CJ Geeza Midterm

CJ Geeza

Introduction

Force production asymmetry between limbs is often used as a predictor of future injury, with asymmetries greater than 10-15% being considered at risk. Increased symmetry as a result of rehabilitation can also be used as a marker of success. Although used as a predictor of future injury, especially in ACL rehab patients, the effect of interlimb force production asymmetry is less clear as it relates to performance. A systematic review evaluating the effects of inter-limb asymmetries on physical and sports performance found a high rate of asymmetry across a range of physical quantities. While not a consistent finding, they also concluded that asymmetry has a detrimental effect on performance. Single leg and horizontal jumps have been shown to detect asymmetries, but their effect on Change of Direction performance appears inconclusive. Some studies have even shown a positive impact of asymmetry in cycling and no effect in swimming. During bilateral jumping tasks, it has been shown that a significant difference in the asymmetry between landing and take off phases exists.

The counter-movement jump is divided into 5 phases with different biomechanical demands: Unweighing, Braking, Propulsion, Flight, and Landing. This project aimed to calculate the average asymmetry between limbs during each phase, as well as visualize the change in asymmetry across the course of the jump. By relating the asymmetry score to the phase being performed, corrective exercise mimicking the demands of that phase may be implemented in an attempt to restore symmetry.

Methods

Data from a single CMJ performed by a 28 year old recreational hockey player was collected on a Vault force plate at a sampling rate of 1000 Hz at the USC Clinical Exercise Research Center on September 2, 2022. The data was exported to a csv file, and initially included 3136 observations and 22 variables. After importing the data in R using the function `read.csv`, the combined function `mean(is.na)` was used to confirm no missing data for the variables representing the force produced in the Z direction for the left and right legs.

```r
rm(list=ls())

library(ggplot2)
library(plotly)
```

```

Attaching package: 'plotly'

The following object is masked from 'package:ggplot2':

    last_plot

The following object is masked from 'package:stats':

    filter

The following object is masked from 'package:graphics':

    layout
```

```r
data <-read.csv("CJ CMJ-Countermovement Jump-2022.09.02-10.33.20-Trial1 - CJ CMJ-Countermo
                skip=9)
mean(is.na(data$Z.Left))
```

```
[1] 0
```

```r
mean(is.na(data$Z.Right))
```

```
[1] 0
```

The variable "time" was then shifted to 0 by subtracting the minimum value from each observation for this variable.

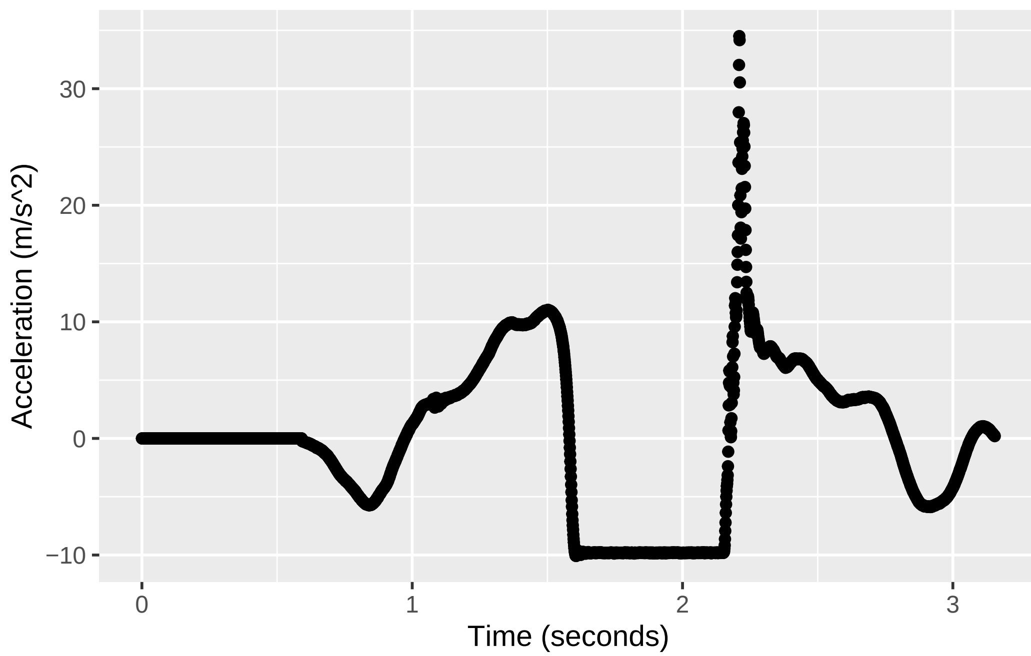
```r
# Change time to normalize to 0
data$Time <- data$Time - min(data$Time)
```

