Joshua Grossman
Tyler Laurisch
Alyssa Siegmann

Stat 139 Final Project
Human Motor Task Analysis

Introduction

The dominant hand is more precise and accurate than the non-dominant hand in most human motor tasks. The cause of this phenomenon is uncertain. There is a hypothesized internal model in the central nervous system that allows prediction of the outcome of initiated movement by neural simulation (Oldfield 1880). The precision of a movement is linked to the strength of this internal model, defined as the accuracy of the model's prediction. We sought to determine whether the increased precision of the dominant hand is due to a stronger internal model for the dominant hand than for the non-dominant hand. Though we did not come to a definitive answer to this question, there were nonetheless interesting results.

Methods

Our experiment had subjects play a soccer-like game where one hand would control a pass across the screen while the other hand pressed a key in order to "kick" the passed ball, as displayed in Figure 1.

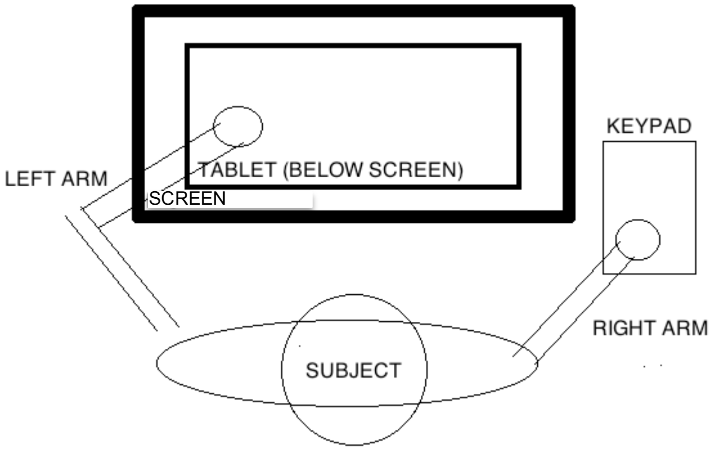


Figure 1
Subjects were elevated above the screen on which the experiment was displayed. Between each block of 70 trials, the subject moved the keypad to the opposite side of the tablet and switched the control stick to the opposite hand.

The object of the game was to score a goal as close as possible to the center of the onscreen net. The ball was either automatically passed by the computer for an “auto pass” trial or the subject would move the ball on his or her own for a “manual pass” trial. For manual pass trials, subjects controlled the ball with a control stick that could be freely moved on a tablet. The tablet recorded the position of the control stick at all times. Subjects were warned if they were passing the ball too slowly (0.4 pixels/millisecond minimum) and encouraged to keep the ball moving through the center of the screen, as to avoid stopping the ball at the perfect location. Auto passes came at a constant speed of 0.9 pixels/millisecond. Subjects were seated in an elevated chair so that they could look down on the mounted screen on which the visuals of the experiment were displayed. The tablet sat below the screen so that the movement of the subject’s hand below the screen would correlate to the movement of the ball on the screen above.

Subjects completed two learning blocks of 20 trials each and then completed an additional 16 blocks of 70 trials each. The sequence of passes were predetermined for each block by generating a random order of 35 auto and 35 manual passes, and holding each block constant between subjects. Our analysis does not take into account the 40 learning trials. Between each block, subjects would alternate hands for kicking and passing. There were four different types of trials, which will be henceforth referred to as RA, LA, RM, and LM, respectively: trials in which the ball was automatically passed from the subject’s right side, automatically passed from the subject’s left side, passed manually by the subject’s right hand, and passed manually by the subject’s left hand.

Time error, position error, and hand speed were the three major measurements recorded for each trial. Time error was measured as the difference in time between the point at which the subject kicked the ball and the time at which the subject’s hand passed the center of the screen (i.e. the location of the "perfect" kick). Position error was measured as the difference between the position at which the subject's hand was located when the ball was kicked and the position of the center of the screen. Both of these error measurements could be positive (if the ball was kicked too late) or negative (if the ball was kicked too early). Hand velocity was the x-component of the subject's hand speed at the point at which the ball was kicked.

Results

Time and position errors were distributed approximately normally for all subjects after extreme outliers were removed (Figures 2 and 3).

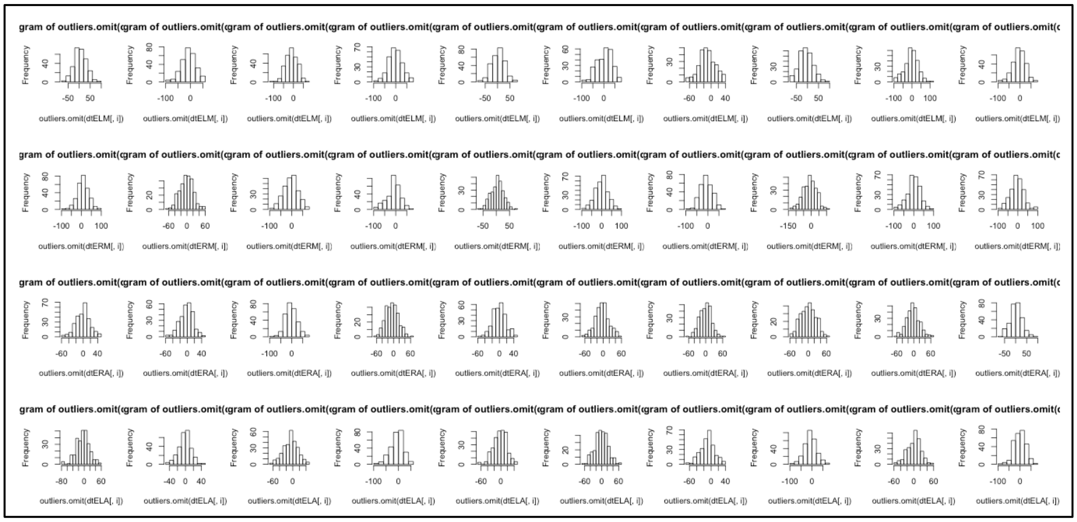


Figure 2 Histograms of time errors for LA, RA, RM, and LM trials with outliers removed. Each column represents one of the ten subjects and each row represents a trial type.

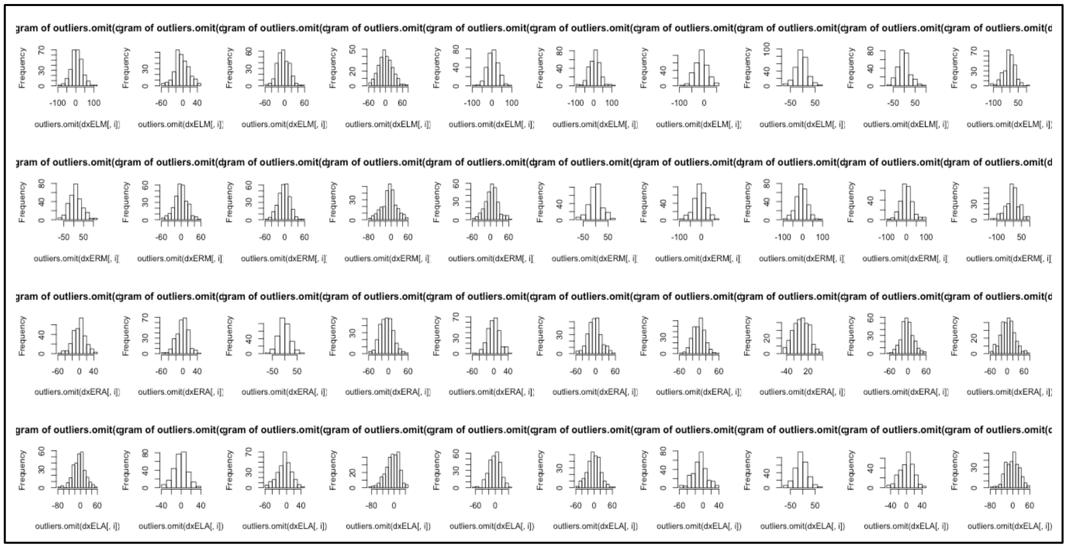


Figure 3 Histograms of position errors for LA, RA, RM, and LM trials with outliers removed. Each column represents one of the ten subjects and each row represents a trial type.

We found no significant difference in the median time or position errors between each type of trial when measured across subjects (paired t-test and permutation test, Bonferroni corrected, p > 0.05 > 0.005 for all four types of trials). Furthermore, we found that the majority of subjects did not exhibit differences in the mean time or position error for every type of trial on an individual basis after outliers were removed (t-tests, Bonferroni corrected, p > 0.05 > 0.005 for the majority of subjects). Thus, we turned to comparison of time error variances, position error variances, and IQRs in order to measure precision rather than accuracy.

Plots of IQR and standard deviation of time and position error for each subject show a pattern of manual passes having significantly larger spreads than auto pass trials (Figures 4-7).

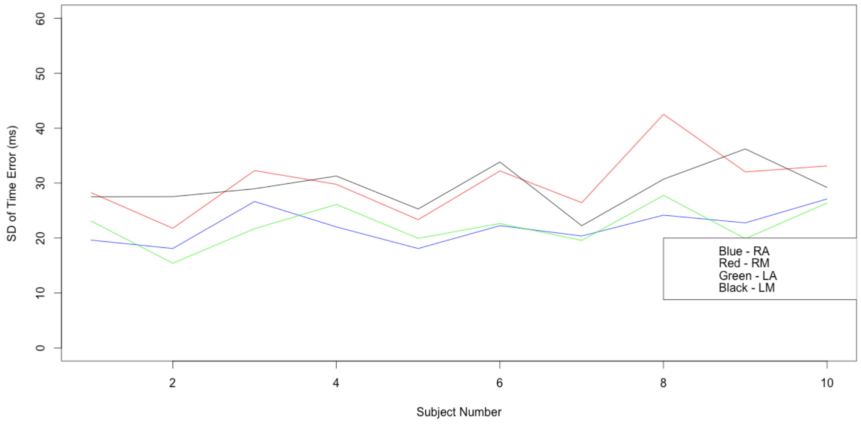


Figure 4
The standard deviation of time error for each type of trial (see key) measured for each subject.

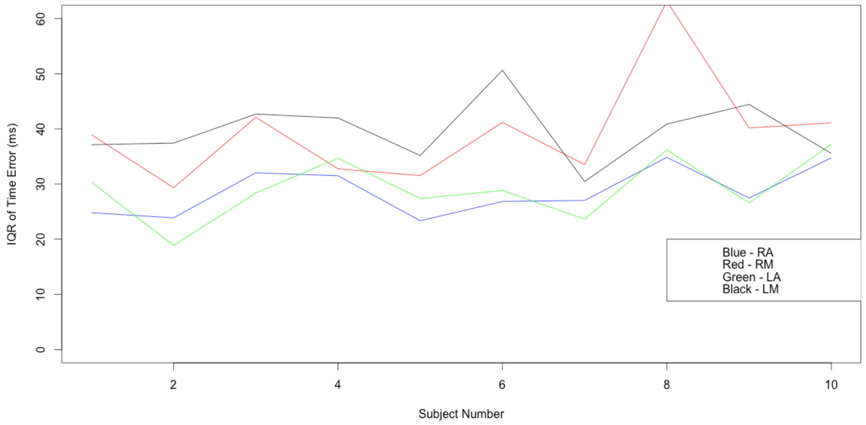


Figure 5
The IQR of time error for each type of trial (see key) measured for each subject.

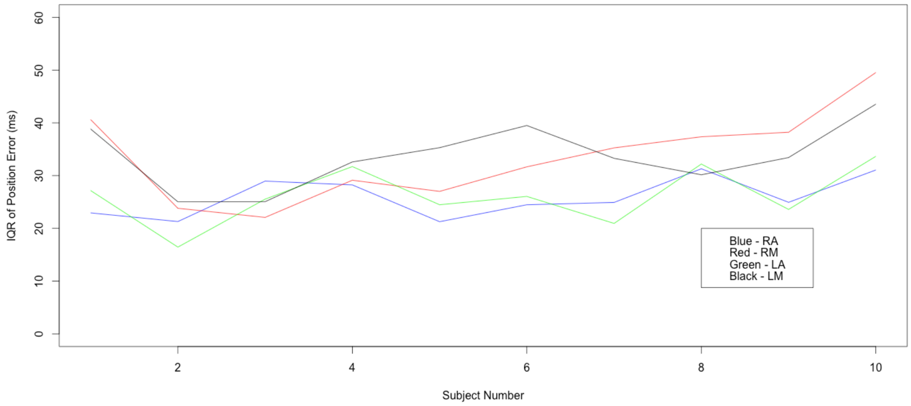


Figure 6
The standard deviation of position error for each type of trial (see key) measured for each subject.

