",t5Run1L,"force @ t5= ",force5Run1L,"veloc. \
@ t5= ",velocity5Run1L,"displace. @ t5= ",displacement5Run1L,"power @ \
t5= ",power5Run1L},{"tmax= ",tmaxRun1L,"force @ tmax= \
",forcemaxRun1L,"veloc. @ tmax= ",velocitymaxRun1L,"displace. @ tmax= \
",displacementmaxRun1L,"power @ tmax= ",powermaxRun1L},{"thalf= \
",thalfRun1L,"force @ thalf= ",forcehalfRun1L,"veloc. @ thalf= \
",velocityhalfRun1L,"displace. @ thalf= \
",displacementhalfRun1L,"power @ thalf= \
",powerhalfRun1L},{},{"impulse t1-t2= ",impulse12Run1L,"impulse \
t2-t3= ",impulse23Run1L,"impulse t3-t4= ",impulse34Run1L,"impulse \
t4-t5= ",impulse45Run1L,"impulse t2-tmax= ",impulse2maxRun1L},{"max \
force t1-t2= ",maxforce12Run1L,"max force t2-t3= \
",maxforce23Run1L,"max force t3-t4= ",maxforce34Run1L,"max force \
t4-t5= ",maxforce45Run1L,"max force t2-tmax= ",maxforce2maxRun1L,"max \
force t1-thalf= ",maxforce1halfRun1L},{"min force t1-t2= \
",minforce12Run1L,"min force t2-t3= ",minforce23Run1L,"min force \
t3-t4= ",minforce34Run1L,"min force t4-t5= ",minforce45Run1L,"min \
force t2-tmax= ",minforce2maxRun1L,"min force t1-thalf= \
",minforce1halfRun1L},{"max veloc. t1-t2= ",maxvelocity12Run1L,"max \
veloc. t2-t3= ",maxvelocity23Run1L,"max veloc. t3-t4= \
",maxvelocity34Run1L,"max veloc. t4-t5= ",maxvelocity45Run1L,"max \
veloc. t2-tmax= ",maxvelocity2maxRun1L,"max veloc. t1-thalf= \
",maxvelocity1halfRun1L},{"min veloc. t1-t2= \
",minvelocity12Run1L,"min veloc. t2-t3= ",minvelocity23Run1L,"min \
veloc. t3-t4= ",minvelocity34Run1L,"min veloc. t4-t5= \
",minvelocity45Run1L,"min veloc. t2-tmax= ",minvelocity2maxRun1L,"min \
veloc. t1-thalf= ",minvelocity1halfRun1L},{"max displace. t1-t2= \
",maxdisplacement12Run1L,"max displace. t2-t3= \
",maxdisplacement23Run1L,"max displace. t3-t4= \
",maxdisplacement34Run1L,"max displace. t4-t5= \
",maxdisplacement45Run1L,"max displace. t2-tmax= \
",maxdisplacement2maxRun1L,"max displace. t1-thalf= \
```

```
",maxdisplacement1halfRun1L},{"min displace. t1-t2= \
",mindisplacement12Run1L,"min displace. t2-t3= \
",mindisplacement23Run1L,"min displace. t3-t4= \
",mindisplacement34Run1L,"min displace. t4-t5= \
",mindisplacement45Run1L,"min displace. t2-tmax= \
",mindisplacement2maxRun1L,"min displace. t1-thalf= \
",mindisplacement1halfRun1L},{"max power t1-t2= \
",maxpower12Run1L,"max power t2-t3= ",maxpower23Run1L,"max power \
t3-t4= ",maxpower34Run1L,"max power t4-t5= ",maxpower45Run1L,"max \
power t2-tmax= ",maxpower2maxRun1L,"max power t1-thalf= \
",maxpower1halfRun1L},
{"min power t1-t2= ",minpower12Run1L,"min power t2-t3= \
",minpower23Run1L,"min power t3-t4= ",minpower34Run1L,"min power \
t4-t5= ",minpower45Run1L,"min power t2-tmax= ",minpower2maxRun1L,"min \
power t1-thalf= ",minpower1halfRun1L}} ;
Grid[TableRun1L,Frame\[Rule]All]*)

(*classify the type of NCMJ based on number of peaks--look at the \
number of maxima/minima of the first derivative*)

localrootsRun1L =
  Length[findAllRoots[rootfuncRun1L[x], {x, t3Run1L, t4Run1L},
    "ShowPlot" -> False]];
If[localrootsRun1L == 1, {runtypeRun1L = 1,
   Print["NCMJ Type 1 (one peak)"]}, {runtypeRun1L = 2,
   Print["NCMJ Type 2 (two or more peaks)"]}];

Print["tsquat= ", tsquatRun1L]
Print["t1= ", t1Run1L]
Print["t2= ", t2Run1L]
Print["t3= ", t3Run1L]
Print["t4= ", t4Run1L]
Print["t5= ", t5Run1L]
Print["thalf= ", thalfRun1L]
Print["tmax= ", tmaxRun1L]

line1Run1L = Line[{{t1Run1L, -5000}, {t1Run1L, 5000}}];
line2Run1L = Line[{{t2Run1L, -5000}, {t2Run1L, 5000}}];
line3Run1L = Line[{{t3Run1L, -5000}, {t3Run1L, 5000}}];
line4Run1L = Line[{{t4Run1L, -5000}, {t4Run1L, 5000}}];
line5Run1L = Line[{{t5Run1L, -5000}, {t5Run1L, 5000}}];
linehalfRun1L = Line[{{thalfRun1L, -5000}, {thalfRun1L, 5000}}];
linemaxRun1L = Line[{{tmaxRun1L, -5000}, {tmaxRun1L, 5000}}];
lineStyle = {Thin, Red, Dashed};
bodyweightlineRun1L =
  Line[{{0, BWforceRun1L}, {tmaxRun1L, BWforceRun1L}}];
cutofflineRun1L = Line[{{cutoffRun1L, -5000}, {cutoffRun1L, 5000}}];
squatavglineRun1L =
  Line[{{0, squatavgRun1L}, {cutoffRun1L, squatavgRun1L}}];
squatlineRun1L = Line[{{tsquatRun1L, -5000}, {tsquatRun1L, 5000}}];

Print["Body Weight= ", BWforceRun1L]
Print["Force in Air= ", noforceRun1L]
Plot[{funcRun1L[x]}, {x, cutoffRun1L, lastTimeRun1L},
 ImageSize -> 600, PlotStyle -> {Automatic},
 Epilog -> {Directive[lineStyle], line1Run1L, line2Run1L, line3Run1L,
   line4Run1L, line5Run1L, Directive[Thin, Blue, Dashed],
   linehalfRun1L, linemaxRun1L, Directive[Thin, Black, Dashed],
   bodyweightlineRun1L}, PlotRange -> All,
 BaseStyle -> { FontSize -> 12}, PlotLabel -> "NCMJ - Normal Force",
 AxesLabel -> {"Time (s)", "Normal Force (N)"} ]
Plot[{velocityRun1L}, {x, cutoffRun1L, lastTimeRun1L},
 ImageSize -> 600, PlotStyle -> {Automatic},
 Epilog -> {Directive[lineStyle], line1Run1L, line2Run1L, line3Run1L,
   line4Run1L, line5Run1L, Directive[Thin, Blue, Dashed],
   linehalfRun1L, linemaxRun1L, Directive[Thin, Black, Dashed],
   bodyweightlineRun1L}, PlotRange -> All,
```

```
 BaseStyle -> { FontSize -> 12}, PlotLabel -> "NCMJ - Velocity",
 AxesLabel -> {"Time (s)", "Velocity {m/s}"} ]
Plot[{funcRun1L[x]}, {x, 0, lastTimeRun1L}, ImageSize -> 600,
 PlotStyle -> {Automatic},
 Epilog -> {Directive[lineStyle], line1Run1L, line2Run1L, line3Run1L,
   line4Run1L, line5Run1L, Directive[Thin, Blue, Dashed],
   linehalfRun1L, linemaxRun1L, Directive[Thin, Black, Dashed],
   cutofflineRun1L, bodyweightlineRun1L, squatavglineRun1L,
   squatlineRun1L}, PlotRange -> All, BaseStyle -> { FontSize -> 12},
 PlotLabel -> "NCMJ - Normal Force",
 AxesLabel -> {"Time (s)", "Normal Force (N)"} ]
(*Plot[{velocityRun1L},{x,0,lastTimeRun1L},ImageSize\[Rule]600,\
PlotStyle \[Rule] \
{Automatic},Epilog\[Rule]{Directive[lineStyle],line1Run1L,line2Run1L,\
line3Run1L,line4Run1L,line5Run1L, \
Directive[Thin,Blue,Dashed],linehalfRun1L,linemaxRun1L, \
Directive[Thin,Black,Dashed],bodyweightlineRun1L}, \
PlotRange\[Rule]All, BaseStyle\[Rule]{ FontSize\[Rule] 12},PlotLabel\
\[Rule] "NCMJ - Velocity", AxesLabel\[Rule]{"Time (s)", "Velocity \
(m/s)"} ]
Plot[{displacementRun1L},{x,0,lastTimeRun1L},ImageSize\[Rule]600,\
PlotStyle \[Rule] \
{Automatic},Epilog\[Rule]{Directive[lineStyle],line1Run1L,line2Run1L,\
line3Run1L,line4Run1L,line5Run1L, \
Directive[Thin,Blue,Dashed],linehalfRun1L,linemaxRun1L},PlotRange\
\[Rule]All, BaseStyle\[Rule]{ FontSize\[Rule] 12},PlotLabel\[Rule] \
"NCMJ - COM Displacement", AxesLabel\[Rule]{"Time (s)", "Normal Force \
(N)"} ]
Plot[{powerRun1L},{x,0,lastTimeRun1L},ImageSize\[Rule]600,PlotStyle \
\[Rule] {Automatic},Epilog\[Rule]{Directive[lineStyle],line1Run1L,\
line2Run1L,line3Run1L,line4Run1L,line5Run1L, \
Directive[Thin,Blue,Dashed],linehalfRun1L,linemaxRun1L, \
Directive[Thin,Black,Dashed],bodyweightlineRun1L,},PlotRange\[Rule]\
All, BaseStyle\[Rule]{ FontSize\[Rule] 12},PlotLabel\[Rule] "NCMJ - \
Power", AxesLabel\[Rule]{"Time (s)", "Normal Force (N)"} ]*)
```

```
(* Run1R *)
  DataRun1Ri =};
DataRun1Ri = DeleteCases[DataRun1Ri, y_ /; y[[1]] > 4];
Clear[x, a, b, c, d]
DataRun1R = Table[DataRun1Ri[[4 i]], {i, 1, Length[DataRun1Ri]/4}];
f = DataRun1R[[All, 2]];
f = GaussianFilter[f, 10];
Datarefined =
  Table[{DataRun1R[[i, 1]], f[[i]]}, {i, 1, Length[DataRun1R]}];
funcRun1R =
  Interpolation[Datarefined, Method -> "Spline",
   InterpolationOrder -> 3];
funcRun1Rp = D[funcRun1R[x], x];
LL = Length[DataRun1R];
lastTimeRun1R = DataRun1R[[LL]][[1]];

Spline3[DataRun1R_] := (k = Length[DataRun1R];
  atable = Table[Subscript[a, i], {i, 1, k - 3}];
  btable = Table[Subscript[b, i], {i, 1, k - 3}];
  ctable = Table[Subscript[c, i], {i, 1, k - 3}];
  dtable = Table[Subscript[d, i], {i, 1, k - 3}];
  Poly = Table[
    atable[[i]] + btable[[i]]*x + ctable[[i]]*x^2 +
     dtable[[i]]*x^3, {i, 1, k - 3}];
  Polyd =
   Table[btable[[i]] + 2*ctable[[i]]*x + 3*dtable[[i]]*x^2, {i, 1,
     k - 3}];
  intermediate = Table[DataRun1R[[i + 2]], {i, 1, k - 4}];
  Equations = Table[0, {i, 1, 4 (k - 3)}];
  Do[Equations[[i]] = (Poly[[i]] /. x -> DataRun1R[[i + 2, 1]]) ==
     DataRun1R[[i + 2, 2]], {i, 1, k - 4}];
  Do[Equations[[
```

```
    i + k - 4]] = (Poly[[i + 1]] /. x -> DataRun1R[[i + 2, 1]]) ==
    DataRun1R[[i + 2, 2]], {i, 1, k - 4}];
  Equations[[
   2 (k - 4) + 1]] = (Poly[[1]] /. x -> DataRun1R[[1, 1]]) ==
   DataRun1R[[1, 2]];
  Equations[[
   2 (k - 4) + 2]] = (Poly[[1]] /. x -> DataRun1R[[2, 1]]) ==
   DataRun1R[[2, 2]];
  Equations[[
   2 (k - 4) + 3]] = (Poly[[k - 3]] /. x -> DataRun1R[[k - 1, 1]]) ==
    DataRun1R[[k - 1, 2]];
  Equations[[
   2 (k - 4) + 4]] = (Poly[[k - 3]] /. x -> DataRun1R[[k, 1]]) ==
   DataRun1R[[k, 2]];
  Do[Equations[[
    2 (k - 3) + 2 +
     i]] = (D[Poly[[i]], x] /.
      x -> intermediate[[i, 1]]) == (D[Poly[[i + 1]], x] /.
      x -> intermediate[[i, 1]]), {i, 1, k - 4}];
  Do[Equations[[
    2 (k - 3) + 2 + k - 4 +
     i]] = (D[Poly[[i]], {x, 2}] /.
      x -> intermediate[[i, 1]]) == (D[Poly[[i + 1]], {x, 2}] /.
      x -> intermediate[[i, 1]]), {i, 1, k - 4}];
  un = Flatten[{atable, btable, ctable, dtable}];
  f = Solve[Equations];
  Poly = Poly /. f[[1]];
  Polyd = Polyd /. f[[1]];
  xs = intermediate;
  xs = Prepend[Append[xs, DataRun1R[[k]]], DataRun1R[[1]]];
  pf = Table[{Poly[[i]], xs[[i, 1]] <= x <= xs[[i + 1, 1]]}, {i, 1,
    k - 3}];
  pdf = Table[{Polyd[[i]], xs[[i, 1]] <= x <= xs[[i + 1, 1]]}, {i, 1,
    k - 3}];
  sol = {Piecewise[pf], Piecewise[pdf]})
Timing[y = Spline3[Datarefined]];

(*Max force and tmax it occurs*)

globalmintimeRun1R = ArgMin[funcRun1R[x], 2 < x < lastTimeRun1R, x];
globalminRun1R = MinValue[funcRun1R[x], 2 < x < lastTimeRun1R, x];
peakMaxRun1R = MaxValue[funcRun1R[x], 2 < x < globalmintimeRun1R, x];
tmaxRun1R = ArgMax[funcRun1R[x], 2 < x < globalmintimeRun1R, x];
(* t4Run1R *)

t4Run1R = ArgMin[funcRun1R[x], tmaxRun1R < x < lastTimeRun1R, x];
While[funcRun1R[t4Run1R] <= Max[0, (1 + globalminRun1R)],
  t4Run1R = ArgMin[funcRun1R[x], tmaxRun1R < x < (t4Run1R - 0.01), x]];
(* t5Run1R *)

LocalMaxx5 =
  ArgMax[{funcRun1R[x], t4Run1R + 0.2 <= x <= lastTimeRun1R - 0.1}, x];
LocalMax5 = funcRun1R[LocalMaxx5];
mlist = Solve[
   y[[2]] == funcRun1R[0]*0.05 && x > t4Run1R + 0.2 &&
    x < LocalMaxx5 && funcRun1R[x] < LocalMax5*0.1, x];
minlist = x /. mlist;
t5Run1R = Last[minlist];
(*force while in air*)

noforceRun1R =
  NIntegrate[
   funcRun1R[
    x], {x, (t4Run1R + 0.05), (t5Run1R - 0.05)}]/((t5Run1R -
      0.05) - (t4Run1R + 0.05));
(* t0Run1R - to find quiet stance mass*)
```

```
earlyGlobalMin = ArgMin[funcRun1R[x], 0.1 < x < tmaxRun1R, x];
slope = (funcRun1R[0] - funcRun1R[earlyGlobalMin])/(0 -
      earlyGlobalMin) + 25;
(* t0list is list of times where derivative = defined threshold*)

t0list = Solve[y[[2]] == slope, x];
v = t0list[[1]][[1]];
t0Run1R = (x /. v);

(*find quiet stance mass by taking average from t=0 to just before \
the jump "starts", but only if it appears to be relatively steady*)

BWforceRun1R =
  NIntegrate[
    funcRun1R[x], {x, 0.05, (t0Run1R - 0.1)}]/((t0Run1R - 0.1) - 0.05);
If[(funcRun1R[0] - funcRun1R[(t0Run1R - 0.1)])/(0 - (t0Run1R - 0.1)) >
    40, (Print["->Issue with quiet stance mass"])];
If[t0Run1R <= 0.15, Print["Check mass calc."]];
massRun1R = BWforceRun1R/9.81;
If[Abs[noforceRun1R] > 0.05*BWforceRun1R,
  Print["Possible error: force while subject is in air 5% body \
weight"]];

(* t1Run1R *)

t1localmin =
  ArgMin[funcRun1R[x], (tmaxRun1R - 0.3) < x < tmaxRun1R, x];
Clear[findAllRoots]
SyntaxInformation[
   findAllRoots] = {"LocalVariables" -> {"Plot", {2, 2}},
   "ArgumentsPattern" -> {_, _, OptionsPattern[]}};
SetAttributes[findAllRoots, HoldAll];
Options[findAllRoots] =
  Join[{"ShowPlot" -> False, PlotRange -> All},
   FilterRules[Options[Plot], Except[PlotRange]]];
findAllRoots[fn_, {l_, lmin_, lmax_}, opts : OptionsPattern[]] :=
 Module[{pl, p, x, localFunction, brackets},
  localFunction = ReleaseHold[Hold[fn] /. HoldPattern[l] :> x];
  If[lmin != lmax,
   pl = Plot[localFunction, {x, lmin, lmax},
     Evaluate@
      FilterRules[Join[{opts}, Options[findAllRoots]], Options[Plot]]];
   p = Cases[pl, Line[{x__}] :> x, Infinity];
   If[OptionValue["ShowPlot"],
    Print[Show[pl, PlotLabel -> "Finding roots for this function",
      ImageSize -> 200, BaseStyle -> {FontSize -> 8}]]], p = {}];
  brackets =
   Map[First,
    Select[(*This Split trick pretends that two points on the curve \
are "equal" if the function values have _opposite _ sign.Pairs of \
such sign-changes form the brackets for the subsequent FindRoot*)
     Split[p, Sign[Last[#2]] == -Sign[Last[#1]] &],
     Length[#1] == 2 &], {2}];
  x /. Apply[FindRoot[localFunction == 0, {x, ##1}] &,
     brackets, {1}] /. x -> {}]
rootfuncRun1R[t_] := funcRun1R"[x] /. x -> t;
t1rootlist =
  findAllRoots[rootfuncRun1R[x], {x, (tmaxRun1R - 0.5), t1localmin},
   "ShowPlot" -> False];
t1rootlistlength = Length[t1rootlist];
While[funcRun1R[t1rootlist[[t1rootlistlength]]] <
   funcRun1R[t1rootlist[[t1rootlistlength - 1]]],
  t1rootlistlength = t1rootlistlength - 1];
t1bound = t1rootlist[[t1rootlistlength]];

t1localmin2 = ArgMin[funcRun1R[x], t1bound < x < tmaxRun1R, x];
slope2 = (funcRun1R[t1bound] - funcRun1R[t1localmin2])/(t1bound -
```

```
      t1localmin2) + 25;
t1list = Solve[
   y[[2]] == slope2 &&
    x > (t1localmin - (tmaxRun1R - t1localmin) - 0.05) &&
    x < t1localmin, x];
v = t1list[[1]][[1]];
t1Run1R = (x /. v);

cutoffRun1R = t1Run1R - 0.2;
thalfRun1R = (0.5*(t4Run1R - t1Run1R)) + t1Run1R;

(*acceleration, velocity, displacement, power*)

accelRun1R[x_] := (funcRun1R[x] - BWforceRun1R)/massRun1R;
velocityRun1R =
  NDSolveValue[{accelRun1R[x] == f2'[x], f2[t1Run1R] == 0},
   f2[x], {x, 0, lastTimeRun1R}];
displacementRun1R =
  NDSolveValue[{accelRun1R[x] == g2''[x],
    g2[t1Run1R] == g2'[t1Run1R] == 0}, g2[x], {x, 0, lastTimeRun1R}];
powerRun1R = funcRun1R[x]*velocityRun1R;

(* t2Run1R *)

t2Run1R = ArgMin[velocityRun1R, t1Run1R < x < tmaxRun1R, x];
(* t3Run1R *)

t3Run1R = ArgMin[displacementRun1R, t2Run1R < x < t4Run1R, x];

(*tsquat \[Rule] average and standard deviation of force during squat*)

tearlymaxRun1R = ArgMax[funcRun1R[x], t0Run1R < x < t1Run1R, x];
tsquatlist =
  findAllRoots[rootfuncRun1R[x], {x, tearlymaxRun1R, cutoffRun1R},
   "ShowPlot" -> False];
tpos = 1;
While[funcRun1R[tsquatlist[[tpos]]] >
   funcRun1R[tsquatlist[[tpos + 1]]], tpos = tpos + 1];
tsquatRun1R = tsquatlist[[tpos]];
squatlistRun1R = DeleteCases[Datarefined, y_ /; y[[1]] > t1Run1R];
squatlistRun1R =
  DeleteCases[squatlistRun1R, y_ /; y[[1]] < tsquatRun1R];
squatlistRun1R = squatlistRun1R[[All, 2]];
squatavgRun1R = Mean[squatlistRun1R];
squatstdevRun1R = StandardDeviation[squatlistRun1R];

(*outputs*)
force1Run1R = funcRun1R[t1Run1R];
velocity1Run1R = velocityRun1R /. x -> t1Run1R;
displacement1Run1R = displacementRun1R /. x -> t1Run1R;
power1Run1R = powerRun1R /. x -> t1Run1R;
force2Run1R = funcRun1R[t2Run1R];
velocity2Run1R = velocityRun1R /. x -> t2Run1R;
displacement2Run1R = displacementRun1R /. x -> t2Run1R;
power2Run1R = powerRun1R /. x -> t2Run1R;
force3Run1R = funcRun1R[t3Run1R];
velocity3Run1R = velocityRun1R /. x -> t3Run1R;
displacement3Run1R = displacementRun1R /. x -> t3Run1R;
power3Run1R = powerRun1R /. x -> t3Run1R;
force4Run1R = funcRun1R[t4Run1R];
velocity4Run1R = velocityRun1R /. x -> t4Run1R;
displacement4Run1R = displacementRun1R /. x -> t4Run1R;
power4Run1R = powerRun1R /. x -> t4Run1R;
force5Run1R = funcRun1R[t5Run1R];
velocity5Run1R = velocityRun1R /. x -> t5Run1R;
displacement5Run1R = displacementRun1R /. x -> t5Run1R;
power5Run1R = powerRun1R /. x -> t5Run1R;
```

