```
# Author: Keith R. Lohse, PhD, PStat
# Date: 2020-05-22
# Title: Supplemental Appendix i: Simulation Code

# A brief illustration of mathematical coupling -------------------------------
# Mathematical Coupling with normal data
set.seed(1)
x<-rnorm(1000, 0, 1)
y<-rnorm(1000, 0, 1)
z<-y-x

cor(x,y)
summary(lm(y~x))
plot(x,y, ylim=c(-5,5), xlim=c(-5,5),cex.lab=1.5, cex.axis=1.5)
abline(a=-0.016187, b=-0.00643, lty=2, col="red")
# z~x
# (y-x)~x
# (-x+y)~X

cor(x,z)
summary(lm(z~x))
plot(x,z, ylim=c(-5,5), xlim=c(-5,5),
     ylab="z (y-x)", cex.lab=1.5, cex.axis=1.5)
abline(a=-0.016, b=-0.99357, lty=2, col="red")

# Mathematical Coupling with Uniform Data data
set.seed(1)
x<-runif(1000, -5, 5)
y<-runif(1000, -5, 5)
z<-y-x

plot(x,y, ylim=c(-10,10), xlim=c(-10,10),cex.lab=1.5, cex.axis=1.5)
abline(a=-0.09713, b=-0.02923, lty=2, col="red")
cor(x,y)
summary(lm(y~x))
# z~x
# (y-x)~x
# (-x+y)~X

plot(x,z, ylim=c(-10,10), xlim=c(-10,10),
     ylab="z (y-x)",cex.lab=1.5, cex.axis=1.5)
abline(a=-0.09713, b=-1.02923, lty=2, col="red")

cor(x,z)
summary(lm(z~x))



# Simulating two different types of recovery ----------------------------------
# Figure 3A Proportional recovery ----
# Set the upper limit on the scale you want to create
UL<-66

# Set the lower limit
LL <- 0

# Set the Population size
```

```r
N = 100

# Create a population of uniform scores from
set.seed(1)
II<-round(UL-runif(N, min=LL, max=UL),digits=0)
summary(II)
CHANGE<-rep(NA, N)
FINAL<-rep(NA, N)

POP<-data.frame(II, CHANGE, FINAL)

set.seed(1)
for (i in c(1:N)){
  POP$CHANGE[i] <- round(0.5*(POP$II[i])+rnorm(n=1,mean=0,sd=6), digits=0)
  POP$CHANGE[i] <- ifelse((66-POP$II[i])+POP$CHANGE[i]>66, 66-(66-POP$II[i]),
POP$CHANGE[i])
  POP$FINAL[i] <- (66-POP$II[i])+POP$CHANGE[i]
  print(i)
}

summary(POP$CHANGE)
summary(POP$FINAL)
pop_mod<-lm(CHANGE~II, data = POP)
summary(pop_mod)
cor(POP$II, POP$CHANGE)

plot(x=POP$II, y=POP$CHANGE, ylab="Change (Final - Baseline FMA)",
     xlab="Initial Impairment (66 - Baseline FMA)",
     xlim=c(0, 66), ylim=c(-10,66), cex.lab=1.5, cex.axis=1.5)
abline(a=0.165, b=0.502, lty=2, lwd=3, col="red")
abline(a=0, b=1, lty=1, lwd=1, col="black")


# Figure 3B Uniform recovery ----
# Set the upper limit on the scale you want to create
UL<-66

# Set the lower limit
LL <- 0

# Set the Population size
N = 100

# Create a population of uniform scores from
set.seed(1)
II<-round(UL-runif(N, min=LL, max=UL),digits=0)
summary(II)
CHANGE<-rep(NA, N)
FINAL<-rep(NA, N)

POP<-data.frame(II, CHANGE, FINAL)

set.seed(3)
for (i in c(1:N)){
  POP$CHANGE[i] <- round(runif(1, min=-5, max=(0+POP$II[i])), digits=0)
  POP$FINAL[i] <- (66-POP$II[i])+POP$CHANGE[i]