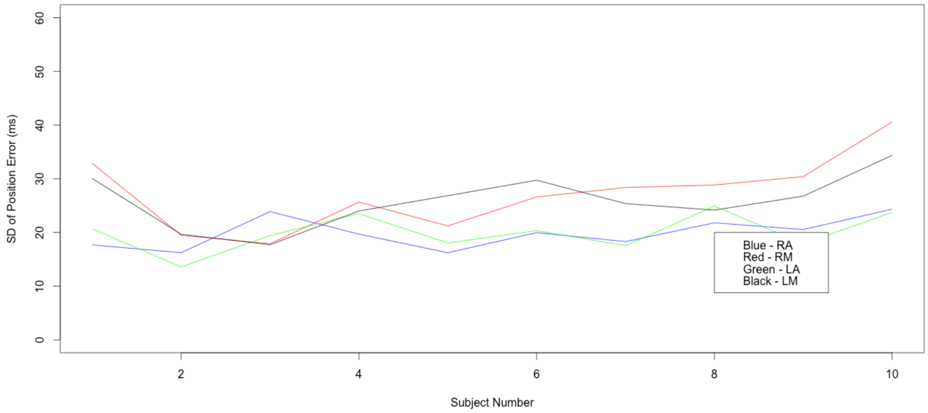


Figure 7
The IQR of position error for each type of trial (see key) measured for each subject.

Statistical testing confirms this result: time error variances of RM and RA trials along with LM and LA trials were significantly different when measured across subjects (paired t-test, p = 0.001166 and p = 0.001449 respectively), though RM/LM and RA/LA variance differences were not significantly different (paired t-test, p > 0.05 for both). The same pattern held for position errors as well, both graphically and statistically (paired t-test, p = 0.008553 and p = 0.00455, respectively). With no significant differences between the left and right hands detected thus far, we turned to analysis of hand speeds to locate any source of handedness-based differences.

Each subject was asked to complete the Edinburgh Handedness Inventory in order to measure his or her degree of handedness (Oldfield 97). Plotting the value of this score against the difference in IQRs of time errors between all LM passes and all RM passes for each subject produced no clear pattern (Figure 8). The same result occurred with position errors. The plots do give a clear indication that subject nine is left-handed.

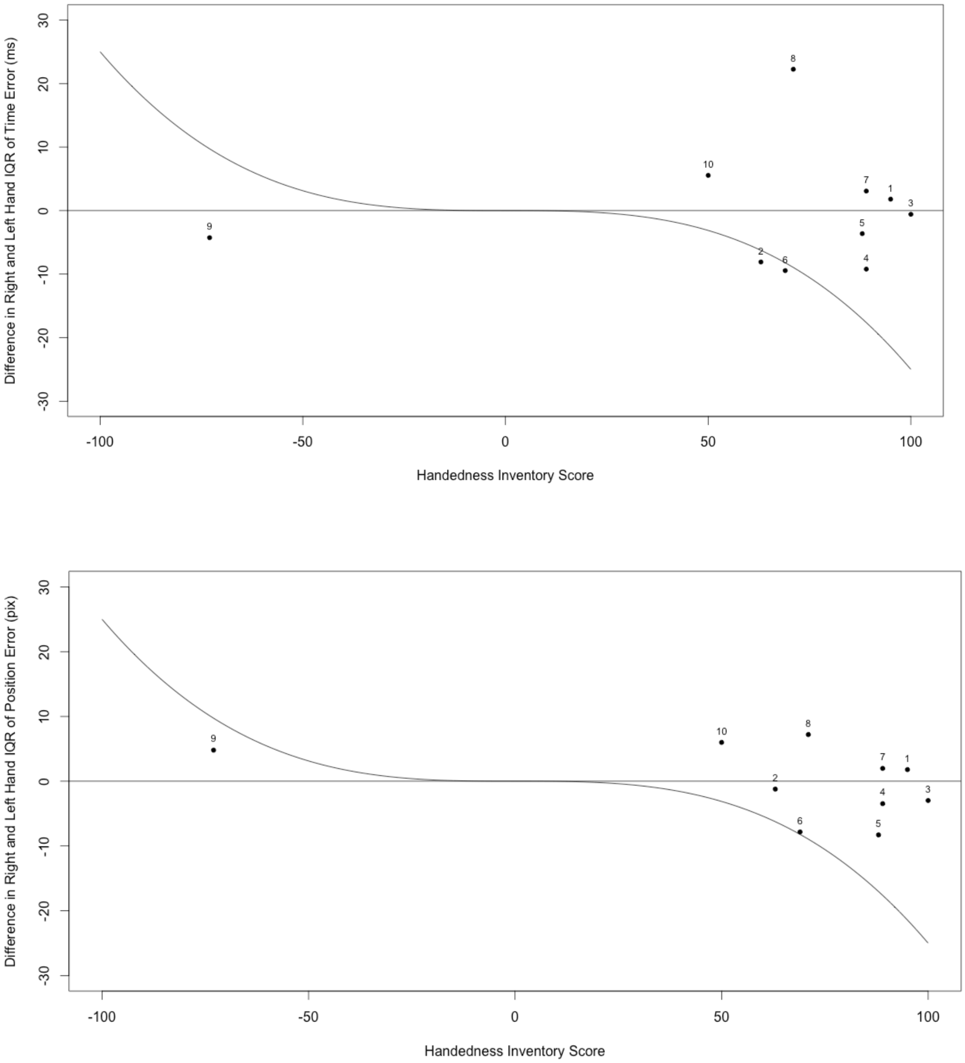


Figure 9
Handedness inventory score plotted against differences in the IQR of the left and right hand for time errors and position errors. Note that the curved lines represent a hypothesized scenario where a stronger right hand would have a presumably smaller IQR for right handed time and position errors than left handed time and position errors and vice versa. The horizontal line represents the null hypothesis of no significant difference in the precision of right and left hands.

While this analysis might lead one to believe that handedness is in no way correlated with time and position errors, the observed lack of difference may simply be due to the confounding factor of hand speed. Each subject may have hand speeds distributed differently for each hand, thus inflating or deflating time and position error and erasing any significant difference in precision between the left and right hands. Plotting the median magnitude of hand speed at the kick against the IQR of time error for left and right manual passes showed a moderate inverse correlation for left passes and almost no correlation for right handed passes (Figure 10).

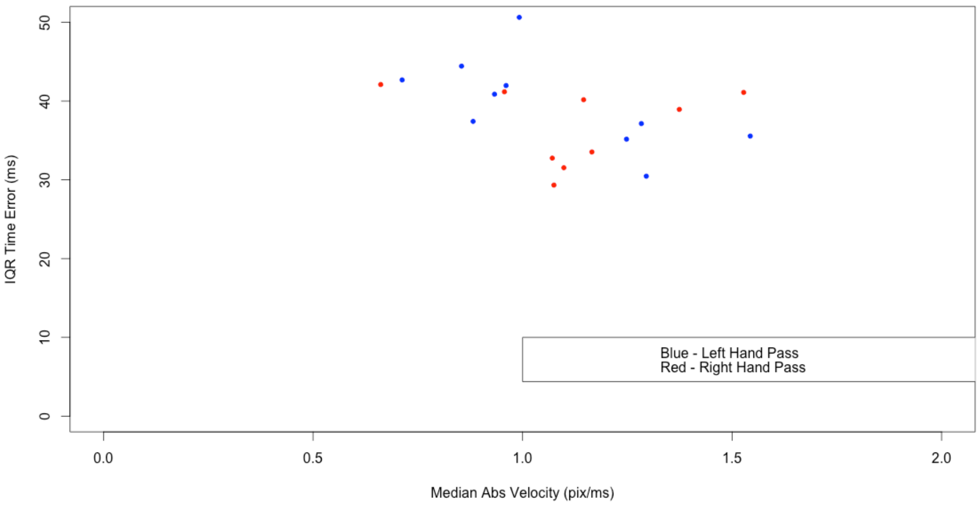


Figure 10
IQR of left and right hand time error as a function of median hand speed at the kick. Each dot represents data from one subject. Note that data is only relevant for manual pass trials during which velocity varies.

A similar plot with the IQR of position error shows a clear positive correlation, as would be expected given that, with a constant value of time error, a faster pass would have a greater position error than a slower pass (Figure 11).

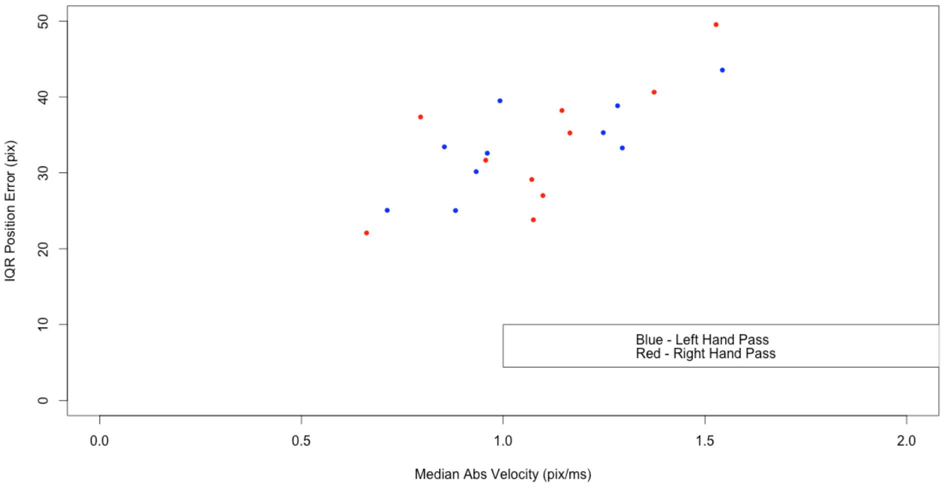


Figure 11
IQR of left and right hand position error as a function of median hand speed at the kick. Each dot represents data from one subject. Note that data is only relevant for manual pass trials during which velocity varies.

Consider histograms of velocity for each hand for each subject binned between 0.4 pixels per millisecond, the minimum speed at which subjects were permitted to move their hands during a given trial, and 1.8 pixels per millisecond, which is double the speed of the automatic pass (Figure 12).

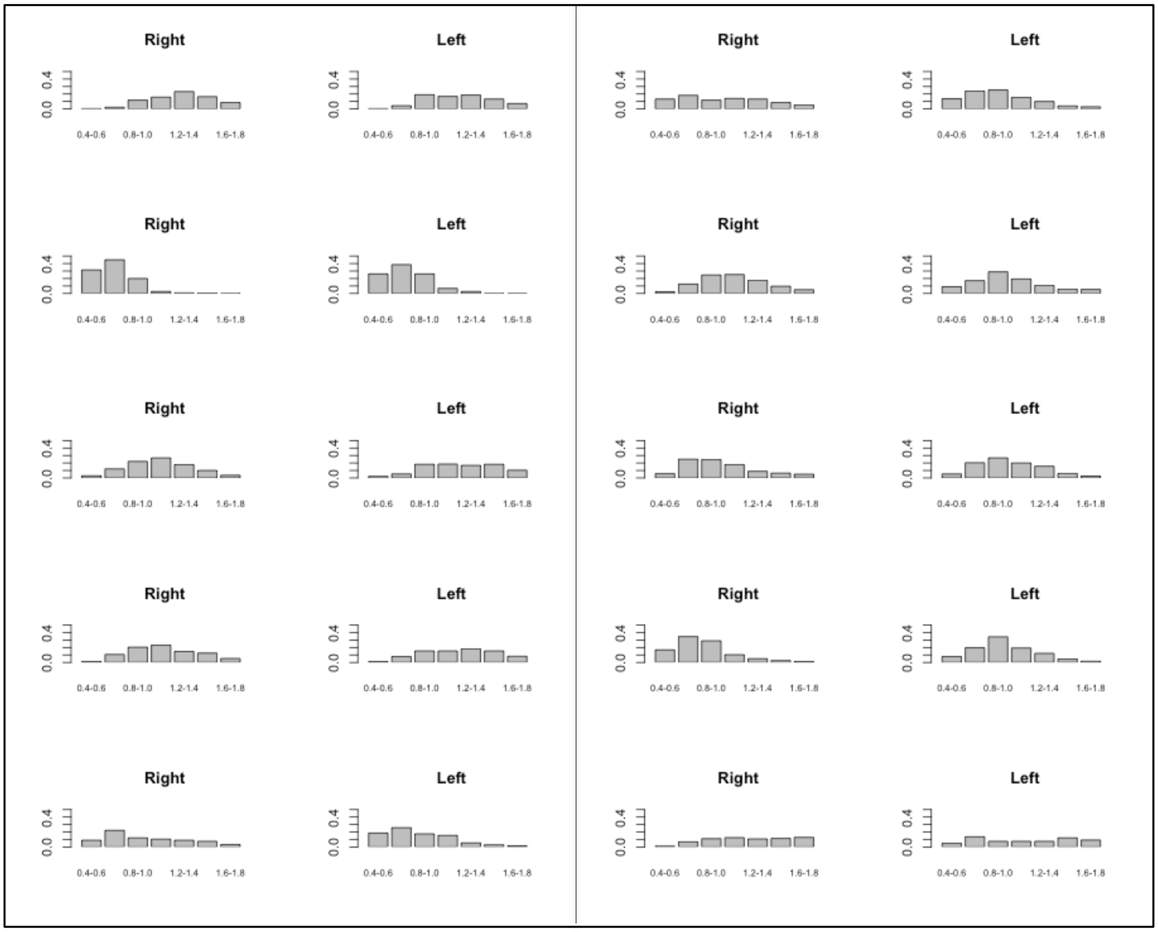


Figure 12 Histograms of pass velocity binned into intervals of 0.2 pixels/ms from 0.4 pixels/ms to 1.8 pixels/ms. Two subjects are shown per row.