```
forcemaxRun1R = funcRun1R[tmaxRun1R];
velocitymaxRun1R = velocityRun1R /. x -> tmaxRun1R;
displacementmaxRun1R = displacementRun1R /. x -> tmaxRun1R;
powermaxRun1R = powerRun1R /. x -> tmaxRun1R;
forcehalfRun1R = funcRun1R[thalfRun1R];
velocityhalfRun1R = velocityRun1R /. x -> thalfRun1R;
displacementhalfRun1R = displacementRun1R /. x -> thalfRun1R;
powerhalfRun1R = powerRun1R /. x -> thalfRun1R;

impulse12Run1R = NIntegrate[funcRun1R[x], {x, t1Run1R, t2Run1R}];
impulse23Run1R = NIntegrate[funcRun1R[x], {x, t2Run1R, t3Run1R}];
impulse34Run1R = NIntegrate[funcRun1R[x], {x, t3Run1R, t4Run1R}];
impulse45Run1R = NIntegrate[funcRun1R[x], {x, t4Run1R, t5Run1R}];
impulse2maxRun1R = NIntegrate[funcRun1R[x], {x, t2Run1R, tmaxRun1R}];
impulse1halfRun1R = NIntegrate[funcRun1R[x], {x, t1Run1R, thalfRun1R}];

maxforce12Run1R = MaxValue[{funcRun1R[x], t1Run1R <= x <= t2Run1R}, x];
minforce12Run1R = MinValue[{funcRun1R[x], t1Run1R <= x <= t2Run1R}, x];
maxvelocity12Run1R =
  MaxValue[{velocityRun1R, t1Run1R <= x <= t2Run1R}, x];
minvelocity12Run1R =
  MinValue[{velocityRun1R, t1Run1R <= x <= t2Run1R}, x];
maxdisplacement12Run1R =
  MaxValue[{displacementRun1R, t1Run1R <= x <= t2Run1R}, x];
mindisplacement12Run1R =
  MinValue[{displacementRun1R, t1Run1R <= x <= t2Run1R}, x];
maxpower12Run1R = MaxValue[{powerRun1R, t1Run1R <= x <= t2Run1R}, x];
minpower12Run1R = MinValue[{powerRun1R, t1Run1R <= x <= t2Run1R}, x];

maxforce23Run1R = MaxValue[{funcRun1R[x], t2Run1R <= x <= t3Run1R}, x];
minforce23Run1R = MinValue[{funcRun1R[x], t2Run1R <= x <= t3Run1R}, x];
maxvelocity23Run1R =
  MaxValue[{velocityRun1R, t2Run1R <= x <= t3Run1R}, x];
minvelocity23Run1R =
  MinValue[{velocityRun1R, t2Run1R <= x <= t3Run1R}, x];
maxdisplacement23Run1R =
  MaxValue[{displacementRun1R, t2Run1R <= x <= t3Run1R}, x];
mindisplacement23Run1R =
  MinValue[{displacementRun1R, t2Run1R <= x <= t3Run1R}, x];
maxpower23Run1R = MaxValue[{powerRun1R, t2Run1R <= x <= t3Run1R}, x];
minpower23Run1R = MinValue[{powerRun1R, t2Run1R <= x <= t3Run1R}, x];

maxforce34Run1R = MaxValue[{funcRun1R[x], t3Run1R <= x <= t4Run1R}, x];
minforce34Run1R = MinValue[{funcRun1R[x], t3Run1R <= x <= t4Run1R}, x];
maxvelocity34Run1R =
  MaxValue[{velocityRun1R, t3Run1R <= x <= t4Run1R}, x];
minvelocity34Run1R =
  MinValue[{velocityRun1R, t3Run1R <= x <= t4Run1R}, x];
maxdisplacement34Run1R =
  MaxValue[{displacementRun1R, t3Run1R <= x <= t4Run1R}, x];
mindisplacement34Run1R =
  MinValue[{displacementRun1R, t3Run1R <= x <= t4Run1R}, x];
maxpower34Run1R = MaxValue[{powerRun1R, t3Run1R <= x <= t4Run1R}, x];
minpower34Run1R = MinValue[{powerRun1R, t3Run1R <= x <= t4Run1R}, x];

maxforce45Run1R = MaxValue[{funcRun1R[x], t4Run1R <= x <= t5Run1R}, x];
minforce45Run1R = MinValue[{funcRun1R[x], t4Run1R <= x <= t5Run1R}, x];
maxvelocity45Run1R =
  MaxValue[{velocityRun1R, t4Run1R <= x <= t5Run1R}, x];
minvelocity45Run1R =
  MinValue[{velocityRun1R, t4Run1R <= x <= t5Run1R}, x];
maxdisplacement45Run1R =
  MaxValue[{displacementRun1R, t4Run1R <= x <= t5Run1R}, x];
mindisplacement45Run1R =
  MinValue[{displacementRun1R, t4Run1R <= x <= t5Run1R}, x];
maxpower45Run1R = MaxValue[{powerRun1R, t4Run1R <= x <= t5Run1R}, x];
minpower45Run1R = MinValue[{powerRun1R, t4Run1R <= x <= t5Run1R}, x];
```

```
maxforce2maxRun1R =
  MaxValue[{funcRun1R[x], t2Run1R <= x <= tmaxRun1R}, x];
minforce2maxRun1R =
  MinValue[{funcRun1R[x], t2Run1R <= x <= tmaxRun1R}, x];
maxvelocity2maxRun1R =
  MaxValue[{velocityRun1R, t2Run1R <= x <= tmaxRun1R}, x];
minvelocity2maxRun1R =
  MinValue[{velocityRun1R, t2Run1R <= x <= tmaxRun1R}, x];
maxdisplacement2maxRun1R =
  MaxValue[{displacementRun1R, t2Run1R <= x <= tmaxRun1R}, x];
mindisplacement2maxRun1R =
  MinValue[{displacementRun1R, t2Run1R <= x <= tmaxRun1R}, x];
maxpower2maxRun1R =
  MaxValue[{powerRun1R, t2Run1R <= x <= tmaxRun1R}, x];
minpower2maxRun1R =
  MinValue[{powerRun1R, t2Run1R <= x <= tmaxRun1R}, x];

maxforce1halfRun1R =
  MaxValue[{funcRun1R[x], t1Run1R <= x <= thalfRun1R}, x];
minforce1halfRun1R =
  MinValue[{funcRun1R[x], t1Run1R <= x <= thalfRun1R}, x];
maxvelocity1halfRun1R =
  MaxValue[{velocityRun1R, t1Run1R <= x <= thalfRun1R}, x];
minvelocity1halfRun1R =
  MinValue[{velocityRun1R, t1Run1R <= x <= thalfRun1R}, x];
maxdisplacement1halfRun1R =
  MaxValue[{displacementRun1R, t1Run1R <= x <= thalfRun1R}, x];
mindisplacement1halfRun1R =
  MinValue[{displacementRun1R, t1Run1R <= x <= thalfRun1R}, x];
maxpower1halfRun1R =
  MaxValue[{powerRun1R, t1Run1R <= x <= thalfRun1R}, x];
minpower1halfRun1R =
  MinValue[{powerRun1R, t1Run1R <= x <= thalfRun1R}, x];

(*TableRun1R={{"mass= ",massRun1R},{},
{"t1= ",t1Run1R,"force @ t1= ",force1Run1R,"veloc. @ t1= \
",velocity1Run1R,"displace. @ t1= ",displacement1Run1R,"power @ t1= \
",power1Run1R},{"t2= ",t2Run1R,"force @ t2= ",force2Run1R,"veloc. @ \
t2= ",velocity2Run1R,"displace. @ t2= ",displacement2Run1R,"power @ \
t2= ",power2Run1R},{"t3= ",t3Run1R,"force @ t3= ",force3Run1R,"veloc. \
@ t3= ",velocity3Run1R,"displace. @ t3= ",displacement3Run1R,"power @ \
t3= ",power3Run1R},{"t4= ",t4Run1R,"force @ t4= ",force4Run1R,"veloc. \
@ t4= ",velocity4Run1R,"displace. @ t4= ",displacement4Run1R,"power @ \
t4= ",power4Run1R},{"t5= ",t5Run1R,"force @ t5= ",force5Run1R,"veloc. \
@ t5= ",velocity5Run1R,"displace. @ t5= ",displacement5Run1R,"power @ \
t5= ",power5Run1R},{"tmax= ",tmaxRun1R,"force @ tmax= \
",forcemaxRun1R,"veloc. @ tmax= ",velocitymaxRun1R,"displace. @ tmax= \
",displacementmaxRun1R,"power @ tmax= ",powermaxRun1R},{"thalf= \
",thalfRun1R,"force @ thalf= ",forcehalfRun1R,"veloc. @ thalf= \
",velocityhalfRun1R,"displace. @ thalf= \
",displacementhalfRun1R,"power @ thalf= \
",powerhalfRun1R},{},{"impulse t1-t2= ",impulse12Run1R,"impulse \
t2-t3= ",impulse23Run1R,"impulse t3-t4= ",impulse34Run1R,"impulse \
t4-t5= ",impulse45Run1R,"impulse t2-tmax= ",impulse2maxRun1R},{"max \
force t1-t2= ",maxforce12Run1R,"max force t2-t3= \
",maxforce23Run1R,"max force t3-t4= ",maxforce34Run1R,"max force \
t4-t5= ",maxforce45Run1R,"max force t2-tmax= ",maxforce2maxRun1R,"max \
force t1-thalf= ",maxforce1halfRun1R},{"min force t1-t2= \
",minforce12Run1R,"min force t2-t3= ",minforce23Run1R,"min force \
t3-t4= ",minforce34Run1R,"min force t4-t5= ",minforce45Run1R,"min \
force t2-tmax= ",minforce2maxRun1R,"min force t1-thalf= \
",minforce1halfRun1R},{"max veloc. t1-t2= ",maxvelocity12Run1R,"max \
veloc. t2-t3= ",maxvelocity23Run1R,"max veloc. t3-t4= \
",maxvelocity34Run1R,"max veloc. t4-t5= ",maxvelocity45Run1R,"max \
veloc. t2-tmax= ",maxvelocity2maxRun1R,"max veloc. t1-thalf= \
",maxvelocity1halfRun1R},{"min veloc. t1-t2= \
```

```
",minvelocity12Run1R,"min veloc. t2-t3= ",minvelocity23Run1R,"min \
veloc. t3-t4= ",minvelocity34Run1R,"min veloc. t4-t5= \
",minvelocity45Run1R,"min veloc. t2-tmax= ",minvelocity2maxRun1R,"min \
veloc. t1-thalf= ",minvelocity1halfRun1R},{"max displace. t1-t2= \
",maxdisplacement12Run1R,"max displace. t2-t3= \
",maxdisplacement23Run1R,"max displace. t3-t4= \
",maxdisplacement34Run1R,"max displace. t4-t5= \
",maxdisplacement45Run1R,"max displace. t2-tmax= \
",maxdisplacement2maxRun1R,"max displace. t1-thalf= \
",maxdisplacement1halfRun1R},{"min displace. t1-t2= \
",mindisplacement12Run1R,"min displace. t2-t3= \
",mindisplacement23Run1R,"min displace. t3-t4= \
",mindisplacement34Run1R,"min displace. t4-t5= \
",mindisplacement45Run1R,"min displace. t2-tmax= \
",mindisplacement2maxRun1R,"min displace. t1-thalf= \
",mindisplacement1halfRun1R},{"max power t1-t2= \
",maxpower12Run1R,"max power t2-t3= ",maxpower23Run1R,"max power \
t3-t4= ",maxpower34Run1R,"max power t4-t5= ",maxpower45Run1R,"max \
power t2-tmax= ",maxpower2maxRun1R,"max power t1-thalf= \
",maxpower1halfRun1R},
{"min power t1-t2= ",minpower12Run1R,"min power t2-t3= \
",minpower23Run1R,"min power t3-t4= ",minpower34Run1R,"min power \
t4-t5= ",minpower45Run1R,"min power t2-tmax= ",minpower2maxRun1R,"min \
power t1-thalf= ",minpower1halfRun1R}} ;
Grid[TableRun1R,Frame\[Rule]All]*)

(*classify the type of NCMJ based on number of peaks--look at the \
number of maxima/minima of the first derivative*)

localrootsRun1R =
  Length[findAllRoots[rootfuncRun1R[x], {x, t3Run1R, t4Run1R},
    "ShowPlot" -> False]];
If[localrootsRun1R == 1, {runtypeRun1R = 1,
   Print["NCMJ Type 1 (one peak)"]}, {runtypeRun1R = 2,
   Print["NCMJ Type 2 (two or more peaks)"]}];

Print["tsquat= ", tsquatRun1R]
Print["t1= ", t1Run1R]
Print["t2= ", t2Run1R]
Print["t3= ", t3Run1R]
Print["t4= ", t4Run1R]
Print["t5= ", t5Run1R]
Print["thalf= ", thalfRun1R]
Print["tmax= ", tmaxRun1R]

line1Run1R = Line[{{t1Run1R, -5000}, {t1Run1R, 5000}}];
line2Run1R = Line[{{t2Run1R, -5000}, {t2Run1R, 5000}}];
line3Run1R = Line[{{t3Run1R, -5000}, {t3Run1R, 5000}}];
line4Run1R = Line[{{t4Run1R, -5000}, {t4Run1R, 5000}}];
line5Run1R = Line[{{t5Run1R, -5000}, {t5Run1R, 5000}}];
linehalfRun1R = Line[{{thalfRun1R, -5000}, {thalfRun1R, 5000}}];
linemaxRun1R = Line[{{tmaxRun1R, -5000}, {tmaxRun1R, 5000}}];
lineStyle = {Thin, Red, Dashed};
bodyweightlineRun1R =
  Line[{{0, BWforceRun1R}, {tmaxRun1R, BWforceRun1R}}];
cutofflineRun1R = Line[{{cutoffRun1R, -5000}, {cutoffRun1R, 5000}}];
squatavglineRun1R =
  Line[{{0, squatavgRun1R}, {cutoffRun1R, squatavgRun1R}}];
squatlineRun1R = Line[{{tsquatRun1R, -5000}, {tsquatRun1R, 5000}}];

Print["Body Weight= ", BWforceRun1R]
Print["Force in Air= ", noforceRun1R]
Plot[{funcRun1R[x]}, {x, cutoffRun1R, lastTimeRun1R},
 ImageSize -> 600, PlotStyle -> {Automatic},
 Epilog -> {Directive[lineStyle], line1Run1R, line2Run1R, line3Run1R,
   line4Run1R, line5Run1R, Directive[Thin, Blue, Dashed],
   linehalfRun1R, linemaxRun1R, Directive[Thin, Black, Dashed],
```

```
   bodyweightlineRun1R}, PlotRange -> All,
  BaseStyle -> { FontSize -> 12}, PlotLabel -> "NCMJ - Normal Force",
  AxesLabel -> {"Time (s)", "Normal Force (N)"} ]
Plot[{velocityRun1R}, {x, cutoffRun1R, lastTimeRun1R},
  ImageSize -> 600, PlotStyle -> {Automatic},
  Epilog -> {Directive[lineStyle], line1Run1R, line2Run1R, line3Run1R,
    line4Run1R, line5Run1R, Directive[Thin, Blue, Dashed],
    linehalfRun1R, linemaxRun1R, Directive[Thin, Black, Dashed],
    bodyweightlineRun1R}, PlotRange -> All,
  BaseStyle -> { FontSize -> 12}, PlotLabel -> "NCMJ - Velocity",
  AxesLabel -> {"Time (s)", "Velocity {m/s}"} ]
Plot[{funcRun1R[x]}, {x, 0, lastTimeRun1R}, ImageSize -> 600,
  PlotStyle -> {Automatic},
  Epilog -> {Directive[lineStyle], line1Run1R, line2Run1R, line3Run1R,
    line4Run1R, line5Run1R, Directive[Thin, Blue, Dashed],
    linehalfRun1R, linemaxRun1R, Directive[Thin, Black, Dashed],
    cutofflineRun1R, bodyweightlineRun1R, squatavglineRun1R,
    squatlineRun1R}, PlotRange -> All, BaseStyle -> { FontSize -> 12},
  PlotLabel -> "NCMJ - Normal Force",
  AxesLabel -> {"Time (s)", "Normal Force (N)"} ]
(*Plot[{velocityRun1R},{x,0,lastTimeRun1R},ImageSize\[Rule]600,\
PlotStyle \[Rule] \
{Automatic},Epilog\[Rule]{Directive[lineStyle],line1Run1R,line2Run1R,\
line3Run1R,line4Run1R,line5Run1R, \
Directive[Thin,Blue,Dashed],linehalfRun1R,linemaxRun1R, \
Directive[Thin,Black,Dashed],bodyweightlineRun1R}, \
PlotRange\[Rule]All, BaseStyle\[Rule]{ FontSize\[Rule] 12},PlotLabel\
\[Rule] "NCMJ - Velocity", AxesLabel\[Rule]{"Time (s)", "Velocity \
(m/s)"} ]
Plot[{displacementRun1R},{x,0,lastTimeRun1R},ImageSize\[Rule]600,\
PlotStyle \[Rule] \
{Automatic},Epilog\[Rule]{Directive[lineStyle],line1Run1R,line2Run1R,\
line3Run1R,line4Run1R,line5Run1R, \
Directive[Thin,Blue,Dashed],linehalfRun1R,linemaxRun1R},PlotRange\
\[Rule]All, BaseStyle\[Rule]{ FontSize\[Rule] 12},PlotLabel\[Rule] \
"NCMJ - COM Displacement", AxesLabel\[Rule]{"Time (s)", "Normal Force \
(N)"} ]
Plot[{powerRun1R},{x,0,lastTimeRun1R},ImageSize\[Rule]600,PlotStyle \
\[Rule] {Automatic},Epilog\[Rule]{Directive[lineStyle],line1Run1R,\
line2Run1R,line3Run1R,line4Run1R,line5Run1R, \
Directive[Thin,Blue,Dashed],linehalfRun1R,linemaxRun1R, \
Directive[Thin,Black,Dashed],bodyweightlineRun1R,},PlotRange\[Rule]\
All, BaseStyle\[Rule]{ FontSize\[Rule] 12},PlotLabel\[Rule] "NCMJ - \
Power", AxesLabel\[Rule]{"Time (s)", "Normal Force (N)"} ]*)
```

```
(* Run1Both *)
  DataRun1Bothi =};
DataRun1Bothi = DeleteCases[DataRun1Bothi, y_ /; y[[1]] > 4];
Clear[x, a, b, c, d]
DataRun1Both =
  Table[DataRun1Bothi[[4 i]], {i, 1, Length[DataRun1Bothi]/4}];
f = DataRun1Both[[All, 2]];
f = GaussianFilter[f, 10];
Datarefined =
  Table[{DataRun1Both[[i, 1]], f[[i]]}, {i, 1,
    Length[DataRun1Both]}];
funcRun1Both =
  Interpolation[Datarefined, Method -> "Spline",
   InterpolationOrder -> 3];
funcRun1Bothp = D[funcRun1Both[x], x];
LL = Length[DataRun1Both];
lastTimeRun1Both = DataRun1Both[[LL]][[1]];

Spline3[DataRun1Both_] := (k = Length[DataRun1Both];
  atable = Table[Subscript[a, i], {i, 1, k - 3}];
  btable = Table[Subscript[b, i], {i, 1, k - 3}];
  ctable = Table[Subscript[c, i], {i, 1, k - 3}];
  dtable = Table[Subscript[d, i], {i, 1, k - 3}];
```

```
  Poly = Table[
    atable[[i]] + btable[[i]]*x + ctable[[i]]*x^2 +
     dtable[[i]]*x^3, {i, 1, k - 3}];
  Polyd =
   Table[btable[[i]] + 2*ctable[[i]]*x + 3*dtable[[i]]*x^2, {i, 1,
     k - 3}];
  intermediate = Table[DataRun1Both[[i + 2]], {i, 1, k - 4}];
  Equations = Table[0, {i, 1, 4 (k - 3)}];
  Do[Equations[[i]] = (Poly[[i]] /. x -> DataRun1Both[[i + 2, 1]]) ==
     DataRun1Both[[i + 2, 2]], {i, 1, k - 4}];
  Do[Equations[[
     i + k - 4]] = (Poly[[i + 1]] /. x -> DataRun1Both[[i + 2, 1]]) ==
      DataRun1Both[[i + 2, 2]], {i, 1, k - 4}];
  Equations[[
    2 (k - 4) + 1]] = (Poly[[1]] /. x -> DataRun1Both[[1, 1]]) ==
    DataRun1Both[[1, 2]];
  Equations[[
    2 (k - 4) + 2]] = (Poly[[1]] /. x -> DataRun1Both[[2, 1]]) ==
    DataRun1Both[[2, 2]];
  Equations[[
    2 (k - 4) +
     3]] = (Poly[[k - 3]] /. x -> DataRun1Both[[k - 1, 1]]) ==
    DataRun1Both[[k - 1, 2]];
  Equations[[
    2 (k - 4) + 4]] = (Poly[[k - 3]] /. x -> DataRun1Both[[k, 1]]) ==
    DataRun1Both[[k, 2]];
  Do[Equations[[
     2 (k - 3) + 2 +
      i]] = (D[Poly[[i]], x] /.
       x -> intermediate[[i, 1]]) == (D[Poly[[i + 1]], x] /.
       x -> intermediate[[i, 1]]), {i, 1, k - 4}];
  Do[Equations[[
     2 (k - 3) + 2 + k - 4 +
      i]] = (D[Poly[[i]], {x, 2}] /.
       x -> intermediate[[i, 1]]) == (D[Poly[[i + 1]], {x, 2}] /.
       x -> intermediate[[i, 1]]), {i, 1, k - 4}];
  un = Flatten[{atable, btable, ctable, dtable}];
  f = Solve[Equations];
  Poly = Poly /. f[[1]];
  Polyd = Polyd /. f[[1]];
  xs = intermediate;
  xs = Prepend[Append[xs, DataRun1Both[[k]]], DataRun1Both[[1]]];
  pf = Table[{Poly[[i]], xs[[i, 1]] <= x <= xs[[i + 1, 1]]}, {i, 1,
     k - 3}];
  pdf = Table[{Polyd[[i]], xs[[i, 1]] <= x <= xs[[i + 1, 1]]}, {i, 1,
     k - 3}];
  sol = {Piecewise[pf], Piecewise[pdf]})
Timing[y = Spline3[Datarefined]];