  print(i)
```

```
}

summary(POP$FINAL)
pop_mod<-lm(CHANGE~II, data = POP)
summary(pop_mod)
cor(POP$II, POP$CHANGE)

plot(x=POP$II, y=POP$CHANGE, ylab="Change (Final - Baseline FMA)",
     xlab="Initial Impairment (66 - Baseline FMA)",
     xlim=c(0, 66), ylim=c(-10,66), cex.lab=1.5, cex.axis=1.5)
abline(a=-3.24, b=0.5, lty=2, lwd=3, col="red")
abline(a=0, b=1, lty=1, lwd=1, col="black")


# Simulating over the Top of Empirical Data ------------------------------------

getwd()
# Set the working directory to the appropriate location for the data file
# Then run the following code:
REAL<-read.csv("./data_EMPIRICAL.csv", header=TRUE)
head(REAL)
hist(REAL$II)
hist(REAL$CHANGE)

# Plotting all of the data
plot(x=REAL$II, y=REAL$CHANGE, ylab="Change (Final - Baseline FMA)",
     xlab="Initial Impairment (66 - Baseline FMA)",
     pch=1, col=REAL$Group,
     xlim=c(0, 66), ylim=c(-5,66), cex.lab=1.5, cex.axis=1.5)
summary(lm(CHANGE~0+II, data=REAL))
abline(a=0, b=0.41865, lty=2, lwd=3, col="red")

# Minimum Effects Test against 0.50:

pt(q=(0.41865-0.50)/0.01767, df=(373-2),
   lower.tail = TRUE)*2

# Plotting data with non-fitters removed
FITTERS <- subset(REAL, Group == "fitters")

# Plotting all of the data
plot(x=FITTERS$II, y=FITTERS$CHANGE, ylab="Change (Final - Baseline FMA)",
     xlab="Initial Impairment (66 - Baseline FMA)",
     xlim=c(0, 66), ylim=c(-0,66), cex.lab=1.5, cex.axis=1.5)
summary(lm(CHANGE~0+II, data=FITTERS))
abline(a=0, b=0.76921, lty=2, lwd=3, col="red")


# Simulating Random Change Scores ----------------------------------------------
# Set the Population size
N = 10000

# Create a population of uniform scores from
II<-rep(NA,N)
CHANGE<-rep(NA, N)
FINAL<-rep(NA, N)
```

```
SIM_POP<-data.frame(II, CHANGE, FINAL)

set.seed(1)
for (i in c(1:N)){
  # Randomly samples from the real distribution of II
  x<-sample(REAL$II, 1, replace=TRUE)

  # Avoid going over the boundaries
  ifelse(x==0,
         SIM_POP$II[i]<-0, # Returns if above is TRUE
         ifelse(x==66, # Returns if the above is FALSE
                SIM_POP$II[i]<-66, # Returns if the above is TRUE
                SIM_POP$II[i]<-x+round(runif(1,-1,1), digits=0)))
  # Generates a random change score somewhere between II and the test max
  SIM_POP$CHANGE[i] <- round(runif(1, min=0, max=(0+SIM_POP$II[i])), digits=0)
  # Add the change score to the baseline score to get a final score
  SIM_POP$FINAL[i] <- (66-SIM_POP$II[i])+SIM_POP$CHANGE[i]
  print(i)
}


# Contrast our simulated density of II with the REAL density
hist(REAL$II, main="Distribution of Real Data",
     xlab="Initial Impairment (66 - Baseline FMA)",
     ylab="Count", cex.lab=1.5, cex.axis=1.5)