There does not appear to be any clear distribution of velocities common to all subjects, and it also appears that there is no consistent or definitive distribution within subjects. Thus, it seems as though exploring precision as a function of both velocity and handedness is appropriate. Consider a plot of the mean IQR of time and position error as a function of velocity for all subjects when binned in the same manner as the histograms above (Figures 13 and 14).

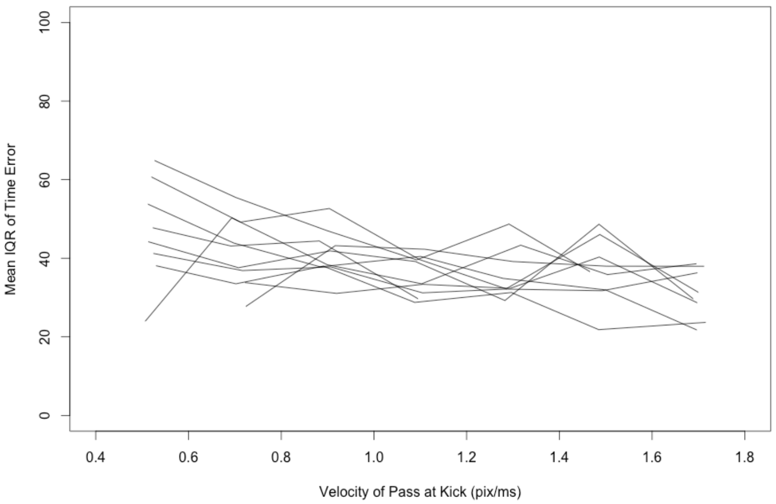


Figure 13
Mean IQR of time error for each velocity bin as measured for all subjects. Each line represents a subject. Discontinuities in any line indicate that less than 10 data points were available for the corresponding bin for that subject.

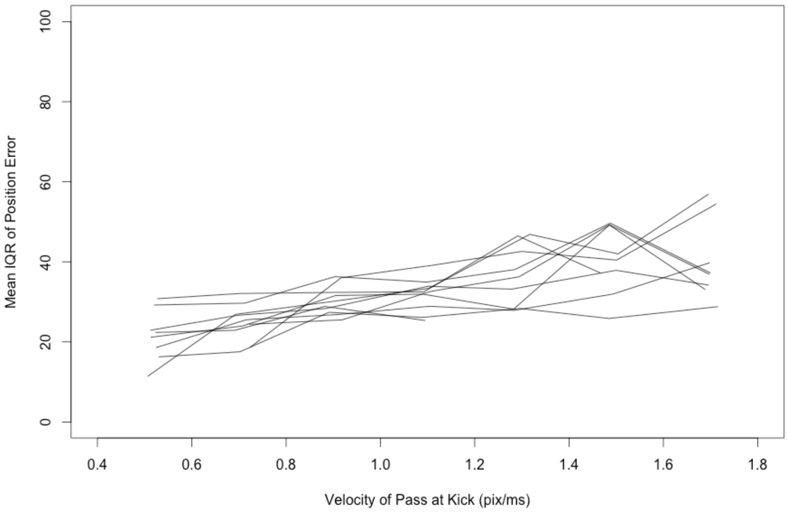


Figure 14
Mean IQR of position error for each velocity bin as measured for all subjects. Each line represents a subject. Discontinuities in any line indicate that less than 10 data points were available for the corresponding bin for that subject.

It appears as though there is an inverse relationship between time error and velocity, as would be expected given that one would be less precise if one were moving faster. The positive relationship between position error and velocity is also expected for the same reasons outlined for the data presented in Figure 11. Consider the same plots in Figures 13 and 14 when averaged across all subjects (Figures 15 and 16).

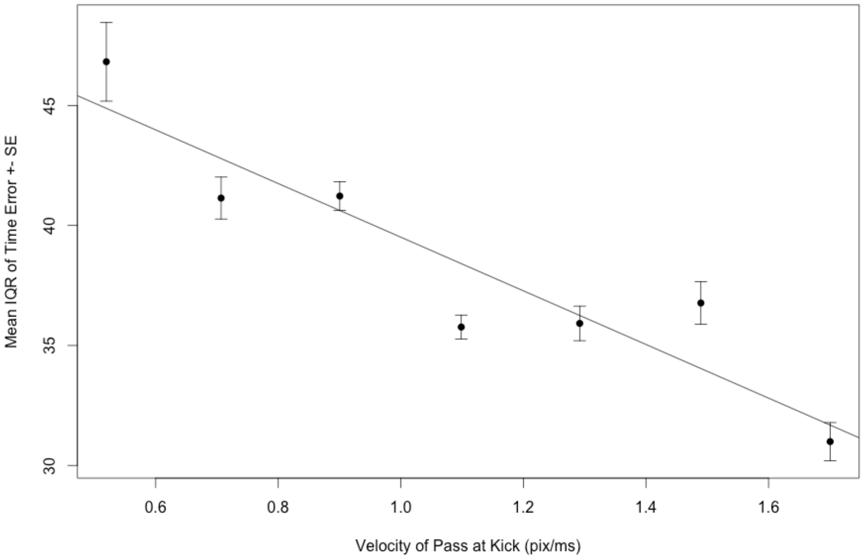


Figure 15
Mean IQR of time error +/- one standard error averaged across all subjects plotted against velocity.

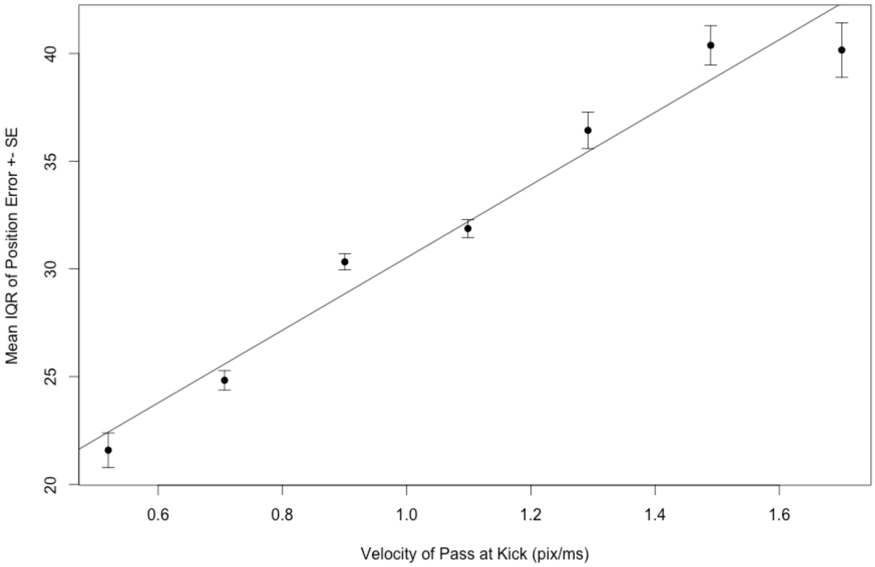


Figure 16
Mean IQR of position error +/- one standard error averaged across all subjects plotted against velocity.

Regression on the average plots for each velocity bin confirms these trends, as both are significant (Linear regression for time error and position error, p = 0.00259 and p = 9.14e-05). Let's now consider the plots of the difference in IQR of time and position error of the left and right hand for each velocity bin (Figures 17 and 18).

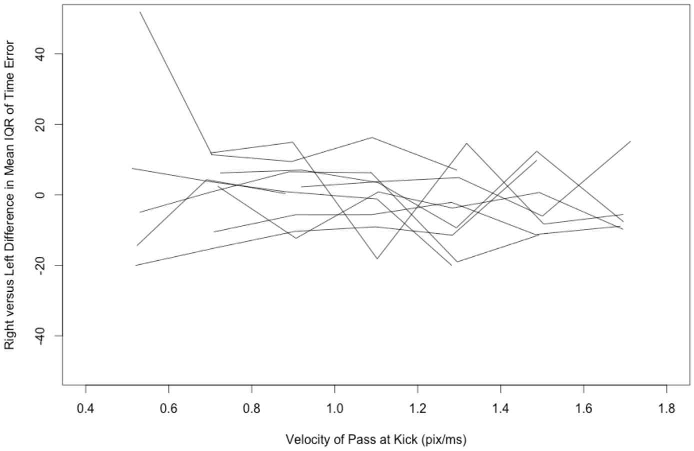


Figure 17
Difference in the right and left hand mean IQR of time error for each velocity bin. Each line represents a subject. If a bin did not contain at least ten data points for a given subject, the line shows a discontinuity at that bin.

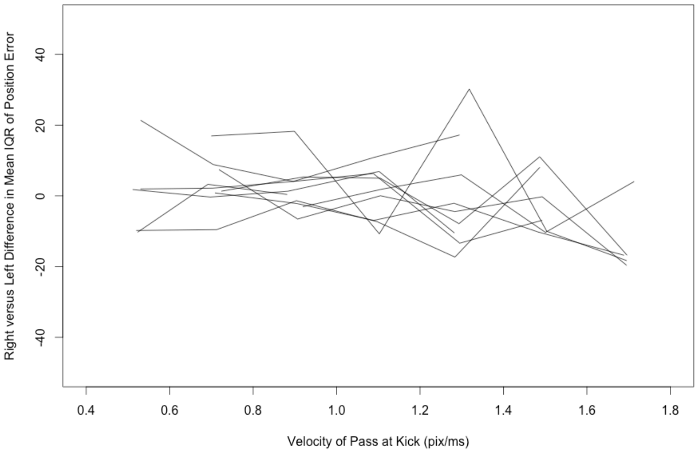


Figure 18
Difference in the right and left hand mean IQR of position error for each velocity bin. Each line represents a subject. If a bin did not contain at least ten data points for a given subject, the line shows a discontinuity at that bin.

Trends are less clear in Figures 17 and 18. Consider the same plots instead averaged over all subjects with error bars (Figures 19 and 20).

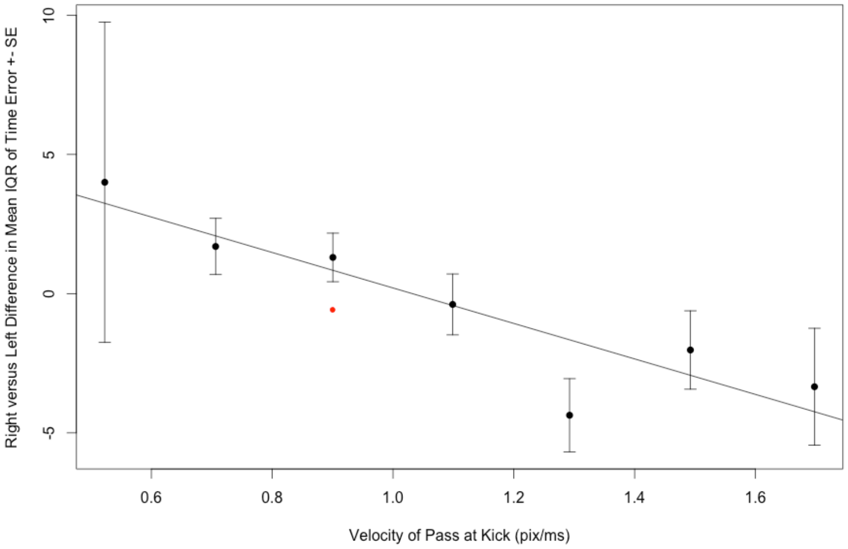


Figure 19
Difference between right and left mean IQR of time error averaged across all subjects +/- one standard error. The red dot represents the difference in right and left hand IQR for the auto pass trials, which had a consistent velocity of 0.9 pixels per second and a negligible standard error.

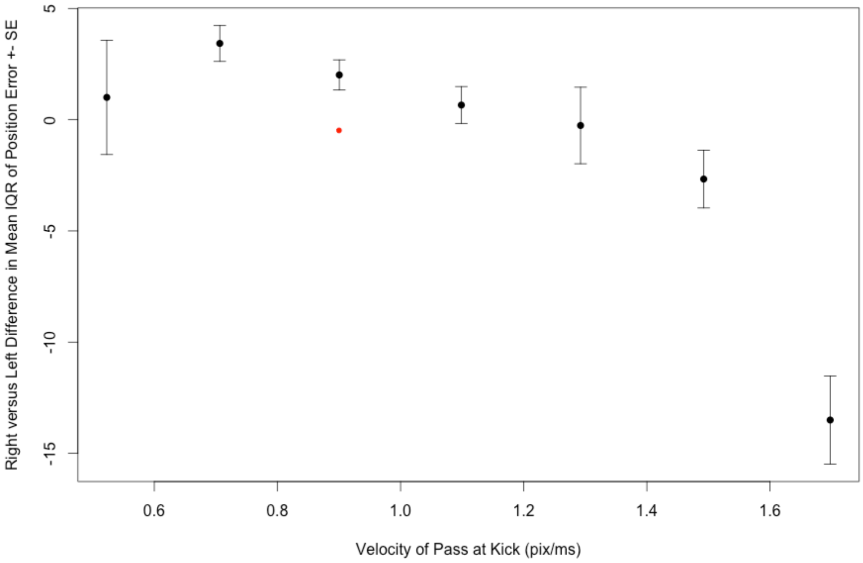


Figure 20
Difference between right and left mean IQR of position error averaged across all subjects +/- one standard error. The red dot represents the difference in right and left hand IQR for the auto pass trials, which had a consistent velocity of 0.9 pixels per second and a negligible standard error.