(*Max force and tmax it occurs*)

globalmintimeRun1Both =
  ArgMin[funcRun1Both[x], 2 < x < lastTimeRun1Both, x];
globalminRun1Both =
  MinValue[funcRun1Both[x], 2 < x < lastTimeRun1Both, x];
peakMaxRun1Both =
  MaxValue[funcRun1Both[x], 2 < x < globalmintimeRun1Both, x];
tmaxRun1Both =
  ArgMax[funcRun1Both[x], 2 < x < globalmintimeRun1Both, x];
(* t4Run1Both *)

t4Run1Both =
  ArgMin[funcRun1Both[x], tmaxRun1Both < x < lastTimeRun1Both, x];
While[funcRun1Both[t4Run1Both] <= Max[0, (1 + globalminRun1Both)],
  t4Run1Both =
   ArgMin[funcRun1Both[x], tmaxRun1Both < x < (t4Run1Both - 0.01), x]];
(* t5Run1Both *)
```

```
LocalMaxx5 =
  ArgMax[{funcRun1Both[x],
    t4Run1Both + 0.2 <= x <= lastTimeRun1Both - 0.1}, x];
LocalMax5 = funcRun1Both[LocalMaxx5];
mlist = Solve[
   y[[2]] == funcRun1Both[0]*0.05 && x > t4Run1Both + 0.2 &&
    x < LocalMaxx5 && funcRun1Both[x] < LocalMax5*0.1, x];
minlist = x /. mlist;
t5Run1Both = Last[minlist];
(*force while in air*)

noforceRun1Both =
  NIntegrate[
    funcRun1Both[
     x], {x, (t4Run1Both + 0.05), (t5Run1Both -
       0.05)}]/((t5Run1Both - 0.05) - (t4Run1Both + 0.05));
(* t0Run1Both - to find quiet stance mass*)

earlyGlobalMin = ArgMin[funcRun1Both[x], 0.1 < x < tmaxRun1Both, x];
slope = (funcRun1Both[0] - funcRun1Both[earlyGlobalMin])/(0 -
      earlyGlobalMin) + 25;
(* t0list is list of times where derivative = defined threshold*)

t0list = Solve[y[[2]] == slope, x];
v = t0list[[1]][[1]];
t0Run1Both = (x /. v);

(*find quiet stance mass by taking average from t=0 to just before \
the jump "starts", but only if it appears to be relatively steady*)

BWforceRun1Both =
  NIntegrate[
    funcRun1Both[x], {x,
     0.05, (t0Run1Both - 0.1)}]/((t0Run1Both - 0.1) - 0.05);
If[(funcRun1Both[0] -
      funcRun1Both[(t0Run1Both - 0.1)])/(0 - (t0Run1Both - 0.1)) >
   40, (Print["->Issue with quiet stance mass"])];
If[t0Run1Both <= 0.15, Print["Check mass calc."]];
massRun1Both = BWforceRun1Both/9.81;
If[Abs[noforceRun1Both] > 0.05*BWforceRun1Both,
  Print["Possible error: force while subject is in air 5% body \
weight"]];

(* t1Run1Both *)

t1localmin =
  ArgMin[funcRun1Both[x], (tmaxRun1Both - 0.3) < x < tmaxRun1Both,
   x];
Clear[findAllRoots]
SyntaxInformation[
   findAllRoots] = {"LocalVariables" -> {"Plot", {2, 2}},
   "ArgumentsPattern" -> {_, _, OptionsPattern[]}};
SetAttributes[findAllRoots, HoldAll];
Options[findAllRoots] =
  Join[{"ShowPlot" -> False, PlotRange -> All},
   FilterRules[Options[Plot], Except[PlotRange]]];
findAllRoots[fn_, {l_, lmin_, lmax_}, opts : OptionsPattern[]] :=
 Module[{pl, p, x, localFunction, brackets},
  localFunction = ReleaseHold[Hold[fn] /. HoldPattern[l] :> x];
  If[lmin != lmax,
   pl = Plot[localFunction, {x, lmin, lmax},
     Evaluate@
      FilterRules[Join[{opts}, Options[findAllRoots]], Options[Plot]]];
   p = Cases[pl, Line[{x__}] :> x, Infinity];
   If[OptionValue["ShowPlot"],
    Print[Show[pl, PlotLabel -> "Finding roots for this function",
```

```
      ImageSize -> 200, BaseStyle -> {FontSize -> 8}]]], p = {}];
  brackets =
   Map[First,
    Select[(*This Split trick pretends that two points on the curve \
are "equal" if the function values have _opposite _ sign.Pairs of \
such sign-changes form the brackets for the subsequent FindRoot*)
     Split[p, Sign[Last[#2]] == -Sign[Last[#1]] &],
     Length[#1] == 2 &], {2}];
  x /. Apply[FindRoot[localFunction == 0, {x, ##1}] &,
     brackets, {1}] /. x -> {}]
rootfuncRun1Both[t_] := funcRun1Both''[x] /. x -> t;
t1rootlist =
  findAllRoots[
   rootfuncRun1Both[x], {x, (tmaxRun1Both - 0.5), t1localmin},
    "ShowPlot" -> False];
t1rootlistlength = Length[t1rootlist];
While[funcRun1Both[t1rootlist[[t1rootlistlength]]] <
   funcRun1Both[t1rootlist[[t1rootlistlength - 1]]],
  t1rootlistlength = t1rootlistlength - 1];
t1bound = t1rootlist[[t1rootlistlength]];

t1localmin2 = ArgMin[funcRun1Both[x], t1bound < x < tmaxRun1Both, x];
slope2 = (funcRun1Both[t1bound] -
      funcRun1Both[t1localmin2])/(t1bound - t1localmin2) + 25;
t1list = Solve[
   y[[2]] == slope2 &&
    x > (t1localmin - (tmaxRun1Both - t1localmin) - 0.05) &&
    x < t1localmin, x];
v = t1list[[1]][[1]];
t1Run1Both = (x /. v);

cutoffRun1Both = t1Run1Both - 0.2;
thalfRun1Both = (0.5*(t4Run1Both - t1Run1Both)) + t1Run1Both;

(*acceleration, velocity, displacement, power*)

accelRun1Both[x_] := (funcRun1Both[x] - BWforceRun1Both)/
   massRun1Both;
velocityRun1Both =
  NDSolveValue[{accelRun1Both[x] == f2'[x], f2[t1Run1Both] == 0},
   f2[x], {x, 0, lastTimeRun1Both}];
displacementRun1Both =
  NDSolveValue[{accelRun1Both[x] == g2''[x],
    g2[t1Run1Both] == g2'[t1Run1Both] == 0},
   g2[x], {x, 0, lastTimeRun1Both}];
powerRun1Both = funcRun1Both[x]*velocityRun1Both;

(* t2Run1Both *)

t2Run1Both =
  ArgMin[velocityRun1Both, t1Run1Both < x < tmaxRun1Both, x];
(* t3Run1Both *)

t3Run1Both =
  ArgMin[displacementRun1Both, t2Run1Both < x < t4Run1Both, x];

(*tsquat \[Rule] average and standard deviation of force during squat*)

tearlymaxRun1Both =
  ArgMax[funcRun1Both[x], t0Run1Both < x < t1Run1Both, x];
tsquatlist =
  findAllRoots[
   rootfuncRun1Both[x], {x, tearlymaxRun1Both, cutoffRun1Both},
    "ShowPlot" -> False];
tpos = 1;
While[funcRun1Both[tsquatlist[[tpos]]] >
   funcRun1Both[tsquatlist[[tpos + 1]]], tpos = tpos + 1];
```

```
tsquatRun1Both = tsquatlist[[tpos]];
squatlistRun1Both =
  DeleteCases[Datarefined, y_ /; y[[1]] > t1Run1Both];
squatlistRun1Both =
  DeleteCases[squatlistRun1Both, y_ /; y[[1]] < tsquatRun1Both];
squatlistRun1Both = squatlistRun1Both[[All, 2]];
squatavgRun1Both = Mean[squatlistRun1Both];
squatstdevRun1Both = StandardDeviation[squatlistRun1Both];

(*outputs*)
force1Run1Both = funcRun1Both[t1Run1Both];
velocity1Run1Both = velocityRun1Both /. x -> t1Run1Both;
displacement1Run1Both = displacementRun1Both /. x -> t1Run1Both;
power1Run1Both = powerRun1Both /. x -> t1Run1Both;
force2Run1Both = funcRun1Both[t2Run1Both];
velocity2Run1Both = velocityRun1Both /. x -> t2Run1Both;
displacement2Run1Both = displacementRun1Both /. x -> t2Run1Both;
power2Run1Both = powerRun1Both /. x -> t2Run1Both;
force3Run1Both = funcRun1Both[t3Run1Both];
velocity3Run1Both = velocityRun1Both /. x -> t3Run1Both;
displacement3Run1Both = displacementRun1Both /. x -> t3Run1Both;
power3Run1Both = powerRun1Both /. x -> t3Run1Both;
force4Run1Both = funcRun1Both[t4Run1Both];
velocity4Run1Both = velocityRun1Both /. x -> t4Run1Both;
displacement4Run1Both = displacementRun1Both /. x -> t4Run1Both;
power4Run1Both = powerRun1Both /. x -> t4Run1Both;
force5Run1Both = funcRun1Both[t5Run1Both];
velocity5Run1Both = velocityRun1Both /. x -> t5Run1Both;
displacement5Run1Both = displacementRun1Both /. x -> t5Run1Both;
power5Run1Both = powerRun1Both /. x -> t5Run1Both;
forcemaxRun1Both = funcRun1Both[tmaxRun1Both];
velocitymaxRun1Both = velocityRun1Both /. x -> tmaxRun1Both;
displacementmaxRun1Both = displacementRun1Both /. x -> tmaxRun1Both;
powermaxRun1Both = powerRun1Both /. x -> tmaxRun1Both;
forcehalfRun1Both = funcRun1Both[thalfRun1Both];
velocityhalfRun1Both = velocityRun1Both /. x -> thalfRun1Both;
displacementhalfRun1Both =
  displacementRun1Both /. x -> thalfRun1Both;
powerhalfRun1Both = powerRun1Both /. x -> thalfRun1Both;

impulse12Run1Both =
  NIntegrate[funcRun1Both[x], {x, t1Run1Both, t2Run1Both}];
impulse23Run1Both =
  NIntegrate[funcRun1Both[x], {x, t2Run1Both, t3Run1Both}];
impulse34Run1Both =
  NIntegrate[funcRun1Both[x], {x, t3Run1Both, t4Run1Both}];
impulse45Run1Both =
  NIntegrate[funcRun1Both[x], {x, t4Run1Both, t5Run1Both}];
impulse2maxRun1Both =
  NIntegrate[funcRun1Both[x], {x, t2Run1Both, tmaxRun1Both}];
impulse1halfRun1Both =
  NIntegrate[funcRun1Both[x], {x, t1Run1Both, thalfRun1Both}];

maxforce12Run1Both =
  MaxValue[{funcRun1Both[x], t1Run1Both <= x <= t2Run1Both}, x];
minforce12Run1Both =
  MinValue[{funcRun1Both[x], t1Run1Both <= x <= t2Run1Both}, x];
maxvelocity12Run1Both =
  MaxValue[{velocityRun1Both, t1Run1Both <= x <= t2Run1Both}, x];
minvelocity12Run1Both =
  MinValue[{velocityRun1Both, t1Run1Both <= x <= t2Run1Both}, x];
maxdisplacement12Run1Both =
  MaxValue[{displacementRun1Both, t1Run1Both <= x <= t2Run1Both}, x];
mindisplacement12Run1Both =
  MinValue[{displacementRun1Both, t1Run1Both <= x <= t2Run1Both}, x];
maxpower12Run1Both =
  MaxValue[{powerRun1Both, t1Run1Both <= x <= t2Run1Both}, x];
```

```
minpower12Run1Both =
  MinValue[{powerRun1Both, t1Run1Both <= x <= t2Run1Both}, x];

maxforce23Run1Both =
  MaxValue[{funcRun1Both[x], t2Run1Both <= x <= t3Run1Both}, x];
minforce23Run1Both =
  MinValue[{funcRun1Both[x], t2Run1Both <= x <= t3Run1Both}, x];
maxvelocity23Run1Both =
  MaxValue[{velocityRun1Both, t2Run1Both <= x <= t3Run1Both}, x];
minvelocity23Run1Both =
  MinValue[{velocityRun1Both, t2Run1Both <= x <= t3Run1Both}, x];
maxdisplacement23Run1Both =
  MaxValue[{displacementRun1Both, t2Run1Both <= x <= t3Run1Both}, x];
mindisplacement23Run1Both =
  MinValue[{displacementRun1Both, t2Run1Both <= x <= t3Run1Both}, x];
maxpower23Run1Both =
  MaxValue[{powerRun1Both, t2Run1Both <= x <= t3Run1Both}, x];
minpower23Run1Both =
  MinValue[{powerRun1Both, t2Run1Both <= x <= t3Run1Both}, x];

maxforce34Run1Both =
  MaxValue[{funcRun1Both[x], t3Run1Both <= x <= t4Run1Both}, x];
minforce34Run1Both =
  MinValue[{funcRun1Both[x], t3Run1Both <= x <= t4Run1Both}, x];
maxvelocity34Run1Both =
  MaxValue[{velocityRun1Both, t3Run1Both <= x <= t4Run1Both}, x];
minvelocity34Run1Both =
  MinValue[{velocityRun1Both, t3Run1Both <= x <= t4Run1Both}, x];
maxdisplacement34Run1Both =
  MaxValue[{displacementRun1Both, t3Run1Both <= x <= t4Run1Both}, x];
mindisplacement34Run1Both =
  MinValue[{displacementRun1Both, t3Run1Both <= x <= t4Run1Both}, x];
maxpower34Run1Both =
  MaxValue[{powerRun1Both, t3Run1Both <= x <= t4Run1Both}, x];
minpower34Run1Both =
  MinValue[{powerRun1Both, t3Run1Both <= x <= t4Run1Both}, x];

maxforce45Run1Both =
  MaxValue[{funcRun1Both[x], t4Run1Both <= x <= t5Run1Both}, x];
minforce45Run1Both =
  MinValue[{funcRun1Both[x], t4Run1Both <= x <= t5Run1Both}, x];
maxvelocity45Run1Both =
  MaxValue[{velocityRun1Both, t4Run1Both <= x <= t5Run1Both}, x];
minvelocity45Run1Both =
  MinValue[{velocityRun1Both, t4Run1Both <= x <= t5Run1Both}, x];
maxdisplacement45Run1Both =
  MaxValue[{displacementRun1Both, t4Run1Both <= x <= t5Run1Both}, x];
mindisplacement45Run1Both =
  MinValue[{displacementRun1Both, t4Run1Both <= x <= t5Run1Both}, x];
maxpower45Run1Both =
  MaxValue[{powerRun1Both, t4Run1Both <= x <= t5Run1Both}, x];
minpower45Run1Both =
  MinValue[{powerRun1Both, t4Run1Both <= x <= t5Run1Both}, x];

maxforce2maxRun1Both =
  MaxValue[{funcRun1Both[x], t2Run1Both <= x <= tmaxRun1Both}, x];
minforce2maxRun1Both =
  MinValue[{funcRun1Both[x], t2Run1Both <= x <= tmaxRun1Both}, x];
maxvelocity2maxRun1Both =
  MaxValue[{velocityRun1Both, t2Run1Both <= x <= tmaxRun1Both}, x];
minvelocity2maxRun1Both =
  MinValue[{velocityRun1Both, t2Run1Both <= x <= tmaxRun1Both}, x];
maxdisplacement2maxRun1Both =
  MaxValue[{displacementRun1Both, t2Run1Both <= x <= tmaxRun1Both}, x];
mindisplacement2maxRun1Both =
  MinValue[{displacementRun1Both, t2Run1Both <= x <= tmaxRun1Both}, x];
maxpower2maxRun1Both =
```

```
  MaxValue[{powerRun1Both, t2Run1Both <= x <= tmaxRun1Both}, x];
minpower2maxRun1Both =
  MinValue[{powerRun1Both, t2Run1Both <= x <= tmaxRun1Both}, x];

maxforce1halfRun1Both =
  MaxValue[{funcRun1Both[x], t1Run1Both <= x <= thalfRun1Both}, x];
minforce1halfRun1Both =
  MinValue[{funcRun1Both[x], t1Run1Both <= x <= thalfRun1Both}, x];
maxvelocity1halfRun1Both =
  MaxValue[{velocityRun1Both, t1Run1Both <= x <= thalfRun1Both}, x];
minvelocity1halfRun1Both =
  MinValue[{velocityRun1Both, t1Run1Both <= x <= thalfRun1Both}, x];
maxdisplacement1halfRun1Both =
  MaxValue[{displacementRun1Both, t1Run1Both <= x <= thalfRun1Both},
   x];
mindisplacement1halfRun1Both =
  MinValue[{displacementRun1Both, t1Run1Both <= x <= thalfRun1Both},
   x];
maxpower1halfRun1Both =
  MaxValue[{powerRun1Both, t1Run1Both <= x <= thalfRun1Both}, x];
minpower1halfRun1Both =
  MinValue[{powerRun1Both, t1Run1Both <= x <= thalfRun1Both}, x];

(*TableRun1Both={{"mass= ",massRun1Both},{},
{"t1= ",t1Run1Both,"force @ t1= ",force1Run1Both,"veloc. @ t1= \
",velocity1Run1Both,"displace. @ t1= ",displacement1Run1Both,"power @ \
t1= ",power1Run1Both},{"t2= ",t2Run1Both,"force @ t2= \
",force2Run1Both,"veloc. @ t2= ",velocity2Run1Both,"displace. @ t2= \
",displacement2Run1Both,"power @ t2= ",power2Run1Both},{"t3= \
",t3Run1Both,"force @ t3= ",force3Run1Both,"veloc. @ t3= \
",velocity3Run1Both,"displace. @ t3= ",displacement3Run1Both,"power @ \
t3= ",power3Run1Both},{"t4= ",t4Run1Both,"force @ t4= \
",force4Run1Both,"veloc. @ t4= ",velocity4Run1Both,"displace. @ t4= \
",displacement4Run1Both,"power @ t4= ",power4Run1Both},{"t5= \
",t5Run1Both,"force @ t5= ",force5Run1Both,"veloc. @ t5= \
",velocity5Run1Both,"displace. @ t5= ",displacement5Run1Both,"power @ \
t5= ",power5Run1Both},{"tmax= ",tmaxRun1Both,"force @ tmax= \
",forcemaxRun1Both,"veloc. @ tmax= ",velocitymaxRun1Both,"displace. @ \
tmax= ",displacementmaxRun1Both,"power @ tmax= \
",powermaxRun1Both},{"thalf= ",thalfRun1Both,"force @ thalf= \
",forcehalfRun1Both,"veloc. @ thalf= \
",velocityhalfRun1Both,"displace. @ thalf= \
",displacementhalfRun1Both,"power @ thalf= \
",powerhalfRun1Both},{},{"impulse t1-t2= ",impulse12Run1Both,"impulse \
t2-t3= ",impulse23Run1Both,"impulse t3-t4= \
",impulse34Run1Both,"impulse t4-t5= ",impulse45Run1Both,"impulse \
t2-tmax= ",impulse2maxRun1Both},{"max force t1-t2= \
",maxforce12Run1Both,"max force t2-t3= ",maxforce23Run1Both,"max \
force t3-t4= ",maxforce34Run1Both,"max force t4-t5= \
",maxforce45Run1Both,"max force t2-tmax= ",maxforce2maxRun1Both,"max \
force t1-thalf= ",maxforce1halfRun1Both},{"min force t1-t2= \
",minforce12Run1Both,"min force t2-t3= ",minforce23Run1Both,"min \
force t3-t4= ",minforce34Run1Both,"min force t4-t5= \
",minforce45Run1Both,"min force t2-tmax= ",minforce2maxRun1Both,"min \
force t1-thalf= ",minforce1halfRun1Both},{"max veloc. t1-t2= \
",maxvelocity12Run1Both,"max veloc. t2-t3= \
",maxvelocity23Run1Both,"max veloc. t3-t4= \
",maxvelocity34Run1Both,"max veloc. t4-t5= \
",maxvelocity45Run1Both,"max veloc. t2-tmax= \
",maxvelocity2maxRun1Both,"max veloc. t1-thalf= \
",maxvelocity1halfRun1Both},{"min veloc. t1-t2= \
",minvelocity12Run1Both,"min veloc. t2-t3= \
",minvelocity23Run1Both,"min veloc. t3-t4= \
",minvelocity34Run1Both,"min veloc. t4-t5= \
",minvelocity45Run1Both,"min veloc. t2-tmax= \
",minvelocity2maxRun1Both,"min veloc. t1-thalf= \
",minvelocity1halfRun1Both},{"max displace. t1-t2= \
```

```
",maxdisplacement12Run1Both,"max displace. t2-t3= \
",maxdisplacement23Run1Both,"max displace. t3-t4= \
",maxdisplacement34Run1Both,"max displace. t4-t5= \
",maxdisplacement45Run1Both,"max displace. t2-tmax= \
",maxdisplacement2maxRun1Both,"max displace. t1-thalf= \
",maxdisplacement1halfRun1Both},{"min displace. t1-t2= \
",mindisplacement12Run1Both,"min displace. t2-t3= \
",mindisplacement23Run1Both,"min displace. t3-t4= \
",mindisplacement34Run1Both,"min displace. t4-t5= \
",mindisplacement45Run1Both,"min displace. t2-tmax= \
",mindisplacement2maxRun1Both,"min displace. t1-thalf= \
",mindisplacement1halfRun1Both},{"max power t1-t2= \
",maxpower12Run1Both,"max power t2-t3= ",maxpower23Run1Both,"max \
power t3-t4= ",maxpower34Run1Both,"max power t4-t5= \
",maxpower45Run1Both,"max power t2-tmax= ",maxpower2maxRun1Both,"max \
power t1-thalf= ",maxpower1halfRun1Both},
{"min power t1-t2= ",minpower12Run1Both,"min power t2-t3= \
",minpower23Run1Both,"min power t3-t4= ",minpower34Run1Both,"min \
power t4-t5= ",minpower45Run1Both,"min power t2-tmax= \
",minpower2maxRun1Both,"min power t1-thalf= ",minpower1halfRun1Both}} ;
Grid[TableRun1Both,Frame\[Rule]All]*)

(*classify the type of NCMJ based on number of peaks--look at the \
number of maxima/minima of the first derivative*)

localrootsRun1Both =
  Length[findAllRoots[
    rootfuncRun1Both[x], {x, t3Run1Both, t4Run1Both},
    "ShowPlot" -> False]];
If[localrootsRun1Both == 1, {runtypeRun1Both = 1,
   Print["NCMJ Type 1 (one peak)"]}, {runtypeRun1Both = 2,
   Print["NCMJ Type 2 (two or more peaks)"]}];