hist(SIM_POP$II, main="Distribution of Simulated Data",
     xlab="Initial Impairment (66 - Baseline FMA)",
     ylab="Count", cex.lab=1.5, cex.axis=1.5)

pop_mod<-lm(CHANGE~0+II, data = SIM_POP)
summary(pop_mod)
cor(SIM_POP$II, SIM_POP$CHANGE)

plot(x=SIM_POP$II, y=SIM_POP$CHANGE,
     ylab="Change (Final - Baseline FMA)",
     xlab="Initial Impairment (66 - Baseline FMA)",
     xlim=c(0, 66), ylim=c(-0,66), cex.lab=1.5, cex.axis=1.5)
abline(a=0, b=0.496501, lty=2, lwd=3, col="red")


# Sampling from the Empirical Data
k_samples <- 10000
sample_size<-30

INT<-rep(NA, k_samples)
SLOPE<-rep(NA, k_samples)

SAMPLES <-data.frame(INT, SLOPE)

set.seed(3)
for (i in c(1:k_samples)){
  samp<-SIM_POP[sample(nrow(SIM_POP), sample_size),]
  x<-lm(CHANGE~II, data = samp)
  SAMPLES$INT[i] <- x$coefficients[1]
  SAMPLES$SLOPE[i] <- x$coefficients[2]
  print(i)
}
```

```
set.seed(5)
samp1<-SIM_POP[sample(nrow(SIM_POP), sample_size),]
head(samp1)

mod01<-lm(CHANGE~II, data=samp1)
plot(mod01)
summary(mod01)
plot(samp1$II, samp1$CHANGE, main=paste("Random Sample of N =", sample_size),
     ylab="Change (Final - Baseline FMA)",
     xlab="Initial Impairment (66 - Baseline FIM)",
     xlim=c(0, 66), ylim=c(0,66), cex.axis=1.5, cex.lab=1.5)
abline(a=-1.0036, b=0.6154, lty=2, lwd=2, col="red")
abline(a=0, b=1, lty=1, lwd=1, col="black")


plot(density(SAMPLES$SLOPE), col="royal blue", lwd=2, lty=2,
     main=paste("Distribution of Sample Slopes, N =", sample_size),
     xlab="Observed Slope",
     xlim=c(-1,1), cex.lab=1.5, cex.axis=1.5)

# Calculate the standard error
SE<-sd(SAMPLES$SLOPE)
alt_dist<-dnorm(c(seq(from=0, to=1, by=0.01)), mean=0.5, sd=SE)
null_dist<-dnorm(c(seq(from=-0.5, to=0.5, by=0.01)), mean=0, sd=SE)
plot(c(seq(from=0, to=1, by=0.01)),
     alt_dist,
     type = "l", col="royal blue", lwd=2, lty=2,
     main=paste("Distribution of Sample Slopes, N =", sample_size),
     xlab="Observed Slope",
     ylab="Density",
     xlim=c(-1,1), cex.lab=1.5, cex.axis=1.5)
lines(c(seq(from=-0.5, to=0.5, by=0.01)),
      null_dist, col="black", lty=2, lwd=2)
abline(v=0.615, col="red", lwd=1)


# Incorrectly doing a one sample t-test against zero:
pt(q=(0.615-0)/0.132, df=(sample_size-2),
   lower.tail = FALSE)*2

# Correctly doing a one sample t-test against the mean of the
# sampling distribution, approximately 0.5
pt(q=(0.615-mean(SAMPLES$SLOPE))/0.132, df=(sample_size-2),
   lower.tail = FALSE)*2



# Plotting a biased distribution wih non-fitters excluded ---------------
# Using Hierarchical Cluster Analysis to Identify Non-Fitters
head(samp1)
samp1$pred<-0.70*samp1$II
samp1$resid<-samp1$CHANGE-samp1$pred
head(samp1)

