SynchronyScoresAnalysis

Thomas Gargot

22 septembre 2016

Clean environment

```
rm(list = ls(all.names = TRUE))
```

Session info

```
sessionInfo()
```

```
## R version 3.2.2 (2015-08-14)
## Platform: x86_64-apple-darwin13.4.0 (64-bit)
## Running under: OS X 10.12.3 (unknown)
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## loaded via a namespace (and not attached):
##  [1] backports_1.0.5 magrittr_1.5    rprojroot_1.2   tools_3.2.2
##  [5] htmltools_0.3.5 yaml_2.1.13     Rcpp_0.12.6     stringi_1.1.1
##  [9] rmarkdown_1.3   knitr_1.15.1    stringr_1.0.0   digest_0.6.9
## [13] evaluate_0.10
```

Preparation of the session

Fixed variables

```
FileExtension <- ".MTS.avi_res.csv"

# working directory
# where this report is
#setwd("./Git/Monrado/Reports/")
setwd("/Users/Ofix/Documents/Fac/internat/Recherche/projets/synchro/synchroData/Git/Monrado/Reports")

# blue will refer to father
# red will refer to mother
# green to child
colOrderList <- c("blue","red", "green")

ParticipantsList <- c("father", "mother", "child")
```

```r
# Import data
cutFrames <- read.csv2("../Data/CSV/Cutframes.csv")
str(cutFrames)
```

```
## 'data.frame':    34 obs. of  7 variables:
##  $ family    : Factor w/ 34 levels "1606","BAJE059",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ CutBefore : Factor w/ 11 levels "00:06","00:07",..: 6 4 3 5 3 7 5 11 7 5 ...
##  $ CutMiddle1: Factor w/ 26 levels "04:43","04:53",..: 21 15 16 16 4 10 17 18 20 17 ...
##  $ CutMiddle2: Factor w/ 31 levels "05:02","05:08",..: 27 20 25 14 2 20 17 26 19 21 ...
##  $ CutFinal  : Factor w/ 28 levels "11:40","14:43",..: 25 26 21 14 6 11 23 22 24 20 ...
##  $ ChildSex  : Factor w/ 2 levels "Female","Male": 1 1 2 1 1 1 1 1 1 2 ...
##  $ ParentSex : Factor w/ 2 levels "Female","Male": 2 2 1 1 1 1 1 1 1 1 ...
```

```r
# Change the vector in character and cut the string in two parts minutes and second for the 4 time labe
cutFrames$CutBefore <- as.character(cutFrames$CutBefore)
cutFramesCB <- strsplit(cutFrames$CutBefore, split=":")

# Compute the time in minutes from time in minutes and seconds for each video for Cut Before
Cut <- c()
for (i in 1:nrow(cutFrames)){
  CutBeforeAlone <- (as.numeric(cutFramesCB[i][[1]][1]) + as.numeric(cutFramesCB[i][[1]][2])/60)
  Cut <- c(Cut, CutBeforeAlone)
  }
cutFrames$CutBeforeMin <- Cut

# Compute the time in minutes from time in minutes and seconds for each video for Cut CutMiddle1
Cut <- c()
cutFrames$CutMiddle1 <- as.character(cutFrames$CutMiddle1)
cutMiddleSplit <- strsplit(cutFrames$CutMiddle1, split=":")
for (i in 1:nrow(cutFrames)){
  CutAlone <- (as.numeric(cutMiddleSplit[i][[1]][1]) + as.numeric(cutMiddleSplit[i][[1]][2])/60)
  Cut <- c(Cut, CutAlone)
  }
cutFrames$CutMiddle1Min <- Cut

# Compute the time in minutes from time in minutes and seconds for each video for Cut CutMiddle2
Cut <- c()
cutFrames$CutMiddle2 <- as.character(cutFrames$CutMiddle2)
cutMiddleSplit <- strsplit(cutFrames$CutMiddle2, split=":")
for (i in 1:nrow(cutFrames)){
  CutAlone <- (as.numeric(cutMiddleSplit[i][[1]][1]) + as.numeric(cutMiddleSplit[i][[1]][2])/60)
  Cut <- c(Cut, CutAlone)
  }
cutFrames$CutMiddle2Min <- Cut

# Compute the time in minutes from time in minutes and seconds for each video for Cut CutFinal
Cut <- c()
cutFrames$CutFinal <- as.character(cutFrames$CutFinal)
cutSplit <- strsplit(cutFrames$CutFinal, split=":")
for (i in 1:nrow(cutFrames)){
  CutAlone <- (as.numeric(cutSplit[i][[1]][1]) + as.numeric(cutSplit[i][[1]][2])/60)
  Cut <- c(Cut, CutAlone)
  }
cutFrames$CutFinalMin <- Cut
```

```
Cut <- c()
```

Data dictionnary of cutFrames dataframe

- **family** : code of the family
- **CutBefore** : when the experimenter leave the room and the interaction begin character string in the form min:sec
- **CutMiddle1** : when the experimenter come back to explain that the participants are asked to have a conflictual discussion, character string in the form min:sec
- **CutMiddle2** : when the experimenter leave and the conflictual discussion begin Between is the conflictual discussion, character string in the form min:sec
- **CutFinal** : when the experimenter come back to shut down the camera, character string in the form min:sec
- **ChildSex** : Factor variable : Male of Female
- **ParentSex** : Factor variable : Male of Female
- **CutBeforeMin** : when the experimenter leave the room and the interaction begin numeric variable in minutes
- **CutMiddle1Min** : when the experimenter come back to explain that the participants are asked to have a conflictual discussion, numeric variable in minutes
- **CutMiddle2Min** : when the experimenter leave and the conflictual discussion begin Between is the conflictual discussion, numeric variable in minutes
- **CutFinalMin** : when the experimenter come back to shut down the camera, numeric variable in minutes

SyncPy utilisation for creating synchrony dataframe

After extracting filtered motion motion history with mean on sliding interval (overlapping interval) of 5 frames And after puting this data on a CSV file slideddata.csv

We import this data on python Script with panda module Call_S_Estimator.py This script will compute the synchrony between each dyad of the interaction and of the whole group It will return a csv file for each video SSIXXXX.csv with XXXX the name of the video (1606, BAJE059 etc) that we can import with R with this following function

```
# Create a list with the file names
print("SSI Files Directory")
```

```
## [1] "SSI Files Directory"
```

```
SSInoLogFilesList <- list.files("../Data/CSV/Synchrony/noLog/S_estimator/noShuffle", full.name=TRUE)
#SSInoLogFilesList
```

```
# Import and merge the data
SSInoLog <- data.frame(video="Name")
for (file in SSInoLogFilesList){
  #  print(file)
    SSIalone <- read.csv(file)
  #  print(str(SSIalone))
    SSInoLog<- rbind.fill(SSInoLog, SSIalone)}
SSInoLog$video <-as.factor(SSInoLog$video)
SSInoLog <- SSInoLog[-which(SSInoLog$video=="Name"),]
SSInoLog$Interval <- NULL
SSInoLog <- rename (SSInoLog, c("video" = "family"))
```

```
SSInoLog <- rename (SSInoLog, c("X" = "SSI-interval"))
SSInoLog$SSI<-rowSums(SSInoLog[, c("SSI_fa_ch", "SSI_mo_ch")], na.rm=T)
```

Annotating data frames noLog

```
SSInoLog <- merge(SSInoLog, cutFrames, by.x="family", by.y="family")
SSInoLog$LabelVideo <- rep(NA, nrow(SSInoLog))
SSInoLog[which(SSInoLog$Time_min <= SSInoLog$CutBeforeMin),]$LabelVideo <- "Cut"

SSInoLog[which(SSInoLog$Time_min > SSInoLog$CutBeforeMin & SSInoLog$Time_min < SSInoLog$CutMiddle1Min),]

SSInoLog[which(SSInoLog$Time_min >= SSInoLog$CutMiddle1Min & SSInoLog$Time_min <= SSInoLog$CutMiddle2Min

SSInoLog[which(SSInoLog$Time_min > SSInoLog$CutMiddle2Min & SSInoLog$Time_min < SSInoLog$CutFinalMin),]$

SSInoLog[which(SSInoLog$Time_min >= SSInoLog$CutFinalMin),]$LabelVideo <- "Cut"
```

Import motion history (MH)

```
MH_list <- list.files("../Data/CSV/filtered/noLog/noshuffled", full.names = TRUE)

MH <- data.frame(video="Name")

for (file in MH_list){
  #  print(file)
    MHalone <- read.csv(file)
  #  print(str(SSIalone))
    MH<- rbind.fill(MH, MHalone)}
MH <- MH[-which(MH$video=="Name"),]
MH$X <- NULL
MH <- rename (MH, c("video" = "family"))
MH$slidedParent<-rowSums(MH[, c("slidedFather", "slidedMother")], na.rm=T)
```

Annotating dataframe MH

```
#View(MH)
MH <- merge(MH, cutFrames, by.x="family", by.y="family")
MH$LabelVideo <- rep(NA, nrow(MH))
#for 25 frmaes by second and 60 seconds by min
MH$Time_min <- MH$frame_index/(25*60)
MH[which(MH$Time_min <= MH$CutBeforeMin),]$LabelVideo <- "Cut"

MH[which(MH$Time_min > MH$CutBeforeMin & MH$Time_min < MH$CutMiddle1Min),]$LabelVideo <- "No Conflict"

MH[which(MH$Time_min >= MH$CutMiddle1Min & MH$Time_min <= MH$CutMiddle2Min),]$LabelVideo <- "Cut"

MH[which(MH$Time_min > MH$CutMiddle2Min & MH$Time_min < MH$CutFinalMin),]$LabelVideo <- "Conflict"
```

```
MH[which(MH$Time_min >= MH$CutFinalMin),]$LabelVideo <- "Cut"
```

Description of noLogSSI data frame

```
str(SSInoLog)
```

```
## 'data.frame':    6404 obs. of  17 variables:
##  $ family       : Factor w/ 35 levels "1606","BAJE059",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ SSI-interval : int  0 1 2 3 4 5 6 7 8 9 ...
##  $ Time_min     : num  0 0.0833 0.1667 0.25 0.3333 ...
##  $ SSI_fa_ch    : num  0.06532 0.00806 0.1161 0.11668 0.18393 ...
##  $ SSI_mo_ch    : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ SSI          : num  0.06532 0.00806 0.1161 0.11668 0.18393 ...
##  $ CutBefore    : chr  "00:11" "00:11" "00:11" "00:11" ...
##  $ CutMiddle1   : chr  "05:30" "05:30" "05:30" "05:30" ...
##  $ CutMiddle2   : chr  "06:16" "06:16" "06:16" "06:16" ...
##  $ CutFinal     : chr  "16:27" "16:27" "16:27" "16:27" ...
##  $ ChildSex     : Factor w/ 2 levels "Female","Male": 1 1 1 1 1 1 1 1 1 1 ...
##  $ ParentSex    : Factor w/ 2 levels "Female","Male": 2 2 2 2 2 2 2 2 2 2 ...
##  $ CutBeforeMin : num  0.183 0.183 0.183 0.183 0.183 ...
##  $ CutMiddle1Min: num  5.5 5.5 5.5 5.5 5.5 5.5 5.5 5.5 5.5 5.5 ...
##  $ CutMiddle2Min: num  6.27 6.27 6.27 6.27 6.27 ...
##  $ CutFinalMin  : num  16.4 16.4 16.4 16.4 16.4 ...
##  $ LabelVideo   : chr  "Cut" "Cut" "Cut" "No Conflict" ...
```

```
#View(SSInoLog)
```

Data dictionnary of SSInolog data frame

- **family** : code of the family
- **SSI-interval** : interval of SSI
- **Time_min** : Time in minutes
- **SSI_fa_ch** : SSI index of Synchrony between father and child
- **SSI_mo_ch** : SSI index of Synchrony between mother and child
- **SSI** : SSI index of the interaction (father-child or mother-child)
- **CutBefore** : when the experimenter leave the room and the interaction begin character string in the form min:sec
- **CutMiddle1** : when the experimenter come back to explain that the participants are asked to have a conflictual discussion, character string in the form min:sec
- **CutMiddle2** : when the experimenter leave and the conflictual discussion begin Between is the conflictual discussion, character string in the form min:sec
- **CutFinal** : when the experimenter come back to shut down the camera, character string in the form min:sec
- **ChildSex** : Factor variable : Male of Female
- **ParentSex** : Factor variable : Male of Female
- **CutBeforeMin** : when the experimenter leave the room and the interaction begin numeric variable in minutes
- **CutMiddle1Min** : when the experimenter come back to explain that the participants are asked to have a conflictual discussion, numeric variable in minutes
- **CutMiddle2Min** : when the experimenter leave and the conflictual discussion begin Between is the conflictual discussion, numeric variable in minutes

- **CutFinalMin** : when the experimenter come back to shut down the camera, numeric variable in minutes
- **LabelVideo** : label of the video if it is related to No Conflict period or Conflict Period

Description of MH data frame

```
str(MH)
```

```
## 'data.frame':    802834 obs. of  18 variables:
##  $ family       : chr  "1606" "1606" "1606" "1606" ...
##  $ slidedFather : num  0.002084 0.001758 0.000729 0.000693 0.000537 ...
##  $ slidedMother : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ slidedChild  : num  2.57e-03 1.55e-03 2.54e-04 2.09e-04 5.28e-05 ...
##  $ frame_index  : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ slidedParent : num  0.002084 0.001758 0.000729 0.000693 0.000537 ...
##  $ CutBefore    : chr  "00:11" "00:11" "00:11" "00:11" ...
##  $ CutMiddle1   : chr  "05:30" "05:30" "05:30" "05:30" ...
##  $ CutMiddle2   : chr  "06:16" "06:16" "06:16" "06:16" ...
##  $ CutFinal     : chr  "16:27" "16:27" "16:27" "16:27" ...
##  $ ChildSex     : Factor w/ 2 levels "Female","Male": 1 1 1 1 1 1 1 1 1 1 ...
##  $ ParentSex    : Factor w/ 2 levels "Female","Male": 2 2 2 2 2 2 2 2 2 2 ...
##  $ CutBeforeMin : num  0.183 0.183 0.183 0.183 0.183 ...
##  $ CutMiddle1Min: num  5.5 5.5 5.5 5.5 5.5 5.5 5.5 5.5 5.5 5.5 ...
##  $ CutMiddle2Min: num  6.27 6.27 6.27 6.27 6.27 ...
##  $ CutFinalMin  : num  16.4 16.4 16.4 16.4 16.4 ...
##  $ LabelVideo   : chr  "Cut" "Cut" "Cut" "Cut" ...
##  $ Time_min     : num  0.000667 0.001333 0.002 0.002667 0.003333 ...
```

```
#View(SSInoLog)
```