Print["tsquat= ", tsquatRun1Both]
Print["t1= ", t1Run1Both]
Print["t2= ", t2Run1Both]
Print["t3= ", t3Run1Both]
Print["t4= ", t4Run1Both]
Print["t5= ", t5Run1Both]
Print["thalf= ", thalfRun1Both]
Print["tmax= ", tmaxRun1Both]

line1Run1Both = Line[{{t1Run1Both, -5000}, {t1Run1Both, 5000}}];
line2Run1Both = Line[{{t2Run1Both, -5000}, {t2Run1Both, 5000}}];
line3Run1Both = Line[{{t3Run1Both, -5000}, {t3Run1Both, 5000}}];
line4Run1Both = Line[{{t4Run1Both, -5000}, {t4Run1Both, 5000}}];
line5Run1Both = Line[{{t5Run1Both, -5000}, {t5Run1Both, 5000}}];
linehalfRun1Both =
  Line[{{thalfRun1Both, -5000}, {thalfRun1Both, 5000}}];
linemaxRun1Both =
  Line[{{tmaxRun1Both, -5000}, {tmaxRun1Both, 5000}}];
lineStyle = {Thin, Red, Dashed};
bodyweightlineRun1Both =
  Line[{{0, BWforceRun1Both}, {tmaxRun1Both, BWforceRun1Both}}];
cutofflineRun1Both =
  Line[{{cutoffRun1Both, -5000}, {cutoffRun1Both, 5000}}];
squatavglineRun1Both =
  Line[{{0, squatavgRun1Both}, {cutoffRun1Both, squatavgRun1Both}}];
squatlineRun1Both =
  Line[{{tsquatRun1Both, -5000}, {tsquatRun1Both, 5000}}];

Print["Body Weight= ", BWforceRun1Both]
Print["Force in Air= ", noforceRun1Both]
Plot[{funcRun1Both[x]}, {x, cutoffRun1Both, lastTimeRun1Both},
 ImageSize -> 600, PlotStyle -> {Automatic},
 Epilog -> {Directive[lineStyle], line1Run1Both, line2Run1Both,
   line3Run1Both, line4Run1Both, line5Run1Both,
```

```
   Directive[Thin, Blue, Dashed], linehalfRun1Both, linemaxRun1Both,
   Directive[Thin, Black, Dashed], bodyweightlineRun1Both},
 PlotRange -> All, BaseStyle -> { FontSize -> 12},
 PlotLabel -> "NCMJ - Normal Force",
 AxesLabel -> {"Time (s)", "Normal Force (N)"} ]
Plot[{velocityRun1Both}, {x, cutoffRun1Both, lastTimeRun1Both},
 ImageSize -> 600, PlotStyle -> {Automatic},
 Epilog -> {Directive[lineStyle], line1Run1Both, line2Run1Both,
   line3Run1Both, line4Run1Both, line5Run1Both,
   Directive[Thin, Blue, Dashed], linehalfRun1Both, linemaxRun1Both,
   Directive[Thin, Black, Dashed], bodyweightlineRun1Both},
 PlotRange -> All, BaseStyle -> { FontSize -> 12},
 PlotLabel -> "NCMJ - Velocity",
 AxesLabel -> {"Time (s)", "Velocity {m/s}"} ]
Plot[{funcRun1Both[x]}, {x, 0, lastTimeRun1Both}, ImageSize -> 600,
 PlotStyle -> {Automatic},
 Epilog -> {Directive[lineStyle], line1Run1Both, line2Run1Both,
   line3Run1Both, line4Run1Both, line5Run1Both,
   Directive[Thin, Blue, Dashed], linehalfRun1Both, linemaxRun1Both,
   Directive[Thin, Black, Dashed], cutofflineRun1Both,
   bodyweightlineRun1Both, squatavglineRun1Both, squatlineRun1Both},
 PlotRange -> All, BaseStyle -> { FontSize -> 12},
 PlotLabel -> "NCMJ - Normal Force",
 AxesLabel -> {"Time (s)", "Normal Force (N)"} ]
(*Plot[{velocityRun1Both},{x,0,lastTimeRun1Both},ImageSize\[Rule]600,\
PlotStyle \[Rule] \
{Automatic},Epilog\[Rule]{Directive[lineStyle],line1Run1Both,\
line2Run1Both,line3Run1Both,line4Run1Both,line5Run1Both, \
Directive[Thin,Blue,Dashed],linehalfRun1Both,linemaxRun1Both, \
Directive[Thin,Black,Dashed],bodyweightlineRun1Both}, \
PlotRange\[Rule]All, BaseStyle\[Rule]{ FontSize\[Rule] 12},PlotLabel\
\[Rule] "NCMJ - Velocity", AxesLabel\[Rule]{"Time (s)", "Velocity \
(m/s)"} ]
Plot[{displacementRun1Both},{x,0,lastTimeRun1Both},ImageSize\[Rule]\
600,PlotStyle \[Rule] \
{Automatic},Epilog\[Rule]{Directive[lineStyle],line1Run1Both,\
line2Run1Both,line3Run1Both,line4Run1Both,line5Run1Both, \
Directive[Thin,Blue,Dashed],linehalfRun1Both,linemaxRun1Both},\
PlotRange\[Rule]All, BaseStyle\[Rule]{ FontSize\[Rule] 12},PlotLabel\
\[Rule] "NCMJ - COM Displacement", AxesLabel\[Rule]{"Time (s)", \
"Normal Force (N)"} ]
Plot[{powerRun1Both},{x,0,lastTimeRun1Both},ImageSize\[Rule]600,\
PlotStyle \[Rule] \
{Automatic},Epilog\[Rule]{Directive[lineStyle],line1Run1Both,\
line2Run1Both,line3Run1Both,line4Run1Both,line5Run1Both, \
Directive[Thin,Blue,Dashed],linehalfRun1Both,linemaxRun1Both, \
Directive[Thin,Black,Dashed],bodyweightlineRun1Both,},PlotRange\[Rule]\
All, BaseStyle\[Rule]{ FontSize\[Rule] 12},PlotLabel\[Rule] "NCMJ - \
Power", AxesLabel\[Rule]{"Time (s)", "Normal Force (N)"} ]*)
```

(Cells for Run 2 through Run 5 inserted here, with all code that includes the naming system “Run1” in it replaced with the corresponding run number (i.e. “Run2” for the second jump trial, etc.)

```
(*ENTER IN NAME AND DATE*)
NameID = "NAME";
DateID = "DATE";

IDRun1 = "" <> NameID <> " NCMJ Run 1 " <> DateID <> "";
IDRun2 = "" <> NameID <> " NCMJ Run 2 " <> DateID <> "";
IDRun3 = "" <> NameID <> " NCMJ Run 3 " <> DateID <> "";
IDRun4 = "" <> NameID <> " NCMJ Run 4 " <> DateID <> "";
IDRun5 = "" <> NameID <> " NCMJ Run 5 " <> DateID <> "";

(* Run1 *)
solutionRun1 = {{IDRun1, "", ""}, {"", "",
    ""}, {"Variable", "Left", "Right"}, {"", "", ""},
```

```
{"run class.:", runtypeRun1L, runtypeRun1R}, {"mass:", massRun1L,
massRun1R}, {"t1:", t1Run1L, t1Run1R}, {"t2:", t2Run1L,
t2Run1R}, {"t3:", t3Run1L, t3Run1R}, {"t4:", t4Run1L,
t4Run1R}, {"t5:", t5Run1L, t5Run1R}, {"tmax:", tmaxRun1L,
tmaxRun1R}, {"thalf:", thalfRun1L, thalfRun1R}, {"", "",
""}, {"force @ t1:", force1Run1L, force1Run1R}, {"force @ t2:",
force2Run1L, force2Run1R}, {"force @ t3:", force3Run1L,
force3Run1R}, {"force @ t4:", force4Run1L,
force4Run1R}, {"force @ t5:", force5Run1L,
force5Run1R}, {"force @ tmax:", forcemaxRun1L,
forcemaxRun1R}, {"force @ thalf:", forcehalfRun1L,
forcehalfRun1R}, {"veloc. @ t1:", velocity1Run1L,
velocity1Run1R}, {"veloc. @ t2:", velocity2Run1L,
velocity2Run1R}, {"veloc. @ t3:", velocity3Run1L,
velocity3Run1R}, {"veloc. @ t4:", velocity4Run1L,
velocity4Run1R}, {"veloc. @ t5:", velocity5Run1L,
velocity5Run1R}, {"veloc. @ tmax:", velocitymaxRun1L,
velocitymaxRun1R}, {"veloc. @ thalf:", velocityhalfRun1L,
velocityhalfRun1R}, {"displace. @ t1:", displacement1Run1L,
displacement1Run1R}, {"displace. @ t2:", displacement2Run1L,
displacement2Run1R}, {"displace. @ t3:", displacement3Run1L,
displacement3Run1R}, {"displace. @ t4:", displacement4Run1L,
displacement4Run1R}, {"displace. @ t5:", displacement5Run1L,
displacement5Run1R}, {"displace. @ tmax:", displacementmaxRun1L,
displacementmaxRun1R}, {"displace. @ thalf:",
displacementhalfRun1L, displacementhalfRun1R}, {"power @ t1:",
power1Run1L, power1Run1R}, {"power @ t2:", power2Run1L,
power2Run1R}, {"power @ t3:", power3Run1L,
power3Run1R}, {"power @ t4:", power4Run1L,
power4Run1R}, {"power @ t5:", power5Run1L,
power5Run1R}, {"power @ tmax:", powermaxRun1L,
powermaxRun1R}, {"power @ thalf:", powerhalfRun1L,
powerhalfRun1R}, {"", "", ""}, {"impulse t1-t2:", impulse12Run1L,
impulse12Run1R}, {"impulse t2-t3:", impulse23Run1L,
impulse23Run1R}, {"impulse t3-t4:", impulse34Run1L,
impulse34Run1R}, {"impulse t4-t5:", impulse45Run1L,
impulse45Run1R}, {"impulse t2-tmax:", impulse2maxRun1L,
impulse2maxRun1R}, {"impulse t1-thalf:", impulse1halfRun1L,
impulse1halfRun1R}, {"max force t1-t2:", maxforce12Run1L,
maxforce12Run1R}, {"max force t2-t3:", maxforce23Run1L,
maxforce23Run1R}, {"max force t3-t4:", maxforce34Run1L,
maxforce34Run1R}, {"max force t4-t5:", maxforce45Run1L,
maxforce45Run1R}, {"max force t2-tmax:", maxforce2maxRun1L,
maxforce2maxRun1R}, {"max force t1-thalf:", maxforce1halfRun1L,
maxforce1halfRun1R}, {"min force t1-t2:", minforce12Run1L,
minforce12Run1R}, {"min force t2-t3:", minforce23Run1L,
minforce23Run1R}, {"min force t3-t4:", minforce34Run1L,
minforce34Run1R}, {"min force t4-t5:", minforce45Run1L,
minforce45Run1R}, {"min force t2-tmax:", minforce2maxRun1L,
minforce2maxRun1R}, {"min force t1-thalf:", minforce1halfRun1L,
minforce1halfRun1R}, {"max veloc. t1-t2:", maxvelocity12Run1L,
maxvelocity12Run1R}, {"max veloc. t2-t3:", maxvelocity23Run1L,
maxvelocity23Run1R}, {"max veloc. t3-t4:", maxvelocity34Run1L,
maxvelocity34Run1R}, {"max veloc. t4-t5:", maxvelocity45Run1L,
maxvelocity45Run1R}, {"max veloc. t2-tmax:", maxvelocity2maxRun1L,
 maxvelocity2maxRun1R}, {"max veloc. t1-thalf:",
maxvelocity1halfRun1L,
maxvelocity1halfRun1R}, {"min veloc. t1-t2:", minvelocity12Run1L,
minvelocity12Run1R}, {"min veloc. t2-t3:", minvelocity23Run1L,
minvelocity23Run1R}, {"min veloc. t3-t4:", minvelocity34Run1L,
minvelocity34Run1R}, {"min veloc. t4-t5:", minvelocity45Run1L,
minvelocity45Run1R}, {"min veloc. t2-tmax:", minvelocity2maxRun1L,
 minvelocity2maxRun1R}, {"min veloc. t1-thalf:",
minvelocity1halfRun1L,
minvelocity1halfRun1R}, {"max displace. t1-t2:",
maxdisplacement12Run1L,
maxdisplacement12Run1R}, {"max displace. t2-t3:",
```

```
maxdisplacement23Run1L,
maxdisplacement23Run1R}, {"max displace. t3-t4:",
maxdisplacement34Run1L,
maxdisplacement34Run1R}, {"max displace. t4-t5:",
maxdisplacement45Run1L,
maxdisplacement45Run1R}, {"max displace. t2-tmax:",
maxdisplacement2maxRun1L,
maxdisplacement2maxRun1R}, {"max displace. t1-thalf:",
maxdisplacement1halfRun1L,
maxdisplacement1halfRun1R}, {"min displace. t1-t2:",
mindisplacement12Run1L,
mindisplacement12Run1R}, {"min displace. t2-t3:",
mindisplacement23Run1L,
mindisplacement23Run1R}, {"min displace. t3-t4:",
mindisplacement34Run1L,
mindisplacement34Run1R}, {"min displace. t4-t5:",
mindisplacement45Run1L,
mindisplacement45Run1R}, {"min displace. t2-tmax:",
mindisplacement2maxRun1L,
mindisplacement2maxRun1R}, {"min displace. t1-thalf:",
mindisplacement1halfRun1L,
mindisplacement1halfRun1R}, {"max power t1-t2:", maxpower12Run1L,
maxpower12Run1R}, {"max power t2-t3:", maxpower23Run1L,
maxpower23Run1R}, {"max power t3-t4:", maxpower34Run1L,
maxpower34Run1R}, {"max power t4-t5:", maxpower45Run1L,
maxpower45Run1R}, {"max power t2-tmax:", maxpower2maxRun1L,
maxpower2maxRun1R}, {"max power t1-thalf:", maxpower1halfRun1L,
maxpower1halfRun1R}, {"min power t1-t2:", minpower12Run1L,
minpower12Run1R}, {"min power t2-t3:", minpower23Run1L,
minpower23Run1R}, {"min power t3-t4:", minpower34Run1L,
minpower34Run1R}, {"min power t4-t5:", minpower45Run1L,
minpower45Run1R}, {"min power t2-tmax:", minpower2maxRun1L,
minpower2maxRun1R}, {"min power t1-thalf:", minpower1halfRun1L,
minpower1halfRun1R}, {"avg. force in air:", noforceRun1L,
noforceRun1R}, {"avg. force in squat:", squatavgRun1L,
squatavgRun1R}, {"st. dev. force in squat:", squatstdevRun1L,
squatstdevRun1R}, {"", "", ""}, {"Legs Analyzed Together:", "",
""}, {"", "", ""}, {"Variable", "Legs Combined", ""}, {"", "",
""}, {"run class.:", runtypeRun1Both, ""}, {"mass:", massRun1Both,
 ""}, {"t1:", t1Run1Both, ""}, {"t2:", t2Run1Both, ""}, {"t3:",
t3Run1Both, ""}, {"t4:", t4Run1Both, ""}, {"t5:", t5Run1Both,
""}, {"tmax:", tmaxRun1Both, ""}, {"", "", ""}, {"force @ t1:",
force1Run1Both, ""}, {"force @ t2:", force2Run1Both,
""}, {"force @ t3:", force3Run1Both, ""}, {"force @ t4:",
force4Run1Both, ""}, {"force @ t5:", force5Run1Both,
""}, {"force @ tmax:", forcemaxRun1Both, ""}, {"force @ thalf:",
forcehalfRun1Both, ""}, {"veloc. @ t1:", velocity1Run1Both,
""}, {"veloc. @ t2:", velocity2Run1Both, ""}, {"veloc. @ t3:",
velocity3Run1Both, ""}, {"veloc. @ t4:", velocity4Run1Both,
""}, {"veloc. @ t5:", velocity5Run1Both, ""}, {"veloc. @ tmax:",
velocitymaxRun1Both, ""}, {"veloc. @ thalf:",
velocityhalfRun1Both, ""}, {"displace. @ t1:",
displacement1Run1Both, ""}, {"displace. @ t2:",
displacement2Run1Both, ""}, {"displace. @ t3:",
displacement3Run1Both, ""}, {"displace. @ t4:",
displacement4Run1Both, ""}, {"displace. @ t5:",
displacement5Run1Both, ""}, {"displace. @ tmax:",
displacementmaxRun1Both, ""}, {"displace. @ thalf:",
displacementhalfRun1Both, ""}, {"power @ t1:", power1Run1Both,
""}, {"power @ t2:", power2Run1Both, ""}, {"power @ t3:",
power3Run1Both, ""}, {"power @ t4:", power4Run1Both,
""}, {"power @ t5:", power5Run1Both, ""}, {"power @ tmax:",
powermaxRun1Both, ""}, {"power @ thalf:", powerhalfRun1Both,
""}, {"", "", ""}, {"impulse t1-t2:", impulse12Run1Both,
""}, {"impulse t2-t3:", impulse23Run1Both, ""}, {"impulse t3-t4:",
 impulse34Run1Both, ""}, {"impulse t4-t5:", impulse45Run1Both,
""}, {"impulse t2-tmax:", impulse2maxRun1Both,
```

```
    ""}, {"impulse t1-thalf:", impulse1halfRun1Both,
    ""}, {"max force t1-t2:", maxforce12Run1Both,
    ""}, {"max force t2-t3:", maxforce23Run1Both,
    ""}, {"max force t3-t4:", maxforce34Run1Both,
    ""}, {"max force t4-t5:", maxforce45Run1Both,
    ""}, {"max force t2-tmax:", maxforce2maxRun1Both,
    ""}, {"max force t1-thalf:", maxforce1halfRun1Both,
    ""}, {"min force t1-t2:", minforce12Run1Both,
    ""}, {"min force t2-t3:", minforce23Run1Both,
    ""}, {"min force t3-t4:", minforce34Run1Both,
    ""}, {"min force t4-t5", minforce45Run1Both,
    ""}, {"min force t2-tmax:", minforce2maxRun1Both,
    ""}, {"min force t1-thalf:", minforce1halfRun1Both,
    ""}, {"max veloc. t1-t2:", maxvelocity12Run1Both,
    ""}, {"max veloc. t2-t3:", maxvelocity23Run1Both,
    ""}, {"max veloc. t3-t4:", maxvelocity34Run1Both,
    ""}, {"max veloc. t4-t5:", maxvelocity45Run1Both,
    ""}, {"max veloc. t2-tmax:", maxvelocity2maxRun1Both,
    ""}, {"max veloc. t1-thalf:", maxvelocity1halfRun1Both,
    ""}, {"min veloc. t1-t2:", minvelocity12Run1Both,
    ""}, {"min veloc. t2-t3:", minvelocity23Run1Both,
    ""}, {"min veloc. t3-t4:", minvelocity34Run1Both,
    ""}, {"min veloc. t4-t5:", minvelocity45Run1Both,
    ""}, {"min veloc. t2-tmax:", minvelocity2maxRun1Both,
    ""}, {"min veloc. t1-thalf:", minvelocity1halfRun1Both,
    ""}, {"max displace. t1-t2:", maxdisplacement12Run1Both,
    ""}, {"max displace. t2-t3:", maxdisplacement23Run1Both,
    ""}, {"max displace. t3-t4:", maxdisplacement34Run1Both,
    ""}, {"max displace. t4-t5:", maxdisplacement45Run1Both,
    ""}, {"max displace. t2-tmax:", maxdisplacement2maxRun1Both,
    ""}, {"max displace. t1-thalf:", maxdisplacement1halfRun1Both,
    ""}, {"min displace. t1-t2:", mindisplacement12Run1Both,
    ""}, {"min displace. t2-t3:", mindisplacement23Run1Both,
    ""}, {"min displace. t3-t4:", mindisplacement34Run1Both,
    ""}, {"min displace. t4-t5:", mindisplacement45Run1Both,
    ""}, {"min displace. t2-tmax:", mindisplacement2maxRun1Both,
    ""}, {"min displace. t1-thalf:", mindisplacement1halfRun1Both,
    ""}, {"max power t1-t2:", maxpower12Run1Both,
    ""}, {"max power t2-t3:", maxpower23Run1Both,
    ""}, {"max power t3-t4:", maxpower34Run1Both,
    ""}, {"max power t4-t5:", maxpower45Run1Both,
    ""}, {"max power t2-tmax:", maxpower2maxRun1Both,
    ""}, {"max power t1-thalf:", maxpower1halfRun1Both,
    ""}, {"min power t1-t2:", minpower12Run1Both,
    ""}, {"min power t2-t3:", minpower23Run1Both,
    ""}, {"min power t3-t4:", minpower34Run1Both,
    ""}, {"min power t4-t5:", minpower45Run1Both,
    ""}, {"min power t2-tmax:", minpower2maxRun1Both,
    ""}, {"min power t1-thalf:", minpower1halfRun1Both,
    ""}, {"avg. force in air:", noforceRun1Both,
    ""}, {"avg. force in squat:", squatavgRun1Both,
    ""}, {"st. dev. force in squat:", squatstdevRun1Both, ""}};

ForceGraphRun1Both =
 Plot[Tooltip[funcRun1Both[x]], {x, cutoffRun1Both, lastTimeRun1Both},
   ImageSize -> 600, PlotStyle -> {Automatic},
  PlotLegends -> {"Legs Combined"},
  Epilog -> {Directive[Blue, Dashed], line1Run1Both, line2Run1Both,
    line3Run1Both, line4Run1Both, line5Run1Both,
    Directive[Blue, Dashing[None]], linemaxRun1Both,
    linehalfRun1Both}, PlotRange -> All,
  PlotLabel -> "Non-Counter Movement Jump - Normal Force",
  AxesLabel -> {"Time (s)", "Normal Force (N)"} ]
ForceGraphRun1 =
 Plot[Tooltip[{funcRun1L[x], funcRun1R[x]}], {x, cutoffRun1Both,
   lastTimeRun1Both}, ImageSize -> 600,
  PlotLegends -> (Placed[#, Right] & /@ {{"Left", "Right"},
```