# Install the ecodist package to get access to mahalanobis distances.
library(ecodist)
```

```r
?ecodist::distance
clusters <- hclust(distance(samp1[, c("CHANGE", "pred")], method="mahalanobis"),
                   method="ward.D2") # In Winters et al. 2015, their reference
suggests Ward 1963
plot(clusters)
cutree(clusters, 2)
samp1$cluster<-cutree(clusters, 2)
samp1


# Note that we arrive at similar outlier exclusions even though the
# population data were random
plot(samp1$II, samp1$CHANGE, main=paste("Random Sample of N =", sample_size),
     ylab="Change (Final - Baseline FMA)",
     xlab="Initial Impairment (66 - Baseline FIM)",
     pch=21, col=samp1$cluster,
     xlim=c(0, 66), ylim=c(0,66), cex.axis=1.5, cex.lab=1.5)

summary(lm(CHANGE~II, data=samp1[samp1$cluster==2,]))
summary(lm(CHANGE~II, data=samp1))

# Biased sampling distribution
sample_size <- 30
k_samples <- 50

INT_TOTAL<-rep(NA, k_samples)
SLOPE_TOTAL<-rep(NA, k_samples)

BIASED <-data.frame(INT_TOTAL, SLOPE_TOTAL)

set.seed(1)
for (i in c(1:k_samples)){
  # Obtain a sample
  samp<-SIM_POP[sample(nrow(SIM_POP), sample_size),]

  # Get the total slopes and intercepts
  tot_mod <-lm(CHANGE~II, data=samp)
  BIASED$INT_TOTAL[i] <- tot_mod$coefficients[1]
  BIASED$SLOPE_TOTAL[i] <- tot_mod$coefficients[2]

  # Get the residuals and fitted values for your model
  samp$pred<-0.70*samp$II
  samp$resid<-samp$CHANGE-samp$pred

  # Get your clusters
  clusters <- hclust(distance(samp[, c("CHANGE", "pred")],
method="mahalanobis"),
                     method="ward.D2")
  samp$cluster<-cutree(clusters, 2)

  # Write the results slopes and intercepts
  # Cluster 1
  BIASED$C1_INT_BIAS[i] <- lm(CHANGE~II, data =
samp[samp$cluster==1,])$coefficients[1]
  BIASED$C1_SLOPE_BIAS[i] <- lm(CHANGE~II, data =
samp[samp$cluster==1,])$coefficients[2]
  BIASED$C1_size[i]<-nrow(samp[samp$cluster==1,])
```

```
  BIASED$C1_min_II[i]<-min(samp$II[samp$cluster==1])
  BIASED$C1_max_II[i]<-max(samp$II[samp$cluster==1])
  BIASED$C1_mean_CHANGE[i]<-mean(samp$CHANGE[samp$cluster==1])


  # Cluster 2
  BIASED$C2_INT_BIAS[i] <- lm(CHANGE~II, data =
samp[samp$cluster==2,])$coefficients[1]
  BIASED$C2_SLOPE_BIAS[i] <- lm(CHANGE~II, data =
samp[samp$cluster==2,])$coefficients[2]
  BIASED$C2_size[i]<-nrow(samp[samp$cluster==2,])
  BIASED$C2_min_II[i]<-min(samp$II[samp$cluster==2])
  BIASED$C2_max_II[i]<-max(samp$II[samp$cluster==2])
  BIASED$C2_mean_CHANGE[i]<-mean(samp$CHANGE[samp$cluster==2])

  # Change the # of samples and uncomment these lines to get images of
classification
  # png(paste("sample", i, ".png"))
  # # Create a plot
  # plot(samp$II, samp$CHANGE, main=paste("Random Sample of N =", sample_size),
  #    ylab="Change (Final - Baseline FMA)",
  #    xlab="Initial Impairment (66 - Baseline FIM)",
  #    pch=21, col=samp$cluster,
  #    xlim=c(0, 66), ylim=c(0,66), cex.axis=1.5, cex.lab=1.5)
  #    # Close the pdf file
  # dev.off()
  #
  # print(i)

}

head(BIASED)
getwd()

# Problematically, the cluster analysis does not identify responders and
# non-responders:
summary(BIASED$C1_SLOPE_BIAS)