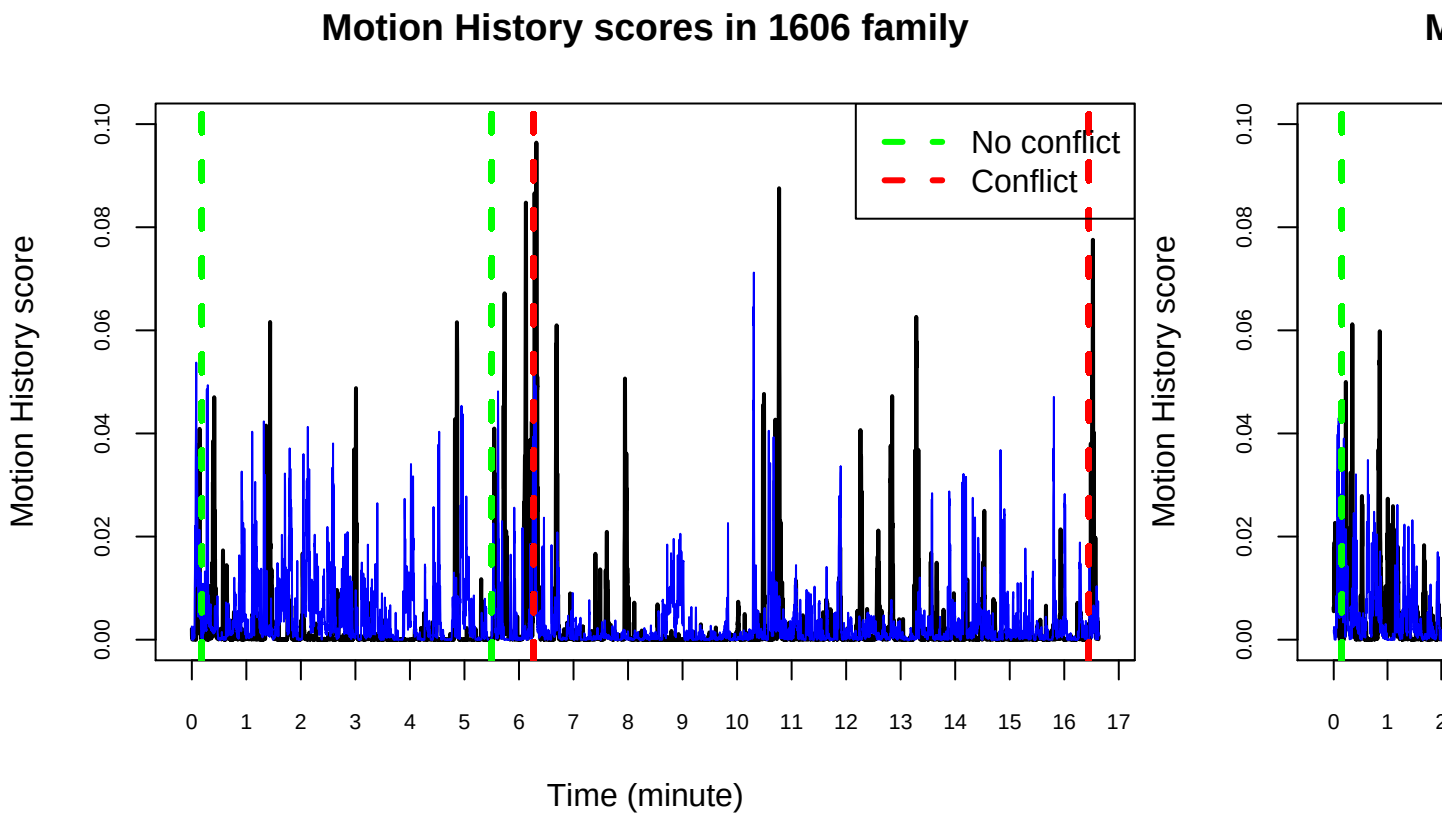
Data dictionnary of SSInolog data frame

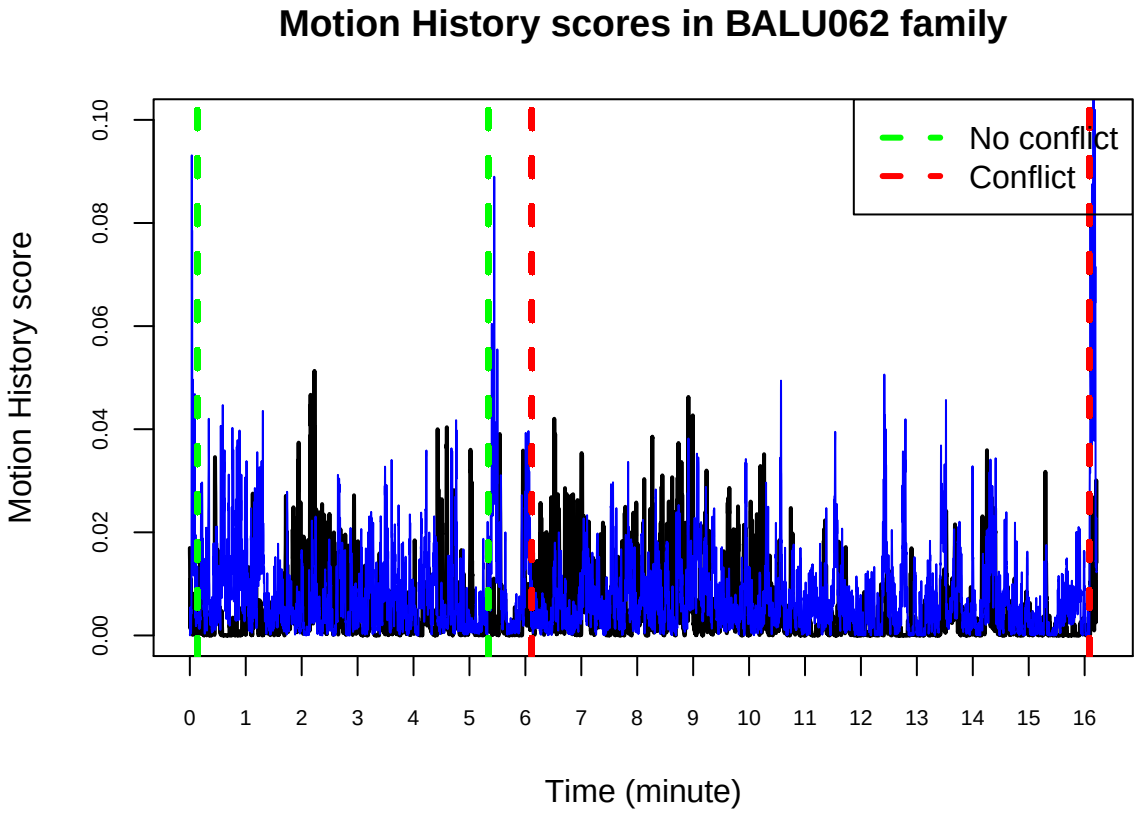
- **family** : code of the family
- **SlidedFather** : Filtered motion History of the father
- **Time_min** : Time in minutes
- **SlidedMother* : Filtered Motion History of the mother
- **SlidedChild** : Filtered Motion History of the child
- **SlidedParent* : Filtered Motion History of the parent (either mother or father)
- **frame index* : index of the frame (from 1st to last frame of each video)
- **CutBefore** : when the experimenter leave the room and the interaction begin character string in the form min:sec
- **CutMiddle1** : when the experimenter come back to explain that the participants are asked to have a conflictual discussion, character string in the form min:sec
- **CutMiddle2** : when the experimenter leave and the conflictual discussion begin Between is the conflictual discussion, character string in the form min:sec
- **CutFinal** : when the experimenter come back to shut down the camera, character string in the form min:sec
- **ChildSex** : Factor variable : Male of Female
- **ParentSex** : Factor variable : Male of Female
- **CutBeforeMin** : when the experimenter leave the room and the interaction begin numeric variable in minutes
- **CutMiddle1Min** : when the experimenter come back to explain that the participants are asked to have a conflictual discussion, numeric variable in minutes

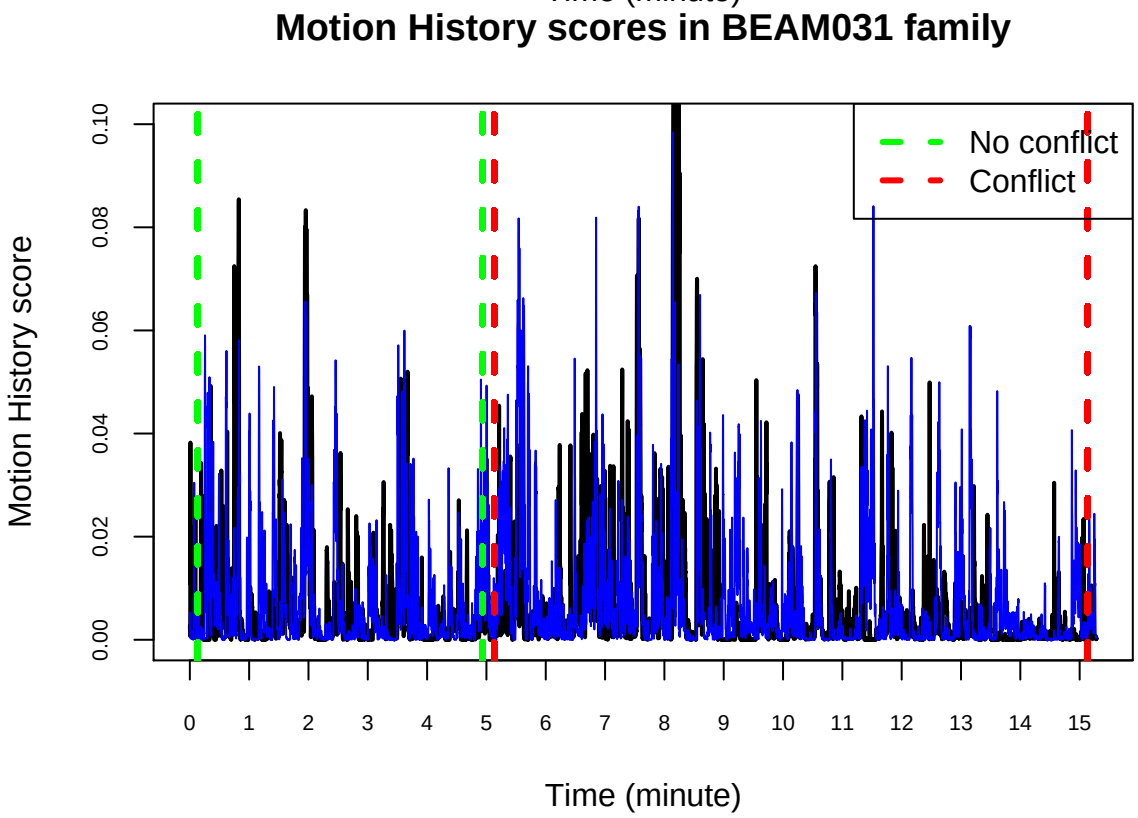
- **CutMiddle2Min** : when the experimenter leave and the conflictual discussion begin Between is the conflictual discussion, numeric variable in minutes
- **CutFinalMin** : when the experimenter come back to shut down the camera, numeric variable in minutes
- **LabelVideo** : label of the video if it is related to No Conflict period or Conflict Period

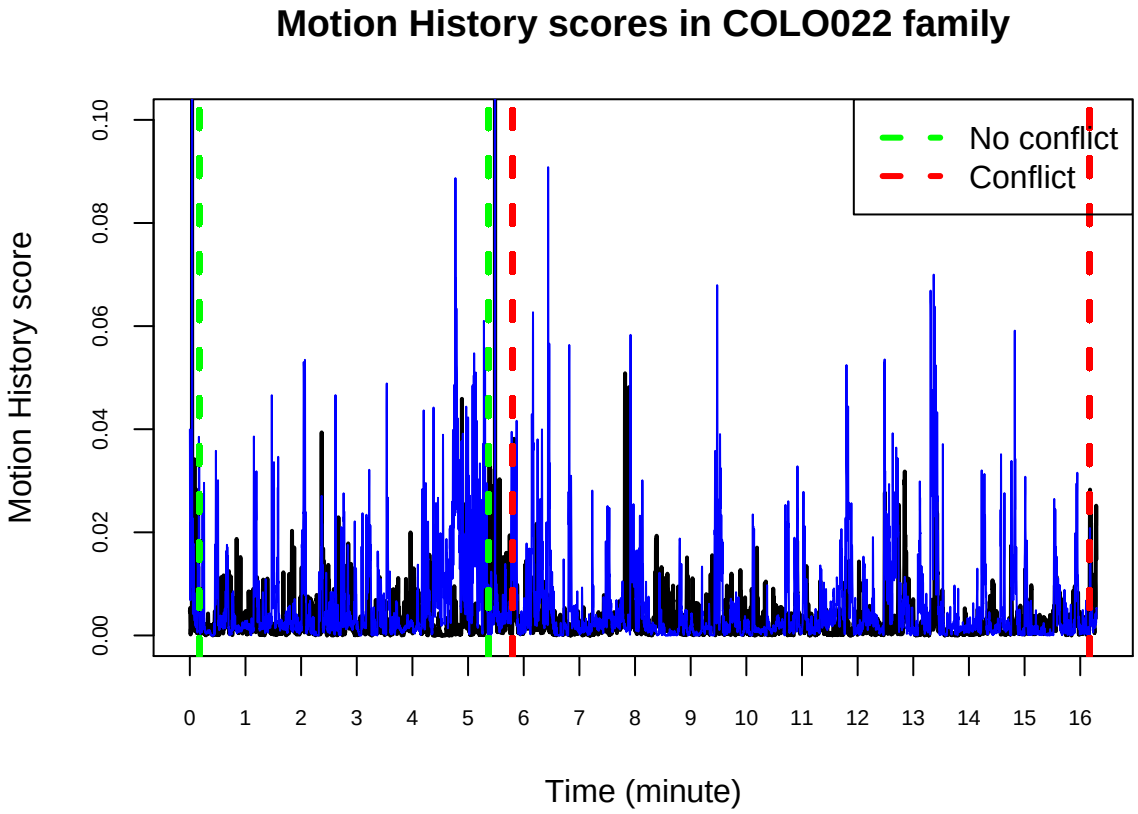
Motion history score for each dyad

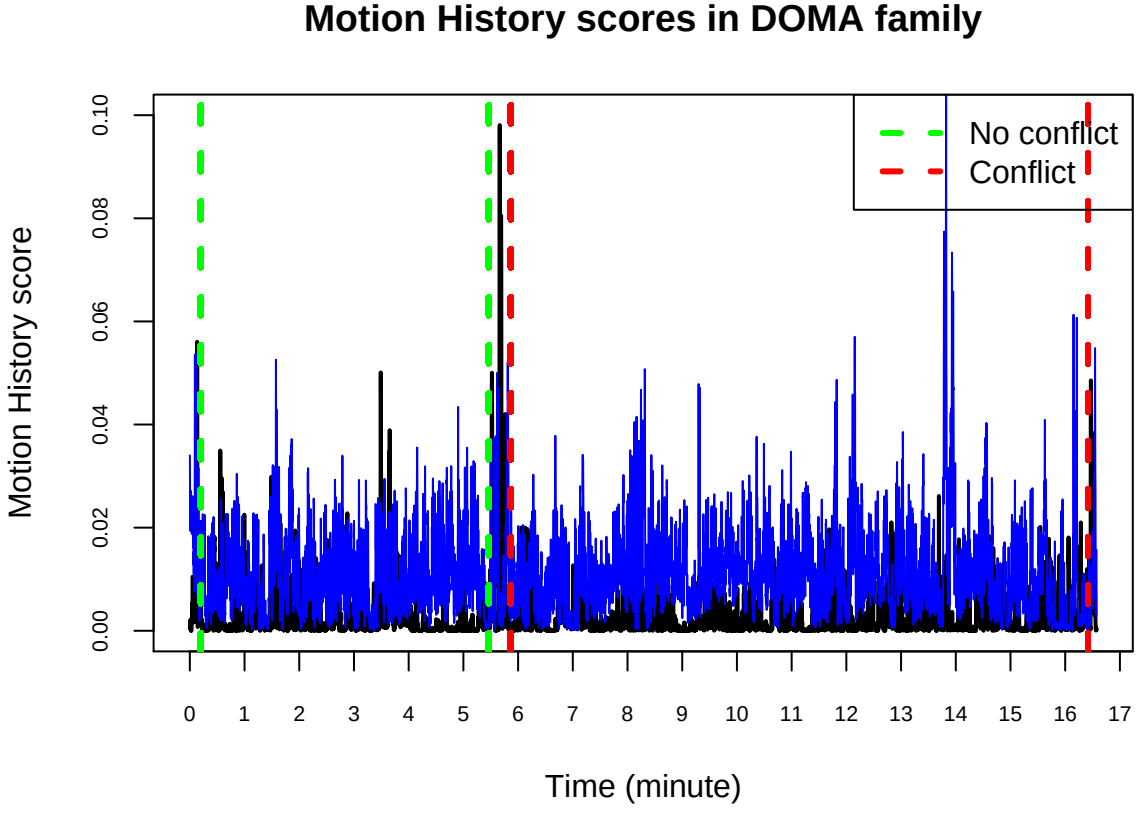
```
par(mar=c(4,4,4,3))
for (i in unique(MH$family)){
        plot(MH[which(MH$family==i),]$Time_min, (MH[which(MH$family==i),]$slidedParent),
             ylim=c(0, 0.1),
             main=paste("Motion History scores in", i, "family"),
             xlab = "Time (minute)", ylab="Motion History score",
                    lwd=2, type="l", cex.axis=0.7, xaxp=c(0,length(MH[which(MH$family==i),]$Time_min),
        lines(MH[which(MH$family==i),]$Time_min, MH[which(MH$family==i),]$slidedChild, col="blue")
        abline(v= MH[which(MH$family==i),]$CutBeforeMin, col="green", lty=2, lwd=3)
        abline(v= MH[which(MH$family==i),]$CutMiddle1Min, col="green", lty=2, lwd=3)
        abline(v= MH[which(MH$family==i),]$CutMiddle2Min
               , col="red", lty=2, lwd=3)
        abline(v= MH[which(MH$family==i),]$CutFinalMin, col="red", lty=2, lwd=3)
        legend("topright", c("No conflict", "Conflict"), lty=c(2,2), lwd=c(3,3), col=c("green", "red"))
#     a <- paste(as.character(i), ".jpeg", sep="")

 #      jpeg(filename = a, width = 560, height = 360, units = "px")
}
```