```
      LineLegend[{{Red, Dashed}, Dashed, Black}, {"Left", "Right"}]}),
    PlotStyle -> {Automatic},
   Epilog -> {Directive[Red, Dashed], line1Run1L, line2Run1L,
     line3Run1L, line4Run1L, line5Run1L, Directive[Red, Dashing[None]],
      linemaxRun1L, linehalfRun1L, Directive[Black, Dashed],
     line1Run1R, line2Run1R, line3Run1R, line4Run1R, line5Run1R,
     Directive[Black, Dashing[None]], linemaxRun1R, linehalfRun1R},
   PlotRange -> All,
   PlotLabel -> "Non-Counter Movement Jump - Normal Force",
   AxesLabel -> {"Time (s)", "Normal Force (N)"} ]
VelocityGraphRun1Both =
 Plot[Tooltip[velocityRun1Both], {x, cutoffRun1Both,
    lastTimeRun1Both}, ImageSize -> 600, PlotStyle -> {Automatic},
   PlotLegends -> {"Legs Combined"},
   Epilog -> {Directive[Blue, Dashed], line1Run1Both, line2Run1Both,
     line3Run1Both, line4Run1Both, line5Run1Both,
     Directive[Blue, Dashing[None]], linemaxRun1Both,
     linehalfRun1Both}, PlotRange -> All,
   PlotLabel -> "Non-Counter Movement Jump - Velocity",
   AxesLabel -> {"Time (s)", "Velocity (m/s)"} ]
VelocityGraphRun1 =
  Plot[Tooltip[{velocityRun1L, velocityRun1R}], {x, cutoffRun1Both,
     lastTimeRun1Both}, ImageSize -> 600,
    PlotLegends -> (Placed[#, Right] & /@ {{"Left", "Right"},
        LineLegend[{{Red, Dashed}, Dashed, Black}, {"Left",
          "Right"}]}), PlotStyle -> {Automatic}, PlotRange -> All,
    Epilog -> {Directive[Red, Dashed], line1Run1L, line2Run1L,
      line3Run1L, line4Run1L, line5Run1L,
      Directive[Red, Dashing[None]], linemaxRun1L, linehalfRun1L,
      Directive[Black, Dashed], line1Run1R, line2Run1R, line3Run1R,
      line4Run1R, line5Run1R, Directive[Black, Dashing[None]],
      linemaxRun1R, linehalfRun1R},
    PlotLabel -> "Non-Counter Movement Jump - Velocity",
    AxesLabel -> {"Time (s)", "Velocity (m/s)"} ];
DisplacementGraphRun1Both =
  Plot[Tooltip[displacementRun1Both], {x, cutoffRun1Both,
     lastTimeRun1Both}, ImageSize -> 600, PlotStyle -> {Automatic},
    PlotLegends -> {"Legs Combined"},
    Epilog -> {Directive[Blue, Dashed], line1Run1Both, line2Run1Both,
      line3Run1Both, line4Run1Both, line5Run1Both,
      Directive[Blue, Dashing[None]], linemaxRun1Both,
      linehalfRun1Both}, PlotRange -> All,
    PlotLabel -> "Non-Counter Movement Jump - Displacement",
    AxesLabel -> {"Time (s)", "Displacement (m)"} ];
DisplacementGraphRun1 =
  Plot[Tooltip[{displacementRun1L, displacementRun1R}], {x,
     cutoffRun1Both, lastTimeRun1Both}, ImageSize -> 600,
    PlotLegends -> (Placed[#, Right] & /@ {{"Left", "Right"},
        LineLegend[{{Red, Dashed}, Dashed, Black}, {"Left",
          "Right"}]}), PlotStyle -> {Automatic}, PlotRange -> All,
    Epilog -> {Directive[Red, Dashed], line1Run1L, line2Run1L,
      line3Run1L, line4Run1L, line5Run1L,
      Directive[Red, Dashing[None]], linemaxRun1L, linehalfRun1L,
      Directive[Black, Dashed], line1Run1R, line2Run1R, line3Run1R,
      line4Run1R, line5Run1R, Directive[Black, Dashing[None]],
      linemaxRun1R, linehalfRun1R},
    PlotLabel -> "Non-Counter Movement Jump - Displacement",
    AxesLabel -> {"Time (s)", "Displacement (m)"} ];
PowerGraphRun1Both =
  Plot[Tooltip[powerRun1Both], {x, cutoffRun1Both, lastTimeRun1Both},
    ImageSize -> 600, PlotStyle -> {Automatic},
    Epilog -> {Directive[Blue, Dashed],
      PlotLegends -> {"Legs Combined"}, line1Run1Both, line2Run1Both,
      line3Run1Both, line4Run1Both, line5Run1Both,
      Directive[Blue, Dashing[None]], linemaxRun1Both,
      linehalfRun1Both}, PlotRange -> All,
    PlotLabel -> "Non-Counter Movement Jump - Power",
```

```
   AxesLabel -> {"Time (s)", "Power (W)"} ];
PowerGraphRun1 =
  Plot[Tooltip[{powerRun1L, powerRun1R}], {x, cutoffRun1Both,
    lastTimeRun1Both}, ImageSize -> 600,
   PlotLegends -> (Placed[#, Right] & /@ {{"Left", "Right"},
       LineLegend[{{Red, Dashed}, Dashed, Black}, {"Left",
         "Right"}]}), PlotStyle -> {Automatic}, PlotRange -> All,
   Epilog -> {Directive[Red, Dashed], line1Run1L, line2Run1L,
     line3Run1L, line4Run1L, line5Run1L,
     Directive[Red, Dashing[None]], linemaxRun1L, linehalfRun1L,
     Directive[Black, Dashed], line1Run1R, line2Run1R, line3Run1R,
     line4Run1R, line5Run1R, Directive[Black, Dashing[None]],
     linemaxRun1R, linehalfRun1R},
   PlotLabel -> "Non-Counter Movement Jump - Power",
   AxesLabel -> {"Time (s)", "Power (W)"} ];

(* Run2 *)
solutionRun2 = {{IDRun2, "", ""}, {"", "",
    ""}, {"Variable", "Left", "Right"}, {"", "", ""},
   {"run class.:", runtypeRun2L, runtypeRun2R}, {"mass:", massRun2L,
    massRun2R}, {"t1:", t1Run2L, t1Run2R}, {"t2:", t2Run2L,
    t2Run2R}, {"t3:", t3Run2L, t3Run2R}, {"t4:", t4Run2L,
    t4Run2R}, {"t5:", t5Run2L, t5Run2R}, {"tmax:", tmaxRun2L,
    tmaxRun2R}, {"thalf:", thalfRun2L, thalfRun2R}, {"", "",
    ""}, {"force @ t1:", force1Run2L, force1Run2R}, {"force @ t2:",
    force2Run2L, force2Run2R}, {"force @ t3:", force3Run2L,
    force3Run2R}, {"force @ t4:", force4Run2L,
    force4Run2R}, {"force @ t5:", force5Run2L,
    force5Run2R}, {"force @ tmax:", forcemaxRun2L,
    forcemaxRun2R}, {"force @ thalf:", forcehalfRun2L,
    forcehalfRun2R}, {"veloc. @ t1:", velocity1Run2L,
    velocity1Run2R}, {"veloc. @ t2:", velocity2Run2L,
    velocity2Run2R}, {"veloc. @ t3:", velocity3Run2L,
    velocity3Run2R}, {"veloc. @ t4:", velocity4Run2L,
    velocity4Run2R}, {"veloc. @ t5:", velocity5Run2L,
    velocity5Run2R}, {"veloc. @ tmax:", velocitymaxRun2L,
    velocitymaxRun2R}, {"veloc. @ thalf:", velocityhalfRun2L,
    velocityhalfRun2R}, {"displace. @ t1:", displacement1Run2L,
    displacement1Run2R}, {"displace. @ t2:", displacement2Run2L,
    displacement2Run2R}, {"displace. @ t3:", displacement3Run2L,
    displacement3Run2R}, {"displace. @ t4:", displacement4Run2L,
    displacement4Run2R}, {"displace. @ t5:", displacement5Run2L,
    displacement5Run2R}, {"displace. @ tmax:", displacementmaxRun2L,
    displacementmaxRun2R}, {"displace. @ thalf:",
    displacementhalfRun2L, displacementhalfRun2R}, {"power @ t1:",
    power1Run2L, power1Run2R}, {"power @ t2:", power2Run2L,
    power2Run2R}, {"power @ t3:", power3Run2L,
    power3Run2R}, {"power @ t4:", power4Run2L,
    power4Run2R}, {"power @ t5:", power5Run2L,
    power5Run2R}, {"power @ tmax:", powermaxRun2L,
    powermaxRun2R}, {"power @ thalf:", powerhalfRun2L,
    powerhalfRun2R}, {"", "", ""}, {"impulse t1-t2:", impulse12Run2L,
    impulse12Run2R}, {"impulse t2-t3:", impulse23Run2L,
    impulse23Run2R}, {"impulse t3-t4:", impulse34Run2L,
    impulse34Run2R}, {"impulse t4-t5:", impulse45Run2L,
    impulse45Run2R}, {"impulse t2-tmax:", impulse2maxRun2L,
    impulse2maxRun2R}, {"impulse t1-thalf:", impulse1halfRun2L,
    impulse1halfRun2R}, {"max force t1-t2:", maxforce12Run2L,
    maxforce12Run2R}, {"max force t2-t3:", maxforce23Run2L,
    maxforce23Run2R}, {"max force t3-t4:", maxforce34Run2L,
    maxforce34Run2R}, {"max force t4-t5:", maxforce45Run2L,
    maxforce45Run2R}, {"max force t2-tmax:", maxforce2maxRun2L,
    maxforce2maxRun2R}, {"max force t1-thalf:", maxforce1halfRun2L,
    maxforce1halfRun2R}, {"min force t1-t2:", minforce12Run2L,
    minforce12Run2R}, {"min force t2-t3:", minforce23Run2L,
    minforce23Run2R}, {"min force t3-t4:", minforce34Run2L,
    minforce34Run2R}, {"min force t4-t5:", minforce45Run2L,
```

```
minforce45Run2R}, {"min force t2-tmax:", minforce2maxRun2L,
minforce2maxRun2R}, {"min force t1-thalf:", minforce1halfRun2L,
minforce1halfRun2R}, {"max veloc. t1-t2:", maxvelocity12Run2L,
maxvelocity12Run2R}, {"max veloc. t2-t3:", maxvelocity23Run2L,
maxvelocity23Run2R}, {"max veloc. t3-t4:", maxvelocity34Run2L,
maxvelocity34Run2R}, {"max veloc. t4-t5:", maxvelocity45Run2L,
maxvelocity45Run2R}, {"max veloc. t2-tmax:", maxvelocity2maxRun2L,
 maxvelocity2maxRun2R}, {"max veloc. t1-thalf:",
maxvelocity1halfRun2L,
maxvelocity1halfRun2R}, {"min veloc. t1-t2:", minvelocity12Run2L,
minvelocity12Run2R}, {"min veloc. t2-t3:", minvelocity23Run2L,
minvelocity23Run2R}, {"min veloc. t3-t4:", minvelocity34Run2L,
minvelocity34Run2R}, {"min veloc. t4-t5:", minvelocity45Run2L,
minvelocity45Run2R}, {"min veloc. t2-tmax:", minvelocity2maxRun2L,
 minvelocity2maxRun2R}, {"min veloc. t1-thalf:",
minvelocity1halfRun2L,
minvelocity1halfRun2R}, {"max displace. t1-t2:",
maxdisplacement12Run2L,
maxdisplacement12Run2R}, {"max displace. t2-t3:",
maxdisplacement23Run2L,
maxdisplacement23Run2R}, {"max displace. t3-t4:",
maxdisplacement34Run2L,
maxdisplacement34Run2R}, {"max displace. t4-t5:",
maxdisplacement45Run2L,
maxdisplacement45Run2R}, {"max displace. t2-tmax:",
maxdisplacement2maxRun2L,
maxdisplacement2maxRun2R}, {"max displace. t1-thalf:",
maxdisplacement1halfRun2L,
maxdisplacement1halfRun2R}, {"min displace. t1-t2:",
mindisplacement12Run2L,
mindisplacement12Run2R}, {"min displace. t2-t3:",
mindisplacement23Run2L,
mindisplacement23Run2R}, {"min displace. t3-t4:",
mindisplacement34Run2L,
mindisplacement34Run2R}, {"min displace. t4-t5:",
mindisplacement45Run2L,
mindisplacement45Run2R}, {"min displace. t2-tmax:",
mindisplacement2maxRun2L,
mindisplacement2maxRun2R}, {"min displace. t1-thalf:",
mindisplacement1halfRun2L,
mindisplacement1halfRun2R}, {"max power t1-t2:", maxpower12Run2L,
maxpower12Run2R}, {"max power t2-t3:", maxpower23Run2L,
maxpower23Run2R}, {"max power t3-t4:", maxpower34Run2L,
maxpower34Run2R}, {"max power t4-t5:", maxpower45Run2L,
maxpower45Run2R}, {"max power t2-tmax:", maxpower2maxRun2L,
maxpower2maxRun2R}, {"max power t1-thalf:", maxpower1halfRun2L,
maxpower1halfRun2R}, {"min power t1-t2:", minpower12Run2L,
minpower12Run2R}, {"min power t2-t3:", minpower23Run2L,
minpower23Run2R}, {"min power t3-t4:", minpower34Run2L,
minpower34Run2R}, {"min power t4-t5:", minpower45Run2L,
minpower45Run2R}, {"min power t2-tmax:", minpower2maxRun2L,
minpower2maxRun2R}, {"min power t1-thalf:", minpower1halfRun2L,
minpower1halfRun2R}, {"avg. force in air:", noforceRun2L,
noforceRun2R}, {"avg. force in squat:", squatavgRun2L,
squatavgRun2R}, {"st. dev. force in squat:", squatstdevRun2L,
squatstdevRun2R}, {"", "", ""}, {"Legs Analyzed Together:", "",
""}, {"", "", ""}, {"Variable", "Legs Combined", ""}, {"", "",
""}, {"run class.:", runtypeRun2Both, ""}, {"mass:", massRun2Both,
 ""}, {"t1:", t1Run2Both, ""}, {"t2:", t2Run2Both, ""}, {"t3:",
t3Run2Both, ""}, {"t4:", t4Run2Both, ""}, {"t5:", t5Run2Both,
""}, {"tmax:", tmaxRun2Both, ""}, {"", "", ""}, {"force @ t1:",
force1Run2Both, ""}, {"force @ t2:", force2Run2Both,
""}, {"force @ t3:", force3Run2Both, ""}, {"force @ t4:",
force4Run2Both, ""}, {"force @ t5:", force5Run2Both,
""}, {"force @ tmax:", forcemaxRun2Both, ""}, {"force @ thalf:",
forcehalfRun2Both, ""}, {"veloc. @ t1:", velocity1Run2Both,
""}, {"veloc. @ t2:", velocity2Run2Both, ""}, {"veloc. @ t3:",
```

```
velocity3Run2Both, ""}, {"veloc. @ t4:", velocity4Run2Both,
""}, {"veloc. @ t5:", velocity5Run2Both, ""}, {"veloc. @ tmax:",
velocitymaxRun2Both, ""}, {"veloc. @ thalf:",
velocityhalfRun2Both, ""}, {"displace. @ t1:",
displacement1Run2Both, ""}, {"displace. @ t2:",
displacement2Run2Both, ""}, {"displace. @ t3:",
displacement3Run2Both, ""}, {"displace. @ t4:",
displacement4Run2Both, ""}, {"displace. @ t5:",
displacement5Run2Both, ""}, {"displace. @ tmax:",
displacementmaxRun2Both, ""}, {"displace. @ thalf:",
displacementhalfRun2Both, ""}, {"power @ t1:", power1Run2Both,
""}, {"power @ t2:", power2Run2Both, ""}, {"power @ t3:",
power3Run2Both, ""}, {"power @ t4:", power4Run2Both,
""}, {"power @ t5:", power5Run2Both, ""}, {"power @ tmax:",
powermaxRun2Both, ""}, {"power @ thalf:", powerhalfRun2Both,
""}, {"", "", ""}, {"impulse t1-t2:", impulse12Run2Both,
""}, {"impulse t2-t3:", impulse23Run2Both, ""}, {"impulse t3-t4:",
 impulse34Run2Both, ""}, {"impulse t4-t5:", impulse45Run2Both,
""}, {"impulse t2-tmax:", impulse2maxRun2Both,
""}, {"impulse t1-thalf:", impulse1halfRun2Both,
""}, {"max force t1-t2:", maxforce12Run2Both,
""}, {"max force t2-t3:", maxforce23Run2Both,
""}, {"max force t3-t4:", maxforce34Run2Both,
""}, {"max force t4-t5:", maxforce45Run2Both,
""}, {"max force t2-tmax:", maxforce2maxRun2Both,
""}, {"max force t1-thalf:", maxforce1halfRun2Both,
""}, {"min force t1-t2:", minforce12Run2Both,
""}, {"min force t2-t3:", minforce23Run2Both,
""}, {"min force t3-t4:", minforce34Run2Both,
""}, {"min force t4-t5", minforce45Run2Both,
""}, {"min force t2-tmax:", minforce2maxRun2Both,
""}, {"min force t1-thalf:", minforce1halfRun2Both,
""}, {"max veloc. t1-t2:", maxvelocity12Run2Both,
""}, {"max veloc. t2-t3:", maxvelocity23Run2Both,
""}, {"max veloc. t3-t4:", maxvelocity34Run2Both,
""}, {"max veloc. t4-t5:", maxvelocity45Run2Both,
""}, {"max veloc. t2-tmax:", maxvelocity2maxRun2Both,
""}, {"max veloc. t1-thalf:", maxvelocity1halfRun2Both,
""}, {"min veloc. t1-t2:", minvelocity12Run2Both,
""}, {"min veloc. t2-t3:", minvelocity23Run2Both,
""}, {"min veloc. t3-t4:", minvelocity34Run2Both,
""}, {"min veloc. t4-t5:", minvelocity45Run2Both,
""}, {"min veloc. t2-tmax:", minvelocity2maxRun2Both,
""}, {"min veloc. t1-thalf:", minvelocity1halfRun2Both,
""}, {"max displace. t1-t2:", maxdisplacement12Run2Both,
""}, {"max displace. t2-t3:", maxdisplacement23Run2Both,
""}, {"max displace. t3-t4:", maxdisplacement34Run2Both,
""}, {"max displace. t4-t5:", maxdisplacement45Run2Both,
""}, {"max displace. t2-tmax:", maxdisplacement2maxRun2Both,
""}, {"max displace. t1-thalf:", maxdisplacement1halfRun2Both,
""}, {"min displace. t1-t2:", mindisplacement12Run2Both,
""}, {"min displace. t2-t3:", mindisplacement23Run2Both,
""}, {"min displace. t3-t4:", mindisplacement34Run2Both,
""}, {"min displace. t4-t5:", mindisplacement45Run2Both,
""}, {"min displace. t2-tmax:", mindisplacement2maxRun2Both,
""}, {"min displace. t1-thalf:", mindisplacement1halfRun2Both,
""}, {"max power t1-t2:", maxpower12Run2Both,
""}, {"max power t2-t3:", maxpower23Run2Both,
""}, {"max power t3-t4:", maxpower34Run2Both,
""}, {"max power t4-t5:", maxpower45Run2Both,
""}, {"max power t2-tmax:", maxpower2maxRun2Both,
""}, {"max power t1-thalf:", maxpower1halfRun2Both,
""}, {"min power t1-t2:", minpower12Run2Both,
""}, {"min power t2-t3:", minpower23Run2Both,
""}, {"min power t3-t4:", minpower34Run2Both,
""}, {"min power t4-t5:", minpower45Run2Both,
""}, {"min power t2-tmax:", minpower2maxRun2Both,
```

```
   ""}, {"min power t1-thalf:", minpower1halfRun2Both,
   ""}, {"avg. force in air:", noforceRun2Both,
   ""}, {"avg. force in squat:", squatavgRun2Both,
   ""}, {"st. dev. force in squat:", squatstdevRun2Both, ""}};

ForceGraphRun2Both =
 Plot[Tooltip[funcRun2Both[x]], {x, cutoffRun2Both, lastTimeRun2Both},
   ImageSize -> 600, PlotStyle -> {Automatic},
  PlotLegends -> {"Legs Combined"},
  Epilog -> {Directive[Blue, Dashed], line1Run2Both, line2Run2Both,
   line3Run2Both, line4Run2Both, line5Run2Both,
   Directive[Blue, Dashing[None]], linemaxRun2Both,
   linehalfRun2Both}, PlotRange -> All,
  PlotLabel -> "Non-Counter Movement Jump - Normal Force",
  AxesLabel -> {"Time (s)", "Normal Force (N)"} ]
ForceGraphRun2 =
 Plot[Tooltip[{funcRun2L[x], funcRun2R[x]}], {x, cutoffRun2Both,
   lastTimeRun2Both}, ImageSize -> 600,
  PlotLegends -> (Placed[#, Right] & /@ {{"Left", "Right"},
     LineLegend[{{Red, Dashed}, Dashed, Black}, {"Left", "Right"}]}),
   PlotStyle -> {Automatic},
  Epilog -> {Directive[Red, Dashed], line1Run2L, line2Run2L,
   line3Run2L, line4Run2L, line5Run2L, Directive[Red, Dashing[None]],
    linemaxRun2L, linehalfRun2L, Directive[Black, Dashed],
   line1Run2R, line2Run2R, line3Run2R, line4Run2R, line5Run2R,
   Directive[Black, Dashing[None]], linemaxRun2R, linehalfRun2R},
  PlotRange -> All,
  PlotLabel -> "Non-Counter Movement Jump - Normal Force",
  AxesLabel -> {"Time (s)", "Normal Force (N)"} ]
VelocityGraphRun2Both =
 Plot[Tooltip[velocityRun2Both], {x, cutoffRun2Both,
   lastTimeRun2Both}, ImageSize -> 600, PlotStyle -> {Automatic},
  PlotLegends -> {"Legs Combined"},
  Epilog -> {Directive[Blue, Dashed], line1Run2Both, line2Run2Both,
   line3Run2Both, line4Run2Both, line5Run2Both,
   Directive[Blue, Dashing[None]], linemaxRun2Both,
   linehalfRun2Both}, PlotRange -> All,
  PlotLabel -> "Non-Counter Movement Jump - Velocity",
  AxesLabel -> {"Time (s)", "Velocity (m/s)"} ]
VelocityGraphRun2 =
  Plot[Tooltip[{velocityRun2L, velocityRun2R}], {x, cutoffRun2Both,
    lastTimeRun2Both}, ImageSize -> 600,
   PlotLegends -> (Placed[#, Right] & /@ {{"Left", "Right"},
      LineLegend[{{Red, Dashed}, Dashed, Black}, {"Left",
        "Right"}]}), PlotStyle -> {Automatic}, PlotRange -> All,
   Epilog -> {Directive[Red, Dashed], line1Run2L, line2Run2L,
    line3Run2L, line4Run2L, line5Run2L,
    Directive[Red, Dashing[None]], linemaxRun2L, linehalfRun2L,
    Directive[Black, Dashed], line1Run2R, line2Run2R, line3Run2R,
    line4Run2R, line5Run2R, Directive[Black, Dashing[None]],
    linemaxRun2R, linehalfRun2R},
   PlotLabel -> "Non-Counter Movement Jump - Velocity",
   AxesLabel -> {"Time (s)", "Velocity (m/s)"} ];
DisplacementGraphRun2Both =
  Plot[Tooltip[displacementRun2Both], {x, cutoffRun2Both,
    lastTimeRun2Both}, ImageSize -> 600, PlotStyle -> {Automatic},
   PlotLegends -> {"Legs Combined"},
   Epilog -> {Directive[Blue, Dashed], line1Run2Both, line2Run2Both,
    line3Run2Both, line4Run2Both, line5Run2Both,
    Directive[Blue, Dashing[None]], linemaxRun2Both,
    linehalfRun2Both}, PlotRange -> All,
   PlotLabel -> "Non-Counter Movement Jump - Displacement",
   AxesLabel -> {"Time (s)", "Displacement (m)"} ];
DisplacementGraphRun2 =
  Plot[Tooltip[{displacementRun2L, displacementRun2R}], {x,
    cutoffRun2Both, lastTimeRun2Both}, ImageSize -> 600,
   PlotLegends -> (Placed[#, Right] & /@ {{"Left", "Right"},
```