It appears as though faster velocities exhibit a better precision with the right hand and slower velocities exhibit a better precision with the left hand. Linear regression on each plot supports this result (p = 0.00533 and p = 0.0329, respectively). Consider a plot of the left and right hand mean IQRs of time and position error on the same graph to better visualize where these differences come from (Figures 21 and 22).

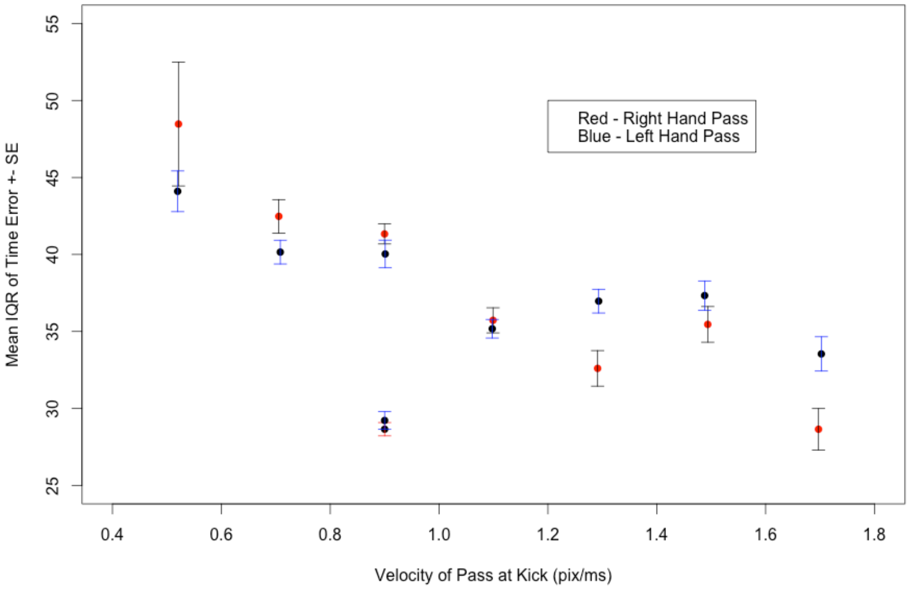


Figure 21
Mean IQR of time error +/- one standard error for left and right hand manual passes averaged over all subjects. The additional points at 0.9 pixels/ms represent the results for auto pass trials.

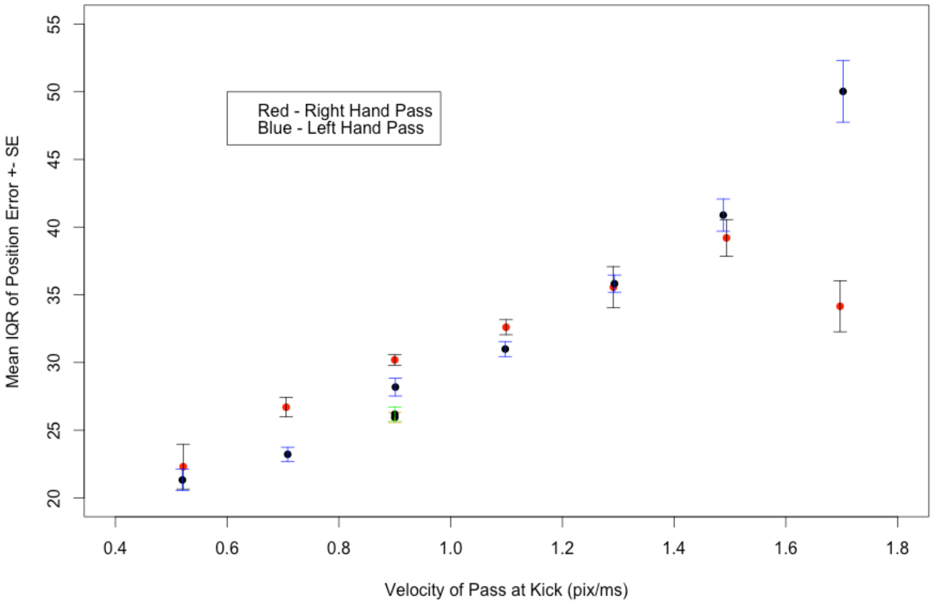


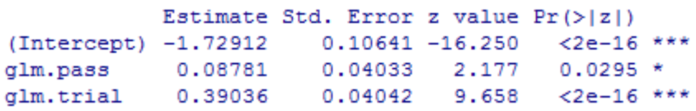
Figure 22
Mean IQR of position error +/- one standard error for left and right hand manual passes averaged over all subjects. The additional points at 0.9 pixels/ms represent the results for auto pass trials.

Although the statistics do not support a difference in IQR of time and position error within each bin (paired t-test, p = [0.684, 0.582, 0.711, 0.834, 0.288, 0.662, 0.329] for each bin in Figure 21 and p = [0.814, 0.196, 0.442, 0.517, 0.952, 0.735, 0.059] for each bin in Figure 22), more data is needed, as the graphics suggest that there might be an interesting relationship between velocity, handedness, and precision.

Finally, we performed a generalized linear model regression (Figure 23) and a mixed-effects model regression (Figure 24) to predict the probability of scoring a goal, based on pass direction and trial type.

Figure 23

Generalized Linear Model



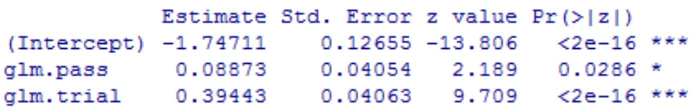
```
            Estimate Std. Error z value Pr(>|z|)    
(Intercept) -1.72912    0.10641 -16.250   <2e-16 ***
glm.pass     0.08781    0.04033   2.177   0.0295 *  
glm.trial    0.39036    0.04042   9.658   <2e-16 ***
```

$$\therefore \quad \hat{p}_{\text{goal}} = \frac{e^{-1.72912+.08781\beta_{pass\ direction}+.39036\beta_{trial\ type}}}{1+e^{-1.72912+.08781\beta_{pass\ direction}+.39036\beta_{trial\ type}}}$$

$$\beta_{\text{pass direction}} = \begin{cases} 0\ if\ left \\ 1\ if\ right \end{cases} \qquad \beta_{\text{trial type}} = \begin{cases} 0\ if\ manual\ pass \\ 1\ if\ automatic\ pass \end{cases}$$

Figure 24

Mixed-Effects Model



```
            Estimate Std. Error z value Pr(>|z|)    
(Intercept) -1.74711    0.12655 -13.806   <2e-16 ***
glm.pass     0.08873    0.04054   2.189   0.0286 *  
glm.trial    0.39443    0.04063   9.709   <2e-16 ***
```

$$\therefore \quad \hat{p}_{\text{goal}} = \frac{e^{-1.74711+.08873\beta_{pass\ direction}+.39443\beta_{trial\ type}}}{1+e^{-1.74711+.08873\beta_{pass\ direction}+.39443\beta_{trial\ type}}}$$

$$\beta_{\text{pass direction}} = \begin{cases} 0\ if\ left \\ 1\ if\ right \end{cases} \qquad \beta_{\text{trial type}} = \begin{cases} 0\ if\ manual\ pass \\ 1\ if\ automatic\ pass \end{cases}$$

GLM assumes that all data points are independent, so we essentially are throwing out any information about correlation between trials of a given individual. The mixed-effects model yields inference for a given individual, rather than for the average of all participants. All coefficients are significant, and the only noteworthy numerical difference in the models is the certainty (i.e., standard error) of the intercepts: 0.10641 vs. 0.12655.

Therefore, our estimates for the predicted probability of an individual scoring a goal and having any particular trial result in a goal are approximately the same (assuming same pass direction and trial type). Furthermore, the coefficients for pass direction and trial type are positive. This means that the predicted probability of scoring a goal (in both models) increases if the ball is passed from the right and increases if the pass is an automatic pass.

Conclusion

In order to truly test the strength of the internal model of the left and rights hands, we need to redesign the experiment with automatic pass trials that model the exact hand movement of corresponding manual pass trials to test if the internal model increases precision in manual pass trials. We also recognize that the dataset includes one left-handed subject, which may be a confounding factor. However, Josh's PI encouraged us to complete the analysis including the left handed subject because the subject pool size was small. Although we cannot make any definitive conclusion as to the strength of the internal model of the left and right hands, the results do bring about an interesting question of whether the right hand is more precise with faster movement speeds while the left hand is more precise with slower movement speeds. Clearly more investigation is needed.

References

1) Oldfield, R.C. “The Assesment and Analysis of Handedness: The Edinburgh Inventory.” *Neuropsychologia* (1971): 97-113. Print.

2) Wolpert, D., Z. Ghahramani, and M. Jordan. “An Interal Model for Sensorimotor Integration.” *Science* (1995): 1880-882. Print.