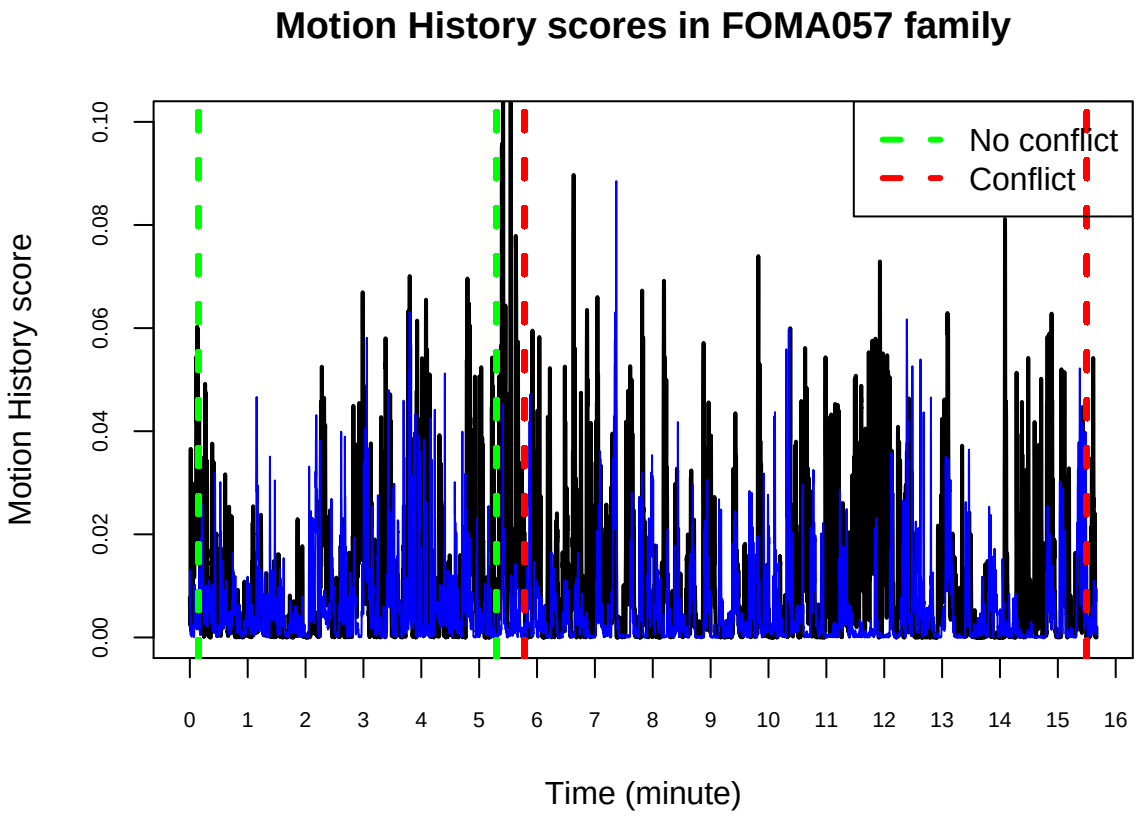


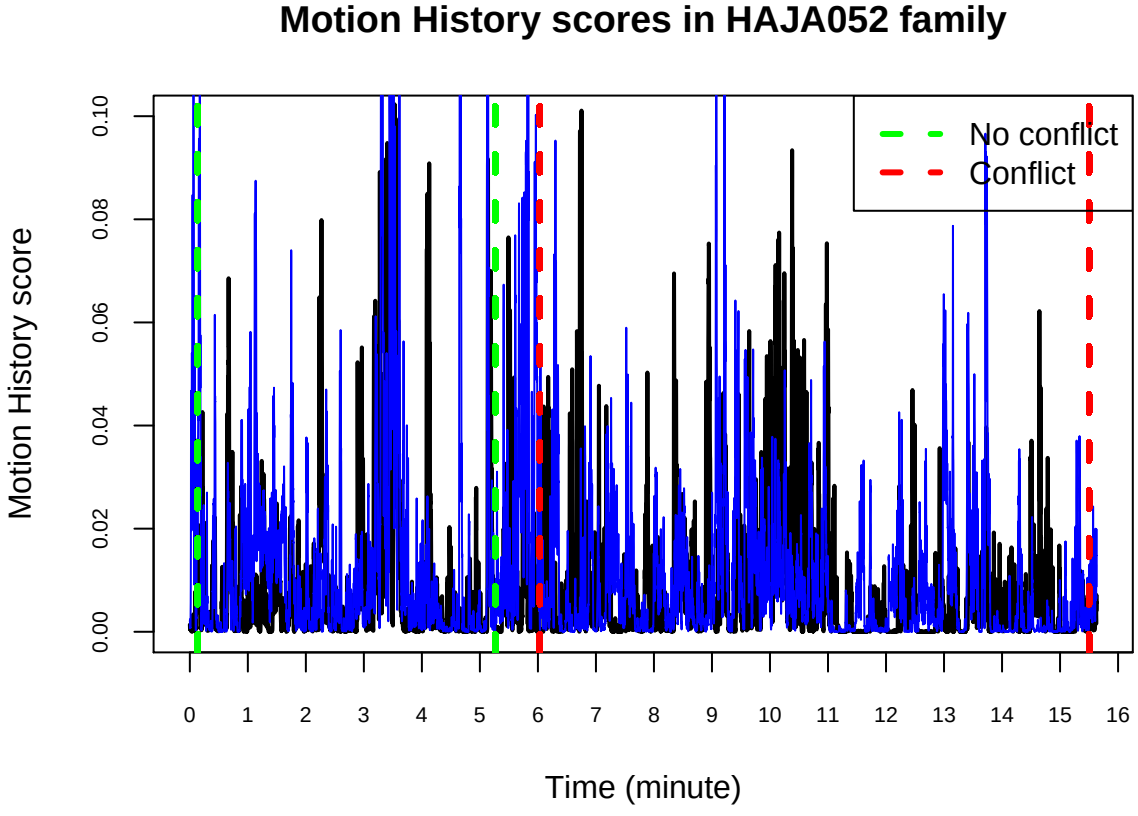

Motion History scores in BALU062 family
Motion History score
0.00
0.02
0.04
0.06
0.08
0.10
No conflict
Conflict
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
Time (minute)

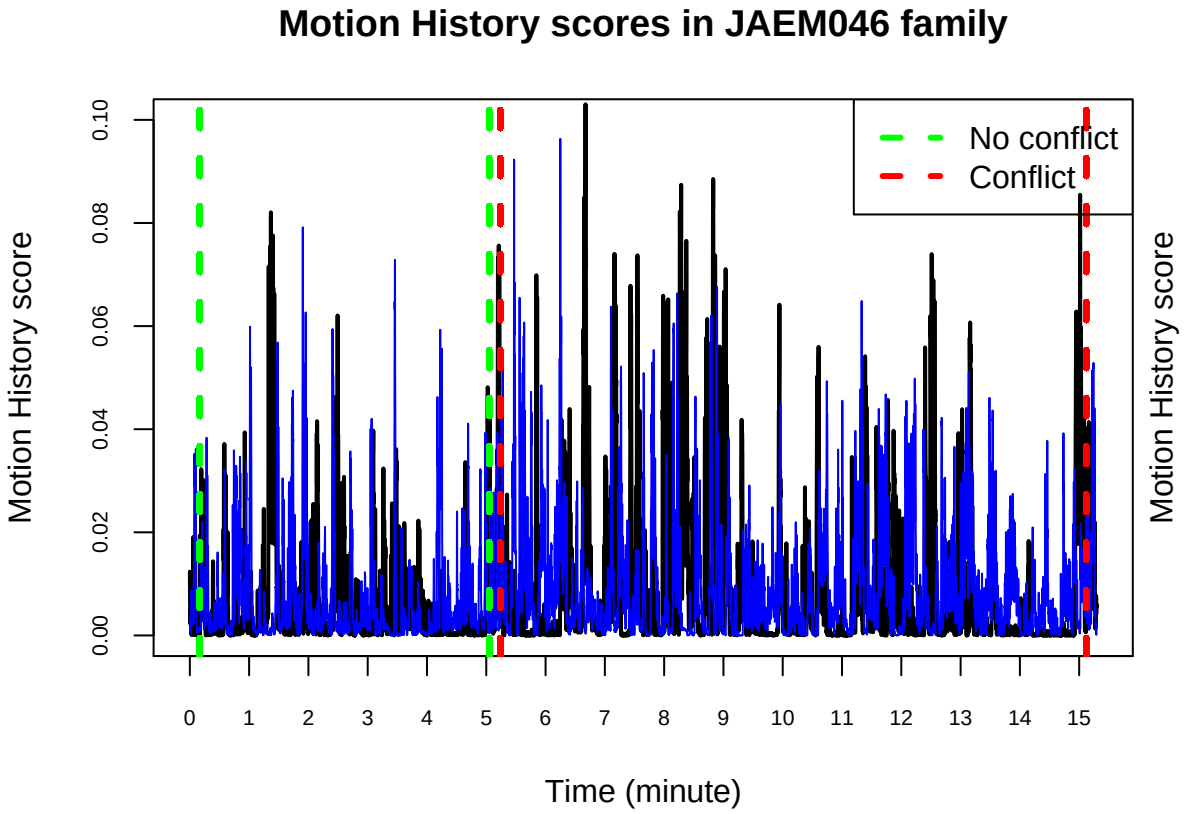

Motion History scores in BEAM031 family
Motion History score
0.00
0.02
0.04
0.06
0.08
0.10
No conflict
Conflict
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
Time (minute)

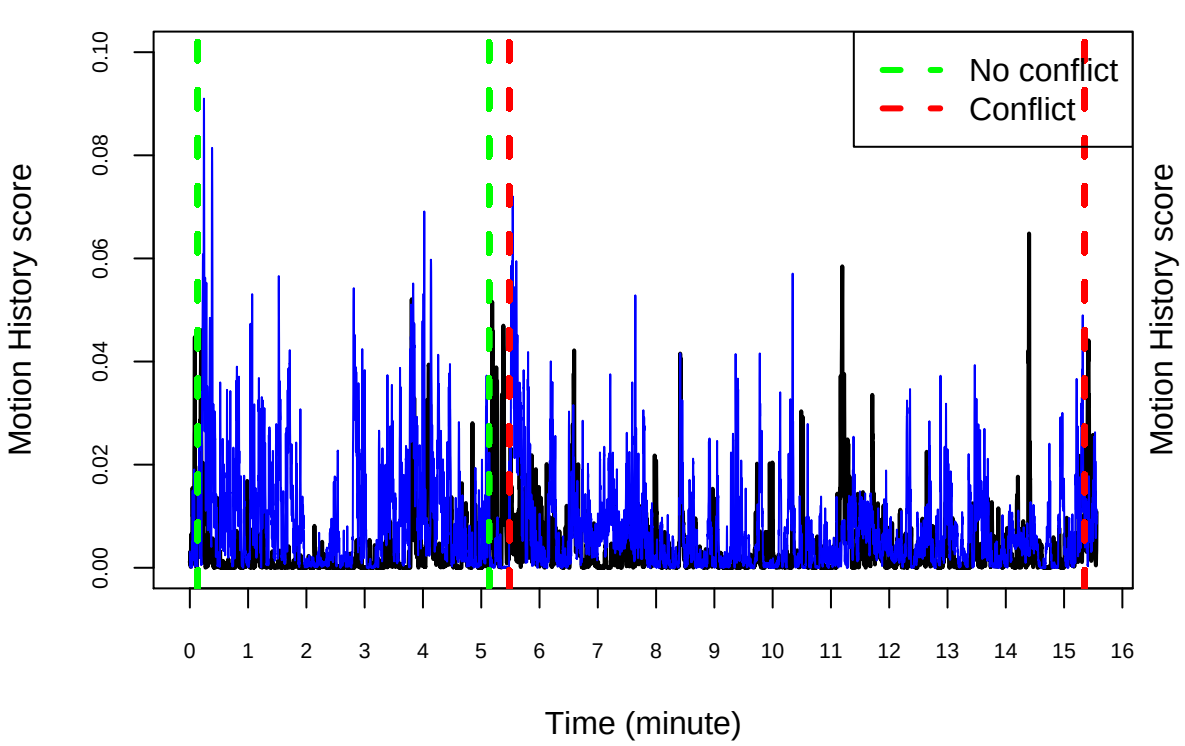

Motion History scores in COLO022 family
Motion History score
0.00
0.02
0.04
0.06
0.08
0.10
No conflict
Conflict
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
Time (minute)

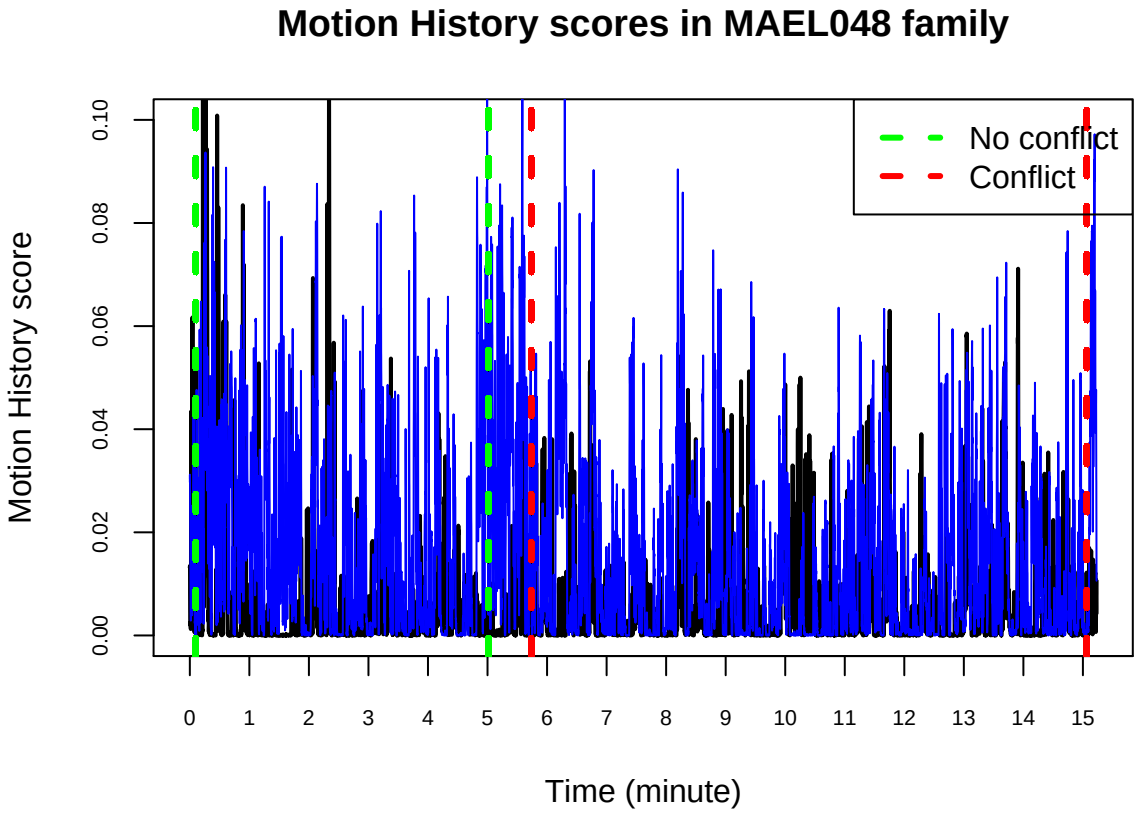

Motion History scores in DOMA family
Motion History score
0.00
0.02
0.04
0.06
0.08
0.10
No conflict
Conflict
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17
Time (minute)

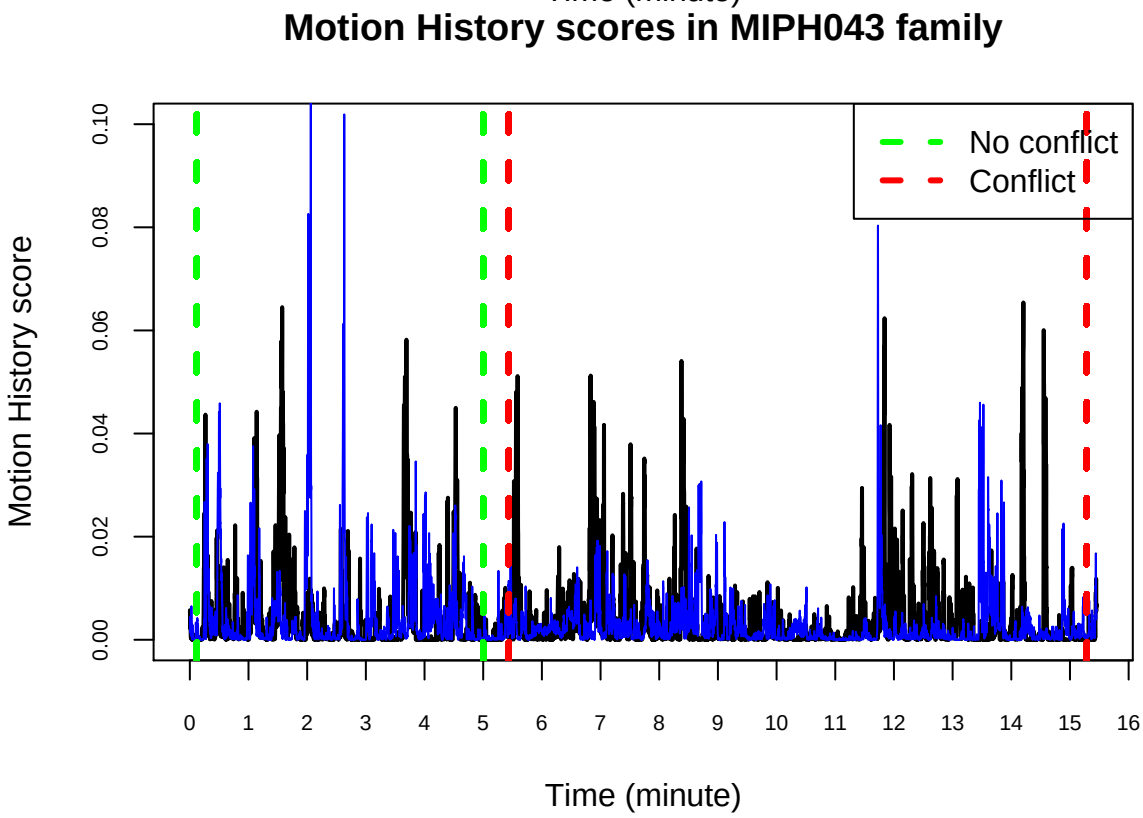

Motion History scores in FOMA057 family
Motion History score
0.00
0.02
0.04
0.06
0.08
0.10
No conflict
Conflict
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
Time (minute)

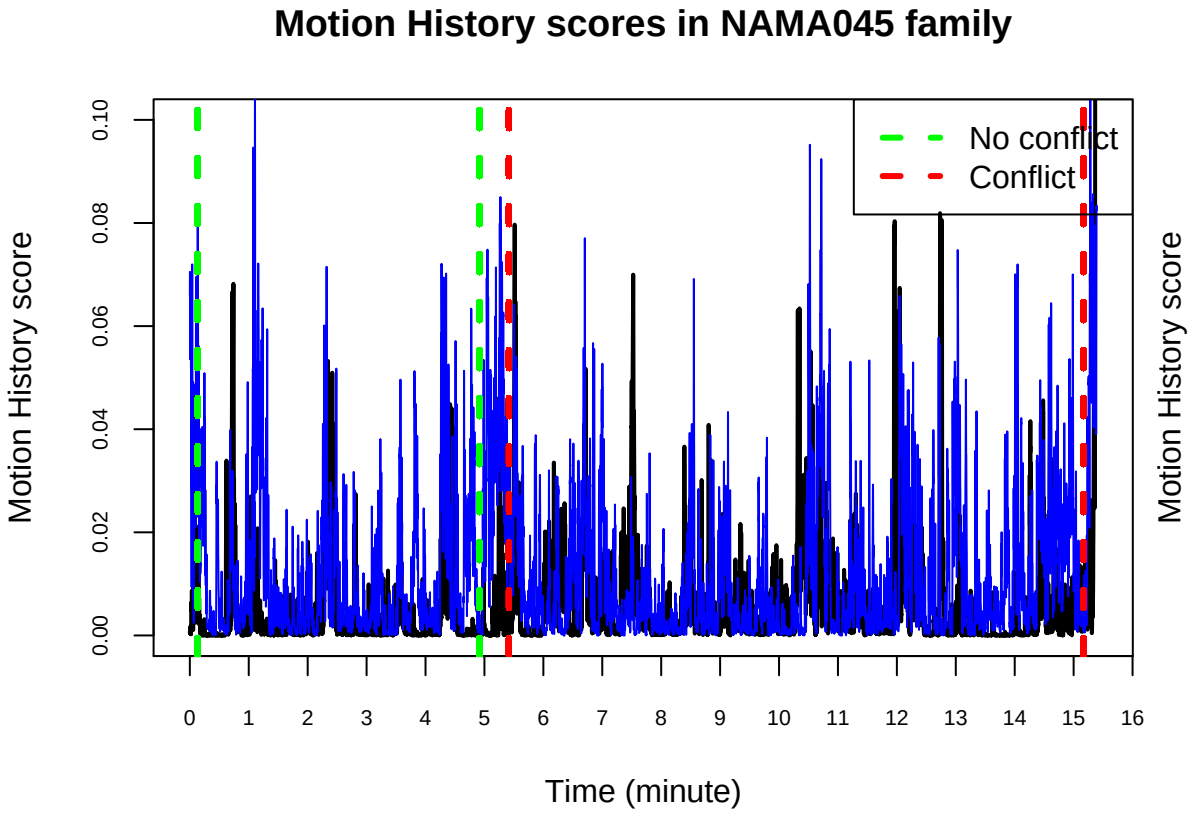

Motion History scores in HAJA052 family
Motion History score
0.00
0.02
0.04
0.06
0.08
0.10
No conflict
Conflict
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
Time (minute)