```
      LineLegend[{{Red, Dashed}, Dashed, Black}, {"Left",
        "Right"}]}), PlotStyle -> {Automatic}, PlotRange -> All,
   Epilog -> {Directive[Red, Dashed], line1Run2L, line2Run2L,
     line3Run2L, line4Run2L, line5Run2L,
     Directive[Red, Dashing[None]], linemaxRun2L, linehalfRun2L,
     Directive[Black, Dashed], line1Run2R, line2Run2R, line3Run2R,
     line4Run2R, line5Run2R, Directive[Black, Dashing[None]],
     linemaxRun2R, linehalfRun2R},
   PlotLabel -> "Non-Counter Movement Jump - Displacement",
   AxesLabel -> {"Time (s)", "Displacement (m)"} ];
PowerGraphRun2Both =
  Plot[Tooltip[powerRun2Both], {x, cutoffRun2Both, lastTimeRun2Both},
   ImageSize -> 600, PlotStyle -> {Automatic},
   Epilog -> {Directive[Blue, Dashed],
     PlotLegends -> {"Legs Combined"}, line1Run2Both, line2Run2Both,
     line3Run2Both, line4Run2Both, line5Run2Both,
     Directive[Blue, Dashing[None]], linemaxRun2Both,
     linehalfRun2Both}, PlotRange -> All,
   PlotLabel -> "Non-Counter Movement Jump - Power",
   AxesLabel -> {"Time (s)", "Power (W)"} ];
PowerGraphRun2 =
  Plot[Tooltip[{powerRun2L, powerRun2R}], {x, cutoffRun2Both,
    lastTimeRun2Both}, ImageSize -> 600,
   PlotLegends -> (Placed[#, Right] & /@ {{"Left", "Right"},
      LineLegend[{{Red, Dashed}, Dashed, Black}, {"Left",
        "Right"}]}), PlotStyle -> {Automatic}, PlotRange -> All,
   Epilog -> {Directive[Red, Dashed], line1Run2L, line2Run2L,
     line3Run2L, line4Run2L, line5Run2L,
     Directive[Red, Dashing[None]], linemaxRun2L, linehalfRun2L,
     Directive[Black, Dashed], line1Run2R, line2Run2R, line3Run2R,
     line4Run2R, line5Run2R, Directive[Black, Dashing[None]],
     linemaxRun2R, linehalfRun2R},
   PlotLabel -> "Non-Counter Movement Jump - Power",
   AxesLabel -> {"Time (s)", "Power (W)"} ];

(* Run3 *)
solutionRun3 = {{IDRun3, "", ""}, {"", "",
    ""}, {"Variable", "Left", "Right"}, {"", "", ""},
   {"run class.:", runtypeRun3L, runtypeRun3R}, {"mass:", massRun3L,
    massRun3R}, {"t1:", t1Run3L, t1Run3R}, {"t2:", t2Run3L,
    t2Run3R}, {"t3:", t3Run3L, t3Run3R}, {"t4:", t4Run3L,
    t4Run3R}, {"t5:", t5Run3L, t5Run3R}, {"tmax:", tmaxRun3L,
    tmaxRun3R}, {"thalf:", thalfRun3L, thalfRun3R}, {"", "",
    ""}, {"force @ t1:", force1Run3L, force1Run3R}, {"force @ t2:",
    force2Run3L, force2Run3R}, {"force @ t3:", force3Run3L,
    force3Run3R}, {"force @ t4:", force4Run3L,
    force4Run3R}, {"force @ t5:", force5Run3L,
    force5Run3R}, {"force @ tmax:", forcemaxRun3L,
    forcemaxRun3R}, {"force @ thalf:", forcehalfRun3L,
    forcehalfRun3R}, {"veloc. @ t1:", velocity1Run3L,
    velocity1Run3R}, {"veloc. @ t2:", velocity2Run3L,
    velocity2Run3R}, {"veloc. @ t3:", velocity3Run3L,
    velocity3Run3R}, {"veloc. @ t4:", velocity4Run3L,
    velocity4Run3R}, {"veloc. @ t5:", velocity5Run3L,
    velocity5Run3R}, {"veloc. @ tmax:", velocitymaxRun3L,
    velocitymaxRun3R}, {"veloc. @ thalf:", velocityhalfRun3L,
    velocityhalfRun3R}, {"displace. @ t1:", displacement1Run3L,
    displacement1Run3R}, {"displace. @ t2:", displacement2Run3L,
    displacement2Run3R}, {"displace. @ t3:", displacement3Run3L,
    displacement3Run3R}, {"displace. @ t4:", displacement4Run3L,
    displacement4Run3R}, {"displace. @ t5:", displacement5Run3L,
    displacement5Run3R}, {"displace. @ tmax:", displacementmaxRun3L,
    displacementmaxRun3R}, {"displace. @ thalf:",
    displacementhalfRun3L, displacementhalfRun3R}, {"power @ t1:",
    power1Run3L, power1Run3R}, {"power @ t2:", power2Run3L,
    power2Run3R}, {"power @ t3:", power3Run3L,
    power3Run3R}, {"power @ t4:", power4Run3L,
```

```
power4Run3R}, {"power @ t5:", power5Run3L,
power5Run3R}, {"power @ tmax:", powermaxRun3L,
powermaxRun3R}, {"power @ thalf:", powerhalfRun3L,
powerhalfRun3R}, {"", "", ""}, {"impulse t1-t2:", impulse12Run3L,
impulse12Run3R}, {"impulse t2-t3:", impulse23Run3L,
impulse23Run3R}, {"impulse t3-t4:", impulse34Run3L,
impulse34Run3R}, {"impulse t4-t5:", impulse45Run3L,
impulse45Run3R}, {"impulse t2-tmax:", impulse2maxRun3L,
impulse2maxRun3R}, {"impulse t1-thalf:", impulse1halfRun3L,
impulse1halfRun3R}, {"max force t1-t2:", maxforce12Run3L,
maxforce12Run3R}, {"max force t2-t3:", maxforce23Run3L,
maxforce23Run3R}, {"max force t3-t4:", maxforce34Run3L,
maxforce34Run3R}, {"max force t4-t5:", maxforce45Run3L,
maxforce45Run3R}, {"max force t2-tmax:", maxforce2maxRun3L,
maxforce2maxRun3R}, {"max force t1-thalf:", maxforce1halfRun3L,
maxforce1halfRun3R}, {"min force t1-t2:", minforce12Run3L,
minforce12Run3R}, {"min force t2-t3:", minforce23Run3L,
minforce23Run3R}, {"min force t3-t4:", minforce34Run3L,
minforce34Run3R}, {"min force t4-t5:", minforce45Run3L,
minforce45Run3R}, {"min force t2-tmax:", minforce2maxRun3L,
minforce2maxRun3R}, {"min force t1-thalf:", minforce1halfRun3L,
minforce1halfRun3R}, {"max veloc. t1-t2:", maxvelocity12Run3L,
maxvelocity12Run3R}, {"max veloc. t2-t3:", maxvelocity23Run3L,
maxvelocity23Run3R}, {"max veloc. t3-t4:", maxvelocity34Run3L,
maxvelocity34Run3R}, {"max veloc. t4-t5:", maxvelocity45Run3L,
maxvelocity45Run3R}, {"max veloc. t2-tmax:", maxvelocity2maxRun3L,
 maxvelocity2maxRun3R}, {"max veloc. t1-thalf:",
maxvelocity1halfRun3L,
maxvelocity1halfRun3R}, {"min veloc. t1-t2:", minvelocity12Run3L,
minvelocity12Run3R}, {"min veloc. t2-t3:", minvelocity23Run3L,
minvelocity23Run3R}, {"min veloc. t3-t4:", minvelocity34Run3L,
minvelocity34Run3R}, {"min veloc. t4-t5:", minvelocity45Run3L,
minvelocity45Run3R}, {"min veloc. t2-tmax:", minvelocity2maxRun3L,
 minvelocity2maxRun3R}, {"min veloc. t1-thalf:",
minvelocity1halfRun3L,
minvelocity1halfRun3R}, {"max displace. t1-t2:",
maxdisplacement12Run3L,
maxdisplacement12Run3R}, {"max displace. t2-t3:",
maxdisplacement23Run3L,
maxdisplacement23Run3R}, {"max displace. t3-t4:",
maxdisplacement34Run3L,
maxdisplacement34Run3R}, {"max displace. t4-t5:",
maxdisplacement45Run3L,
maxdisplacement45Run3R}, {"max displace. t2-tmax:",
maxdisplacement2maxRun3L,
maxdisplacement2maxRun3R}, {"max displace. t1-thalf:",
maxdisplacement1halfRun3L,
maxdisplacement1halfRun3R}, {"min displace. t1-t2:",
mindisplacement12Run3L,
mindisplacement12Run3R}, {"min displace. t2-t3:",
mindisplacement23Run3L,
mindisplacement23Run3R}, {"min displace. t3-t4:",
mindisplacement34Run3L,
mindisplacement34Run3R}, {"min displace. t4-t5:",
mindisplacement45Run3L,
mindisplacement45Run3R}, {"min displace. t2-tmax:",
mindisplacement2maxRun3L,
mindisplacement2maxRun3R}, {"min displace. t1-thalf:",
mindisplacement1halfRun3L,
mindisplacement1halfRun3R}, {"max power t1-t2:", maxpower12Run3L,
maxpower12Run3R}, {"max power t2-t3:", maxpower23Run3L,
maxpower23Run3R}, {"max power t3-t4:", maxpower34Run3L,
maxpower34Run3R}, {"max power t4-t5:", maxpower45Run3L,
maxpower45Run3R}, {"max power t2-tmax:", maxpower2maxRun3L,
maxpower2maxRun3R}, {"max power t1-thalf:", maxpower1halfRun3L,
maxpower1halfRun3R}, {"min power t1-t2:", minpower12Run3L,
minpower12Run3R}, {"min power t2-t3:", minpower23Run3L,
```

```
minpower23Run3R}, {"min power t3-t4:", minpower34Run3L,
minpower34Run3R}, {"min power t4-t5:", minpower45Run3L,
minpower45Run3R}, {"min power t2-tmax:", minpower2maxRun3L,
minpower2maxRun3R}, {"min power t1-thalf:", minpower1halfRun3L,
minpower1halfRun3R}, {"avg. force in air:", noforceRun3L,
noforceRun3R}, {"avg. force in squat:", squatavgRun3L,
squatavgRun3R}, {"st. dev. force in squat:", squatstdevRun3L,
squatstdevRun3R}, {"", "", ""}, {"Legs Analyzed Together:", "",
""}, {"", "", ""}, {"Variable", "Legs Combined", ""}, {"", "",
""}, {"run class.:", runtypeRun3Both, ""}, {"mass:", massRun3Both,
 ""}, {"t1:", t1Run3Both, ""}, {"t2:", t2Run3Both, ""}, {"t3:",
t3Run3Both, ""}, {"t4:", t4Run3Both, ""}, {"t5:", t5Run3Both,
""}, {"tmax:", tmaxRun3Both, ""}, {"", "", ""}, {"force @ t1:",
force1Run3Both, ""}, {"force @ t2:", force2Run3Both,
""}, {"force @ t3:", force3Run3Both, ""}, {"force @ t4:",
force4Run3Both, ""}, {"force @ t5:", force5Run3Both,
""}, {"force @ tmax:", forcemaxRun3Both, ""}, {"force @ thalf:",
forcehalfRun3Both, ""}, {"veloc. @ t1:", velocity1Run3Both,
""}, {"veloc. @ t2:", velocity2Run3Both, ""}, {"veloc. @ t3:",
velocity3Run3Both, ""}, {"veloc. @ t4:", velocity4Run3Both,
""}, {"veloc. @ t5:", velocity5Run3Both, ""}, {"veloc. @ tmax:",
velocitymaxRun3Both, ""}, {"veloc. @ thalf:",
velocityhalfRun3Both, ""}, {"displace. @ t1:",
displacement1Run3Both, ""}, {"displace. @ t2:",
displacement2Run3Both, ""}, {"displace. @ t3:",
displacement3Run3Both, ""}, {"displace. @ t4:",
displacement4Run3Both, ""}, {"displace. @ t5:",
displacement5Run3Both, ""}, {"displace. @ tmax:",
displacementmaxRun3Both, ""}, {"displace. @ thalf:",
displacementhalfRun3Both, ""}, {"power @ t1:", power1Run3Both,
""}, {"power @ t2:", power2Run3Both, ""}, {"power @ t3:",
power3Run3Both, ""}, {"power @ t4:", power4Run3Both,
""}, {"power @ t5:", power5Run3Both, ""}, {"power @ tmax:",
powermaxRun3Both, ""}, {"power @ thalf:", powerhalfRun3Both,
""}, {"", "", ""}, {"impulse t1-t2:", impulse12Run3Both,
""}, {"impulse t2-t3:", impulse23Run3Both, ""}, {"impulse t3-t4:",
 impulse34Run3Both, ""}, {"impulse t4-t5:", impulse45Run3Both,
""}, {"impulse t2-tmax:", impulse2maxRun3Both,
""}, {"impulse t1-thalf:", impulse1halfRun3Both,
""}, {"max force t1-t2:", maxforce12Run3Both,
""}, {"max force t2-t3:", maxforce23Run3Both,
""}, {"max force t3-t4:", maxforce34Run3Both,
""}, {"max force t4-t5:", maxforce45Run3Both,
""}, {"max force t2-tmax:", maxforce2maxRun3Both,
""}, {"max force t1-thalf:", maxforce1halfRun3Both,
""}, {"min force t1-t2:", minforce12Run3Both,
""}, {"min force t2-t3:", minforce23Run3Both,
""}, {"min force t3-t4:", minforce34Run3Both,
""}, {"min force t4-t5", minforce45Run3Both,
""}, {"min force t2-tmax:", minforce2maxRun3Both,
""}, {"min force t1-thalf:", minforce1halfRun3Both,
""}, {"max veloc. t1-t2:", maxvelocity12Run3Both,
""}, {"max veloc. t2-t3:", maxvelocity23Run3Both,
""}, {"max veloc. t3-t4:", maxvelocity34Run3Both,
""}, {"max veloc. t4-t5:", maxvelocity45Run3Both,
""}, {"max veloc. t2-tmax:", maxvelocity2maxRun3Both,
""}, {"max veloc. t1-thalf:", maxvelocity1halfRun3Both,
""}, {"min veloc. t1-t2:", minvelocity12Run3Both,
""}, {"min veloc. t2-t3:", minvelocity23Run3Both,
""}, {"min veloc. t3-t4:", minvelocity34Run3Both,
""}, {"min veloc. t4-t5:", minvelocity45Run3Both,
""}, {"min veloc. t2-tmax:", minvelocity2maxRun3Both,
""}, {"min veloc. t1-thalf:", minvelocity1halfRun3Both,
""}, {"max displace. t1-t2:", maxdisplacement12Run3Both,
""}, {"max displace. t2-t3:", maxdisplacement23Run3Both,
""}, {"max displace. t3-t4:", maxdisplacement34Run3Both,
""}, {"max displace. t4-t5:", maxdisplacement45Run3Both,
```

```
  ""}, {"max displace. t2-tmax:", maxdisplacement2maxRun3Both,
  ""}, {"max displace. t1-thalf:", maxdisplacement1halfRun3Both,
  ""}, {"min displace. t1-t2:", mindisplacement12Run3Both,
  ""}, {"min displace. t2-t3:", mindisplacement23Run3Both,
  ""}, {"min displace. t3-t4:", mindisplacement34Run3Both,
  ""}, {"min displace. t4-t5:", mindisplacement45Run3Both,
  ""}, {"min displace. t2-tmax:", mindisplacement2maxRun3Both,
  ""}, {"min displace. t1-thalf:", mindisplacement1halfRun3Both,
  ""}, {"max power t1-t2:", maxpower12Run3Both,
  ""}, {"max power t2-t3:", maxpower23Run3Both,
  ""}, {"max power t3-t4:", maxpower34Run3Both,
  ""}, {"max power t4-t5:", maxpower45Run3Both,
  ""}, {"max power t2-tmax:", maxpower2maxRun3Both,
  ""}, {"max power t1-thalf:", maxpower1halfRun3Both,
  ""}, {"min power t1-t2:", minpower12Run3Both,
  ""}, {"min power t2-t3:", minpower23Run3Both,
  ""}, {"min power t3-t4:", minpower34Run3Both,
  ""}, {"min power t4-t5:", minpower45Run3Both,
  ""}, {"min power t2-tmax:", minpower2maxRun3Both,
  ""}, {"min power t1-thalf:", minpower1halfRun3Both,
  ""}, {"avg. force in air:", noforceRun3Both,
  ""}, {"avg. force in squat:", squatavgRun3Both,
  ""}, {"st. dev. force in squat:", squatstdevRun3Both, ""}};

ForceGraphRun3Both =
 Plot[Tooltip[funcRun3Both[x]], {x, cutoffRun3Both, lastTimeRun3Both},
   ImageSize -> 600, PlotStyle -> {Automatic},
  PlotLegends -> {"Legs Combined"},
  Epilog -> {Directive[Blue, Dashed], line1Run3Both, line2Run3Both,
    line3Run3Both, line4Run3Both, line5Run3Both,
    Directive[Blue, Dashing[None]], linemaxRun3Both,
    linehalfRun3Both}, PlotRange -> All,
  PlotLabel -> "Non-Counter Movement Jump - Normal Force",
  AxesLabel -> {"Time (s)", "Normal Force (N)"} ]
ForceGraphRun3 =
 Plot[Tooltip[{funcRun3L[x], funcRun3R[x]}], {x, cutoffRun3Both,
   lastTimeRun3Both}, ImageSize -> 600,
  PlotLegends -> (Placed[#, Right] & /@ {{"Left", "Right"},
      LineLegend[{{Red, Dashed}, Dashed, Black}, {"Left", "Right"}]}),
   PlotStyle -> {Automatic},
  Epilog -> {Directive[Red, Dashed], line1Run3L, line2Run3L,
    line3Run3L, line4Run3L, line5Run3L, Directive[Red, Dashing[None]],
     linemaxRun3L, linehalfRun3L, Directive[Black, Dashed],
    line1Run3R, line2Run3R, line3Run3R, line4Run3R, line5Run3R,
    Directive[Black, Dashing[None]], linemaxRun3R, linehalfRun3R},
  PlotRange -> All,
  PlotLabel -> "Non-Counter Movement Jump - Normal Force",
  AxesLabel -> {"Time (s)", "Normal Force (N)"} ]
VelocityGraphRun3Both =
 Plot[Tooltip[velocityRun3Both], {x, cutoffRun3Both,
   lastTimeRun3Both}, ImageSize -> 600, PlotStyle -> {Automatic},
  PlotLegends -> {"Legs Combined"},
  Epilog -> {Directive[Blue, Dashed], line1Run3Both, line2Run3Both,
    line3Run3Both, line4Run3Both, line5Run3Both,
    Directive[Blue, Dashing[None]], linemaxRun3Both,
    linehalfRun3Both}, PlotRange -> All,
  PlotLabel -> "Non-Counter Movement Jump - Velocity",
  AxesLabel -> {"Time (s)", "Velocity (m/s)"} ]
VelocityGraphRun3 =
  Plot[Tooltip[{velocityRun3L, velocityRun3R}], {x, cutoffRun3Both,
    lastTimeRun3Both}, ImageSize -> 600,
   PlotLegends -> (Placed[#, Right] & /@ {{"Left", "Right"},
       LineLegend[{{Red, Dashed}, Dashed, Black}, {"Left",
         "Right"}]}), PlotStyle -> {Automatic}, PlotRange -> All,
   Epilog -> {Directive[Red, Dashed], line1Run3L, line2Run3L,
     line3Run3L, line4Run3L, line5Run3L,
     Directive[Red, Dashing[None]], linemaxRun3L, linehalfRun3L,
```

```
     Directive[Black, Dashed], line1Run3R, line2Run3R, line3Run3R,
     line4Run3R, line5Run3R, Directive[Black, Dashing[None]],
     linemaxRun3R, linehalfRun3R},
   PlotLabel -> "Non-Counter Movement Jump - Velocity",
   AxesLabel -> {"Time (s)", "Velocity (m/s)"} ];
DisplacementGraphRun3Both =
  Plot[Tooltip[displacementRun3Both], {x, cutoffRun3Both,
    lastTimeRun3Both}, ImageSize -> 600, PlotStyle -> {Automatic},
   PlotLegends -> {"Legs Combined"},
   Epilog -> {Directive[Blue, Dashed], line1Run3Both, line2Run3Both,
     line3Run3Both, line4Run3Both, line5Run3Both,
     Directive[Blue, Dashing[None]], linemaxRun3Both,
     linehalfRun3Both}, PlotRange -> All,
   PlotLabel -> "Non-Counter Movement Jump - Displacement",
   AxesLabel -> {"Time (s)", "Displacement (m)"} ];
DisplacementGraphRun3 =
  Plot[Tooltip[{displacementRun3L, displacementRun3R}], {x,
    cutoffRun3Both, lastTimeRun3Both}, ImageSize -> 600,
   PlotLegends -> (Placed[#, Right] & /@ {{"Left", "Right"},
       LineLegend[{{Red, Dashed}, Dashed, Black}, {"Left",
         "Right"}]}), PlotStyle -> {Automatic}, PlotRange -> All,
   Epilog -> {Directive[Red, Dashed], line1Run3L, line2Run3L,
     line3Run3L, line4Run3L, line5Run3L,
     Directive[Red, Dashing[None]], linemaxRun3L, linehalfRun3L,
     Directive[Black, Dashed], line1Run3R, line2Run3R, line3Run3R,
     line4Run3R, line5Run3R, Directive[Black, Dashing[None]],
     linemaxRun3R, linehalfRun3R},
   PlotLabel -> "Non-Counter Movement Jump - Displacement",
   AxesLabel -> {"Time (s)", "Displacement (m)"} ];
PowerGraphRun3Both =
  Plot[Tooltip[powerRun3Both], {x, cutoffRun3Both, lastTimeRun3Both},
   ImageSize -> 600, PlotStyle -> {Automatic},
   Epilog -> {Directive[Blue, Dashed],
     PlotLegends -> {"Legs Combined"}, line1Run3Both, line2Run3Both,
     line3Run3Both, line4Run3Both, line5Run3Both,
     Directive[Blue, Dashing[None]], linemaxRun3Both,
     linehalfRun3Both}, PlotRange -> All,
   PlotLabel -> "Non-Counter Movement Jump - Power",
   AxesLabel -> {"Time (s)", "Power (W)"} ];
PowerGraphRun3 =
  Plot[Tooltip[{powerRun3L, powerRun3R}], {x, cutoffRun3Both,
    lastTimeRun3Both}, ImageSize -> 600,
   PlotLegends -> (Placed[#, Right] & /@ {{"Left", "Right"},
       LineLegend[{{Red, Dashed}, Dashed, Black}, {"Left",
         "Right"}]}), PlotStyle -> {Automatic}, PlotRange -> All,
   Epilog -> {Directive[Red, Dashed], line1Run3L, line2Run3L,
     line3Run3L, line4Run3L, line5Run3L,
     Directive[Red, Dashing[None]], linemaxRun3L, linehalfRun3L,
     Directive[Black, Dashed], line1Run3R, line2Run3R, line3Run3R,
     line4Run3R, line5Run3R, Directive[Black, Dashing[None]],
     linemaxRun3R, linehalfRun3R},
   PlotLabel -> "Non-Counter Movement Jump - Power",
   AxesLabel -> {"Time (s)", "Power (W)"} ];