In order to first visualize the data, forces produced by both the right and left legs in the Z direction were plotted. This visualization was accomplished by first rearranging the data to only include the forces produced by the two limbs in the Z direction, as well as an added variable that differentiated between right and left limb. This rearrangement was accomplished by using the function `data.frame()` to create two separate data frames for the right and left limbs, each containing the variables `Time`, `Force`, and `Leg`. These data frames were then stacked together using the function `rbind()`. After this data wrangling, `ggplot2` was used to convert the stacked data frame into a line graph that plotted the forces produced by both limbs against time, differentiating by color. Individual point plots for the right and left limbs were also generated for reference.

```
T <- nrow(data)

# re-arrange data to be able to plot left leg against right leg
data_tidy <- data.frame(Time=data$Time, Force=data$Z.Left, Leg="Left")

data_tidy <- rbind(data_tidy, data.frame(Time=data$Time, Force=data$Z.Right,
                                         Leg="Right"))

# Create column for difference between legs
data$difference <- data$Z.Right - data$Z.Left

plot_left <- ggplot(data, aes(x=Time, y=Z.Left)) + geom_point() +
  ylab("Force from Left Leg (Newtons)") + xlab("Time (seconds)")

plot_right <- ggplot(data, aes(x=Time, y=Z.Right)) + geom_point() +
  ylab("Force from Right Leg (Newtons)") + xlab("Time (seconds)")

plot_both <- ggplot(data_tidy, aes(x=Time, y=Force, color=Leg)) + geom_line() +
  ylab("Force (Newtons)") + xlab("Time (seconds)")
```

Next, the goal was to first break down the data into Standing, Unweighing, Braking, Propulsion, and Landing phases, and calculate the average absolute asymmetry for each phase. The phases were defined based on values from the `acceleration` variable in the dataset. First, an acceleration vs time plot was generated using `ggplot2` to visualize how the acceleration changed throughout the CMJ.

```
# Acceleration plot

plot_acc <- ggplot(data, aes(x=Time, y=Acceleration)) + geom_point() +
  ylab("Acceleration (m/s^2)") + xlab("Time (seconds)")

plot_acc
```



Using this plot as a guide to read through the data, observations with an acceleration value of 0 were characterized as Standing (indices 1-592), negative accelerations as Unweighing (indices 593-973), and positive accelerations as Braking (indices 974- 1374). The Propulsion phase was defined as the time points between when the acceleration reached its first peak and then started the decline until the time in air (indices 1375:1601). The flight phase was omitted due to lack of relevance. The Landing phase was defined as when the acceleration started meaningfully increasing from -9.8 m/s^2 until the last observation (indices 2155 - `nrow(data)`).

```
# Find standing phase: acceleration 0
# Looks like rows 1 - 592
# data[2100:2600, c("Time", "Acceleration")]
standing_inds <- 1:592

# Find unweighing phase: acceleration negative
# Rows 593 - 973
unweighting_inds <- 593:973

# Find breaking phase: acceleration positive
# Rows 974 - 1374
braking_inds <- 974:1374

plot_diff <- ggplot(data[braking_inds, ], aes(x=Time, y=difference)) + geom_point() +
```

```
  ylab("Differences in Forces (Right Leg - Left) (Newtons)") + xlab("Time (seconds)")

# Find propulsion phase: acceleration peaks and starts declining
# Rows 1375 - 1601
propulsion_inds <- 1375:1601

# In the air: acceleration -9.8

# Landing rows: 2155 - T
landing_inds <- 2155:T
```