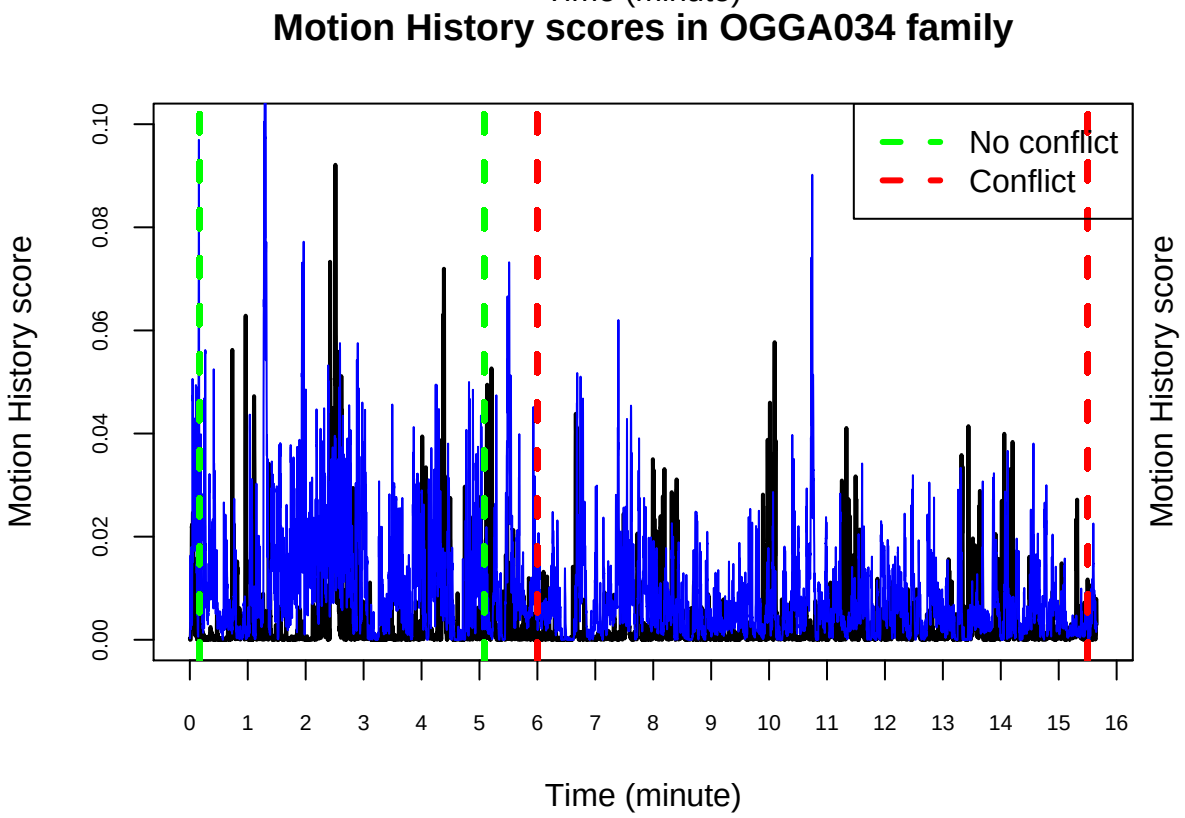

Motion History scores in JAEM046 family
Motion History score
Time (minute)
No conflict
Conflict
0.00
0.02
0.04
0.06
0.08
0.10
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15

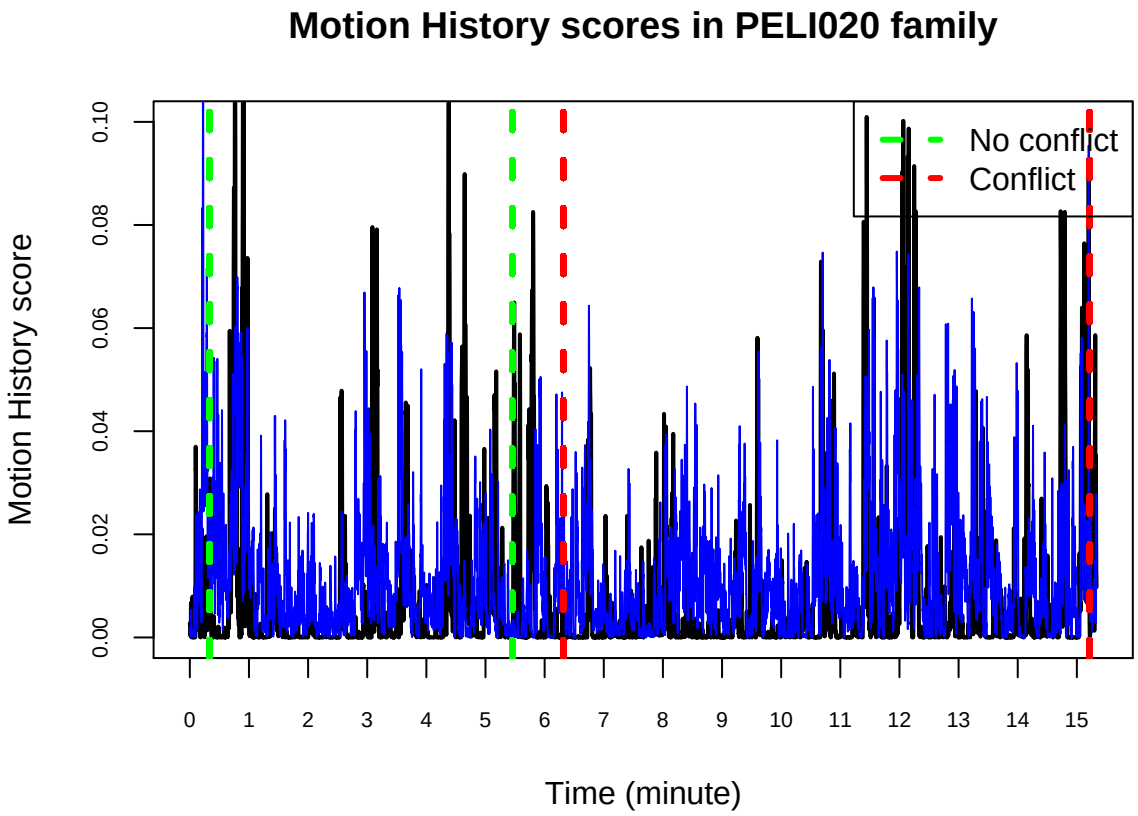

Motion History scores in JOCE014 family
Motion History score
Time (minute)
No conflict
Conflict
0.00
0.02
0.04
0.06
0.08
0.10
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

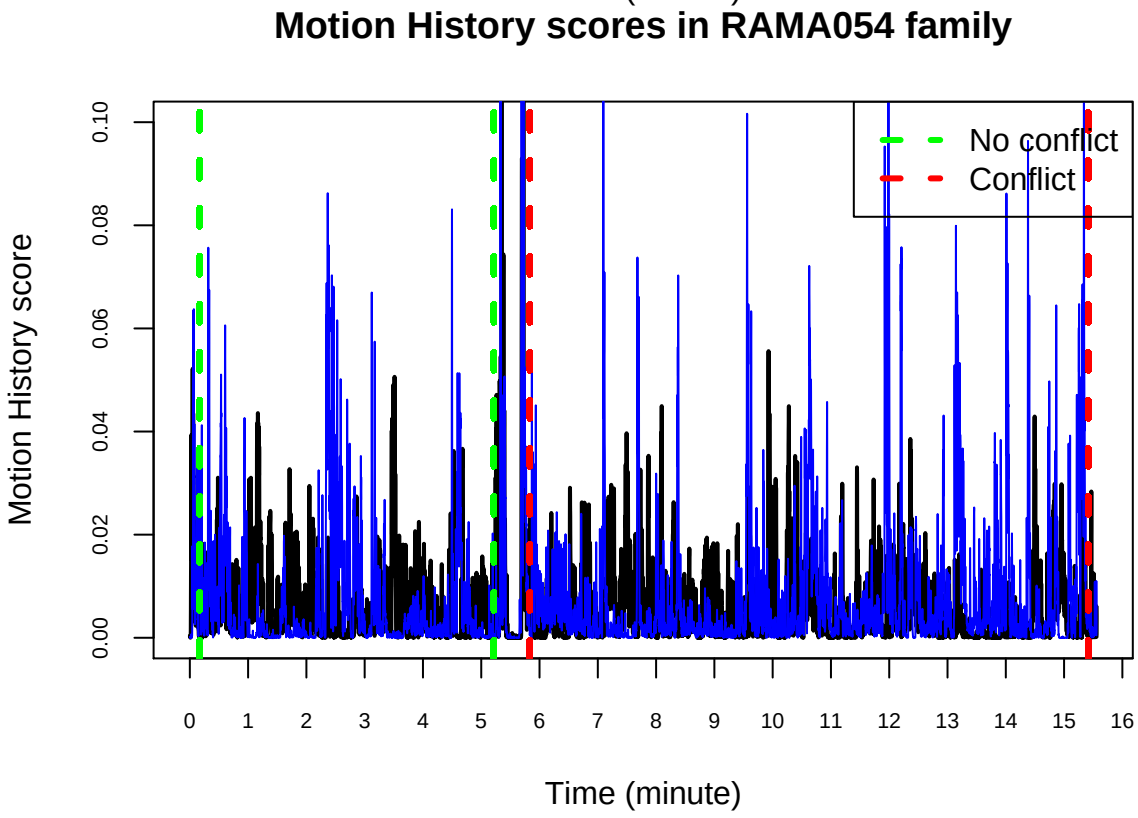

Motion History scores in MAEL048 family
Motion History score
0.00
0.02
0.04
0.06
0.08
0.10
No conflict
Conflict
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
Time (minute)

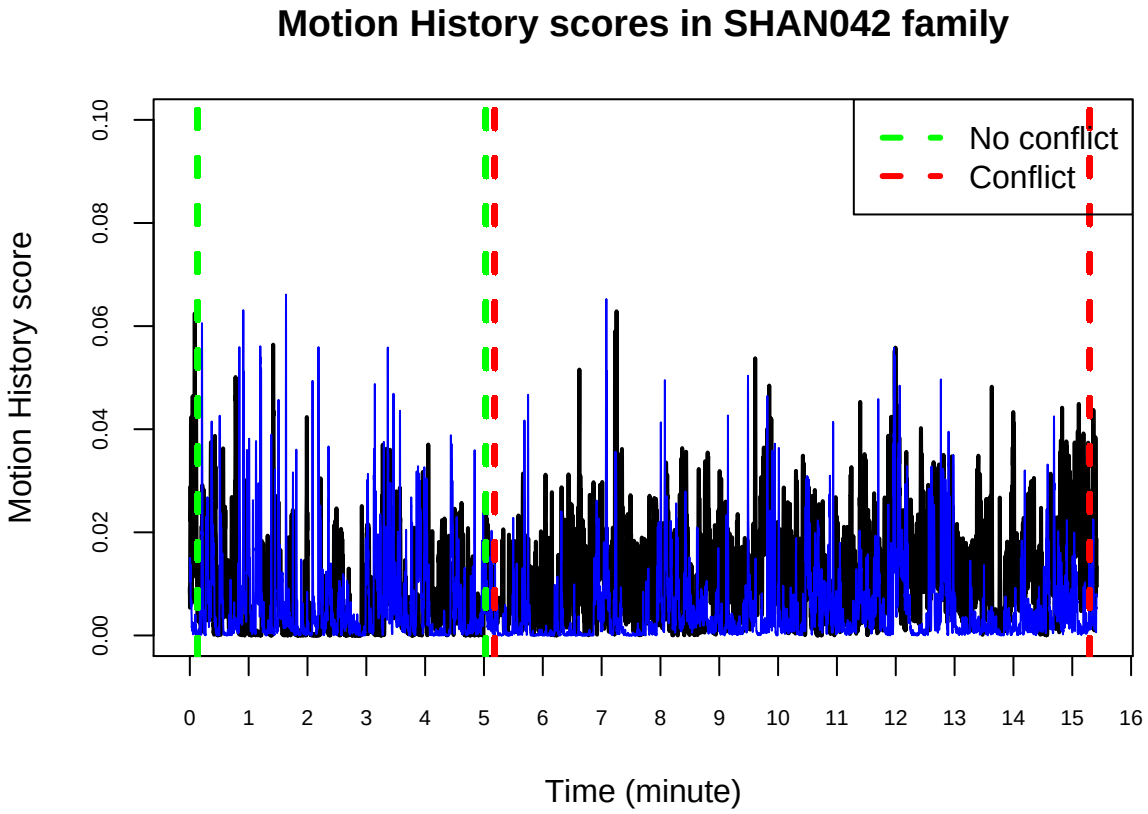

Motion History scores in MIPH043 family
Motion History score
0.00
0.02
0.04
0.06
0.08
0.10
No conflict
Conflict
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
Time (minute)

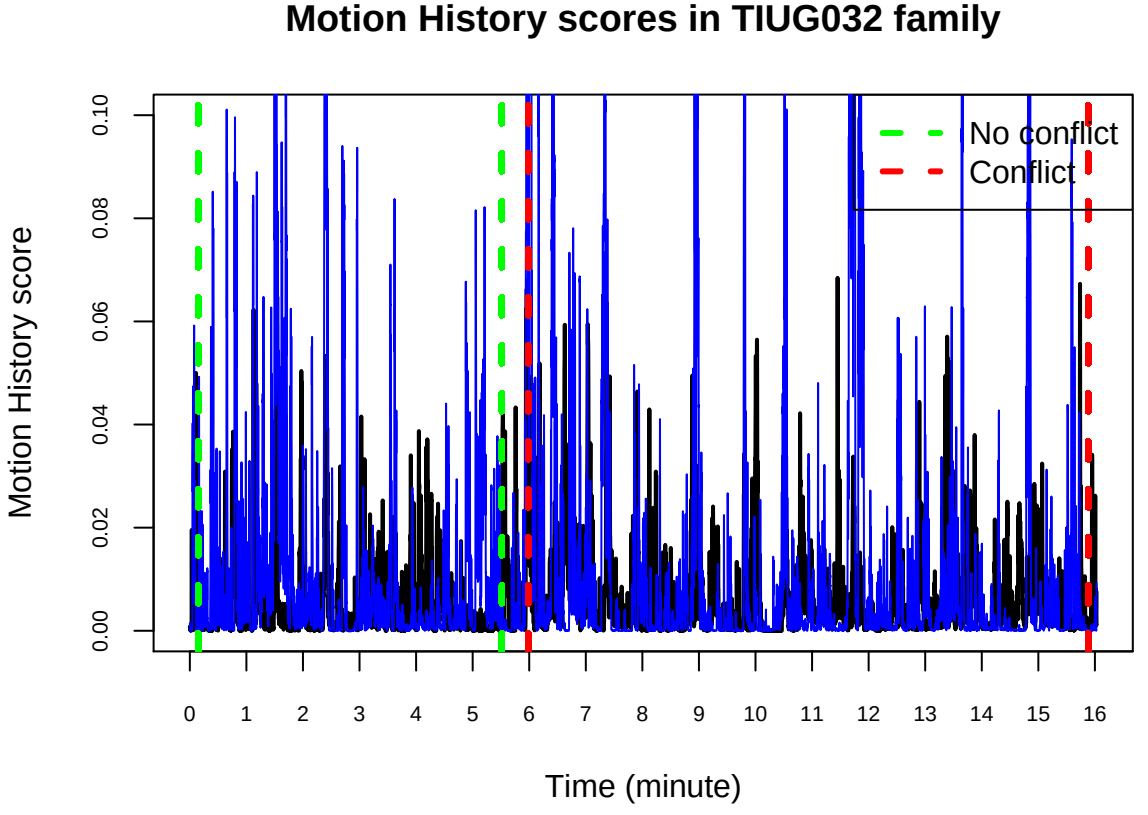

Motion History scores in NAMA045 family
Motion History score
Time (minute)
No conflict
Conflict
0.00
0.02
0.04
0.06
0.08
0.10
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16


Motion History scores in OGGA034 family
Motion History score
Time (minute)
No conflict
Conflict
0.00
0.02
0.04
0.06
0.08
0.10
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16


Motion History scores in PELI020 family
Motion History score
0.00
0.02
0.04
0.06
0.08
0.10
No conflict
Conflict
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
Time (minute)