(* Run4 *)
solutionRun4 = {{IDRun4, "", ""}, {"", "",
    ""}, {"Variable", "Left", "Right"}, {"", "", ""},
   {"run class.:", runtypeRun4L, runtypeRun4R}, {"mass:", massRun4L,
    massRun4R}, {"t1:", t1Run4L, t1Run4R}, {"t2:", t2Run4L,
    t2Run4R}, {"t3:", t3Run4L, t3Run4R}, {"t4:", t4Run4L,
    t4Run4R}, {"t5:", t5Run4L, t5Run4R}, {"tmax:", tmaxRun4L,
    tmaxRun4R}, {"thalf:", thalfRun4L, thalfRun4R}, {"", "",
    ""}, {"force @ t1:", force1Run4L, force1Run4R}, {"force @ t2:",
    force2Run4L, force2Run4R}, {"force @ t3:", force3Run4L,
    force3Run4R}, {"force @ t4:", force4Run4L,
    force4Run4R}, {"force @ t5:", force5Run4L,
    force5Run4R}, {"force @ tmax:", forcemaxRun4L,
```

```
forcemaxRun4R}, {"force @ thalf:", forcehalfRun4L,
forcehalfRun4R}, {"veloc. @ t1:", velocity1Run4L,
velocity1Run4R}, {"veloc. @ t2:", velocity2Run4L,
velocity2Run4R}, {"veloc. @ t3:", velocity3Run4L,
velocity3Run4R}, {"veloc. @ t4:", velocity4Run4L,
velocity4Run4R}, {"veloc. @ t5:", velocity5Run4L,
velocity5Run4R}, {"veloc. @ tmax:", velocitymaxRun4L,
velocitymaxRun4R}, {"veloc. @ thalf:", velocityhalfRun4L,
velocityhalfRun4R}, {"displace. @ t1:", displacement1Run4L,
displacement1Run4R}, {"displace. @ t2:", displacement2Run4L,
displacement2Run4R}, {"displace. @ t3:", displacement3Run4L,
displacement3Run4R}, {"displace. @ t4:", displacement4Run4L,
displacement4Run4R}, {"displace. @ t5:", displacement5Run4L,
displacement5Run4R}, {"displace. @ tmax:", displacementmaxRun4L,
displacementmaxRun4R}, {"displace. @ thalf:",
displacementhalfRun4L, displacementhalfRun4R}, {"power @ t1:",
power1Run4L, power1Run4R}, {"power @ t2:", power2Run4L,
power2Run4R}, {"power @ t3:", power3Run4L,
power3Run4R}, {"power @ t4:", power4Run4L,
power4Run4R}, {"power @ t5:", power5Run4L,
power5Run4R}, {"power @ tmax:", powermaxRun4L,
powermaxRun4R}, {"power @ thalf:", powerhalfRun4L,
powerhalfRun4R}, {"", "", ""}, {"impulse t1-t2:", impulse12Run4L,
impulse12Run4R}, {"impulse t2-t3:", impulse23Run4L,
impulse23Run4R}, {"impulse t3-t4:", impulse34Run4L,
impulse34Run4R}, {"impulse t4-t5:", impulse45Run4L,
impulse45Run4R}, {"impulse t2-tmax:", impulse2maxRun4L,
impulse2maxRun4R}, {"impulse t1-thalf:", impulse1halfRun4L,
impulse1halfRun4R}, {"max force t1-t2:", maxforce12Run4L,
maxforce12Run4R}, {"max force t2-t3:", maxforce23Run4L,
maxforce23Run4R}, {"max force t3-t4:", maxforce34Run4L,
maxforce34Run4R}, {"max force t4-t5:", maxforce45Run4L,
maxforce45Run4R}, {"max force t2-tmax:", maxforce2maxRun4L,
maxforce2maxRun4R}, {"max force t1-thalf:", maxforce1halfRun4L,
maxforce1halfRun4R}, {"min force t1-t2:", minforce12Run4L,
minforce12Run4R}, {"min force t2-t3:", minforce23Run4L,
minforce23Run4R}, {"min force t3-t4:", minforce34Run4L,
minforce34Run4R}, {"min force t4-t5:", minforce45Run4L,
minforce45Run4R}, {"min force t2-tmax:", minforce2maxRun4L,
minforce2maxRun4R}, {"min force t1-thalf:", minforce1halfRun4L,
minforce1halfRun4R}, {"max veloc. t1-t2:", maxvelocity12Run4L,
maxvelocity12Run4R}, {"max veloc. t2-t3:", maxvelocity23Run4L,
maxvelocity23Run4R}, {"max veloc. t3-t4:", maxvelocity34Run4L,
maxvelocity34Run4R}, {"max veloc. t4-t5:", maxvelocity45Run4L,
maxvelocity45Run4R}, {"max veloc. t2-tmax:", maxvelocity2maxRun4L,
 maxvelocity2maxRun4R}, {"max veloc. t1-thalf:",
maxvelocity1halfRun4L,
maxvelocity1halfRun4R}, {"min veloc. t1-t2:", minvelocity12Run4L,
minvelocity12Run4R}, {"min veloc. t2-t3:", minvelocity23Run4L,
minvelocity23Run4R}, {"min veloc. t3-t4:", minvelocity34Run4L,
minvelocity34Run4R}, {"min veloc. t4-t5:", minvelocity45Run4L,
minvelocity45Run4R}, {"min veloc. t2-tmax:", minvelocity2maxRun4L,
 minvelocity2maxRun4R}, {"min veloc. t1-thalf:",
minvelocity1halfRun4L,
minvelocity1halfRun4R}, {"max displace. t1-t2:",
maxdisplacement12Run4L,
maxdisplacement12Run4R}, {"max displace. t2-t3:",
maxdisplacement23Run4L,
maxdisplacement23Run4R}, {"max displace. t3-t4:",
maxdisplacement34Run4L,
maxdisplacement34Run4R}, {"max displace. t4-t5:",
maxdisplacement45Run4L,
maxdisplacement45Run4R}, {"max displace. t2-tmax:",
maxdisplacement2maxRun4L,
maxdisplacement2maxRun4R}, {"max displace. t1-thalf:",
maxdisplacement1halfRun4L,
maxdisplacement1halfRun4R}, {"min displace. t1-t2:",
```

```
mindisplacement12Run4L,
mindisplacement12Run4R}, {"min displace. t2-t3:",
mindisplacement23Run4L,
mindisplacement23Run4R}, {"min displace. t3-t4:",
mindisplacement34Run4L,
mindisplacement34Run4R}, {"min displace. t4-t5:",
mindisplacement45Run4L,
mindisplacement45Run4R}, {"min displace. t2-tmax:",
mindisplacement2maxRun4L,
mindisplacement2maxRun4R}, {"min displace. t1-thalf:",
mindisplacement1halfRun4L,
mindisplacement1halfRun4R}, {"max power t1-t2:", maxpower12Run4L,
maxpower12Run4R}, {"max power t2-t3:", maxpower23Run4L,
maxpower23Run4R}, {"max power t3-t4:", maxpower34Run4L,
maxpower34Run4R}, {"max power t4-t5:", maxpower45Run4L,
maxpower45Run4R}, {"max power t2-tmax:", maxpower2maxRun4L,
maxpower2maxRun4R}, {"max power t1-thalf:", maxpower1halfRun4L,
maxpower1halfRun4R}, {"min power t1-t2:", minpower12Run4L,
minpower12Run4R}, {"min power t2-t3:", minpower23Run4L,
minpower23Run4R}, {"min power t3-t4:", minpower34Run4L,
minpower34Run4R}, {"min power t4-t5:", minpower45Run4L,
minpower45Run4R}, {"min power t2-tmax:", minpower2maxRun4L,
minpower2maxRun4R}, {"min power t1-thalf:", minpower1halfRun4L,
minpower1halfRun4R}, {"avg. force in air:", noforceRun4L,
noforceRun4R}, {"avg. force in squat:", squatavgRun4L,
squatavgRun4R}, {"st. dev. force in squat:", squatstdevRun4L,
squatstdevRun4R}, {"", "", ""}, {"Legs Analyzed Together:", "",
""}, {"", "", ""}, {"Variable", "Legs Combined", ""}, {"", "",
""}, {"run class.:", runtypeRun4Both, ""}, {"mass:", massRun4Both,
 ""}, {"t1:", t1Run4Both, ""}, {"t2:", t2Run4Both, ""}, {"t3:",
t3Run4Both, ""}, {"t4:", t4Run4Both, ""}, {"t5:", t5Run4Both,
""}, {"tmax:", tmaxRun4Both, ""}, {"", "", ""}, {"force @ t1:",
force1Run4Both, ""}, {"force @ t2:", force2Run4Both,
""}, {"force @ t3:", force3Run4Both, ""}, {"force @ t4:",
force4Run4Both, ""}, {"force @ t5:", force5Run4Both,
""}, {"force @ tmax:", forcemaxRun4Both, ""}, {"force @ thalf:",
forcehalfRun4Both, ""}, {"veloc. @ t1:", velocity1Run4Both,
""}, {"veloc. @ t2:", velocity2Run4Both, ""}, {"veloc. @ t3:",
velocity3Run4Both, ""}, {"veloc. @ t4:", velocity4Run4Both,
""}, {"veloc. @ t5:", velocity5Run4Both, ""}, {"veloc. @ tmax:",
velocitymaxRun4Both, ""}, {"veloc. @ thalf:",
velocityhalfRun4Both, ""}, {"displace. @ t1:",
displacement1Run4Both, ""}, {"displace. @ t2:",
displacement2Run4Both, ""}, {"displace. @ t3:",
displacement3Run4Both, ""}, {"displace. @ t4:",
displacement4Run4Both, ""}, {"displace. @ t5:",
displacement5Run4Both, ""}, {"displace. @ tmax:",
displacementmaxRun4Both, ""}, {"displace. @ thalf:",
displacementhalfRun4Both, ""}, {"power @ t1:", power1Run4Both,
""}, {"power @ t2:", power2Run4Both, ""}, {"power @ t3:",
power3Run4Both, ""}, {"power @ t4:", power4Run4Both,
""}, {"power @ t5:", power5Run4Both, ""}, {"power @ tmax:",
powermaxRun4Both, ""}, {"power @ thalf:", powerhalfRun4Both,
""}, {"", "", ""}, {"impulse t1-t2:", impulse12Run4Both,
""}, {"impulse t2-t3:", impulse23Run4Both, ""}, {"impulse t3-t4:",
 impulse34Run4Both, ""}, {"impulse t4-t5:", impulse45Run4Both,
""}, {"impulse t2-tmax:", impulse2maxRun4Both,
""}, {"impulse t1-thalf:", impulse1halfRun4Both,
""}, {"max force t1-t2:", maxforce12Run4Both,
""}, {"max force t2-t3:", maxforce23Run4Both,
""}, {"max force t3-t4:", maxforce34Run4Both,
""}, {"max force t4-t5:", maxforce45Run4Both,
""}, {"max force t2-tmax:", maxforce2maxRun4Both,
""}, {"max force t1-thalf:", maxforce1halfRun4Both,
""}, {"min force t1-t2:", minforce12Run4Both,
""}, {"min force t2-t3:", minforce23Run4Both,
""}, {"min force t3-t4:", minforce34Run4Both,
```

```
    ""}, {"min force t4-t5", minforce45Run4Both,
    ""}, {"min force t2-tmax:", minforce2maxRun4Both,
    ""}, {"min force t1-thalf:", minforce1halfRun4Both,
    ""}, {"max veloc. t1-t2:", maxvelocity12Run4Both,
    ""}, {"max veloc. t2-t3:", maxvelocity23Run4Both,
    ""}, {"max veloc. t3-t4:", maxvelocity34Run4Both,
    ""}, {"max veloc. t4-t5:", maxvelocity45Run4Both,
    ""}, {"max veloc. t2-tmax:", maxvelocity2maxRun4Both,
    ""}, {"max veloc. t1-thalf:", maxvelocity1halfRun4Both,
    ""}, {"min veloc. t1-t2:", minvelocity12Run4Both,
    ""}, {"min veloc. t2-t3:", minvelocity23Run4Both,
    ""}, {"min veloc. t3-t4:", minvelocity34Run4Both,
    ""}, {"min veloc. t4-t5:", minvelocity45Run4Both,
    ""}, {"min veloc. t2-tmax:", minvelocity2maxRun4Both,
    ""}, {"min veloc. t1-thalf:", minvelocity1halfRun4Both,
    ""}, {"max displace. t1-t2:", maxdisplacement12Run4Both,
    ""}, {"max displace. t2-t3:", maxdisplacement23Run4Both,
    ""}, {"max displace. t3-t4:", maxdisplacement34Run4Both,
    ""}, {"max displace. t4-t5:", maxdisplacement45Run4Both,
    ""}, {"max displace. t2-tmax:", maxdisplacement2maxRun4Both,
    ""}, {"max displace. t1-thalf:", maxdisplacement1halfRun4Both,
    ""}, {"min displace. t1-t2:", mindisplacement12Run4Both,
    ""}, {"min displace. t2-t3:", mindisplacement23Run4Both,
    ""}, {"min displace. t3-t4:", mindisplacement34Run4Both,
    ""}, {"min displace. t4-t5:", mindisplacement45Run4Both,
    ""}, {"min displace. t2-tmax:", mindisplacement2maxRun4Both,
    ""}, {"min displace. t1-thalf:", mindisplacement1halfRun4Both,
    ""}, {"max power t1-t2:", maxpower12Run4Both,
    ""}, {"max power t2-t3:", maxpower23Run4Both,
    ""}, {"max power t3-t4:", maxpower34Run4Both,
    ""}, {"max power t4-t5:", maxpower45Run4Both,
    ""}, {"max power t2-tmax:", maxpower2maxRun4Both,
    ""}, {"max power t1-thalf:", maxpower1halfRun4Both,
    ""}, {"min power t1-t2:", minpower12Run4Both,
    ""}, {"min power t2-t3:", minpower23Run4Both,
    ""}, {"min power t3-t4:", minpower34Run4Both,
    ""}, {"min power t4-t5:", minpower45Run4Both,
    ""}, {"min power t2-tmax:", minpower2maxRun4Both,
    ""}, {"min power t1-thalf:", minpower1halfRun4Both,
    ""}, {"avg. force in air:", noforceRun4Both,
    ""}, {"avg. force in squat:", squatavgRun4Both,
    ""}, {"st. dev. force in squat:", squatstdevRun4Both, ""}};

ForceGraphRun4Both =
 Plot[Tooltip[funcRun4Both[x]], {x, cutoffRun4Both, lastTimeRun4Both},
   ImageSize -> 600, PlotStyle -> {Automatic},
  PlotLegends -> {"Legs Combined"},
  Epilog -> {Directive[Blue, Dashed], line1Run4Both, line2Run4Both,
    line3Run4Both, line4Run4Both, line5Run4Both,
    Directive[Blue, Dashing[None]], linemaxRun4Both,
    linehalfRun4Both}, PlotRange -> All,
  PlotLabel -> "Non-Counter Movement Jump - Normal Force",
  AxesLabel -> {"Time (s)", "Normal Force (N)"} ]
ForceGraphRun4 =
 Plot[Tooltip[{funcRun4L[x], funcRun4R[x]}], {x, cutoffRun4Both,
   lastTimeRun4Both}, ImageSize -> 600,
  PlotLegends -> (Placed[#, Right] & /@ {{"Left", "Right"},
      LineLegend[{{Red, Dashed}, Dashed, Black}, {"Left", "Right"}]}),
   PlotStyle -> {Automatic},
  Epilog -> {Directive[Red, Dashed], line1Run4L, line2Run4L,
    line3Run4L, line4Run4L, line5Run4L, Directive[Red, Dashing[None]],
     linemaxRun4L, linehalfRun4L, Directive[Black, Dashed],
    line1Run4R, line2Run4R, line3Run4R, line4Run4R, line5Run4R,
    Directive[Black, Dashing[None]], linemaxRun4R, linehalfRun4R},
  PlotRange -> All,
  PlotLabel -> "Non-Counter Movement Jump - Normal Force",
  AxesLabel -> {"Time (s)", "Normal Force (N)"} ]
```

```
VelocityGraphRun4Both =
 Plot[Tooltip[velocityRun4Both], {x, cutoffRun4Both,
   lastTimeRun4Both}, ImageSize -> 600, PlotStyle -> {Automatic},
  PlotLegends -> {"Legs Combined"},
  Epilog -> {Directive[Blue, Dashed], line1Run4Both, line2Run4Both,
    line3Run4Both, line4Run4Both, line5Run4Both,
    Directive[Blue, Dashing[None]], linemaxRun4Both,
    linehalfRun4Both}, PlotRange -> All,
  PlotLabel -> "Non-Counter Movement Jump - Velocity",
  AxesLabel -> {"Time (s)", "Velocity (m/s)"} ]
VelocityGraphRun4 =
  Plot[Tooltip[{velocityRun4L, velocityRun4R}], {x, cutoffRun4Both,
    lastTimeRun4Both}, ImageSize -> 600,
   PlotLegends -> (Placed[#, Right] & /@ {{"Left", "Right"},
      LineLegend[{{Red, Dashed}, Dashed, Black}, {"Left",
        "Right"}]}), PlotStyle -> {Automatic}, PlotRange -> All,
   Epilog -> {Directive[Red, Dashed], line1Run4L, line2Run4L,
     line3Run4L, line4Run4L, line5Run4L,
     Directive[Red, Dashing[None]], linemaxRun4L, linehalfRun4L,
     Directive[Black, Dashed], line1Run4R, line2Run4R, line3Run4R,
     line4Run4R, line5Run4R, Directive[Black, Dashing[None]],
     linemaxRun4R, linehalfRun4R},
   PlotLabel -> "Non-Counter Movement Jump - Velocity",
   AxesLabel -> {"Time (s)", "Velocity (m/s)"} ];
DisplacementGraphRun4Both =
  Plot[Tooltip[displacementRun4Both], {x, cutoffRun4Both,
    lastTimeRun4Both}, ImageSize -> 600, PlotStyle -> {Automatic},
   PlotLegends -> {"Legs Combined"},
   Epilog -> {Directive[Blue, Dashed], line1Run4Both, line2Run4Both,
     line3Run4Both, line4Run4Both, line5Run4Both,
     Directive[Blue, Dashing[None]], linemaxRun4Both,
     linehalfRun4Both}, PlotRange -> All,
   PlotLabel -> "Non-Counter Movement Jump - Displacement",
   AxesLabel -> {"Time (s)", "Displacement (m)"} ];
DisplacementGraphRun4 =
  Plot[Tooltip[{displacementRun4L, displacementRun4R}], {x,
    cutoffRun4Both, lastTimeRun4Both}, ImageSize -> 600,
   PlotLegends -> (Placed[#, Right] & /@ {{"Left", "Right"},
      LineLegend[{{Red, Dashed}, Dashed, Black}, {"Left",
        "Right"}]}), PlotStyle -> {Automatic}, PlotRange -> All,
   Epilog -> {Directive[Red, Dashed], line1Run4L, line2Run4L,
     line3Run4L, line4Run4L, line5Run4L,
     Directive[Red, Dashing[None]], linemaxRun4L, linehalfRun4L,
     Directive[Black, Dashed], line1Run4R, line2Run4R, line3Run4R,
     line4Run4R, line5Run4R, Directive[Black, Dashing[None]],
     linemaxRun4R, linehalfRun4R},
   PlotLabel -> "Non-Counter Movement Jump - Displacement",
   AxesLabel -> {"Time (s)", "Displacement (m)"} ];
PowerGraphRun4Both =
  Plot[Tooltip[powerRun4Both], {x, cutoffRun4Both, lastTimeRun4Both},
   ImageSize -> 600, PlotStyle -> {Automatic},
   Epilog -> {Directive[Blue, Dashed],
     PlotLegends -> {"Legs Combined"}, line1Run4Both, line2Run4Both,
     line3Run4Both, line4Run4Both, line5Run4Both,
     Directive[Blue, Dashing[None]], linemaxRun4Both,
     linehalfRun4Both}, PlotRange -> All,
   PlotLabel -> "Non-Counter Movement Jump - Power",
   AxesLabel -> {"Time (s)", "Power (W)"} ];
PowerGraphRun4 =
  Plot[Tooltip[{powerRun4L, powerRun4R}], {x, cutoffRun4Both,
    lastTimeRun4Both}, ImageSize -> 600,
   PlotLegends -> (Placed[#, Right] & /@ {{"Left", "Right"},
      LineLegend[{{Red, Dashed}, Dashed, Black}, {"Left",
        "Right"}]}), PlotStyle -> {Automatic}, PlotRange -> All,
   Epilog -> {Directive[Red, Dashed], line1Run4L, line2Run4L,
     line3Run4L, line4Run4L, line5Run4L,
     Directive[Red, Dashing[None]], linemaxRun4L, linehalfRun4L,
```

```
     Directive[Black, Dashed], line1Run4R, line2Run4R, line3Run4R,
     line4Run4R, line5Run4R, Directive[Black, Dashing[None]],
     linemaxRun4R, linehalfRun4R},
   PlotLabel -> "Non-Counter Movement Jump - Power",
   AxesLabel -> {"Time (s)", "Power (W)"} ];