The average absolute asymmetry for each phase was calculated by first inserting a new variable into the data that described the absolute difference between the force produced in the right and left limbs. This was accomplished by taking the absolute value of the force produced by the left limb subtracted from the right limb, and dividing this quantity by the maximum of these two values for each observation (using the function `pmax`). This quantity was then multiplied by 100 to give a percent value. The average asymmetry for each phase was then calculated by using the `mean` function on data extracted by each phase's corresponding indices.

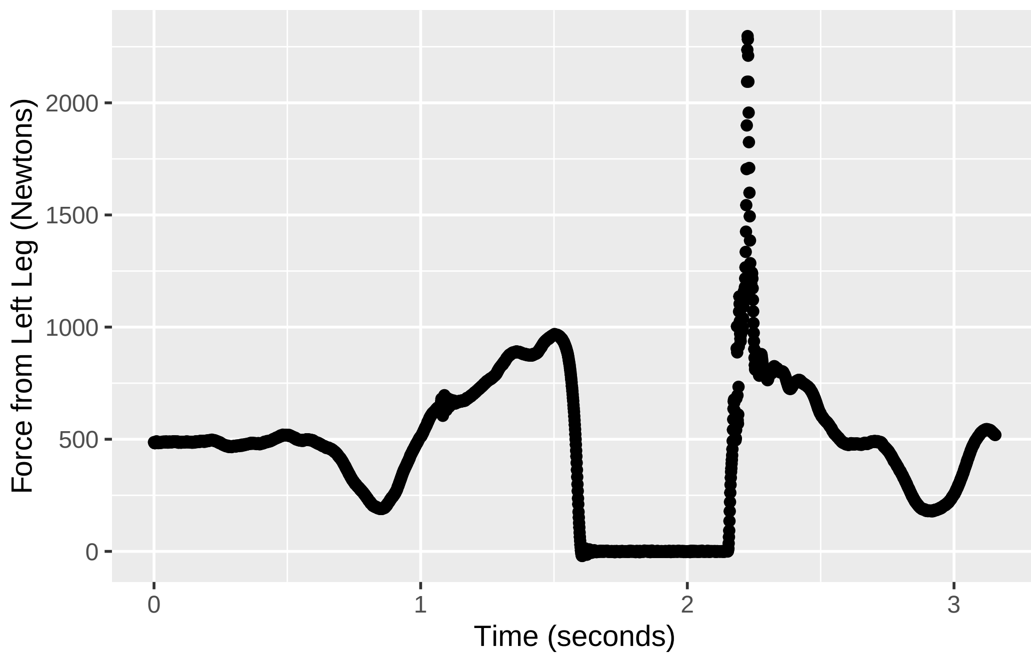
```
# Calculate the percentage difference between force from left and right legs
# in the z direction at each time
data$pct_diff <- (data$Z.Left - data$Z.Right)/pmax(data$Z.Left, data$Z.Right)*100
data$abs_pct_diff <- abs(data$Z.Left - data$Z.Right)/pmax(data$Z.Left, data$Z.Right)*100

# Average absolute asymmetry during standing phase:
avg_standing_asymmetry <- mean(data[standing_inds, "abs_pct_diff"])

# Average absolute asymmetry during unweighting phase:
avg_unweighting_asymmetry <- mean(data[unweighting_inds, "abs_pct_diff"])

# Average absolute asymmetry during breaking phase:
avg_braking_asymmetry <- mean(data[braking_inds, "abs_pct_diff"])

# Average absolute asymmetry during propulsion phase:
avg_propulsion_asymmetry <- mean(data[propulsion_inds, "abs_pct_diff"])

# Average absolute asymmetry during braking phase
avg_landing_asymmetry <- mean(data[landing_inds, "abs_pct_diff"])
```

Finally, the absolute percent asymmetry as a function of time was plotted. For better readability, one plot was generated for the Standing, Unweighing, Braking, and Propulsion phases, and another for the Landing phase. These plots were generated by concatenating the relevant phases from the data and using `ggplot2`. Dotted red lines at y-intercepts of 10 and -10 were inserted to better visualize asymmetries greater than 10%.