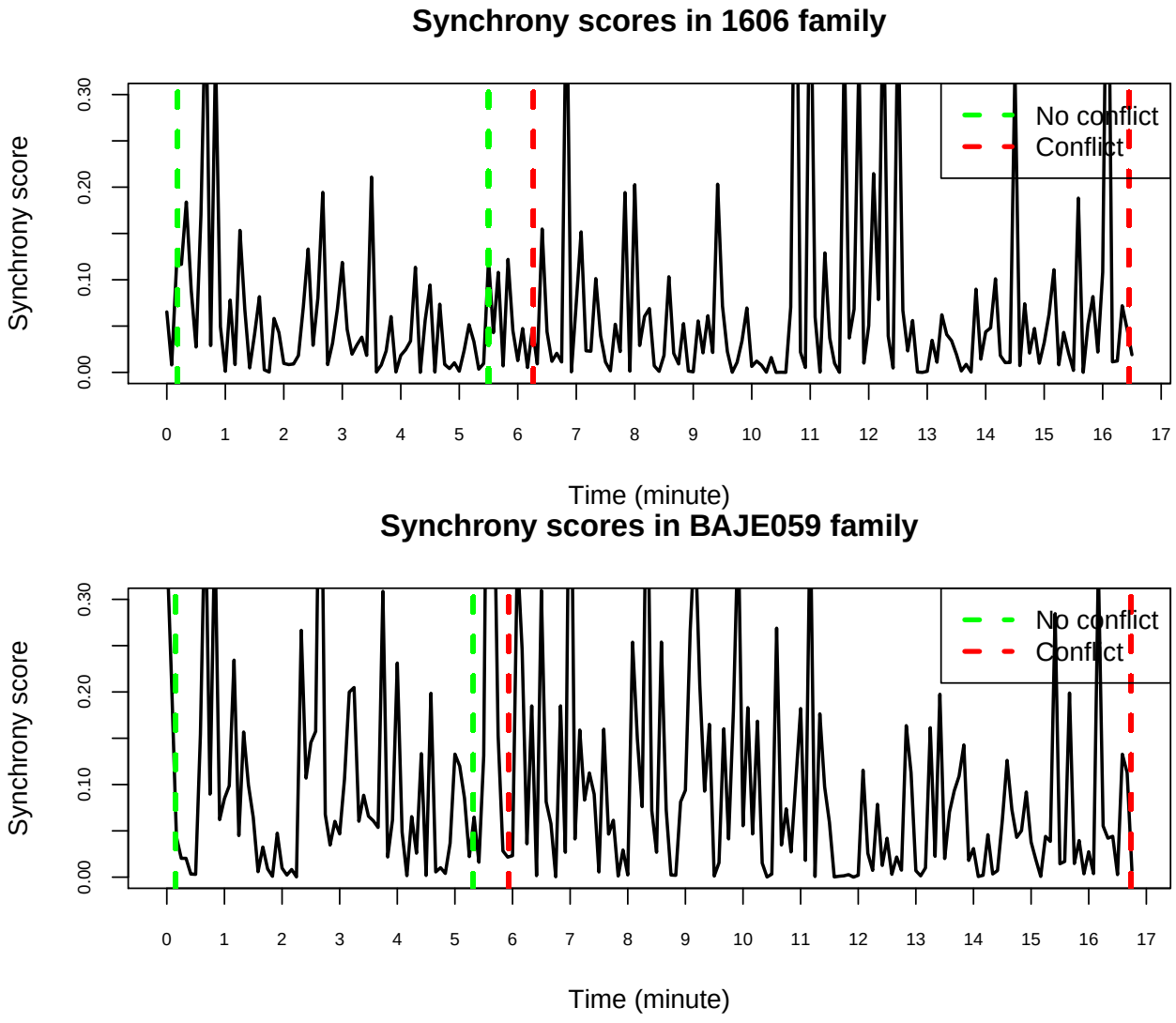

Motion History scores in RAMA054 family
Motion History score
0.00
0.02
0.04
0.06
0.08
0.10
No conflict
Conflict
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
Time (minute)

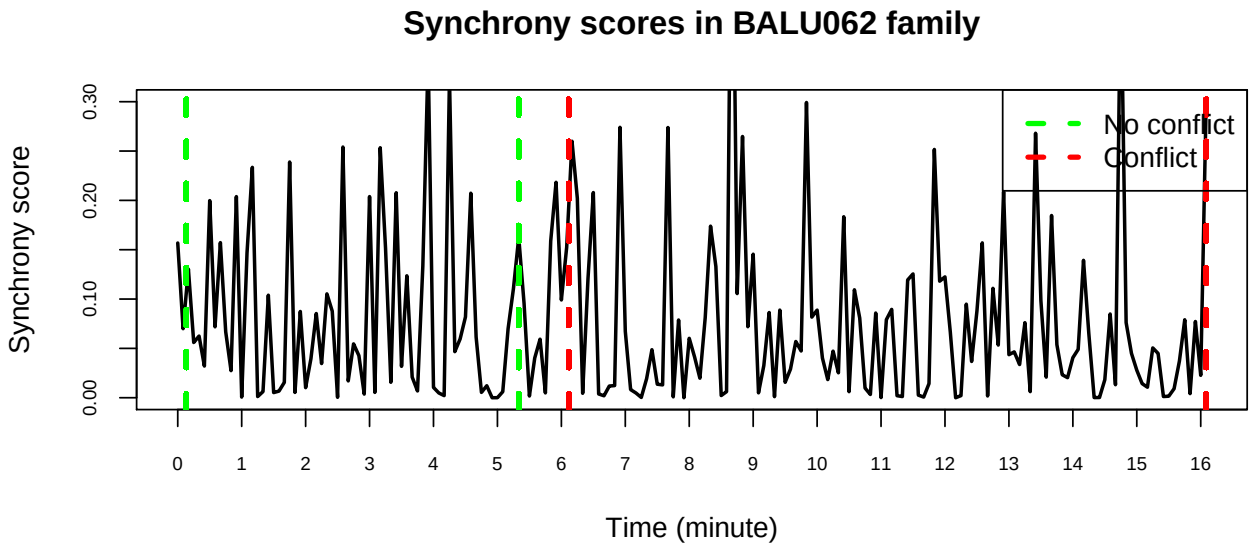

Motion History scores in SHAN042 family
Motion History score
0.00
0.02
0.04
0.06
0.08
0.10
No conflict
Conflict
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
Time (minute)

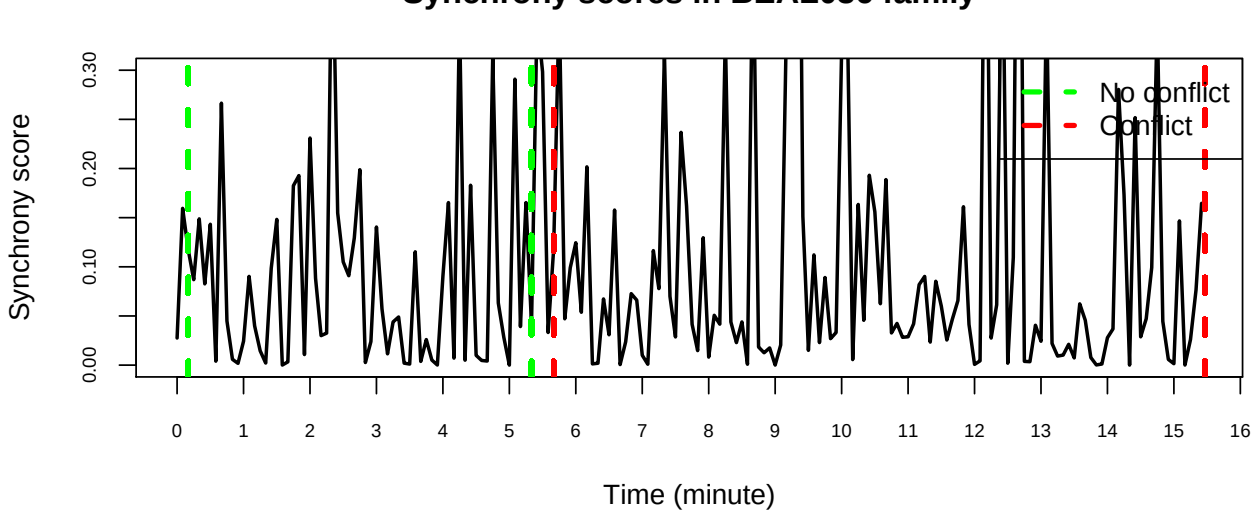

Motion History scores in TIUG032 family
Motion History score
0.00
0.02
0.04
0.06
0.08
0.10
No conflict
Conflict
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
Time (minute)

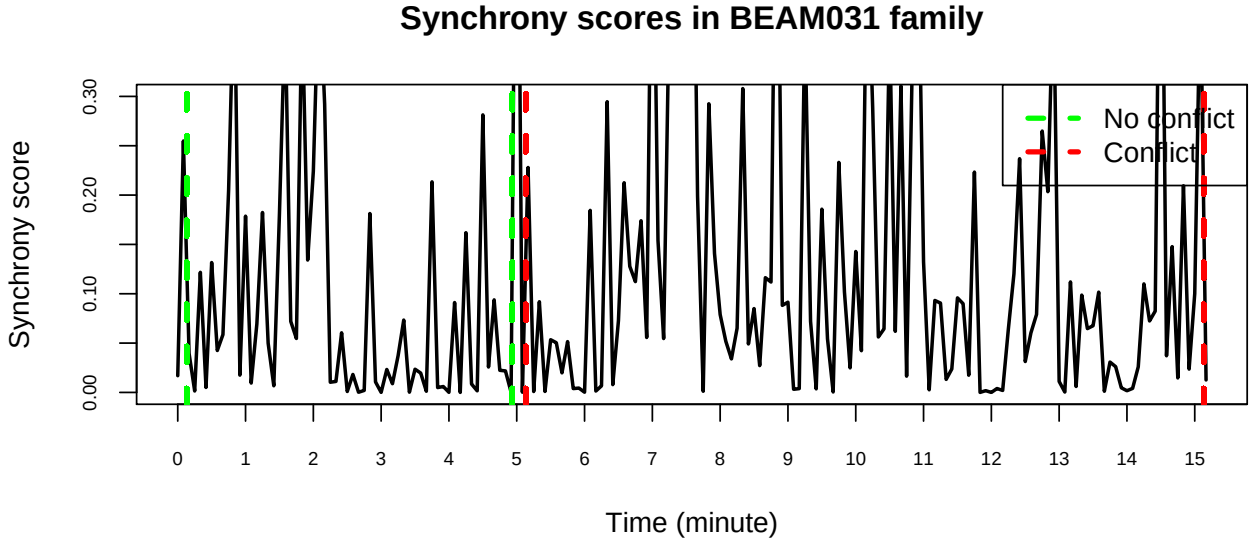
Synchrony scores noLog for each dyad

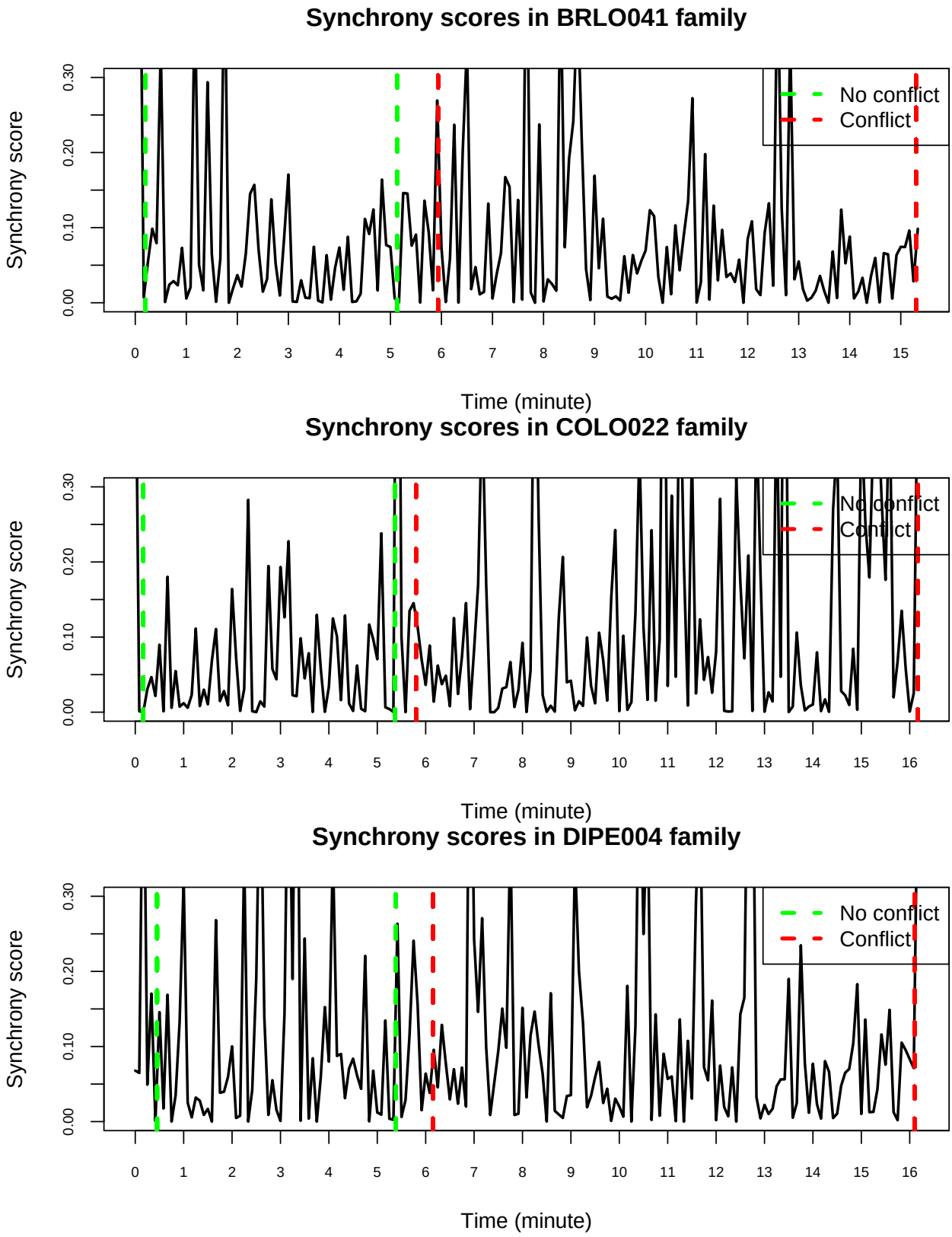
```
par(mar=c(4,4,4,3), mfrow=c(1,1))
for (i in unique(SSInoLog$family)){
        plot(SSInoLog[which(SSInoLog$family==i),]$Time_min,
             SSInoLog[which(SSInoLog$family==i),]$SSI,
             ylim=c(0, 0.3),
             main=paste("Synchrony scores in", i, "family"),
             xlab = "Time (minute)", ylab="Synchrony score",
             lwd=2, type="l", cex.axis=0.7, xaxp=c(0,length(SSInoLog[which(SSInoLog$family==i),]$Time_mi
abline(v= SSInoLog[which(SSInoLog$family==i),]$CutBeforeMin, col="green", lty=2, lwd=3)
abline(v= SSInoLog[which(SSInoLog$family==i),]$CutMiddle1Min, col="green", lty=2, lwd=3)
abline(v= SSInoLog[which(SSInoLog$family==i),]$CutMiddle2Min
       , col="red", lty=2, lwd=3)
abline(v= SSInoLog[which(SSInoLog$family==i),]$CutFinalMin, col="red", lty=2, lwd=3)
legend("topright", c("No conflict", "Conflict"), lty=c(2,2), lwd=c(3,3), col=c("green", "red"))
#max(SSInoLog[which(SSInoLog$family==i),]$SSI)), col="green")

  }
```