plot(density(BIASED$C1_SLOPE_BIAS, na.rm=TRUE), col="navy", lwd=2, lty=2,
     main=paste("Distribution of Sample Slopes, N =", sample_size),
     xlab="Observed Slope",
     xlim=c(-2,2), cex.lab=1.5, cex.axis=1.5)
abline(v=0.7, col="red", lwd=1)

summary(BIASED$C2_SLOPE_BIAS)
plot(density(BIASED$C2_SLOPE_BIAS, na.rm=TRUE), col="navy", lwd=2, lty=2,
     main=paste("Distribution of Sample Slopes, N =", sample_size),
     xlab="Observed Slope",
     xlim=c(-2,2), cex.lab=1.5, cex.axis=1.5)
abline(v=0.7, col="red", lwd=1)

plot(y=BIASED$C1_SLOPE_BIAS, x=BIASED$C1_mean_CHANGE,
     col="black", lwd=2,
     cex.lab=1.5, cex.axis=1.5)

plot(y=BIASED$C1_SLOPE_BIAS, x=BIASED$C1_size,
     col="black", lwd=2,
     cex.lab=1.5, cex.axis=1.5)
```

```r
# Processing the data to remove poorly obtained clusters
# Based on Size
BIASED_FILTER<-subset(BIASED,
                      C1_size >=(0.05*sample_size) & C2_size >=
(0.05*sample_size))
# Based on Initial Impairment
BIASED_FILTER<-subset(BIASED_FILTER,
                      C1_min_II < C2_max_II & C2_min_II < C1_max_II)

plot(y=BIASED_FILTER$C1_SLOPE_BIAS, x=BIASED_FILTER$C1_size,
     col="black", lwd=2,
     cex.lab=1.5, cex.axis=1.5)

plot(y=BIASED_FILTER$C2_mean_CHANGE, x=BIASED_FILTER$C1_mean_CHANGE,
     col="black", lwd=2,
     cex.lab=1.5, cex.axis=1.5)

plot(density(BIASED_FILTER$C1_SLOPE_BIAS, na.rm=TRUE), col="navy", lwd=2,
lty=2,
     main=paste("Distribution of Sample Slopes, N =", sample_size),
     xlab="Observed Slope",
     xlim=c(-5,5), cex.lab=1.5, cex.axis=1.5)



# Identifying the fitter group based on their Change scores
BIASED_FILTER$INT_FITTERS<-ifelse(BIASED_FILTER$C1_mean_CHANGE>BIASED_FILTER$
C2_mean_CHANGE,
        BIASED_FILTER$C1_INT_BIAS, # Returns if above is TRUE
        BIASED_FILTER$C2_INT_BIAS) # Return if above is FALSE

BIASED_FILTER$SLOPE_FITTERS<-ifelse(BIASED_FILTER$C1_mean_CHANGE>BIASED_FILTE
R$C2_mean_CHANGE,
                                    BIASED_FILTER$C1_SLOPE_BIAS, # Returns if
above is TRUE
                                    BIASED_FILTER$C2_SLOPE_BIAS) # Return if
above is FALSE

plot(density(BIASED_FILTER$SLOPE_FITTERS, na.rm=TRUE), col="navy", lwd=2,
lty=2,
     main=paste("Distribution of Sample Slopes, N =", sample_size),
     xlab="Observed Slope",
     xlim=c(-2,2),
     cex.lab=1.5, cex.axis=1.5)

head(BIASED_FILTER)
write.csv(BIASED_FILTER, paste("results_HCL_n",sample_size,".csv"))

# Testing the slope observed in the REAL data
summary(lm(CHANGE~0+II, data=FITTERS))
summary(BIASED_FILTER$SLOPE_FITTERS)
sum(BIASED_FILTER$SLOPE_FITTERS>=0.76921, na.rm=TRUE)

plot(density(BIASED_FILTER$SLOPE_FITTERS, na.rm=TRUE), col="navy", lwd=2,
lty=2,
     main=paste("Distribution of Sample Slopes, N =", sample_size),
```

```
     xlab="Observed Slope",
     xlim=c(0,2), cex.lab=1.5, cex.axis=1.5)
abline(v=0.76921, col="red", lwd=1)


# Figure 6: Reframing the problem ------------------------------------------
# Plotting all of the data
head(REAL)
# Figure 6A
plot(x=REAL$BASE, y=REAL$FINAL,