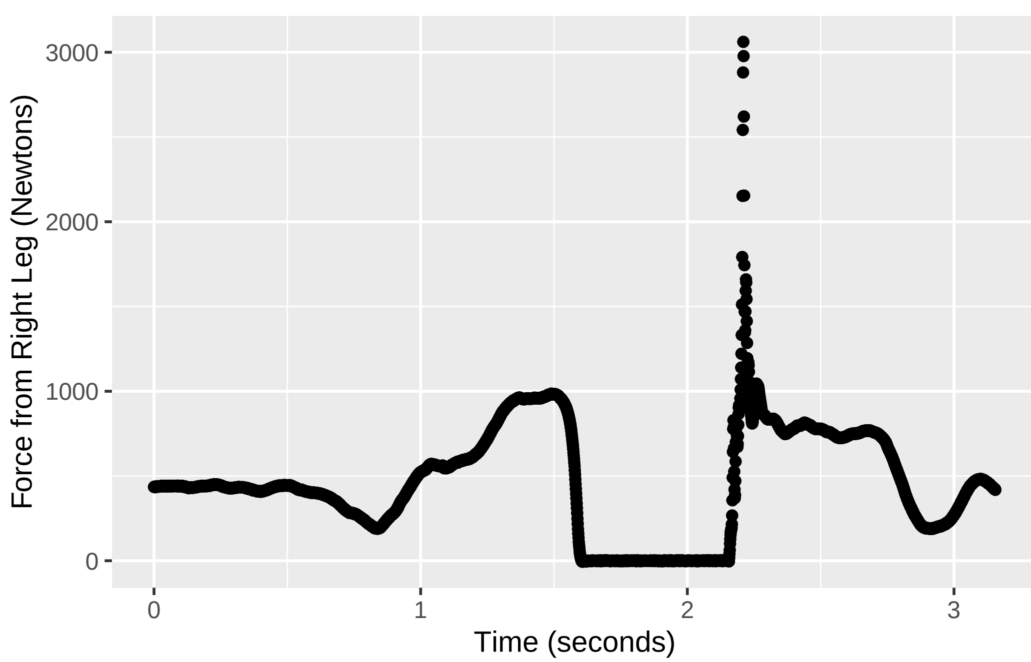
```
df_plot <- data[c(standing_inds, unweighting_inds, braking_inds,
                  propulsion_inds), ]

plot_pct_diff <- ggplot(df_plot, aes(x=Time, y= pct_diff)) + geom_point() +
  geom_hline(yintercept=10, color="red", linetype = "dashed") +
  geom_hline(yintercept=-10, color="red", linetype = "dashed") +
  ylab("Percent Difference") + ggtitle("Percent Asymmetry in Leg Force Over Time")

landing_plot <- data[landing_inds,]

landing_plot_pct_diff <- ggplot(landing_plot, aes(x=Time, y= pct_diff)) + geom_point() +
  geom_hline(yintercept=10, color="red", linetype = "dashed") +
  geom_hline(yintercept=-10, color="red", linetype = "dashed") +
  ylab("Percent Difference") + ggtitle("Percent Asymmetry in Leg Force Over Time")
```