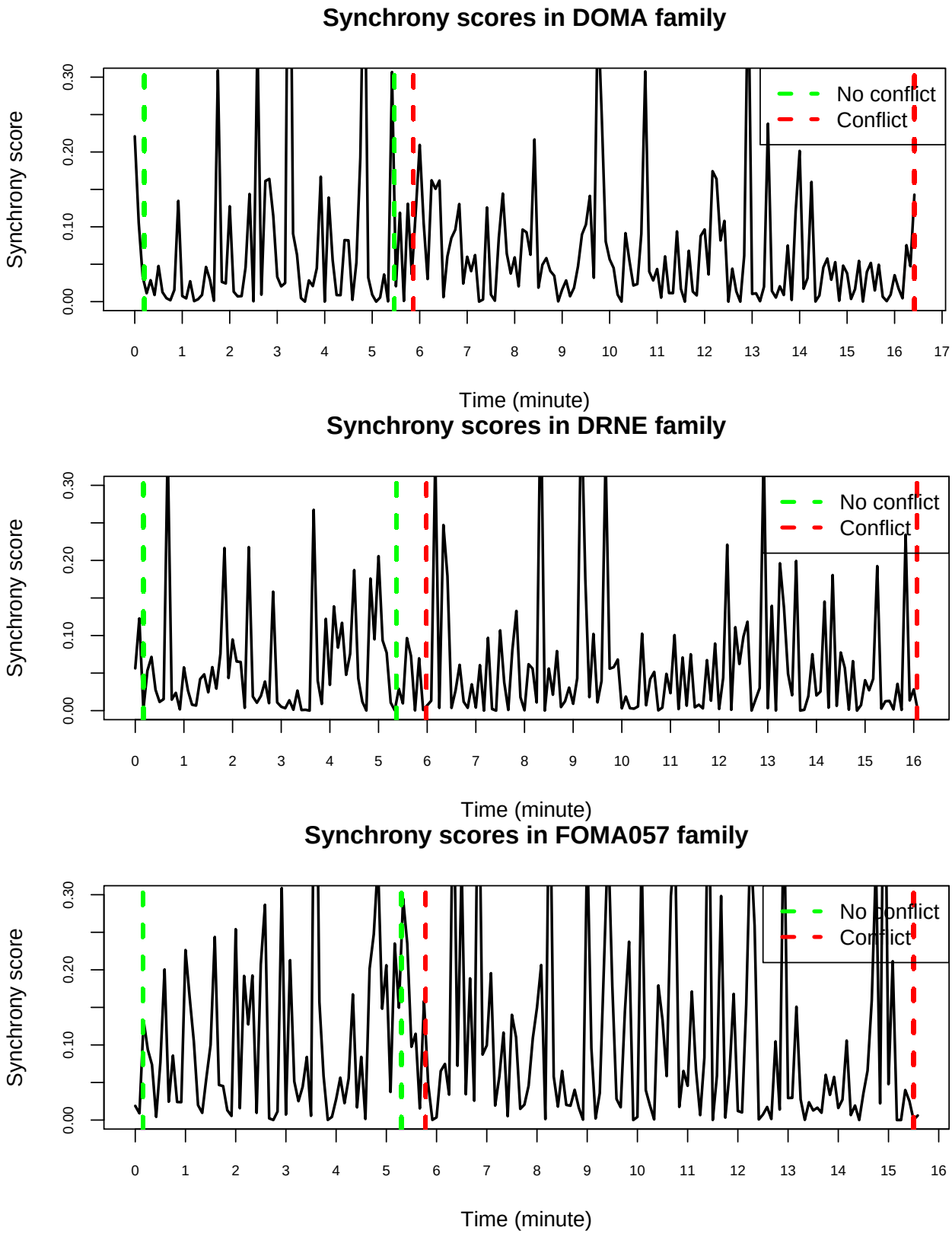


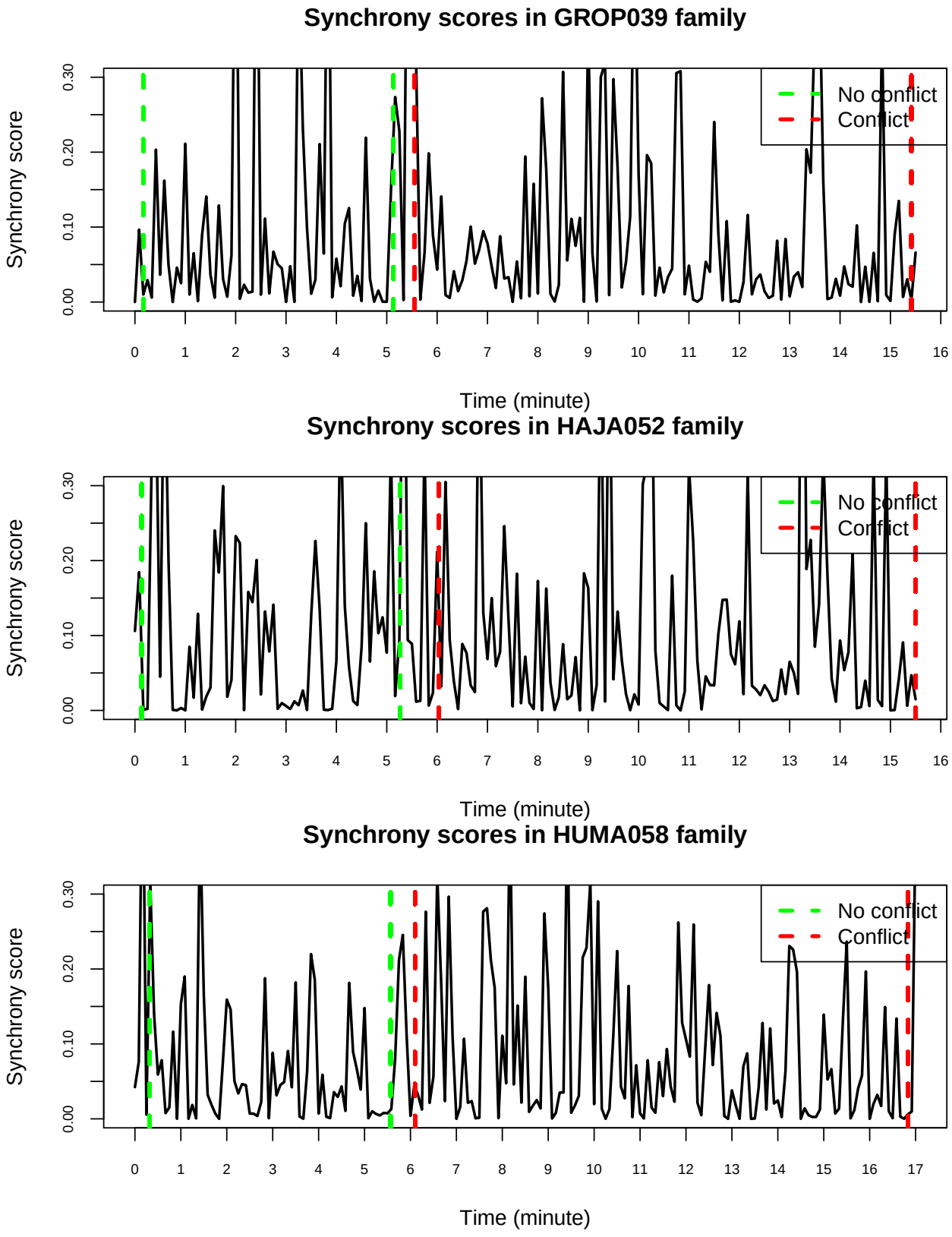

Synchrony scores in BALU062 family
Synchrony score
0.00
0.10
0.20
0.30
No conflict
Conflict
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
Time (minute)

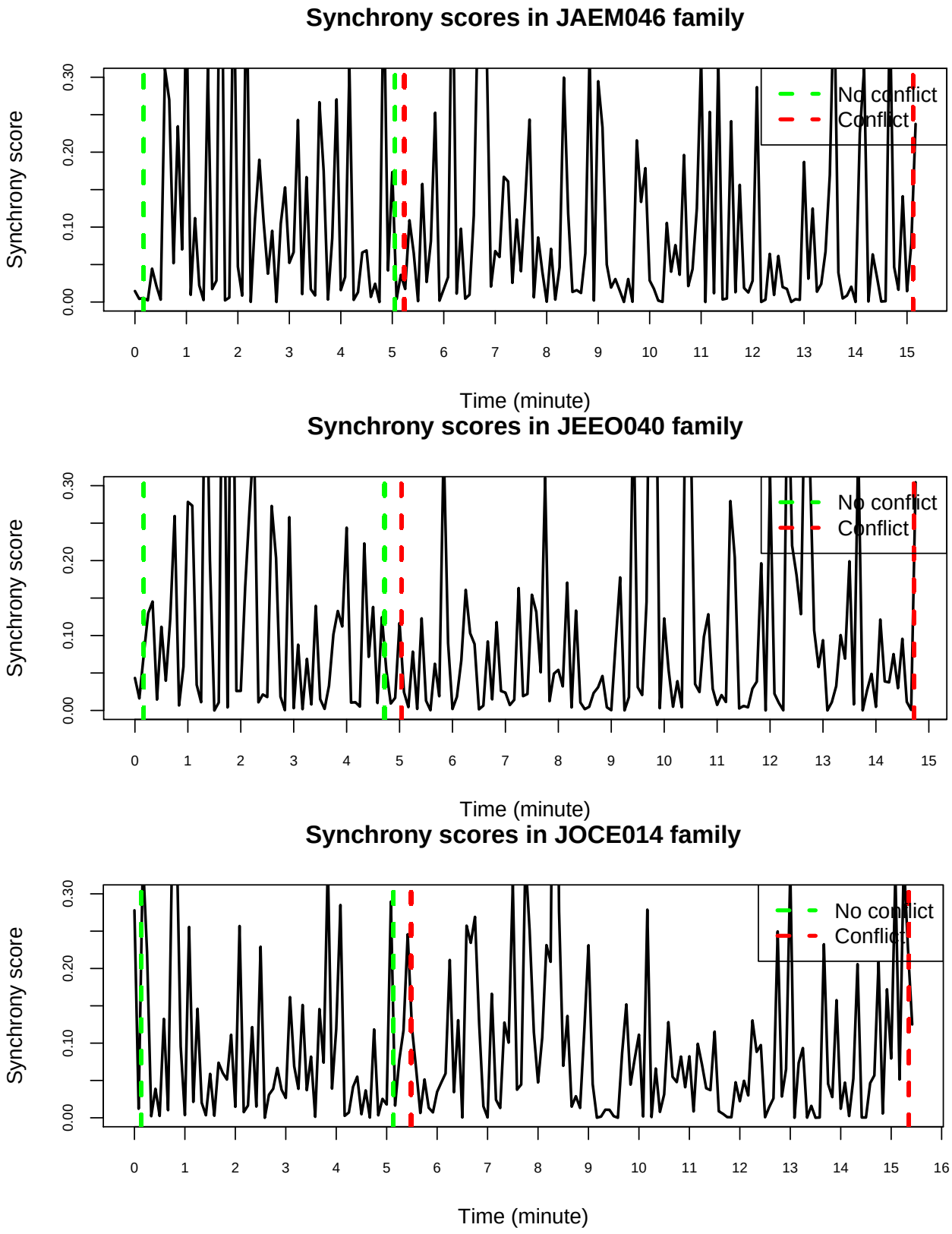

Synchrony scores in BEAL036 family
Synchrony score
0.00
0.10
0.20
0.30
No conflict
Conflict
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
Time (minute)

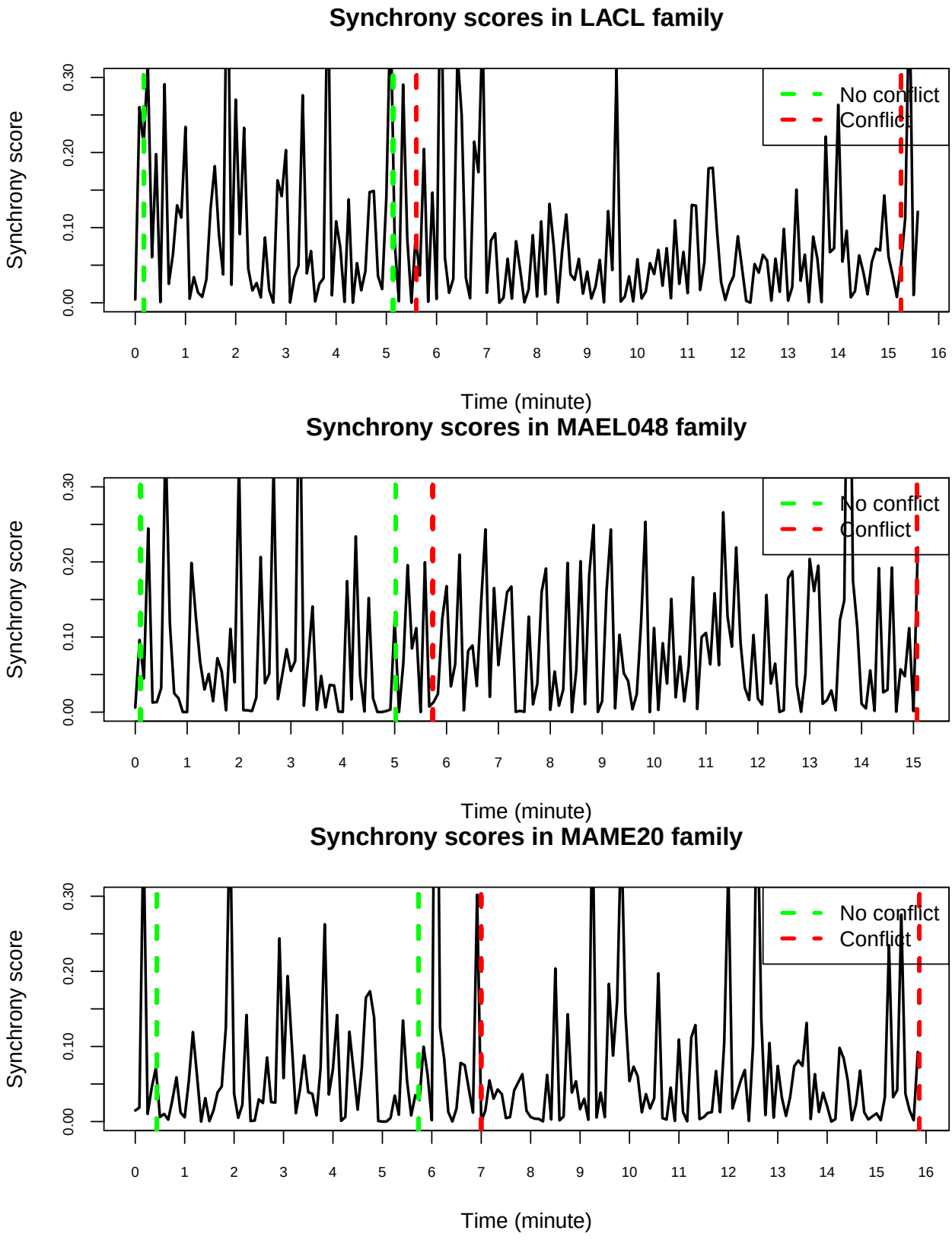

Synchrony scores in BEAM031 family
Synchrony score
0.00
0.10
0.20
0.30
No conflict
Conflict
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
Time (minute)

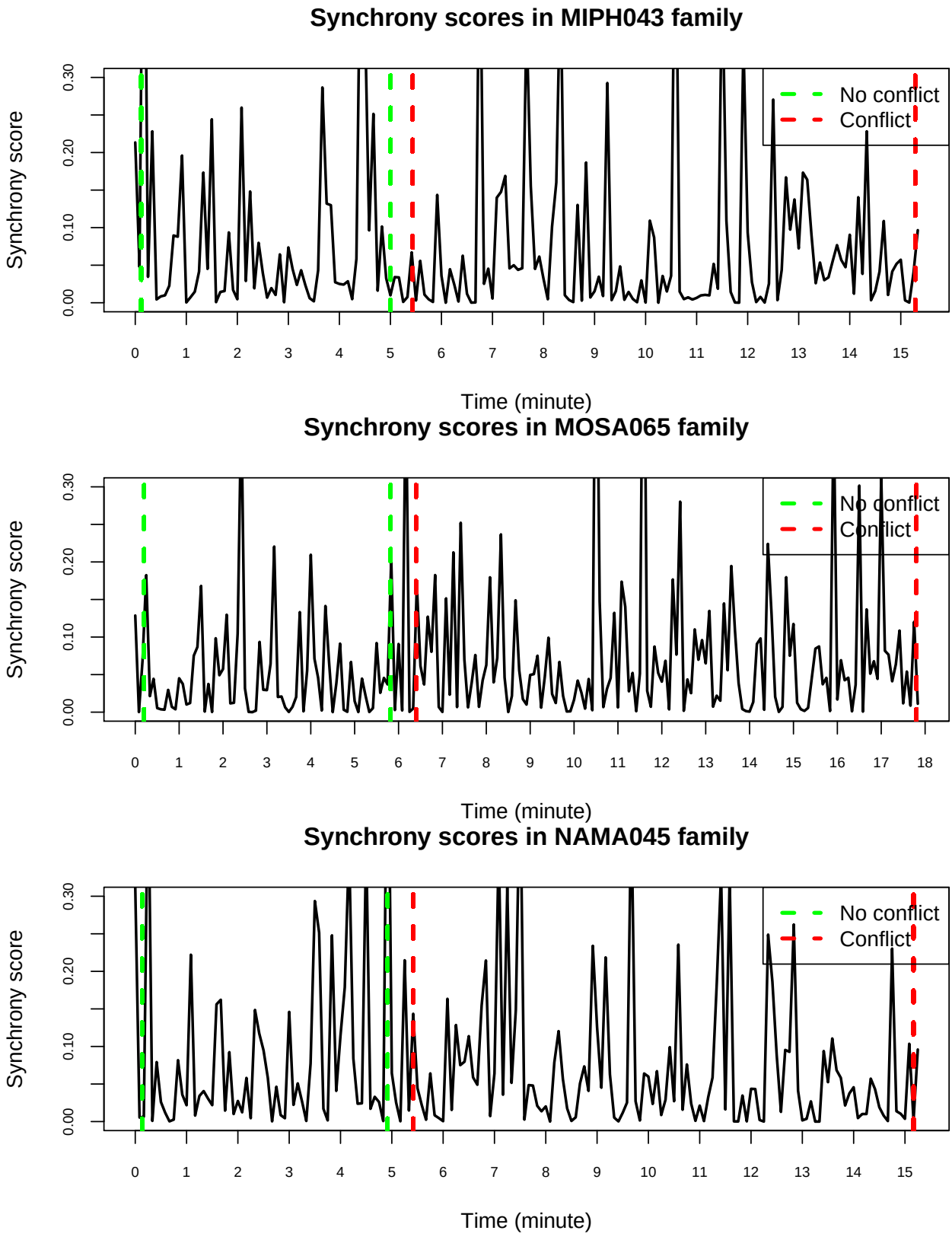

Synchrony scores in BRLO041 family
Synchrony score
0.00
0.10
0.20
0.30
No conflict
Conflict
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
Time (minute)
Synchrony scores in COLO022 family
Synchrony score
0.00
0.10
0.20
0.30
No conflict
Conflict
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
Time (minute)
Synchrony scores in DIPE004 family
Synchrony score
0.00
0.10
0.20
0.30
No conflict
Conflict
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
Time (minute)