Appendix

```
####################
# Stat 139 Project #
####################

# Alyssa Siegmann, TJ Laurisch, Josh Grossman

library(ggplot2) # Unused
library(Hmisc) # For error bars
library(lme4) # for glm/glmer

setwd("/Users/JoshGrossman/Google Drive/Harvard/Current School Work/Stat 139/Stat 139
 Project/data")

# All 1120 trials by 10 subjects matrices
goals = read.csv("goals") #0 for miss, 1 for goal
no_follow_through = read.csv("no_follow_through") #1 for no follow through
pass_direction = read.csv("pass_direction") #0 for left, 1 for right
trial_type = read.csv("trial_type") #2 for manual, 3 for auto
thetas = read.csv("thetas") #theta of kick in radians, nonreal if missed kick

# LETTER CODES
# t stands for time
# x stands for position
# E stands for error
# A stands for Auto
# M stands for manual
# L stands for left
# R stands for Right
# d before a variables indicates that NaN are removed and thus indexing is lost

# Useful definitions:

# time error : The difference between the time that the subject pressed the key to kick the ball
# and the time when the subject's hand passes the center of the screen

# position error: The difference between the position that the subject's hand was located when
 the kicked
# the ball and the position of the center of the screen

# Outliers: Outliers in the experiment will be defined with a lenient 2 IQR instead of 1.5 IQR in
 order to preserve
# data. We will use the following function to remove outliers when needed.

outliers.omit = function(x) {
  quant = quantile(na.omit(x), probs=c(.25, .75))
  def = 2*IQR(na.omit(x))
  y = x
  y[x < (quant[1] - def)] = NA
  y[x > (quant[2] + def)] = NA
  y = na.omit(y)
  return(y)
}

tEA = read.csv("tA")      # time error of auto passes
tEM = read.csv("tM")      # time error of manual passes
tELA = read.csv("tELA")   # time error of left auto passes
tELM = read.csv("tELM")   # time error of left manual passes
tERA = read.csv("tERA")   # time error of right auto passes
tERM = read.csv("tERM")   # time error of right manual passes

vM = read.csv('v')        # velocities of manual passes (nothing for auto, all auto passes are
 900 pix/ms)
vM = abs(vM)              # only absolute value of velocity is useful
vLM = read.csv("vLM")     # left manual pass velocities
vLM = abs(vLM)
vRM = read.csv("vRM")     # right manual pass velocities
vRM = abs(vRM)
dvLM = na.omit(vLM)       # remove auto pass trials because no user created velocity
dvRM = na.omit(vRM)
```

```
xEA = read.csv("xA")      # position " " [see time variables above]
xEM = read.csv("xM")
xELA = read.csv("xELA")
xELM = read.csv("xELM")
xERA = read.csv("xERA")
xERM = read.csv("xERM")

dtEA = na.omit(tEA)
dtEM = na.omit(tEA)
dtELA = na.omit(tELA)
dtELM = na.omit(tELM)
dtERA = na.omit(tERA)
dtERM = na.omit(tERM)
dxEA = na.omit(xEA)
dxEM = na.omit(xEM)
dxELA = na.omit(xELA)
dxELM = na.omit(xELM)
dxERA = na.omit(xERA)
dxERM = na.omit(xERM)
dvLM = na.omit(vLM)
dvRM = na.omit(vRM)

colSums(goals)            # indicates total number of goals produced by each subject



########################################################
# T-tests for L,R,A,M time differences within subjects #
########################################################

# First, we are going to see whether subjects' mean/median time errors differ
# between LA,RA,LM, and RM trials. Afterwards, we will look at position errors

# There will be four main analyses where we control for left passes (LA versus LM)
# right (RA versus RM), manual (LM versus RM), and auto (LA versus RA).

# This analysis will be conducted within subjects and across subjects, in that order.
# The null hypothesis is that the means do not differ for within subjects, and that
# the medians do not differ across subjects. The alternative is that they do differ.

# Let's first plot our data to ensure that it is roughly normal. We will plot
# the time errors for each condition.

{
par(mfrow = c(4,10))
for (i in 1:10){
  hist(outliers.omit(dtELM[,i]))
  }

#par(mfrow = c(4,3))
for (i in 1:10){
  hist(outliers.omit(dtERM[,i]))
  }

#par(mfrow = c(4,3))
for (i in 1:10){
  hist(outliers.omit(dtERA[,i]))
  }

#par(mfrow = c(4,3))
for (i in 1:10){
  hist(outliers.omit(dtELA[,i]))
  }
}

# All of the time errors look roughly normally distributed for each subject.

# We can assume independence from the experimental design, as 40 trials are provided for
# learning before the actual experiment begins. Thus time error residuals should be
# randomly distributed throughout the experiment, as should position errors.
# Bonferroni Correction for n=10 subjects and one comparison per subject
alpha = 0.05
alpha = alpha / 10

# Alpha value
alpha

# First look at left manual versus right manual time error for each subject

ttesttM = rep(NA,10)
for (i in 1:10){
  if ((var(outliers.omit(dtELM[,i]))/var(outliers.omit(dtERM[,i]))) <= 2 &&
 (var(outliers.omit(dtERM[,i]))/var(outliers.omit(dtELM[,i]))) <= 2){
       ttesttM[i] = t.test(outliers.omit(dtELM[,i]), outliers.omit(dtERM[,i]),
 var.equal=T)$p.value
  }
  else{
    ttesttM[i] = t.test(outliers.omit(dtELM[,i]), outliers.omit(dtERM[,i]))$p.value
  }
}

# Tests where p-value < alpha (Bonferroni Corrected)
ttesttM < alpha
# This vector shows the subjects for whom right manual and left manual trials was significantly
 different.
# TRUE represents a rejection of the null hypothesis that the means do not differ.


# Next look at left auto versus right auto mean time error for each subject

ttesttA = rep(NA,10)
for (i in 1:10){
  if ((var(outliers.omit(dtELA[,i]))/var(outliers.omit(dtERA[,i]))) <= 2 &&
 (var(outliers.omit(dtERA[,i]))/var(outliers.omit(dtELA[,i]))) <= 2){
    ttesttA[i] = t.test(outliers.omit(dtELA[,i]), outliers.omit(dtERA[,i]), var.equal=T)$p.value
  }
  else{
    ttesttA[i] = t.test(outliers.omit(dtELA[,i]), outliers.omit(dtERA[,i]))$p.value
  }
}
# Tests where p-value < alpha (Bonferroni Corrected)
ttesttA < alpha
# This vector shows the subjects for whom right auto and left auto trials was significantly
 different.
# TRUE represents a rejection of the null hypothesis that the means do not differ.



# Next look at left auto versus left manual time error for each subject

ttesttL = rep(NA,10)
for (i in 1:10){
  if ((var(outliers.omit(dtELM[,i]))/var(outliers.omit(dtELA[,i]))) <= 2 &&
 (var(outliers.omit(dtELA[,i]))/var(outliers.omit(dtELM[,i]))) <= 2){
    ttesttL[i] = t.test(outliers.omit(dtELM[,i]), outliers.omit(dtELA[,i]), var.equal=T)$p.value
  }
  else{
    ttesttL[i] = t.test(outliers.omit(dtELM[,i]), outliers.omit(dtELA[,i]))$p.value
  }
}

# Tests where p-value < alpha (Bonferroni Corrected)
ttesttL < alpha
# This vector shows the subjects for whom left manual and left auto trials was significantly
 different.
# TRUE represents a rejection of the null hypothesis that the means do not differ.



# Last look at right auto versus right manual mean time error for each subject

ttesttR = rep(NA,10)
for (i in 1:10){
  if ((var(outliers.omit(dtERA[,i]))/var(outliers.omit(dtERM[,i]))) <= 2 &&
 (var(outliers.omit(dtERM[,i]))/var(outliers.omit(dtERA[,i]))) <= 2){
    ttesttR[i] = t.test(outliers.omit(dtERA[,i]), outliers.omit(dtERM[,i]), var.equal=T)$p.value
  }
  else{
    ttesttR[i] = t.test(outliers.omit(dtERA[,i]), outliers.omit(dtERM[,i]))$p.value
  }
}


# Tests where p-value < alpha (Bonferroni Corrected)
ttesttR < alpha
# This vector shows the subjects for whom right auto and left auto trials was significantly
 different.
# TRUE represents a rejection of the null hypothesis that the means do not differ.


# Conclusion:
# Looks as though within subjects, there doesn't seem to be a predictable significant difference
 in the mean time
# error for each condition. Let's look at mean position error instead of time error and see if
 that creates
# any predictable results.


################################################################
# t-tests for L,R,A,M position error differences within subjects
################################################################


# Let's first plot our data to ensure that it is roughly normal. We will plot
# the position errors for each condition.
{par(mfrow = c(4,10))
for (i in 1:10){
   hist(outliers.omit(dxELM[,i]))
}

# par(mfrow = c(4,3))
for (i in 1:10){
  hist(outliers.omit(dxERM[,i]))
}

# par(mfrow = c(4,3))
for (i in 1:10){
  hist(outliers.omit(dxERA[,i]))
}

# par(mfrow = c(4,3))
for (i in 1:10){
  hist(outliers.omit(dxELA[,i]))
}
}

par(mfrow = c(1,1))

# All of the position errors look roughly normally distributed for each subject. Thus we can t
 test.

# Bonferroni Correction for n=10 subjects and one comparison per subject
alpha = 0.05
alpha = alpha / 10

# Again, first look at left manual versus right manual mean position error

ttestxM = rep(NA,10)
for (i in 1:10){
  if ((var(outliers.omit(dxELM[,i]))/var(outliers.omit(dxERM[,i]))) <= 2 &&
 (var(outliers.omit(dxERM[,i]))/var(outliers.omit(dxELM[,i]))) <= 2){
    ttestxM[i] = t.test(outliers.omit(dxELM[,i]), outliers.omit(dxERM[,i]), var.equal=T)$p.value
  }
  else{
    ttestxM[i] = t.test(outliers.omit(dxELM[,i]), outliers.omit(dxERM[,i]))$p.value
  }
```

```
}

# Tests where p-value < alpha (Bonferroni Corrected)
ttestxM < alpha

# This vector shows the subjects for whom left manual and right manual trials was significantly
 different.
# TRUE represents a rejection of the null hypothesis that the means do not differ.


# Next look at left auto versus right auto position error

ttestxA = rep(NA,10)
for (i in 1:10){
  if ((var(outliers.omit(dxELA[,i]))/var(outliers.omit(dxERA[,i]))) <= 2 &&
 (var(outliers.omit(dxERA[,i]))/var(outliers.omit(dxELA[,i]))) <= 2){
    ttestxA[i] = t.test(outliers.omit(dxELA[,i]), outliers.omit(dxERA[,i]), var.equal=T)$p.value
  }
  else{
    ttestxA[i] = t.test(outliers.omit(dxELA[,i]), outliers.omit(dxERA[,i]))$p.value
  }
}


# Tests where p-value < alpha (Bonferroni Corrected)
ttestxA < alpha

# This vector shows the subjects for whom left auto and right auto trials was significantly
 different.
# TRUE represents a rejection of the null hypothesis that the means do not differ.


# Next look at left auto versus left manual position error

ttestxL = rep(NA,10)
for (i in 1:10){
  if ((var(outliers.omit(dxELM[,i]))/var(outliers.omit(dxELA[,i]))) <= 2 &&
 (var(outliers.omit(dxELA[,i]))/var(outliers.omit(dxELM[,i]))) <= 2){
    ttestxL[i] = t.test(outliers.omit(dxELM[,i]), outliers.omit(dxELA[,i]), var.equal=T)$p.value
  }
  else{
    ttestxL[i] = t.test(outliers.omit(dxELM[,i]), outliers.omit(dxELA[,i]))$p.value
  }
}


# Tests where p-value < alpha (Bonferroni Corrected)
ttestxL < alpha

# This vector shows the subjects for whom left manual and left auto trials was significantly
 different.
# TRUE represents a rejection of the null hypothesis that the means do not differ.


# Next look at right auto versus right manual position error

ttestxR = rep(NA,10)
for (i in 1:10){
  if ((var(outliers.omit(dxERA[,i]))/var(outliers.omit(dxERM[,i]))) <= 2 &&