Results

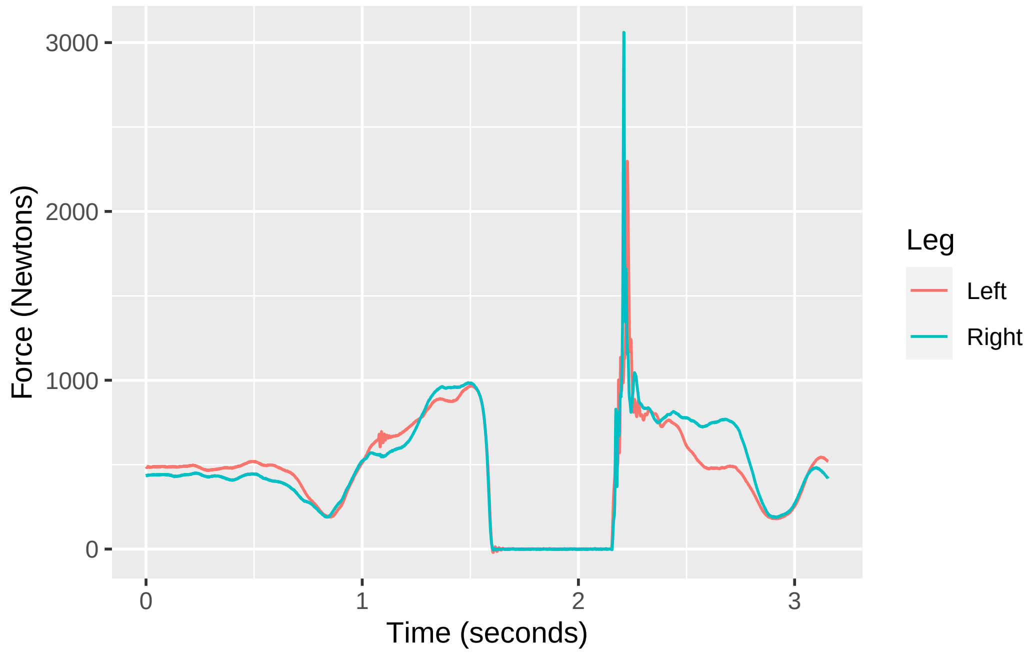
```
# ggplotly(plot_left)
#
# ggplotly(plot_right)
#
# ggplotly(plot_both)
#
# ggplotly(plot_diff)
#
# ggplotly(plot_pct_diff)
#
# ggplotly(landing_plot_pct_diff)

plot_left
```


Force from Left Leg (Newtons)
2000
1500
1000
500
0
0
1
2
3
Time (seconds)