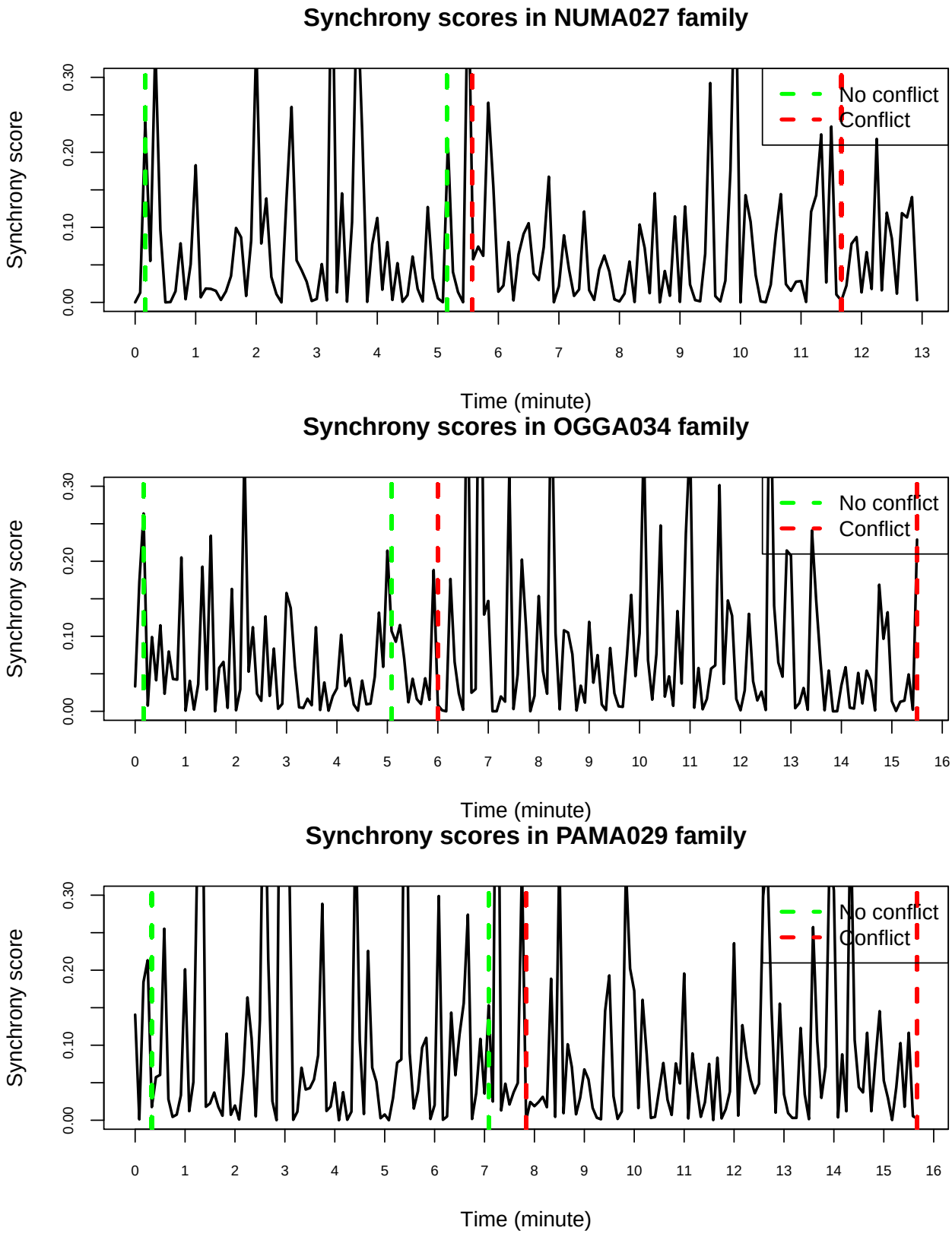

Synchrony scores in DOMA family
Synchrony score
0.00
0.10
0.20
0.30
No conflict
Conflict
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17
Time (minute)
Synchrony scores in DRNE family
Synchrony score
0.00
0.10
0.20
0.30
No conflict
Conflict
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
Time (minute)
Synchrony scores in FOMA057 family
Synchrony score
0.00
0.10
0.20
0.30
No conflict
Conflict
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
Time (minute)

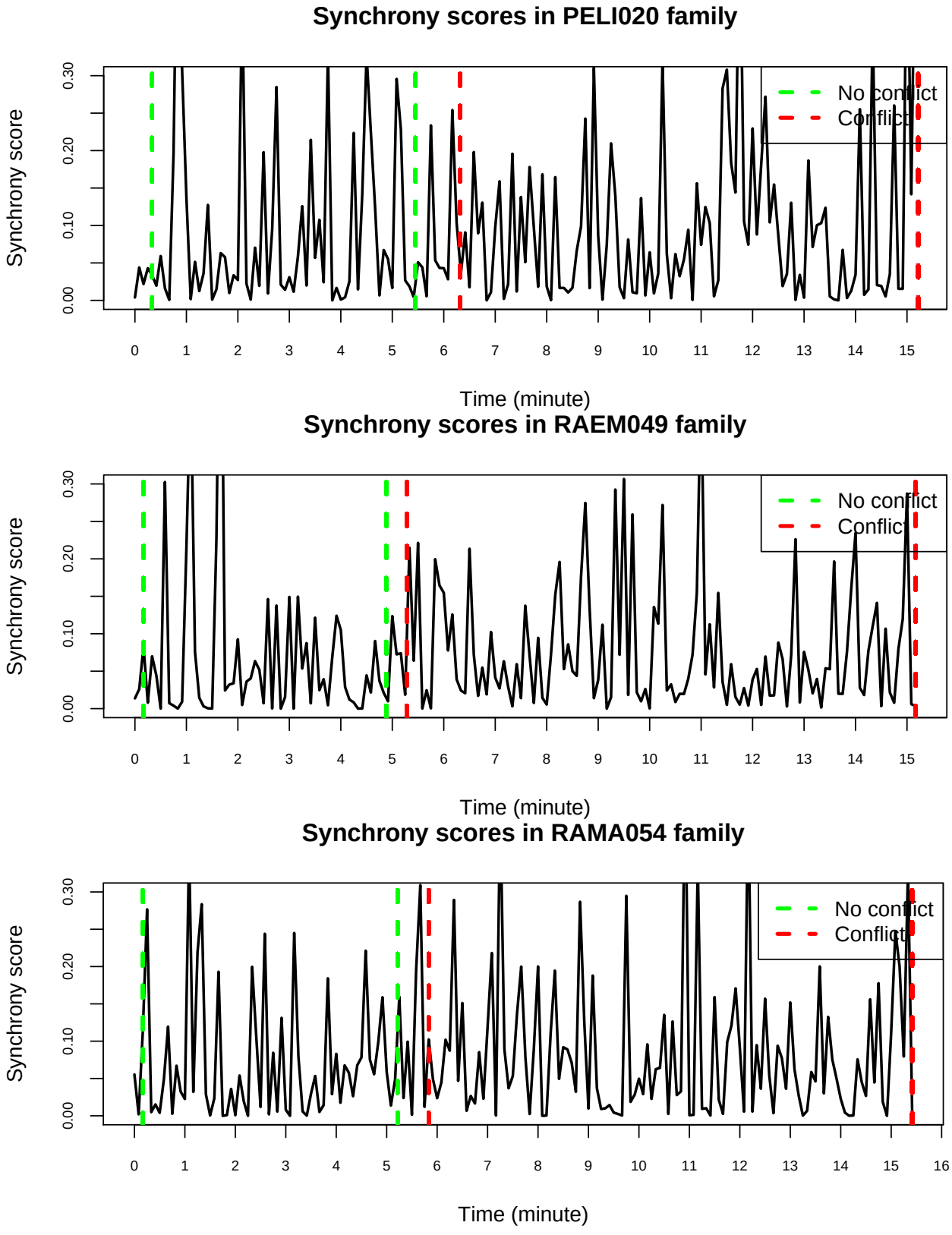

Synchrony scores in GROP039 family
Synchrony score
0.00
0.10
0.20
0.30
No conflict
Conflict
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
Time (minute)
Synchrony scores in HAJA052 family
Synchrony score
0.00
0.10
0.20
0.30
No conflict
Conflict
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
Time (minute)
Synchrony scores in HUMA058 family
Synchrony score
0.00
0.10
0.20
0.30
No conflict
Conflict
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17
Time (minute)

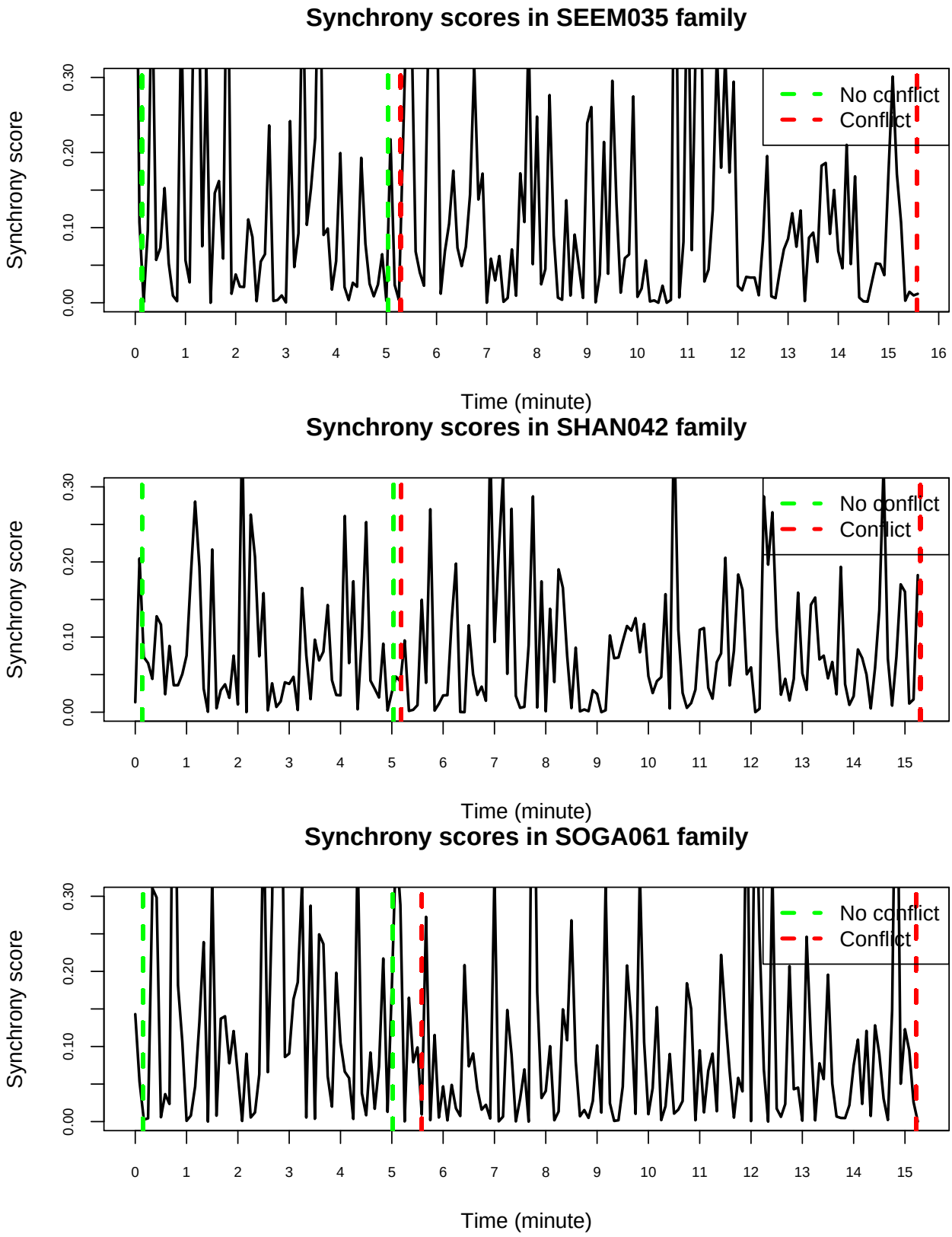

Synchrony scores in JAEM046 family
Synchrony score
0.00
0.10
0.20
0.30
No conflict
Conflict
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
Time (minute)
Synchrony scores in JEEO040 family
Synchrony score
0.00
0.10
0.20
0.30
No conflict
Conflict
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
Time (minute)
Synchrony scores in JOCE014 family
Synchrony score
0.00
0.10
0.20
0.30
No conflict
Conflict
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
Time (minute)

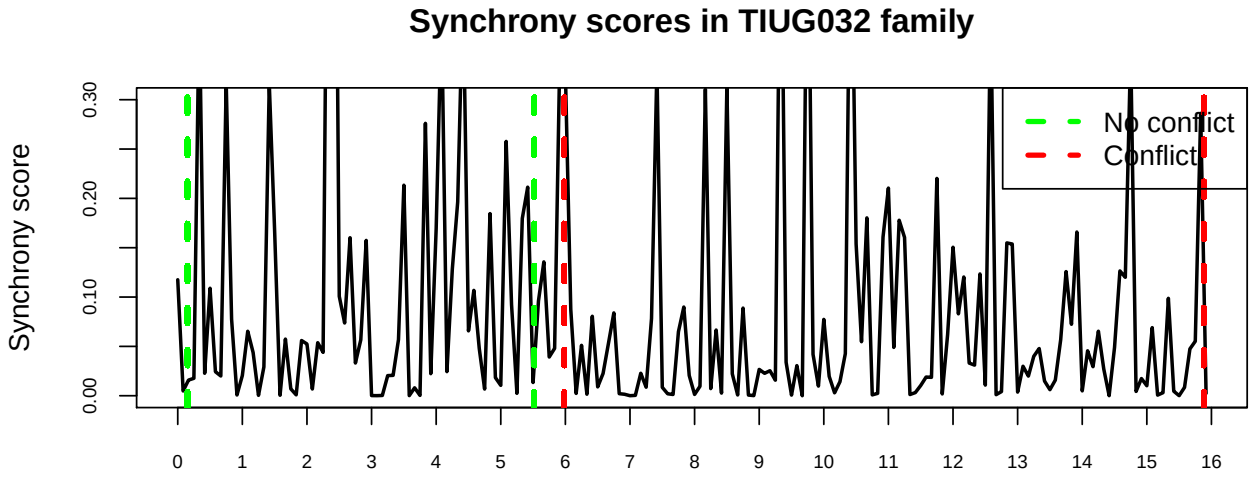

Synchrony scores in LACL family
Synchrony score
0.00
0.10
0.20
0.30
No conflict
Conflict
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
Time (minute)
Synchrony scores in MAEL048 family
Synchrony score
0.00
0.10
0.20
0.30
No conflict
Conflict
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
Time (minute)
Synchrony scores in MAME20 family
Synchrony score
0.00
0.10
0.20
0.30
No conflict
Conflict
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
Time (minute)

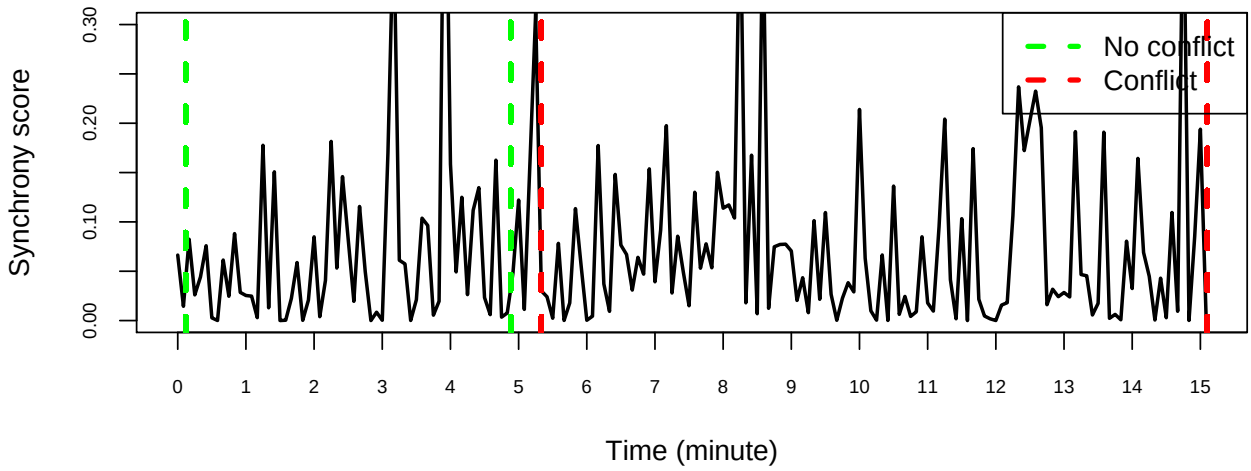

Synchrony scores in MIPH043 family
Synchrony score
0.00
0.10
0.20
0.30
No conflict
Conflict
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
Time (minute)
Synchrony scores in MOSA065 family
Synchrony score
0.00
0.10
0.20
0.30
No conflict
Conflict
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18
Time (minute)
Synchrony scores in NAMA045 family
Synchrony score
0.00
0.10
0.20
0.30
No conflict
Conflict
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
Time (minute)


Synchrony scores in NUMA027 family
Synchrony score
0.00
0.10
0.20
0.30
No conflict
Conflict
0 1 2 3 4 5 6 7 8 9 10 11 12 13
Time (minute)
Synchrony scores in OGGA034 family
Synchrony score
0.00
0.10
0.20
0.30
No conflict
Conflict
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
Time (minute)
Synchrony scores in PAMA029 family
Synchrony score
0.00
0.10
0.20
0.30
No conflict
Conflict
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
Time (minute)


Synchrony scores in PELI020 family
Synchrony score
0.00
0.10
0.20
0.30
No conflict
Conflict
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
Time (minute)
Synchrony scores in RAEM049 family
Synchrony score
0.00
0.10
0.20
0.30
No conflict
Conflict
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
Time (minute)
Synchrony scores in RAMA054 family
Synchrony score
0.00
0.10
0.20
0.30
No conflict
Conflict
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
Time (minute)


Synchrony scores in SEEM035 family
Synchrony score
0.00
0.10
0.20
0.30
No conflict
Conflict
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
Time (minute)
Synchrony scores in SHAN042 family
Synchrony score
0.00
0.10
0.20
0.30
No conflict
Conflict
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
Time (minute)
Synchrony scores in SOGA061 family
Synchrony score
0.00
0.10
0.20
0.30
No conflict
Conflict
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
Time (minute)