 (var(outliers.omit(dxERM[,i]))/var(outliers.omit(dxERA[,i]))) <= 2){
    ttestxR[i] = t.test(outliers.omit(dxERA[,i]), outliers.omit(dxERM[,i]), var.equal=T)$p.value
  }
  else{
    ttestxR[i] = t.test(outliers.omit(dxERA[,i]), outliers.omit(dxERM[,i]))$p.value
  }
}


# Tests where p-value < alpha (Bonferroni Corrected)
ttestxR < alpha

# This vector shows the subjects for whom right auto and right manual trials was significantly
 different.
```

```
# TRUE represents a rejection of the null hypothesis that the means do not differ.


# Looks as though within subjects, there doesn't seem to be a predictable difference in the mean
 position
# error for each condition. Let's look at time and position error across subjects and
# see if there's any overall significant differences. We'd assume that there will not be a
 significant difference,
# but we will used a paired t test and permutation test for both time and position error for each
 condition to check
# if this is indeed the case. We will look at the difference in medians for each subject in order
 to account for outliers.

###################################################################################
# t-tests and permutation tests for L,R,A,M time error differences across subjects
###################################################################################

medianstR = matrix(NA,nrow = 10, ncol = 2)
medianstL = matrix(NA,nrow = 10, ncol = 2)
medianstA = matrix(NA,nrow = 10, ncol = 2)
medianstM = matrix(NA,nrow = 10, ncol = 2)

for (i in 1:10){
  medianstR[i,1] = median(dtERA[,i])
  medianstR[i,2] = median(dtERM[,i])
  medianstL[i,1] = median(dtELA[,i])
  medianstL[i,2] = median(dtELM[,i])
  medianstA[i,1] = median(dtELA[,i])
  medianstA[i,2] = median(dtERA[,i])
  medianstM[i,1] = median(dtELM[,i])
  medianstM[i,2] = median(dtERM[,i])
}

# t-test for L,R,A,M time differences Across Subjects
t.test(medianstR[,1],medianstR[,2],paired = T)
t.test(medianstL[,1],medianstL[,2],paired = T)
t.test(medianstA[,1],medianstA[,2],paired = T)
t.test(medianstM[c(1:7,9:10),1],medianstM[c(1:7,9:10),2],paired = T)

# No significant differences in time error, as expected.

# Permutation Test for L,R,A,M time differences across subjects
nsims = 10000
dR = mean(medianstR[,1]-medianstR[,2]) #observed differences
dL = mean(medianstL[,1]-medianstL[,2])
dA = mean(medianstA[,1]-medianstA[,2])
dM = mean(medianstM[,1]-medianstM[,2])

pR = rep(NA,nsims) #permuted differences
pL = rep(NA,nsims)
pA = rep(NA,nsims)
pM = rep(NA,nsims)

sR = matrix(NA,nrow=10,ncol=2) #shuffled dummy variables for loop
sL = matrix(NA,nrow=10,ncol=2)
sA = matrix(NA,nrow=10,ncol=2)
sM = matrix(NA,nrow=10,ncol=2)

for (i in 1:nsims){
  for (j in 1:10){#number of rows in all means matrices
    sR[j,] = sample(medianstR[j,])
    sL[j,] = sample(medianstL[j,])
    sA[j,] = sample(medianstA[j,])
    sM[j,] = sample(medianstM[j,])
  }
  pR[i] = mean(sR[,1]-sR[,2])
  pL[i] = mean(sL[,1]-sL[,2])
  pA[i] = mean(sA[,1]-sA[,2])
  pM[i] = mean(sM[,1]-sM[,2])
}

# p-values, looks like similar results to t-tests
mean(abs(pR)>abs(dR))
mean(abs(pL)>abs(dL))
mean(abs(pA)>abs(dA))
mean(abs(pM)>abs(dM))

# Same conclusion as paired t test, no significant differences in median time errors.


################################################################################
# t-tests and permutation tests for L,R,A,M position differences across subjects
################################################################################

mediansxR = matrix(NA,nrow = 10, ncol = 2)
mediansxL = matrix(NA,nrow = 10, ncol = 2)
mediansxA = matrix(NA,nrow = 10, ncol = 2)
mediansxM = matrix(NA,nrow = 10, ncol = 2)

for (i in 1:10){
  mediansxR[i,1] = median(dxERA[,i])
  mediansxR[i,2] = median(dxERM[,i])
  mediansxL[i,1] = median(dxELA[,i])
  mediansxL[i,2] = median(dxELM[,i])
  mediansxA[i,1] = median(dxELA[,i])
  mediansxA[i,2] = median(dxERA[,i])
  mediansxM[i,1] = median(dxELM[,i])
  mediansxM[i,2] = median(dxERM[,i])
}

# t-test for L,R,A,M time differences Across Subjects
t.test(mediansxR[,1],mediansxR[,2],paired = T)
t.test(mediansxL[,1],mediansxL[,2],paired = T)
t.test(mediansxA[,1],mediansxA[,2],paired = T)
t.test(mediansxM[c(1:7,9:10),1],mediansxM[c(1:7,9:10),2],paired = T)

# Permutation Test for L,R,A,M time differences across subjects
nsims = 10000
dR = mean(mediansxR[,1]-mediansxR[,2]) #observed differences
dL = mean(mediansxL[,1]-mediansxL[,2])
dA = mean(mediansxA[,1]-mediansxA[,2])
dM = mean(mediansxM[,1]-mediansxM[,2])

pR = rep(NA,nsims) #permuted differences
pL = rep(NA,nsims)
pA = rep(NA,nsims)
pM = rep(NA,nsims)

sR = matrix(NA,nrow=10,ncol=2) #shuffled dummy variables for loop
sL = matrix(NA,nrow=10,ncol=2)
sA = matrix(NA,nrow=10,ncol=2)
sM = matrix(NA,nrow=10,ncol=2)

for (i in 1:nsims){
  for (j in 1:10){#number of rows in all means matrices
    sR[j,] = sample(mediansxR[j,])
    sL[j,] = sample(mediansxL[j,])
    sA[j,] = sample(mediansxA[j,])
    sM[j,] = sample(mediansxM[j,])
  }
  pR[i] = mean(sR[,1]-sR[,2])
  pL[i] = mean(sL[,1]-sL[,2])
  pA[i] = mean(sA[,1]-sA[,2])
  pM[i] = mean(sM[,1]-sM[,2])
}

# p-values, looks like similar results to t-tests
mean(abs(pR)>abs(dR))
mean(abs(pL)>abs(dL))
mean(abs(pA)>abs(dA))
mean(abs(pM)>abs(dM))

# Same results. It seems as though accuracy isn't the thing to test here, but rather precision.
# Thus the remainder of tests will focus on the differences in IQRs of time and position error
 between subjects
# (to account for outliers) along with F tests of the variances (with outliers from data removed).
# Here is a function that we will use to remove outliers. We are using a lenient 2IQR instead of
 1.5IQR
# in order to preserve data.

outliers.omit = function(x) {
  quant = quantile(na.omit(x), probs=c(.25, .75))
  def = 2*IQR(na.omit(x))
  y = x
  y[x < (quant[1] - def)] = NA
  y[x > (quant[2] + def)] = NA
  y = na.omit(y)
  return(y)
}

##################################################################################################
 ####
# t tests for L,R,A,M time error variance differences across subjects and F tests for within
 subjects
##################################################################################################
 ####

# Consider transforming with a square root

vartR = matrix(NA,nrow = 10, ncol = 2)
vartL = matrix(NA,nrow = 10, ncol = 2)
vartA = matrix(NA,nrow = 10, ncol = 2)
vartM = matrix(NA,nrow = 10, ncol = 2)

for (i in 1:10){
  vartR[i,1] = var(outliers.omit(dtERA[,i]))
  vartR[i,2] = var(outliers.omit(dtERM[,i]))
  vartL[i,1] = var(outliers.omit(dtELA[,i]))
  vartL[i,2] = var(outliers.omit(dtELM[,i]))
  vartA[i,1] = var(outliers.omit(dtELA[,i]))
  vartA[i,2] = var(outliers.omit(dtERA[,i]))
  vartM[i,1] = var(outliers.omit(dtELM[,i]))
  vartM[i,2] = var(outliers.omit(dtERM[,i]))
}

# F tests for within subject time error variance differences
fmat = matrix(NA,nrow=4,ncol=10)
for (i in 1:10){
  fmat[1,i] = var.test(outliers.omit(dtERA[,i]),outliers.omit(dtERM[,i]))$p.value
  fmat[2,i] = var.test(outliers.omit(dtELA[,i]),outliers.omit(dtELM[,i]))$p.value
  fmat[3,i] = var.test(outliers.omit(dtELA[,i]),outliers.omit(dtERA[,i]))$p.value
  fmat[4,i] = var.test(outliers.omit(dtELM[,i]),outliers.omit(dtERM[,i]))$p.value
}

# Bonferroni Correction for n=10 subjects and one comparison per subject
alpha = 0.05
alpha = alpha / 10

fmat < alpha

# t test for L,R,A,M time error variances Across Subjects

t.test(vartR[,1],vartR[,2],paired=T)
t.test(vartL[,1],vartL[,2],paired=T)
t.test(vartA[,1],vartA[,2],paired=T)
t.test(vartM[,1],vartM[,2],paired=T)



##########################################################
# t test for L,R,A,M position error variance differences
# across subjects and F tests for within subjects
##########################################################

varxR = matrix(NA,nrow = 10, ncol = 2)
varxL = matrix(NA,nrow = 10, ncol = 2)
varxA = matrix(NA,nrow = 10, ncol = 2)
varxM = matrix(NA,nrow = 10, ncol = 2)
```

```
for (i in 1:10){
  varxR[i,1] = var(outliers.omit(dxERA[,i]))
  varxR[i,2] = var(outliers.omit(dxERM[,i]))
  varxL[i,1] = var(outliers.omit(dxELA[,i]))
  varxL[i,2] = var(outliers.omit(dxELM[,i]))
  varxA[i,1] = var(outliers.omit(dxELA[,i]))
  varxA[i,2] = var(outliers.omit(dxERA[,i]))
  varxM[i,1] = var(outliers.omit(dxELM[,i]))
  varxM[i,2] = var(outliers.omit(dxERM[,i]))
}

# F tests for within subject position error variance differences
fmat = matrix(NA,nrow=4,ncol=10)
for (i in 1:10){
  fmat[1,i] = var.test(outliers.omit(dxERA[,i]),outliers.omit(dxERM[,i]))$p.value
  fmat[2,i] = var.test(outliers.omit(dxELA[,i]),outliers.omit(dxELM[,i]))$p.value
  fmat[3,i] = var.test(outliers.omit(dxELA[,i]),outliers.omit(dxERA[,i]))$p.value
  fmat[4,i] = var.test(outliers.omit(dxELM[,i]),outliers.omit(dxERM[,i]))$p.value
}

# Bonferroni Correction for n=10 subjects and one comparison per subject
alpha = 0.05
alpha = alpha / 10

fmat < alpha

# test for L,R,A,M time error variances Across Subjects
# Again, is this Kosher? Can mean variances be compared with a t test?
t.test(varxR[,1],varxR[,2],paired=T)
t.test(varxL[,1],varxL[,2],paired=T)
t.test(varxA[,1],varxA[,2],paired=T)
t.test(varxM[,1],varxM[,2],paired=T)



######################################################
# Between subjects plots of IQR time error differences
######################################################


medianstR = matrix(NA,nrow = 10, ncol = 2)
medianstL = matrix(NA,nrow = 10, ncol = 2)
medianstA = matrix(NA,nrow = 10, ncol = 2)
medianstM = matrix(NA,nrow = 10, ncol = 2)
SDtR = matrix(NA,nrow = 10, ncol = 2)
SDtL = matrix(NA,nrow = 10, ncol = 2)
SDtA = matrix(NA,nrow = 10, ncol = 2)
SDtM = matrix(NA,nrow = 10, ncol = 2)
IQRtR = matrix(NA,nrow = 10, ncol = 2)
IQRtL = matrix(NA,nrow = 10, ncol = 2)
IQRtA = matrix(NA,nrow = 10, ncol = 2)
IQRtM = matrix(NA,nrow = 10, ncol = 2)


for (i in 1:10){
  IQRtR[i,1] = IQR(dtERA[,i])
  IQRtR[i,2] = IQR(dtERM[,i])
  IQRtL[i,1] = IQR(dtELA[,i])
  IQRtL[i,2] = IQR(dtELM[,i])
  IQRtA[i,1] = IQR(dtELA[,i])
  IQRtA[i,2] = IQR(dtERA[,i])
  IQRtM[i,1] = IQR(dtELM[,i])
  IQRtM[i,2] = IQR(dtERM[,i])

  SDtR[i,1] = sd(outliers.omit(dtERA[,i]))
  SDtR[i,2] = sd(outliers.omit(dtERM[,i]))
  SDtL[i,1] = sd(outliers.omit(dtELA[,i]))
  SDtL[i,2] = sd(outliers.omit(dtELM[,i]))
  SDtA[i,1] = sd(outliers.omit(dtELA[,i]))
  SDtA[i,2] = sd(outliers.omit(dtERA[,i]))
  SDtM[i,1] = sd(outliers.omit(dtELM[,i]))
  SDtM[i,2] = sd(outliers.omit(dtERM[,i]))

  medianstR[i,1] = median(dtERA[,i])
  medianstR[i,2] = median(dtERM[,i])
  medianstL[i,1] = median(dtELA[,i])
  medianstL[i,2] = median(dtELM[,i])
  medianstA[i,1] = median(dtELA[,i])
  medianstA[i,2] = median(dtERA[,i])
  medianstM[i,1] = median(dtELM[,i])
  medianstM[i,2] = median(dtERM[,i])
}

plot(NA,xlim = c(1,10),ylim = c(0,60), xlab = 'Subject Number', ylab = 'IQR of Time Error (ms)')
lines(1:10, IQRtR[,1],col='blue')
lines(1:10, IQRtR[,2],col='red')
lines(1:10, IQRtL[,1],col='green')
lines(1:10, IQRtL[,2],col='black')