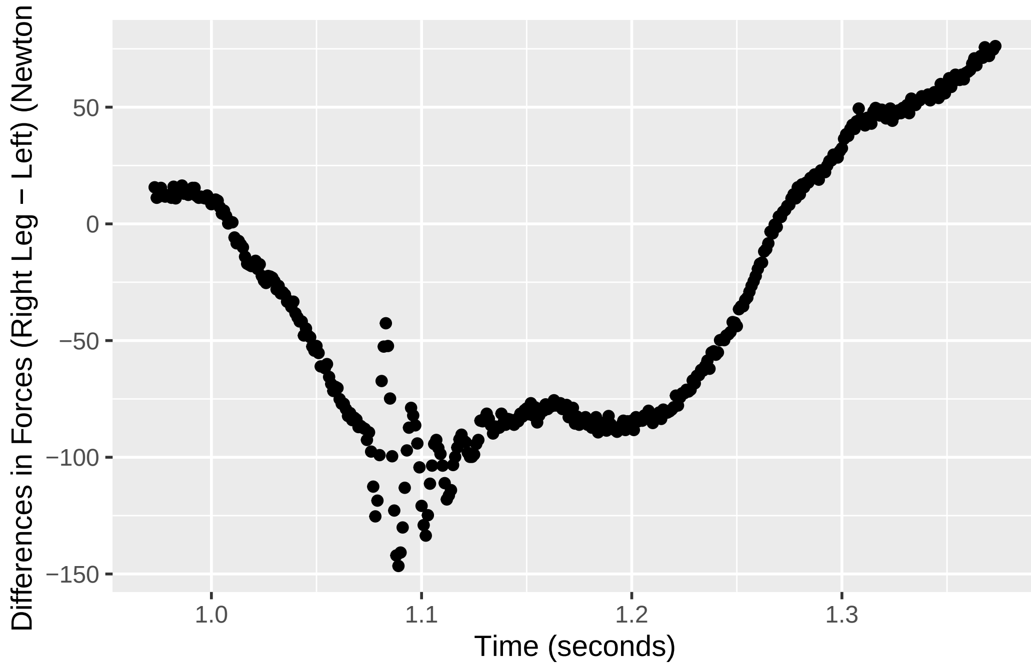
```
plot_right
```


Force from Right Leg (Newtons)
3000
2000
1000
0
0
1
2
3
Time (seconds)