(* Run5 *)
solutionRun5 = {{IDRun5, "", ""}, {"", "",
    ""}, {"Variable", "Left", "Right"}, {"", "", ""},
   {"run class.:", runtypeRun5L, runtypeRun5R}, {"mass:", massRun5L,
    massRun5R}, {"t1:", t1Run5L, t1Run5R}, {"t2:", t2Run5L, t2Run5R},
   {"t3:", t3Run5L, t3Run5R}, {"t4:", t4Run5L, t4Run5R}, {"t5:",
    t5Run5L, t5Run5R}, {"tmax:", tmaxRun5L, tmaxRun5R}, {"thalf:",
    thalfRun5L, thalfRun5R}, {"", "", ""}, {"force @ t1:",
    force1Run5L, force1Run5R}, {"force @ t2:", force2Run5L,
    force2Run5R}, {"force @ t3:", force3Run5L,
    force3Run5R}, {"force @ t4:", force4Run5L,
    force4Run5R}, {"force @ t5:", force5Run5L,
    force5Run5R}, {"force @ tmax:", forcemaxRun5L,
    forcemaxRun5R}, {"force @ thalf:", forcehalfRun5L,
    forcehalfRun5R}, {"veloc. @ t1:", velocity1Run5L,
    velocity1Run5R}, {"veloc. @ t2:", velocity2Run5L,
    velocity2Run5R}, {"veloc. @ t3:", velocity3Run5L,
    velocity3Run5R}, {"veloc. @ t4:", velocity4Run5L,
    velocity4Run5R}, {"veloc. @ t5:", velocity5Run5L,
    velocity5Run5R}, {"veloc. @ tmax:", velocitymaxRun5L,
    velocitymaxRun5R}, {"veloc. @ thalf:", velocityhalfRun5L,
    velocityhalfRun5R}, {"displace. @ t1:", displacement1Run5L,
    displacement1Run5R}, {"displace. @ t2:", displacement2Run5L,
    displacement2Run5R}, {"displace. @ t3:", displacement3Run5L,
    displacement3Run5R}, {"displace. @ t4:", displacement4Run5L,
    displacement4Run5R}, {"displace. @ t5:", displacement5Run5L,
    displacement5Run5R}, {"displace. @ tmax:", displacementmaxRun5L,
    displacementmaxRun5R}, {"displace. @ thalf:",
    displacementhalfRun5L, displacementhalfRun5R}, {"power @ t1:",
    power1Run5L, power1Run5R}, {"power @ t2:", power2Run5L,
    power2Run5R}, {"power @ t3:", power3Run5L,
    power3Run5R}, {"power @ t4:", power4Run5L,
    power4Run5R}, {"power @ t5:", power5Run5L,
    power5Run5R}, {"power @ tmax:", powermaxRun5L,
    powermaxRun5R}, {"power @ thalf:", powerhalfRun5L,
    powerhalfRun5R}, {"", "", ""}, {"impulse t1-t2:", impulse12Run5L,
    impulse12Run5R}, {"impulse t2-t3:", impulse23Run5L,
    impulse23Run5R}, {"impulse t3-t4:", impulse34Run5L,
    impulse34Run5R}, {"impulse t4-t5:", impulse45Run5L,
    impulse45Run5R}, {"impulse t2-tmax:", impulse2maxRun5L,
    impulse2maxRun5R}, {"impulse t1-thalf:", impulse1halfRun5L,
    impulse1halfRun5R}, {"max force t1-t2:", maxforce12Run5L,
    maxforce12Run5R}, {"max force t2-t3:", maxforce23Run5L,
    maxforce23Run5R}, {"max force t3-t4:", maxforce34Run5L,
    maxforce34Run5R}, {"max force t4-t5:", maxforce45Run5L,
    maxforce45Run5R}, {"max force t2-tmax:", maxforce2maxRun5L,
    maxforce2maxRun5R}, {"max force t1-thalf:", maxforce1halfRun5L,
    maxforce1halfRun5R}, {"min force t1-t2:", minforce12Run5L,
    minforce12Run5R}, {"min force t2-t3:", minforce23Run5L,
    minforce23Run5R}, {"min force t3-t4:", minforce34Run5L,
    minforce34Run5R}, {"min force t4-t5:", minforce45Run5L,
    minforce45Run5R}, {"min force t2-tmax:", minforce2maxRun5L,
    minforce2maxRun5R}, {"min force t1-thalf:", minforce1halfRun5L,
    minforce1halfRun5R}, {"max veloc. t1-t2:", maxvelocity12Run5L,
    maxvelocity12Run5R}, {"max veloc. t2-t3:", maxvelocity23Run5L,
    maxvelocity23Run5R}, {"max veloc. t3-t4:", maxvelocity34Run5L,
    maxvelocity34Run5R}, {"max veloc. t4-t5:", maxvelocity45Run5L,
    maxvelocity45Run5R}, {"max veloc. t2-tmax:", maxvelocity2maxRun5L,
     maxvelocity2maxRun5R}, {"max veloc. t1-thalf:",
    maxvelocity1halfRun5L,
    maxvelocity1halfRun5R}, {"min veloc. t1-t2:", minvelocity12Run5L,
```

```
minvelocity12Run5R}, {"min veloc. t2-t3:", minvelocity23Run5L,
minvelocity23Run5R}, {"min veloc. t3-t4:", minvelocity34Run5L,
minvelocity34Run5R}, {"min veloc. t4-t5:", minvelocity45Run5L,
minvelocity45Run5R}, {"min veloc. t2-tmax:", minvelocity2maxRun5L,
 minvelocity2maxRun5R}, {"min veloc. t1-thalf:",
minvelocity1halfRun5L,
minvelocity1halfRun5R}, {"max displace. t1-t2:",
maxdisplacement12Run5L,
maxdisplacement12Run5R}, {"max displace. t2-t3:",
maxdisplacement23Run5L,
maxdisplacement23Run5R}, {"max displace. t3-t4:",
maxdisplacement34Run5L,
maxdisplacement34Run5R}, {"max displace. t4-t5:",
maxdisplacement45Run5L,
maxdisplacement45Run5R}, {"max displace. t2-tmax:",
maxdisplacement2maxRun5L,
maxdisplacement2maxRun5R}, {"max displace. t1-thalf:",
maxdisplacement1halfRun5L,
maxdisplacement1halfRun5R}, {"min displace. t1-t2:",
mindisplacement12Run5L,
mindisplacement12Run5R}, {"min displace. t2-t3:",
mindisplacement23Run5L,
mindisplacement23Run5R}, {"min displace. t3-t4:",
mindisplacement34Run5L,
mindisplacement34Run5R}, {"min displace. t4-t5:",
mindisplacement45Run5L,
mindisplacement45Run5R}, {"min displace. t2-tmax:",
mindisplacement2maxRun5L,
mindisplacement2maxRun5R}, {"min displace. t1-thalf:",
mindisplacement1halfRun5L,
mindisplacement1halfRun5R}, {"max power t1-t2:", maxpower12Run5L,
maxpower12Run5R}, {"max power t2-t3:", maxpower23Run5L,
maxpower23Run5R}, {"max power t3-t4:", maxpower34Run5L,
maxpower34Run5R}, {"max power t4-t5:", maxpower45Run5L,
maxpower45Run5R}, {"max power t2-tmax:", maxpower2maxRun5L,
maxpower2maxRun5R}, {"max power t1-thalf:", maxpower1halfRun5L,
maxpower1halfRun5R}, {"min power t1-t2:", minpower12Run5L,
minpower12Run5R}, {"min power t2-t3:", minpower23Run5L,
minpower23Run5R}, {"min power t3-t4:", minpower34Run5L,
minpower34Run5R}, {"min power t4-t5:", minpower45Run5L,
minpower45Run5R}, {"min power t2-tmax:", minpower2maxRun5L,
minpower2maxRun5R}, {"min power t1-thalf:", minpower1halfRun5L,
minpower1halfRun5R}, {"avg. force in air:", noforceRun5L,
noforceRun5R}, {"avg. force in squat:", squatavgRun5L,
squatavgRun5R}, {"st. dev. force in squat:", squatstdevRun5L,
squatstdevRun5R}, {"", "", ""}, {"Legs Analyzed Together:", "",
""}, {"", "", ""}, {"Variable", "Legs Combined", ""}, {"", "",
""}, {"run class.:", runtypeRun5Both, ""}, {"mass:", massRun5Both,
 ""}, {"t1:", t1Run5Both, ""}, {"t2:", t2Run5Both, ""}, {"t3:",
t3Run5Both, ""}, {"t4:", t4Run5Both, ""}, {"t5:", t5Run5Both,
""}, {"tmax:", tmaxRun5Both, ""}, {"", "", ""}, {"force @ t1:",
force1Run5Both, ""}, {"force @ t2:", force2Run5Both,
""}, {"force @ t3:", force3Run5Both, ""}, {"force @ t4:",
force4Run5Both, ""}, {"force @ t5:", force5Run5Both,
""}, {"force @ tmax:", forcemaxRun5Both, ""}, {"force @ thalf:",
forcehalfRun5Both, ""}, {"veloc. @ t1:", velocity1Run5Both,
""}, {"veloc. @ t2:", velocity2Run5Both, ""}, {"veloc. @ t3:",
velocity3Run5Both, ""}, {"veloc. @ t4:", velocity4Run5Both,
""}, {"veloc. @ t5:", velocity5Run5Both, ""}, {"veloc. @ tmax:",
velocitymaxRun5Both, ""}, {"veloc. @ thalf:",
velocityhalfRun5Both, ""}, {"displace. @ t1:",
displacement1Run5Both, ""}, {"displace. @ t2:",
displacement2Run5Both, ""}, {"displace. @ t3:",
displacement3Run5Both, ""}, {"displace. @ t4:",
displacement4Run5Both, ""}, {"displace. @ t5:",
displacement5Run5Both, ""}, {"displace. @ tmax:",
displacementmaxRun5Both, ""}, {"displace. @ thalf:",
```

```
   displacementhalfRun5Both, ""}, {"power @ t1:", power1Run5Both,
   ""}, {"power @ t2:", power2Run5Both, ""}, {"power @ t3:",
   power3Run5Both, ""}, {"power @ t4:", power4Run5Both,
   ""}, {"power @ t5:", power5Run5Both, ""}, {"power @ tmax:",
   powermaxRun5Both, ""}, {"power @ thalf:", powerhalfRun5Both,
   ""}, {"", "", ""}, {"impulse t1-t2:", impulse12Run5Both,
   ""}, {"impulse t2-t3:", impulse23Run5Both, ""}, {"impulse t3-t4:",
    impulse34Run5Both, ""}, {"impulse t4-t5:", impulse45Run5Both,
   ""}, {"impulse t2-tmax:", impulse2maxRun5Both,
   ""}, {"impulse t1-thalf:", impulse1halfRun5Both,
   ""}, {"max force t1-t2:", maxforce12Run5Both,
   ""}, {"max force t2-t3:", maxforce23Run5Both,
   ""}, {"max force t3-t4:", maxforce34Run5Both,
   ""}, {"max force t4-t5:", maxforce45Run5Both,
   ""}, {"max force t2-tmax:", maxforce2maxRun5Both,
   ""}, {"max force t1-thalf:", maxforce1halfRun5Both,
   ""}, {"min force t1-t2:", minforce12Run5Both,
   ""}, {"min force t2-t3:", minforce23Run5Both,
   ""}, {"min force t3-t4:", minforce34Run5Both,
   ""}, {"min force t4-t5", minforce45Run5Both,
   ""}, {"min force t2-tmax:", minforce2maxRun5Both,
   ""}, {"min force t1-thalf:", minforce1halfRun5Both,
   ""}, {"max veloc. t1-t2:", maxvelocity12Run5Both,
   ""}, {"max veloc. t2-t3:", maxvelocity23Run5Both,
   ""}, {"max veloc. t3-t4:", maxvelocity34Run5Both,
   ""}, {"max veloc. t4-t5:", maxvelocity45Run5Both,
   ""}, {"max veloc. t2-tmax:", maxvelocity2maxRun5Both,
   ""}, {"max veloc. t1-thalf:", maxvelocity1halfRun5Both,
   ""}, {"min veloc. t1-t2:", minvelocity12Run5Both,
   ""}, {"min veloc. t2-t3:", minvelocity23Run5Both,
   ""}, {"min veloc. t3-t4:", minvelocity34Run5Both,
   ""}, {"min veloc. t4-t5:", minvelocity45Run5Both,
   ""}, {"min veloc. t2-tmax:", minvelocity2maxRun5Both,
   ""}, {"min veloc. t1-thalf:", minvelocity1halfRun5Both,
   ""}, {"max displace. t1-t2:", maxdisplacement12Run5Both,
   ""}, {"max displace. t2-t3:", maxdisplacement23Run5Both,
   ""}, {"max displace. t3-t4:", maxdisplacement34Run5Both,
   ""}, {"max displace. t4-t5:", maxdisplacement45Run5Both,
   ""}, {"max displace. t2-tmax:", maxdisplacement2maxRun5Both,
   ""}, {"max displace. t1-thalf:", maxdisplacement1halfRun5Both,
   ""}, {"min displace. t1-t2:", mindisplacement12Run5Both,
   ""}, {"min displace. t2-t3:", mindisplacement23Run5Both,
   ""}, {"min displace. t3-t4:", mindisplacement34Run5Both,
   ""}, {"min displace. t4-t5:", mindisplacement45Run5Both,
   ""}, {"min displace. t2-tmax:", mindisplacement2maxRun5Both,
   ""}, {"min displace. t1-thalf:", mindisplacement1halfRun5Both,
   ""}, {"max power t1-t2:", maxpower12Run5Both,
   ""}, {"max power t2-t3:", maxpower23Run5Both,
   ""}, {"max power t3-t4:", maxpower34Run5Both,
   ""}, {"max power t4-t5:", maxpower45Run5Both,
   ""}, {"max power t2-tmax:", maxpower2maxRun5Both,
   ""}, {"max power t1-thalf:", maxpower1halfRun5Both,
   ""}, {"min power t1-t2:", minpower12Run5Both,
   ""}, {"min power t2-t3:", minpower23Run5Both,
   ""}, {"min power t3-t4:", minpower34Run5Both,
   ""}, {"min power t4-t5:", minpower45Run5Both,
   ""}, {"min power t2-tmax:", minpower2maxRun5Both,
   ""}, {"min power t1-thalf:", minpower1halfRun5Both,
   ""}, {"avg. force in air:", noforceRun5Both,
   ""}, {"avg. force in squat:", squatavgRun5Both,
   ""}, {"st. dev. force in squat:", squatstdevRun5Both, ""}};

ForceGraphRun5Both =
 Plot[Tooltip[funcRun5Both[x]], {x, cutoffRun5Both, lastTimeRun5Both},
   ImageSize -> 600, PlotStyle -> {Automatic},
  PlotLegends -> {"Legs Combined"},
  Epilog -> {Directive[Blue, Dashed], line1Run5Both, line2Run5Both,
```

```
   line3Run5Both, line4Run5Both, line5Run5Both,
   Directive[Blue, Dashing[None]], linemaxRun5Both,
   linehalfRun5Both}, PlotRange -> All,
  PlotLabel -> "Non-Counter Movement Jump - Normal Force",
  AxesLabel -> {"Time (s)", "Normal Force (N)"} ]
ForceGraphRun5 =
 Plot[Tooltip[{funcRun5L[x], funcRun5R[x]}], {x, cutoffRun5Both,
   lastTimeRun5Both}, ImageSize -> 600,
  PlotLegends -> (Placed[#, Right] & /@ {{"Left", "Right"},
     LineLegend[{{Red, Dashed}, Dashed, Black}, {"Left", "Right"}]}),
   PlotStyle -> {Automatic},
  Epilog -> {Directive[Red, Dashed], line1Run5L, line2Run5L,
   line3Run5L, line4Run5L, line5Run5L, Directive[Red, Dashing[None]],
    linemaxRun5L, linehalfRun5L, Directive[Black, Dashed],
   line1Run5R, line2Run5R, line3Run5R, line4Run5R, line5Run5R,
   Directive[Black, Dashing[None]], linemaxRun5R, linehalfRun5R},
  PlotRange -> All,
  PlotLabel -> "Non-Counter Movement Jump - Normal Force",
  AxesLabel -> {"Time (s)", "Normal Force (N)"} ]
VelocityGraphRun5Both =
 Plot[Tooltip[velocityRun5Both], {x, cutoffRun5Both,
   lastTimeRun5Both}, ImageSize -> 600, PlotStyle -> {Automatic},
  PlotLegends -> {"Legs Combined"},
  Epilog -> {Directive[Blue, Dashed], line1Run5Both, line2Run5Both,
   line3Run5Both, line4Run5Both, line5Run5Both,
   Directive[Blue, Dashing[None]], linemaxRun5Both,
   linehalfRun5Both}, PlotRange -> All,
  PlotLabel -> "Non-Counter Movement Jump - Velocity",
  AxesLabel -> {"Time (s)", "Velocity (m/s)"} ]
VelocityGraphRun5 =
  Plot[Tooltip[{velocityRun5L, velocityRun5R}], {x, cutoffRun5Both,
    lastTimeRun5Both}, ImageSize -> 600,
   PlotLegends -> (Placed[#, Right] & /@ {{"Left", "Right"},
      LineLegend[{{Red, Dashed}, Dashed, Black}, {"Left",
        "Right"}]}), PlotStyle -> {Automatic}, PlotRange -> All,
   Epilog -> {Directive[Red, Dashed], line1Run5L, line2Run5L,
     line3Run5L, line4Run5L, line5Run5L,
     Directive[Red, Dashing[None]], linemaxRun5L, linehalfRun5L,
     Directive[Black, Dashed], line1Run5R, line2Run5R, line3Run5R,
     line4Run5R, line5Run5R, Directive[Black, Dashing[None]],
     linemaxRun5R, linehalfRun5R},
   PlotLabel -> "Non-Counter Movement Jump - Velocity",
   AxesLabel -> {"Time (s)", "Velocity (m/s)"} ];
DisplacementGraphRun5Both =
  Plot[Tooltip[displacementRun5Both], {x, cutoffRun5Both,
    lastTimeRun5Both}, ImageSize -> 600, PlotStyle -> {Automatic},
   PlotLegends -> {"Legs Combined"},
   Epilog -> {Directive[Blue, Dashed], line1Run5Both, line2Run5Both,
     line3Run5Both, line4Run5Both, line5Run5Both,
     Directive[Blue, Dashing[None]], linemaxRun5Both,
     linehalfRun5Both}, PlotRange -> All,
   PlotLabel -> "Non-Counter Movement Jump - Displacement",
   AxesLabel -> {"Time (s)", "Displacement (m)"} ];
DisplacementGraphRun5 =
  Plot[Tooltip[{displacementRun5L, displacementRun5R}], {x,
    cutoffRun5Both, lastTimeRun5Both}, ImageSize -> 600,
   PlotLegends -> (Placed[#, Right] & /@ {{"Left", "Right"},
      LineLegend[{{Red, Dashed}, Dashed, Black}, {"Left",
        "Right"}]}), PlotStyle -> {Automatic}, PlotRange -> All,
   Epilog -> {Directive[Red, Dashed], line1Run5L, line2Run5L,
     line3Run5L, line4Run5L, line5Run5L,
     Directive[Red, Dashing[None]], linemaxRun5L, linehalfRun5L,
     Directive[Black, Dashed], line1Run5R, line2Run5R, line3Run5R,
     line4Run5R, line5Run5R, Directive[Black, Dashing[None]],
     linemaxRun5R, linehalfRun5R},
   PlotLabel -> "Non-Counter Movement Jump - Displacement",
   AxesLabel -> {"Time (s)", "Displacement (m)"} ];
```

```
PowerGraphRun5Both =
  Plot[Tooltip[powerRun5Both], {x, cutoffRun5Both, lastTimeRun5Both},
   ImageSize -> 600, PlotStyle -> {Automatic},
   Epilog -> {Directive[Blue, Dashed],
     PlotLegends -> {"Legs Combined"}, line1Run5Both, line2Run5Both,
     line3Run5Both, line4Run5Both, line5Run5Both,
     Directive[Blue, Dashing[None]], linemaxRun5Both,
     linehalfRun5Both}, PlotRange -> All,
   PlotLabel -> "Non-Counter Movement Jump - Power",
   AxesLabel -> {"Time (s)", "Power (W)"} ];
PowerGraphRun5 =
  Plot[Tooltip[{powerRun5L, powerRun5R}], {x, cutoffRun5Both,
    lastTimeRun5Both}, ImageSize -> 600,
   PlotLegends -> (Placed[#, Right] & /@ {{"Left", "Right"},
      LineLegend[{{Red, Dashed}, Dashed, Black}, {"Left",
        "Right"}]}), PlotStyle -> {Automatic}, PlotRange -> All,
   Epilog -> {Directive[Red, Dashed], line1Run5L, line2Run5L,
     line3Run5L, line4Run5L, line5Run5L,
     Directive[Red, Dashing[None]], linemaxRun5L, linehalfRun5L,
     Directive[Black, Dashed], line1Run5R, line2Run5R, line3Run5R,
     line4Run5R, line5Run5R, Directive[Black, Dashing[None]],
     linemaxRun5R, linehalfRun5R},
   PlotLabel -> "Non-Counter Movement Jump - Power",
   AxesLabel -> {"Time (s)", "Power (W)"} ];

(*EXPORT ALL RUNS*)
Export[
 "C:\\Users\\Admin\\Desktop\\Mathematica Analysis Exports\\" <>
  NameID <> " " <> DateID <> ".xlsx", {solutionRun1, solutionRun2,
  solutionRun3, solutionRun4, solutionRun5}]
Export["C:\\Users\\Admin\\Desktop\\Mathematica Analysis Exports\\" <>
  IDRun1 <> ".png", {{"
    " ForceGraphRun1Both}, {"
    " ForceGraphRun1}, {"
    " VelocityGraphRun1Both}, {"
    " VelocityGraphRun1}, {"
    " DisplacementGraphRun1Both}, {"
    " DisplacementGraphRun1}, {"
    " PowerGraphRun1Both}, {"
    " PowerGraphRun1}} // MatrixForm]
Export["C:\\Users\\Admin\\Desktop\\Mathematica Analysis Exports\\" <>
  IDRun2 <> ".png", {{"
    " ForceGraphRun2Both}, {"
    " ForceGraphRun2}, {"
    " VelocityGraphRun2Both}, {"
    " VelocityGraphRun2}, {"
    " DisplacementGraphRun2Both}, {"
    " DisplacementGraphRun2}, {"
    " PowerGraphRun2Both}, {"
    " PowerGraphRun2}} // MatrixForm]
Export["C:\\Users\\Admin\\Desktop\\Mathematica Analysis Exports\\" <>
  IDRun3 <> ".png", {{"
    " ForceGraphRun3Both}, {"
    " ForceGraphRun3}, {"
    " VelocityGraphRun3Both}, {"
    " VelocityGraphRun3}, {"
    " DisplacementGraphRun3Both}, {"
    " DisplacementGraphRun3}, {"
    " PowerGraphRun3Both}, {"
    " PowerGraphRun3}} // MatrixForm]
Export["C:\\Users\\Admin\\Desktop\\Mathematica Analysis Exports\\" <>
  IDRun4 <> ".png", {{"
    " ForceGraphRun4Both}, {"
    " ForceGraphRun4}, {"
    " VelocityGraphRun4Both}, {"
    " VelocityGraphRun4}, {"
    " DisplacementGraphRun4Both}, {"
```

```
    " DisplacementGraphRun4}, {"
    " PowerGraphRun4Both}, {"
    " PowerGraphRun4}} // MatrixForm]
Export["C:\\Users\\Admin\\Desktop\\Mathematica Analysis Exports\\" <>
  IDRun5 <> ".png", {{"
    " ForceGraphRun5Both}, {"
    " ForceGraphRun5}, {"
    " VelocityGraphRun5Both}, {"
    " VelocityGraphRun5}, {"
    " DisplacementGraphRun5Both}, {"
    " DisplacementGraphRun5}, {"
    " PowerGraphRun5Both}, {"
    " PowerGraphRun5}} // MatrixForm]
```