legend(8,20,c('Blue - RA','Red - RM','Green - LA','Black - LM'))

plot(NA,xlim = c(1,10),ylim = c(0,60), xlab = 'Subject Number', ylab = 'SD of Time Error (ms)')
lines(1:10, SDtR[,1],col='blue')
lines(1:10, SDtR[,2],col='red')
lines(1:10, SDtL[,1],col='green')
lines(1:10, SDtL[,2],col='black')
legend(8,20,c('Blue - RA','Red - RM','Green - LA','Black - LM'))

plot(NA,xlim = c(1,10),ylim = c(-20,20), xlab = 'Subject Number', ylab = 'Median of Time Error
 (ms)')
lines(1:10, medianstR[,1],col='blue')
lines(1:10, medianstR[,2],col='red')
lines(1:10, medianstL[,1],col='green')
lines(1:10, medianstL[,2],col='black')
legend(8,20,c('Blue - RA','Red - RM','Green - LA','Black - LM'))
##########################################################
# Between subjects plots of IQR position error differences
##########################################################

mediansxR = matrix(NA,nrow = 10, ncol = 2)
mediansxL = matrix(NA,nrow = 10, ncol = 2)
mediansxA = matrix(NA,nrow = 10, ncol = 2)
mediansxM = matrix(NA,nrow = 10, ncol = 2)
SDxR = matrix(NA,nrow = 10, ncol = 2)
SDxL = matrix(NA,nrow = 10, ncol = 2)
SDxA = matrix(NA,nrow = 10, ncol = 2)
SDxM = matrix(NA,nrow = 10, ncol = 2)
IQRxR = matrix(NA,nrow = 10, ncol = 2)
IQRxL = matrix(NA,nrow = 10, ncol = 2)
IQRxA = matrix(NA,nrow = 10, ncol = 2)
IQRxM = matrix(NA,nrow = 10, ncol = 2)


for (i in 1:10){
  IQRxR[i,1] = IQR(dxERA[,i])
  IQRxR[i,2] = IQR(dxERM[,i])
  IQRxL[i,1] = IQR(dxELA[,i])
  IQRxL[i,2] = IQR(dxELM[,i])
  IQRxA[i,1] = IQR(dxELA[,i])
  IQRxA[i,2] = IQR(dxERA[,i])
  IQRxM[i,1] = IQR(dxELM[,i])
  IQRxM[i,2] = IQR(dxERM[,i])

  SDxR[i,1] = sd(outliers.omit(dxERA[,i]))
  SDxR[i,2] = sd(outliers.omit(dxERM[,i]))
  SDxL[i,1] = sd(outliers.omit(dxELA[,i]))
  SDxL[i,2] = sd(outliers.omit(dxELM[,i]))
  SDxA[i,1] = sd(outliers.omit(dxELA[,i]))
  SDxA[i,2] = sd(outliers.omit(dxERA[,i]))
  SDxM[i,1] = sd(outliers.omit(dxELM[,i]))
  SDxM[i,2] = sd(outliers.omit(dxERM[,i]))

  mediansxR[i,1] = median(dxERA[,i])
  mediansxR[i,2] = median(dxERM[,i])
  mediansxL[i,1] = median(dxELA[,i])
  mediansxL[i,2] = median(dxELM[,i])
  mediansxA[i,1] = median(dxELA[,i])
  mediansxA[i,2] = median(dxERA[,i])
  mediansxM[i,1] = median(dxELM[,i])
  mediansxM[i,2] = median(dxERM[,i])
}

plot(NA,xlim = c(1,10),ylim = c(0,60), xlab = 'Subject Number', ylab = 'IQR of Position Error
 (ms)')
lines(1:10, IQRxR[,1],col='blue')
lines(1:10, IQRxR[,2],col='red')
lines(1:10, IQRxL[,1],col='green')
lines(1:10, IQRxL[,2],col='black')
legend(8,20,c('Blue - RA','Red - RM','Green - LA','Black - LM'))

plot(NA,xlim = c(1,10),ylim = c(0,60), xlab = 'Subject Number', ylab = 'SD of Position Error
 (ms)')
lines(1:10, SDxR[,1],col='blue')
lines(1:10, SDxR[,2],col='red')
lines(1:10, SDxL[,1],col='green')
lines(1:10, SDxL[,2],col='black')
legend(8,20,c('Blue - RA','Red - RM','Green - LA','Black - LM'))

plot(NA,xlim = c(1,10),ylim = c(-20,20), xlab = 'Subject Number', ylab = 'Median of Position
 Error (ms)')
lines(1:10, mediansxR[,1],col='blue')
lines(1:10, mediansxR[,2],col='red')
lines(1:10, mediansxL[,1],col='green')
lines(1:10, mediansxL[,2],col='black')
legend(8,20,c('Blue - RA','Red - RM','Green - LA','Black - LM'))

################################################################################
# Does handedness correlate with differences in IQR of time and position error?
################################################################################

hand = c(95,63,100,89,88,69,89,71,-73,50)
plot((IQRtM[,2]-IQRtM[,1])~hand,xlab = 'Handedness Inventory Score',
     ylab = 'Difference in Right and Left Hand IQR of Time Error (ms)', col = 'black',
     pch = 20,ylim = c(-30,30),xlim = c(-100,100))
lines(-100:100,-(-100:100)^3/40000)
abline(0,0)
text(hand,(IQRtM[,2]-IQRtM[,1]),1:10,cex=0.7,pos=3)

plot((IQRxM[,2]-IQRxM[,1])~hand,xlab = 'Handedness Inventory Score',
     ylab = 'Difference in Right and Left Hand IQR of Position Error (pix)', col = 'black',
     pch = 20,ylim = c(-30,30),xlim = c(-100,100))
lines(-100:100,-(-100:100)^3/40000)
abline(0,0)
text(hand,(IQRxM[,2]-IQRxM[,1]),1:10,cex=0.7,pos=3)

plot(NA, xlim = c(0,2), ylim = c(0,50), xlab = 'Median Abs Velocity (pix/ms)', ylab = 'IQR Time
 Error (ms)')
for (i in 1:10){
  points(median(abs(dvLM[,i])),IQR(dtELM[,i]), col = 'blue', pch = 20)
  points(median(abs(dvRM[,i])),IQR(dtERM[,i]), col = 'red', pch = 20)
}
legend(1,10,c('Blue - Left Hand Pass','Red - Right Hand Pass'))

plot(NA, xlim = c(0,2), ylim = c(0,50), xlab = 'Median Abs Velocity (pix/ms)', ylab = 'IQR
 Position Error (pix)')
for (i in 1:10){
  points(median(abs(dvLM[,i])),IQR(dxELM[,i]), col = 'blue', pch = 20)
  points(median(abs(dvRM[,i])),IQR(dxERM[,i]), col = 'red', pch = 20)
}
legend(1,10,c('Blue - Left Hand Pass','Red - Right Hand Pass'))



# Seems like no obvious correlation between handedness and differences in IQR of time and
 position errors!
# Notice subjects 8 (high error) and subject 9 (left handed).
# Maybe this is more of an issue with different velocities? A faster velocity would presumably
 have lower
```

```
# time errors but high position errors given the same accuracy. Let's explore IQR differences for
 each
# bin of abs velocity from the minimum (400 pix/s) to double the auto pass speed (1800 pix/s).
 Let's arbitrarily set
# bin number to 7.

# This analysis should be completed for overall IQR of time and position error and difference in
 IQR between
# left and right.

# Let's create a matrix of 6 rows and 10 columns where each row represents the value of
# time or position error IQR for a given bin of velocity (0.6 to 0.8, 0.8 to 1.0, etc.) and each
 column
# represents a subject. Let's also make sure that each bin has at least ten observations,
 otherwise
# we will leave it blank.

########################################
# Handedness Differences with Time Error
########################################

nsubs = 10
nbins = 7
iqr.mat = matrix(NA,nrow=nbins,ncol=nsubs)
v.mat = matrix(NA,nrow=nbins,ncol=nsubs)
iqr.sd.mat = matrix(NA,nrow=nbins,ncol=nsubs)

# This loop will also create a matrix of velocities (vbins.mat) for each subject that will
 complement the following
# plots. The matrix will allow us to calculate the percent of velocities in each bin for each
 subject by tabulating the total
# number of velocities recorded that fall each bin.
vbins.mat = matrix(0,nrow=nbins,ncol=nsubs)
vLbins.mat = matrix(0,nrow=nbins,ncol=nsubs)
vRbins.mat = matrix(0,nrow=nbins,ncol=nsubs)


for (i in 1:dim(iqr.mat)[1]){
  for (j in 1:dim(iqr.mat)[2]){
    index = which(abs(vM[,j]) > 0.4+(i-1)*0.2 & abs(vM[,j]) <= i*0.2+0.4 & is.nan(abs(vM[,j])) ==
 0)
    vbins.mat[i,j] = length(index)
    if (length(index) >= 10){
      iqr.mat[i,j] = IQR(na.omit(tEM[index,j]))
      v.mat[i,j] = mean(abs(vM[index,j]))
      iqr.sd.mat[i,j] = sd(na.omit(tEM[index,j]))/length(index)
    }
  }
}

vbins.mat = vbins.mat/(1120/2) #turns into a proportion of all velocities, 1120/2 is number of
 manual trials per subject

par(mfrow=c(3,4))
for (i in 1:10){
  barplot(vbins.mat[,i],width = 0.2, xlim = c(0,1.8),ylim = c(0,0.5),xpd=F,axes=T,names.arg =
 c('0.4-0.6','0.6-0.8'
                                                                                                ,
 '0.8-1.0','1.0-1.2','1.2-1.4','1.4-1.6','1.6-1.8'),cex.names = 0.7)
}

par(mfrow = c(1,1))

for (i in 1:dim(iqr.mat)[1]){
  for (j in 1:dim(iqr.mat)[2]){
    index = which(abs(vLM[,j]) > 0.4+(i-1)*0.2 & abs(vLM[,j]) <= i*0.2+0.4 & is.nan(abs(vLM[,j]))
 == 0)
    vLbins.mat[i,j] = length(index)
  }
}
vLbins.mat = vLbins.mat/(1120/4)

for (i in 1:dim(iqr.mat)[1]){
  for (j in 1:dim(iqr.mat)[2]){
    index = which(abs(vRM[,j]) > 0.4+(i-1)*0.2 & abs(vRM[,j]) <= i*0.2+0.4 & is.nan(abs(vRM[,j]))
 == 0)
    vRbins.mat[i,j] = length(index)
  }
}
vRbins.mat = vRbins.mat/(1120/4)

par(mfrow=c(5,4))
for (i in 1:10){
  barplot(vRbins.mat[,i],width = 0.2, main = 'Right', xlim = c(0,1.8),ylim =
 c(0,0.5),xpd=F,axes=T,names.arg = c('0.4-0.6','0.6-0.8'
                                                    ,'0.8-1.0','1.0-1.2','1.2-1.4','1.4-
 1.6','1.6-1.8'),cex.names = 0.7)
  barplot(vLbins.mat[,i],width = 0.2, main = 'Left', xlim = c(0,1.8),ylim =
 c(0,0.5),xpd=F,axes=T,names.arg = c('0.4-0.6','0.6-0.8'
                                                    ,'0.8-1.0','1.0-1.2','1.2-1.4','1.4-
 1.6','1.6-1.8'),cex.names = 0.7)
}

par(mfrow = c(1,1))

# For individual subjects, all manual passes
plot(NA,xlim = c(0.4,1.8),ylim = c(0,100),xlab = 'Velocity of Pass at Kick (pix/ms)',ylab = 'Mean
 IQR of Time Error')
for (j in 1:dim(iqr.mat)[2]){
  lines(v.mat[,j],iqr.mat[,j])
}

# For subjects overall, all manual passes
# plot(NA,xlim = c(0.4,1.8),ylim = c(0,100))
x = rep(NA,nbins)
y = rep(NA,nbins)
se = rep(NA,nbins)
for (i in 1:dim(iqr.mat)[1]){
  x[i] = mean(na.omit(v.mat[i,]))
  y[i] = mean(na.omit(iqr.mat[i,]))
  #points(x[i],y[i],pch = 20)
  se[i] = sd(na.omit(iqr.mat[i,]))/length(na.omit(iqr.mat[i,]))
}
errbar(x,y,y+se,y-se,xlab = 'Velocity of Pass at Kick (pix/ms)',ylab = 'Mean IQR of Time Error +-
 SE')
abline(lm(y~x))

# Faster velocities seem to be INCREASING the precision of the kick! Better yet, the constant
# variance assumption for a regression analysis looks to be relatively well met. Let's run a
 regression
# analysis to see if this effect is significant.

# summary(lm(na.omit(as.vector(iqr.mat))~na.omit(as.vector(v.mat))))
# INDEPENDENCE issue just use y~x
summary(lm(y~x))


# Let's now run the difference in IQRs instead. We will not throw out the data from subject
 number 9, as
# he is left handed and could exhibit a bias toward the left hand in time error, but we can't say
 for sure.

nsubs = 10
nbins = 7
iqrL.mat = matrix(NA,nrow=nbins,ncol=nsubs)
vL.mat = matrix(NA,nrow=nbins,ncol=nsubs)
iqrL.sd.mat = matrix(NA,nrow=nbins,ncol=nsubs)
iqrR.mat = matrix(NA,nrow=nbins,ncol=nsubs)
vR.mat = matrix(NA,nrow=nbins,ncol=nsubs)
iqrR.sd.mat = matrix(NA,nrow=nbins,ncol=nsubs)

for (i in 1:dim(iqrL.mat)[1]){
  for (j in 1:dim(iqrL.mat)[2]){
    index = which(abs(vLM[,j]) > 0.4+(i-1)*0.2 & abs(vLM[,j]) <= i*0.2+0.4 & is.nan(abs(vLM[,j]))
 == 0)
    if (length(index) >= 10){
      iqrL.mat[i,j] = IQR(na.omit(tELM[index,j]))
      vL.mat[i,j] = mean(abs(vLM[index,j]))
      iqrL.sd.mat[i,j] = sd(na.omit(tELM[index,j]))/length(index)
    }
    index = which(abs(vRM[,j]) > 0.4+(i-1)*0.2 & abs(vRM[,j]) <= i*0.2+0.4 & is.nan(abs(vRM[,j]))
 == 0)
    if (length(index) >= 10){
      iqrR.mat[i,j] = IQR(na.omit(tERM[index,j]))
      vR.mat[i,j] = mean(abs(vRM[index,j]))
      iqrR.sd.mat[i,j] = sd(na.omit(tERM[index,j]))/length(index)
    }
  }
}

# Difference Matrices
vD.mat = (vR.mat + vL.mat)/2
iqrD.mat = iqrR.mat - iqrL.mat
# Re-rerun analysis with these new matrices


# For individual subjects, right versus left manual passes
plot(NA,xlim = c(0.4,1.8),ylim = c(-50,50),xlab = 'Velocity of Pass at Kick (pix/ms)',
     ylab = 'Right versus Left Difference in Mean IQR of Time Error')
for (j in 1:dim(iqrD.mat)[2]){
  lines(vD.mat[,j],iqrD.mat[,j])
}