     ylab="Final FMA",
     xlab="Baseline FMA",
     pch=1, col=REAL$Group,
     xlim=c(0, 66), ylim=c(-5,66), cex.lab=1.5, cex.axis=1.5)
summary(lm(FINAL~1+BASE, data=REAL))
summary(lm(FINAL~1+BASE, data=REAL[REAL$Group=="fitters",]))

abline(a=51.20, b=0.21, lty=2, lwd=3, col="black")

plot(lm(FINAL~1+BASE, data=REAL))
plot(lm(FINAL~1+BASE, data=REAL[REAL$Group=="fitters",]))

head(REAL)
library(tidyverse); library(lme4); library(ggthemes)

REAL_LONG <- REAL %>% select(subID, Group, BASE, FINAL) %>%
  gather(Time, FMA, BASE:FINAL, factor_key=TRUE)

head(REAL_LONG)
REAL_LONG$Time <- ifelse(REAL_LONG$Time=="BASE", "Baseline", "Final")

REAL_LONG_MEAN <- REAL_LONG %>% group_by(Group, Time) %>%
  summarize(FMA=mean(FMA))

REAL_LONG_MEAN$Colors <- c("black", "black", "red", "red")
REAL_LONG_MEAN

#Figure 6B
ggplot(data=REAL_LONG, aes(x=Time, y=FMA)) +
  geom_line(aes(group = subID, col=Group),
            lwd=1, alpha=0.3) +
  geom_point(aes(col=Group), shape=21) +
  scale_color_manual(values=c("dark grey","tomato"))+
  geom_line(data=REAL_LONG_MEAN, aes(group=Group),
            col=REAL_LONG_MEAN$Colors,
            lwd=1.2, lty=2, alpha=1)+
  geom_point(data=REAL_LONG_MEAN, aes(group=Group),
             col= REAL_LONG_MEAN$Colors,
             shape=21, stroke=1.2, size=2, fill="white") +
  theme_bw()+
  theme(axis.text=element_text(size=18, color="black"),
      axis.title=element_text(size=18),
        plot.title=element_text(size=18, hjust=0.5),
        panel.grid.minor = element_blank(),
        strip.text = element_text(size=18),
        panel.grid.major = element_blank(),
        legend.position = "none")+
```

```
  scale_x_discrete(name="Assessment") +
  scale_y_continuous(name="FMA Upper Extremity Score",
                     breaks=c(0,10,20,30,40,50,60,70))



mod1<-lmer(FMA~1+Time*Group+(1|subID), REML=FALSE, data=REAL_LONG)
summary(mod1)


# Figure 6C
head(REAL_LONG)
BASE <- subset(REAL_LONG, Time =="Baseline")
summary(BASE$FMA)

BASE$Class<-factor(ntile(BASE$FMA, 5))
head(BASE)

REAL_LONG<-merge(BASE[,c(1,5)], REAL_LONG, by="subID")

REAL_LONG_MEAN <- REAL_LONG %>% group_by(Time, Class) %>%
  summarize(FMA=mean(FMA))



REAL_LONG_MEAN

ggplot(data=REAL_LONG, aes(x=Time, y=FMA)) +
  geom_line(aes(group = subID, col=Class),
            lwd=1, alpha=0.3) +
  geom_point(aes(col=Class), shape=21) +
  geom_line(data=REAL_LONG_MEAN, aes(group=Class),
             lwd=1.2, lty=2, alpha=1, col="black")+
  geom_point(data=REAL_LONG_MEAN, aes(group=Class, col=Class),
              shape=21, stroke=1.2, size=2, fill="white") +
  theme_bw()+
  theme(axis.text=element_text(size=18, color="black"),
        axis.title=element_text(size=18),
        plot.title=element_text(size=18, hjust=0.5),
        panel.grid.minor = element_blank(),
        strip.text = element_text(size=18),
        panel.grid.major = element_blank(),
        legend.position = "none")+
  scale_x_discrete(name="Assessment") +
  scale_y_continuous(name="FMA Upper Extremity Score",
                     breaks=c(0,10,20,30,40,50,60,70))
```