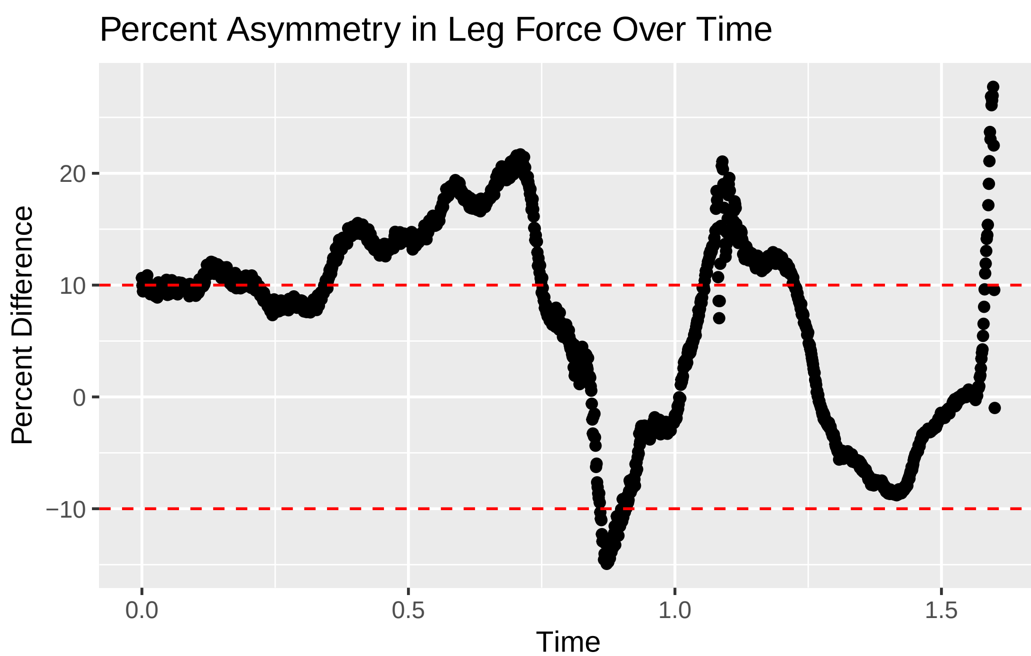
```
plot_both
```



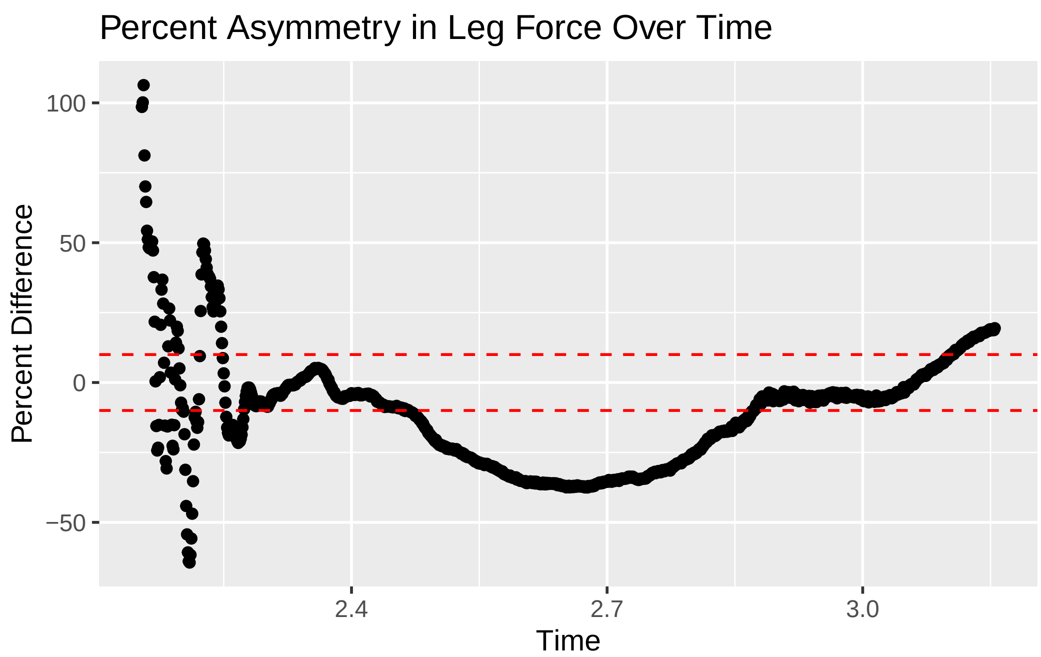

```
plot_diff
```




```
plot_pct_diff
```



```r
landing_plot_pct_diff
```




```r
# Average absolute asymmetry during standing phase:
print(paste(round(avg_standing_asymmetry, 4), "%", sep=""))
```

```
[1] "11.6348%"
```

```r
# Average absolute asymmetry during unweighting phase:
print(paste(round(avg_unweighting_asymmetry, 4), "%", sep=""))
```

```
[1] "11.4569%"
```

```r
# Average absolute asymmetry during breaking phase:
print(paste(round(avg_braking_asymmetry, 4), "%", sep=""))
```

```
[1] "8.1898%"
```

```r
# Average absolute asymmetry during propulsion phase:
print(paste(round(avg_propulsion_asymmetry, 4), "%", sep=""))
```

```
[1] "5.3161%"
```

```r
# Average absolute asymmetry during braking phase
print(paste(round(avg_landing_asymmetry, 4), "%", sep=""))
```

```
[1] "17.8508%"
```

```r
# Summary statistics
print(summary(data[standing_inds, "abs_pct_diff"]))
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  7.317   9.432  10.643  11.635  14.089  19.390
```

```r
print(sd(data[standing_inds, "abs_pct_diff"]))
```

```
[1] 2.843784
```

```r
print(summary(data[standing_inds, "Z.Left"]))
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  465.0   482.4   488.0   489.4   495.0   520.3
```

```r
print(sd(data[standing_inds, "Z.Left"]))
```

```
[1] 12.27255
```

```r
# Unweighting
print(summary(data[unweighting_inds, "abs_pct_diff"]))
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.5603  5.3698 11.1250 11.4569 17.7465 21.6908
```

```r
print(sd(data[unweighting_inds, "abs_pct_diff"]))
```

```
[1] 6.587761
```

```r
# Breaking
print(summary(data[braking_inds, "abs_pct_diff"]))
```

```
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max. 
 0.03345  4.13252  7.41280  8.18975 12.24946 21.05138 
```

```r
print(sd(data[braking_inds, "abs_pct_diff"]))
```

```
[1] 4.829208
```

```r
# Propulsion
print(summary(data[propulsion_inds, "abs_pct_diff"]))
```

```
    Min.  1st Qu.   Median     Mean  3rd Qu.     Max. 
 0.01957  1.04534  3.67047  5.31612  8.13031 27.73442 
```

```r
print(sd(data[propulsion_inds, "abs_pct_diff"]))
```

```
[1] 5.467503
```

```r
# Landing
avg_landing_asymmetry <- mean(data[landing_inds, "abs_pct_diff"])
print(paste(round(avg_landing_asymmetry, 4), "%", sep=""))
```

```
[1] "17.8508%"
```

Conclusion