# For subjects overall, right versus left manual passes
# plot(NA,xlim = c(0.4,1.8),ylim = c(0,100))
x = rep(NA,nbins)
y = rep(NA,nbins)
se = rep(NA,nbins)
for (i in 1:dim(iqrD.mat)[1]){
  x[i] = mean(na.omit(vD.mat[i,]))
  y[i] = mean(na.omit(iqrD.mat[i,]))
  # points(x[i],y[i],pch = 20)
  se[i] = sd(na.omit(iqrD.mat[i,]))/length(na.omit(iqrD.mat[i,]))
}
errbar(x,y,y+se,y-se,xlab = 'Velocity of Pass at Kick (pix/ms)',
       ylab = 'Right versus Left Difference in Mean IQR of Time Error +- SE')
points(0.9,IQR(na.omit(c(data.matrix(dtERA))))-IQR(na.omit(c(data.matrix(dtELA)))),col =
 'red',pch=20)
abline(lm(y~x))

# summary(lm(na.omit(as.vector(iqrD.mat[1:7,]))~na.omit(as.vector(vD.mat[1:7,]))))
summary(lm(y~x))

######################################################
# Right and Left Time Error Separate on the same graph
######################################################

plot(NA,xlim = c(0.4,1.8),ylim = c(25,55),xlab = 'Velocity of Pass at Kick (pix/ms)',
     ylab = 'Mean IQR of Time Error +- SE')
xR = rep(NA,nbins)
yR = rep(NA,nbins)
seR = rep(NA,nbins)
for (i in 1:dim(iqrL.mat)[1]){
  xR[i] = mean(na.omit(vR.mat[i,]))
  yR[i] = mean(na.omit(iqrR.mat[i,]))
  # points(xR[i],yR[i],pch = 20,col='red')
  seR[i] = sd(na.omit(iqrR.mat[i,]))/length(na.omit(iqrR.mat[i,]))
}
errbar(xR,yR,yR+seR,yR-seR,xlab = 'Velocity of Pass at Kick (pix/ms)',
       ylab = 'Right Mean IQR of Time Error +- SE',add=T,col = 'red')
#points(0.9,IQR(na.omit(c(data.matrix(dtERA)))),col = 'red',pch=20)
#abline(lm(y~x))
yAR = rep(NA,length(dtERA))
sdAR = rep(NA,length(dtERA))
for (i in 1:length(dtELA)){
  yAR[i] = IQR(na.omit(dtERA[,i]))
  sdAR[i] = sd(na.omit(dtERA[,i]))
}
seAR = sd(yAR)/length(yAR)
```

```
# yAL = IQR(na.omit(c(data.matrix(dtELA))))
# sdAL = sd(na.omit(c(data.matrix(dtELA))))/10
# points(0.9,mean(yAR),col = 'orange',pch=20)
errbar(0.9,mean(yAR),mean(yAR)+seAR,mean(yAR)-seAR,add=T,errbar.col='red')

xL = rep(NA,nbins)
yL = rep(NA,nbins)
seL = rep(NA,nbins)
for (i in 1:dim(iqrL.mat)[1]){
  xL[i] = mean(na.omit(vL.mat[i,]))
  yL[i] = mean(na.omit(iqrL.mat[i,]))
  points(xL[i],yL[i],pch = 20,col='blue')
  seL[i] = sd(na.omit(iqrL.mat[i,]))/length(na.omit(iqrL.mat[i,]))
}
errbar(xL,yL,yL+seL,yL-seL,add=T,errbar.col='blue')
yAL = rep(NA,length(dtELA))
sdAL = rep(NA,length(dtELA))
for (i in 1:length(dtELA)){
  yAL[i] = IQR(na.omit(dtELA[,i]))
  sdAL[i] = sd(na.omit(dtELA[,i]))
}
seAL = sd(yAL)/length(yAL)
# yAL = IQR(na.omit(c(data.matrix(dtELA))))
# sdAL = sd(na.omit(c(data.matrix(dtELA))))/10
# points(0.9,mean(yAL),col = 'green',pch=20)
errbar(0.9,mean(yAL),mean(yAL)+seAL,mean(yAL)-seAL,add=T,errbar.col='blue')
# abline(lm(y~x))
legend(1.2,50,c('Red - Right Hand Pass','Blue - Left Hand Pass'))


# It seems faster velocities tend to produce an increased left hand precision and slower
# velocities produce a weaker increase in right hand precision! Weird! Let's run a regression
# analysis to check if this is really something significant. Looks like the constant variance
 assumption
# doesn't hold well for the lowest velocity bin.


# Permutation test for each velocity bin

pval.vec = rep(NA,dim(iqrL.mat)[1])
nsims = 1000
for (i in 1:dim(iqrL.mat)[1]){
  L = na.omit(iqrL.mat[i,])
  R = na.omit(iqrR.mat[i,])
  d = abs(mean(L)-mean(R))
  diffs = rep(NA,nsims)
  for (j in 1:nsims){
    mix = sample(c(L,R))
    Lp = mix[1:length(L)]
    Rp = mix[(length(L)+1):(length(L)+length(R))]
    diffs[j] = abs(mean(Lp)-mean(Rp))
  }
  pval.vec[i] = mean(diffs>d)
}

##############################################
# Handedness Differences with X position Error
##############################################

nsubs = 10
nbins = 7
iqr.mat = matrix(NA,nrow=nbins,ncol=nsubs)
v.mat = matrix(NA,nrow=nbins,ncol=nsubs)
iqr.sd.mat = matrix(NA,nrow=nbins,ncol=nsubs)

for (i in 1:dim(iqr.mat)[1]){
  for (j in 1:dim(iqr.mat)[2]){
    index = which(abs(vM[,j]) > 0.4+(i-1)*0.2 & abs(vM[,j]) <= i*0.2+0.4 & is.nan(abs(vM[,j])) ==
 0)
    if (length(index) >= 10){
      iqr.mat[i,j] = IQR(na.omit(xEM[index,j]))
      v.mat[i,j] = mean(abs(vM[index,j]))
      iqr.sd.mat[i,j] = sd(na.omit(xEM[index,j]))/length(index)
    }
  }
}

# For individual subjects, all manual passes
plot(NA,xlim = c(0.4,1.8),ylim = c(0,100),xlab = 'Velocity of Pass at Kick (pix/ms)',ylab = 'Mean
 IQR of Position Error')
for (j in 1:dim(iqr.mat)[2]){
  lines(v.mat[,j],iqr.mat[,j])
}

# For subjects overall, all manual passes
# plot(NA,xlim = c(0.4,1.8),ylim = c(0,100))
x = rep(NA,nbins)
y = rep(NA,nbins)
se = rep(NA,nbins)
for (i in 1:dim(iqr.mat)[1]){
  x[i] = mean(na.omit(v.mat[i,]))
  y[i] = mean(na.omit(iqr.mat[i,]))
  # points(x[i],y[i],pch = 20)
  se[i] = sd(na.omit(iqr.mat[i,]))/length(na.omit(iqr.mat[i,]))
}
errbar(x,y,y+se,y-se,xlab = 'Velocity of Pass at Kick (pix/ms)',ylab = 'Mean IQR of Position
 Error +- SE')
abline(lm(y~x))

# Faster velocities seem to be INCREASING the precision of the kick! Better yet, the constant
# variance assumption for a regression analysis looks to be relatively well met. Let's run a
 regression
# analysis to see if this effect is significant.

summary(lm(y~x))

# Let's now run the difference in IQRs instead. We will not throw out the data from subject
 number 9, as
# he is left handed and could exhibit a bias toward the left hand in time error, but we can't say
 for sure.

nsubs = 10
nbins = 7
iqrL.mat = matrix(NA,nrow=nbins,ncol=nsubs)
vL.mat = matrix(NA,nrow=nbins,ncol=nsubs)
iqrL.sd.mat = matrix(NA,nrow=nbins,ncol=nsubs)
iqrR.mat = matrix(NA,nrow=nbins,ncol=nsubs)
vR.mat = matrix(NA,nrow=nbins,ncol=nsubs)
iqrR.sd.mat = matrix(NA,nrow=nbins,ncol=nsubs)

for (i in 1:dim(iqrL.mat)[1]){
  for (j in 1:dim(iqrL.mat)[2]){
    index = which(abs(vLM[,j]) > 0.4+(i-1)*0.2 & abs(vLM[,j]) <= i*0.2+0.4 & is.nan(abs(vLM[,j]))
 == 0)
    if (length(index) >= 10){
      iqrL.mat[i,j] = IQR(na.omit(xELM[index,j]))
      vL.mat[i,j] = mean(abs(vLM[index,j]))
      iqrL.sd.mat[i,j] = sd(na.omit(xELM[index,j]))/length(index)
    }
    index = which(abs(vRM[,j]) > 0.4+(i-1)*0.2 & abs(vRM[,j]) <= i*0.2+0.4 & is.nan(abs(vRM[,j]))
 == 0)
    if (length(index) >= 10){
      iqrR.mat[i,j] = IQR(na.omit(xERM[index,j]))
      vR.mat[i,j] = mean(abs(vRM[index,j]))
      iqrR.sd.mat[i,j] = sd(na.omit(xERM[index,j]))/length(index)
    }
  }
}

# Difference Matrices
vD.mat = (vR.mat + vL.mat)/2
iqrD.mat = iqrR.mat - iqrL.mat
# Re-rerun analysis with these new matrices


# For individual subjects, right versus left manual passes
```

```r
plot(NA,xlim = c(0.4,1.8),ylim = c(-50,50),xlab = 'Velocity of Pass at Kick (pix/ms)',
     ylab = 'Right versus Left Difference in Mean IQR of Position Error')
for (j in 1:dim(iqrD.mat)[2]){
  lines(vD.mat[,j],iqrD.mat[,j])
}

# For subjects overall, right versus left manual passes
# plot(NA,xlim = c(0.4,1.8),ylim = c(0,100))
x = rep(NA,nbins)
y = rep(NA,nbins)
se = rep(NA,nbins)
for (i in 1:dim(iqrD.mat)[1]){
  x[i] = mean(na.omit(vD.mat[i,]))
  y[i] = mean(na.omit(iqrD.mat[i,]))
  # points(x[i],y[i],pch = 20)
  se[i] = sd(na.omit(iqrD.mat[i,]))/length(na.omit(iqrD.mat[i,]))
}
errbar(x,y,y+se,y-se,xlab = 'Velocity of Pass at Kick (pix/ms)',
       ylab = 'Right versus Left Difference in Mean IQR of Position Error +- SE')
points(0.9,IQR(na.omit(c(data.matrix(dxERA))))-IQR(na.omit(c(data.matrix(dxELA)))),col =
 'red',pch=20)
abline(y~x)

# It seems faster velocities tend to produce an increased left hand precision and slower
# velocities produce a weaker increase in right hand precision! Weird! Let's run a regression
# analysis to check if this is really something significant.

summary(lm(y~x))


##########################################################
# Right and Left Position Error Separate on the same graph
##########################################################

plot(NA,xlim = c(0.4,1.8),ylim = c(20,55),xlab = 'Velocity of Pass at Kick (pix/ms)',
     ylab = 'Mean IQR of Position Error +- SE')
xR = rep(NA,nbins)
yR = rep(NA,nbins)
seR = rep(NA,nbins)
for (i in 1:dim(iqrL.mat)[1]){
  xR[i] = mean(na.omit(vR.mat[i,]))
  yR[i] = mean(na.omit(iqrR.mat[i,]))
  # points(xR[i],yR[i],pch = 20,col='red')
  seR[i] = sd(na.omit(iqrR.mat[i,]))/length(na.omit(iqrR.mat[i,]))
}
errbar(xR,yR,yR+seR,yR-seR,xlab = 'Velocity of Pass at Kick (pix/ms)',
       ylab = 'Right Mean IQR of Position Error +- SE',add=T,col = 'red')
# points(0.9,IQR(na.omit(c(data.matrix(dtERA)))),col = 'red',pch=20)
# abline(lm(y~x))
yAR = rep(NA,length(dxERA))
sdAR = rep(NA,length(dxERA))
for (i in 1:length(dxELA)){
  yAR[i] = IQR(na.omit(dxERA[,i]))
  sdAR[i] = sd(na.omit(dxERA[,i]))
}
seAR = sd(yAR)/length(yAR)
# yAL = IQR(na.omit(c(data.matrix(dtELA))))
# sdAL = sd(na.omit(c(data.matrix(dtELA))))/10
# points(0.9,mean(yAR),col = 'orange',pch=20)
errbar(0.9,mean(yAR),mean(yAR)+seAR,mean(yAR)-seAR,add=T,errbar.col='orange')

xL = rep(NA,nbins)
yL = rep(NA,nbins)
seL = rep(NA,nbins)
for (i in 1:dim(iqrL.mat)[1]){
  xL[i] = mean(na.omit(vL.mat[i,]))
  yL[i] = mean(na.omit(iqrL.mat[i,]))
  points(xL[i],yL[i],pch = 20,col='blue')
  seL[i] = sd(na.omit(iqrL.mat[i,]))/length(na.omit(iqrL.mat[i,]))
}
errbar(xL,yL,yL+seL,yL-seL,add=T,errbar.col='blue')
yAL = rep(NA,length(dxELA))
sdAL = rep(NA,length(dxELA))
```

```
for (i in 1:length(dxELA)){
  yAL[i] = IQR(na.omit(dxELA[,i]))
  sdAL[i] = sd(na.omit(dxELA[,i]))
}
seAL = sd(yAL)/length(yAL)
# yAL = IQR(na.omit(c(data.matrix(dtELA))))
# sdAL = sd(na.omit(c(data.matrix(dtELA))))/10
# points(0.9,mean(yAL),col = 'green',pch=20)
errbar(0.9,mean(yAL),mean(yAL)+seAL,mean(yAL)-seAL,add=T,errbar.col='green')
# abline(lm(y~x))
legend(0.6,50,c('Red - Right Hand Pass','Blue - Left Hand Pass'))

# Permutation test for each bin

pval.vec = rep(NA,dim(iqrL.mat)[1])
nsims = 1000
for (i in 1:dim(iqrL.mat)[1]){
  L = na.omit(iqrL.mat[i,])
  R = na.omit(iqrR.mat[i,])
  d = abs(mean(L)-mean(R))
  diffs = rep(NA,nsims)
  for (j in 1:nsims){
    mix = sample(c(L,R))
    Lp = mix[1:length(L)]
    Rp = mix[(length(L)+1):(length(L)+length(R))]
    diffs[j] = abs(mean(Lp)-mean(Rp))
  }
  pval.vec[i] = mean(diffs>d)
}

################################################################
# Goal prediction based on demographic and basic experiment data
################################################################

# Predicting Goal based on direction of pass, auto/manual

goals = read.csv("goals")                               #0 for miss, 1 for goal
pass_direction = read.csv("pass_direction")             #0 for left, 1 for right
trial_type = read.csv("trial_type")                     #2 for manual, 3 for auto

glm.goals = c(data.matrix(goals))
glm.pass = c(data.matrix(pass_direction))
glm.trial = c(data.matrix(trial_type))
id=c(rep(1,1120),rep(2,1120),rep(3,1120),rep(4,1120),rep(5,1120),
     rep(6,1120),rep(7,1120),rep(8,1120),rep(9,1120),rep(10,1120))

GLM.model = glm(glm.goals ~ glm.pass + glm.trial, family = 'binomial')
summary(GLM.model)
GLMER.model = glmer(glm.goals ~ glm.pass + glm.trial + (1|id), family = 'binomial')
summary(GLMER.model)
```