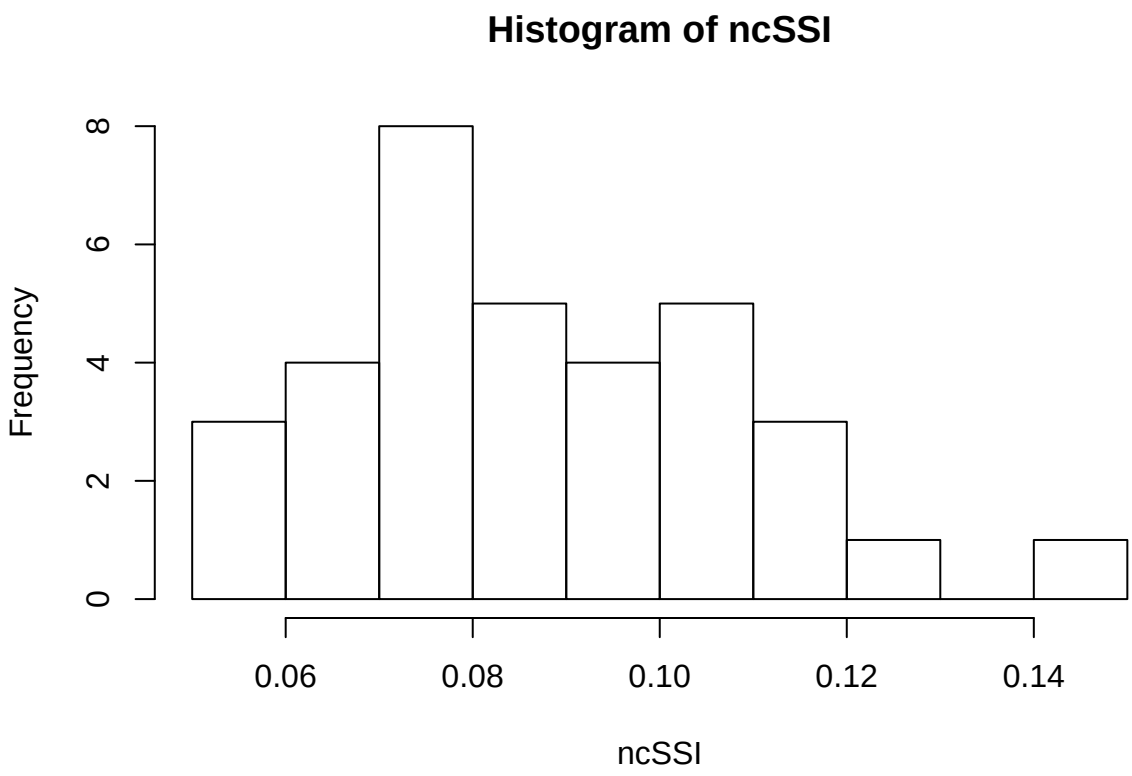




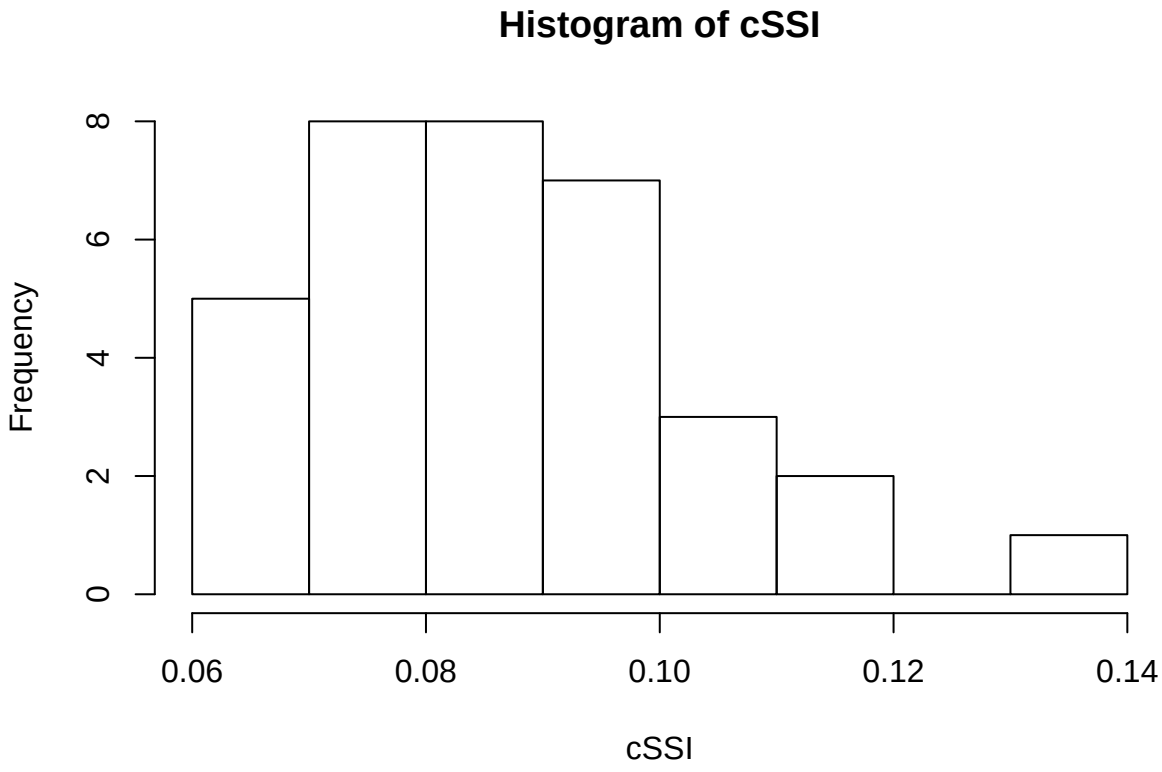

```
#SSInoLog$SSI <- as.ts(SSInoLog$SSI)

#for (i in unique(SSInoLog$family)[5:35]){
#  print(i)
#  SSI.decompose <- #decompose(SSInoLog[(which(SSInoLog$family==i)),]$SSI, type="mult")
#}
```

```
ncSSI <- c()
cSSI <- c()

for (i in unique(SSInoLog$family)){
#  print (i)
  nc <- mean(SSInoLog[which(SSInoLog$LabelVideo=="No Conflict" & SSInoLog$family==i),]$SSI)
 # print(nc)
  ncSSI <- c(ncSSI, nc)
  co <-mean(SSInoLog[which(SSInoLog$LabelVideo=="Conflict"  & SSInoLog$family==i),]$SSI)
 # print (co)
  cSSI <- c(cSSI, co)
}
```

```
hist(ncSSI)
```



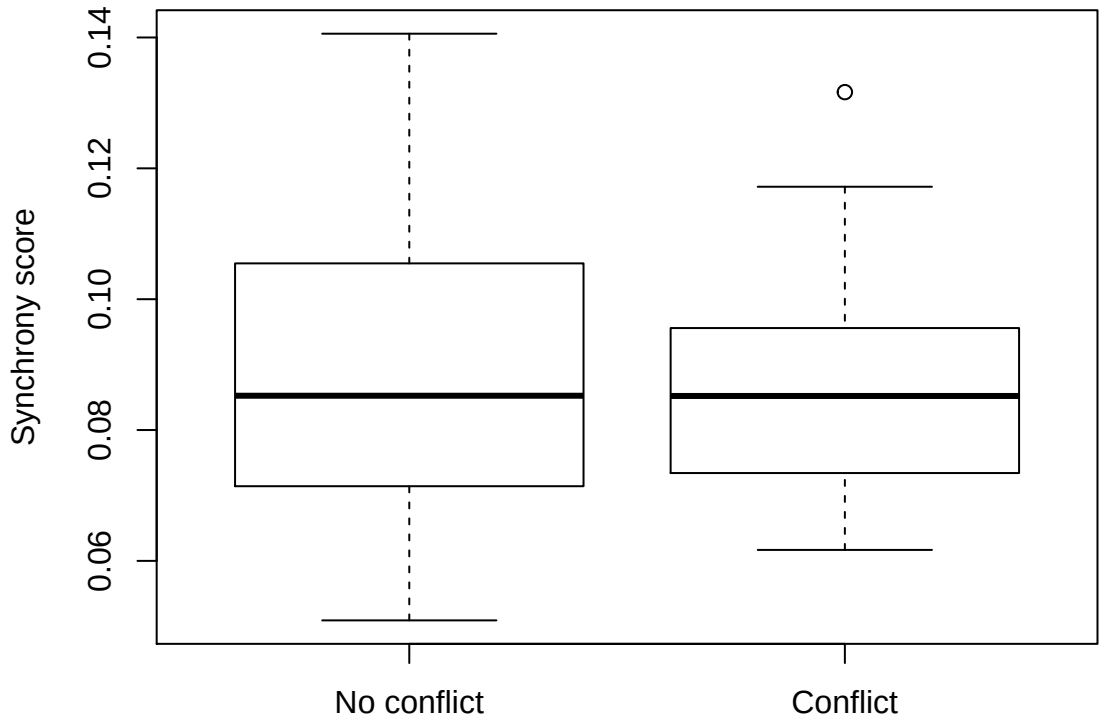

```
hist(cSSI)
```




```
sd(ncSSI)/sd(cSSI)
```

```
## [1] 1.280335
```

```
SSImean <- data.frame(sort(ncSSI, decreasing = TRUE), sort(cSSI, decreasing = TRUE))
par(mar=c(3,5,3,3))

boxplot(SSImean, names= c("No conflict", "Conflict"), ylab="Synchrony score")
```




```
#efectifs <30 donc pas de t-test
#t.test(ncSSI, cSSI, alternative = c("two.sided"),
#paired=TRUE, conf.level = 0.95)

wilcox.test(ncSSI, cSSI)
```

```
##
##  Wilcoxon rank sum test
##
## data:  ncSSI and cSSI
## W = 578, p-value = 1
## alternative hypothesis: true location shift is not equal to 0
```

There isn't any difference between the mean SSI before and after the conflict.

```
SSI <- SSInoLog[-which(SSInoLog$LabelVideo=="Cut"),]
names(SSI)
```

```
##  [1] "family"        "SSI-interval"  "Time_min"      "SSI_fa_ch"
##  [5] "SSI_mo_ch"     "SSI"           "CutBefore"     "CutMiddle1"
##  [9] "CutMiddle2"    "CutFinal"      "ChildSex"      "ParentSex"
## [13] "CutBeforeMin"  "CutMiddle1Min" "CutMiddle2Min" "CutFinalMin"
## [17] "LabelVideo"
```

```
SSIagg <- subset(SSI, select=c(family, SSI, LabelVideo))
#View(SSIagg)
names(SSIagg)
```

```
## [1] "family"     "SSI"        "LabelVideo"
```

```r
str(SSIagg)
```

```
## 'data.frame':    6040 obs. of  3 variables:
##  $ family    : Factor w/ 35 levels "1606","BAJE059",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ SSI       : num  0.1167 0.1839 0.0893 0.0274 0.1701 ...
##  $ LabelVideo: chr  "No Conflict" "No Conflict" "No Conflict" "No Conflict" ...
```

```r
SSIagg$LabelVideo <- as.factor(SSIagg$LabelVideo)
names(SSIagg)
```

```
## [1] "family"     "SSI"        "LabelVideo"
```

```r
SSIagg <-aggregate(SSIagg, by=list(SSIagg$family, SSIagg$LabelVideo), FUN=mean)
```

```
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
```

```
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
```

```
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
```

```
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
```

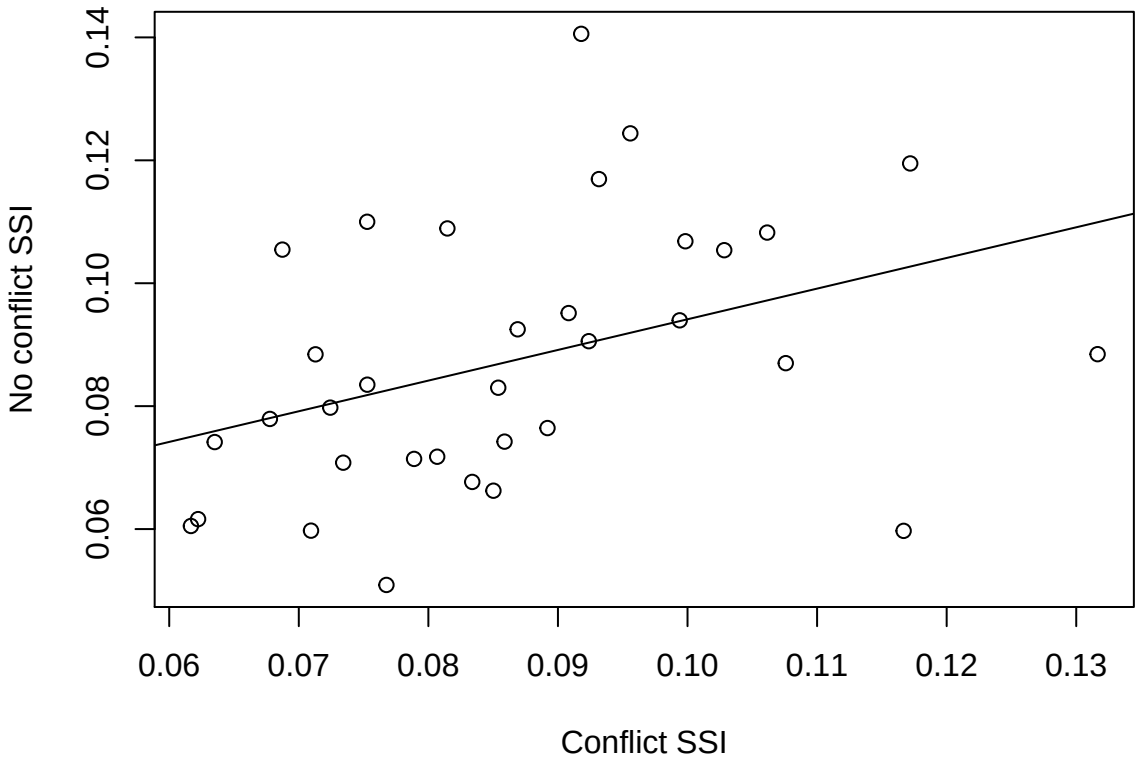
```
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
```

```
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
```

```
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
```

```
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
```

```
#View(SSIagg)
SSIagg$family <-NULL
SSIagg$LabelVideo <-NULL
names(SSIagg) <-c("family","LabelVideo", "SSI")
par(mar= c(5,5,2,2))
plot(SSIagg[which(SSIagg$LabelVideo=="Conflict"),]$SSI,SSIagg[which(SSIagg$LabelVideo=="No Conflict"),]$
model <- lm(SSIagg[which(SSIagg$LabelVideo=="No Conflict"),]$SSI ~ SSIagg[which(SSIagg$LabelVideo=="Con
abline(model)
```



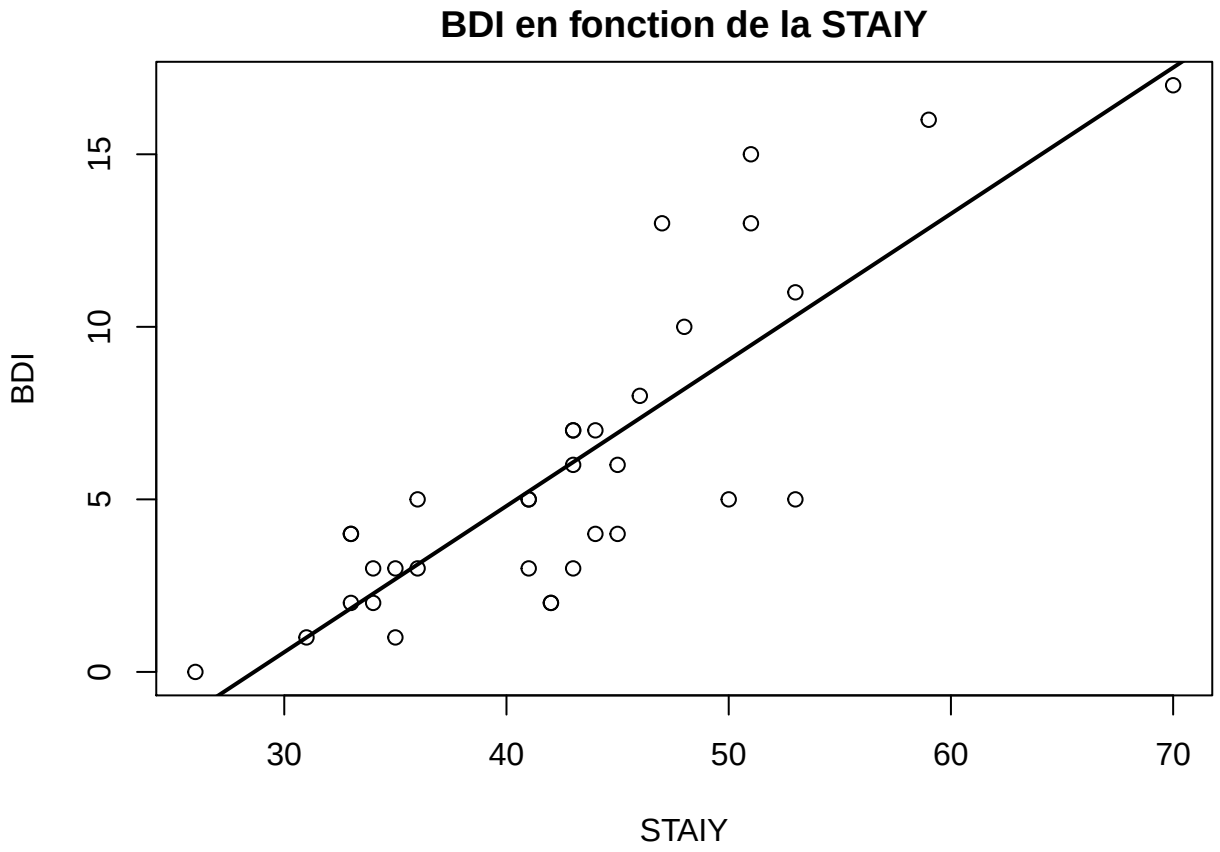

```
cor.test(SSIagg[which(SSIagg$LabelVideo=="Conflict"),]$SSI,SSIagg[which(SSIagg$LabelVideo=="No Conflict
```

```
##
##  Pearson's product-moment correlation
##
## data:  SSIagg[which(SSIagg$LabelVideo == "Conflict"), ]$SSI and SSIagg[which(SSIagg$LabelVideo == "N
## t = 2.3934, df = 32, p-value = 0.02273
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.05929851 0.64307996
## sample estimates:
##       cor
## 0.3896506
```

```
#View(SSIagg)
```

There is a correlation between the synchrony before and after the conflict.

```
psycho<-read.csv2("/Users/Ofix/Documents/Fac/internat/Recherche/projets/synchro/synchroData/Monrado/Data
#View(psycho)
psycho <- subset(psycho, select=c(Num._ident_videos, attachement_cluster, insecurite_level, TAS_total, g
#View(psycho)

par(mar=c(4,4,2,1))
plot(psycho$STAIYB_total, psycho$BDI_total, main="BDI en fonction de la STAIY", xlab="STAIY", ylab="BDI
model <- lm(psycho$BDI_total~psycho$STAIYB_total)
abline(model, lwd=2)
```



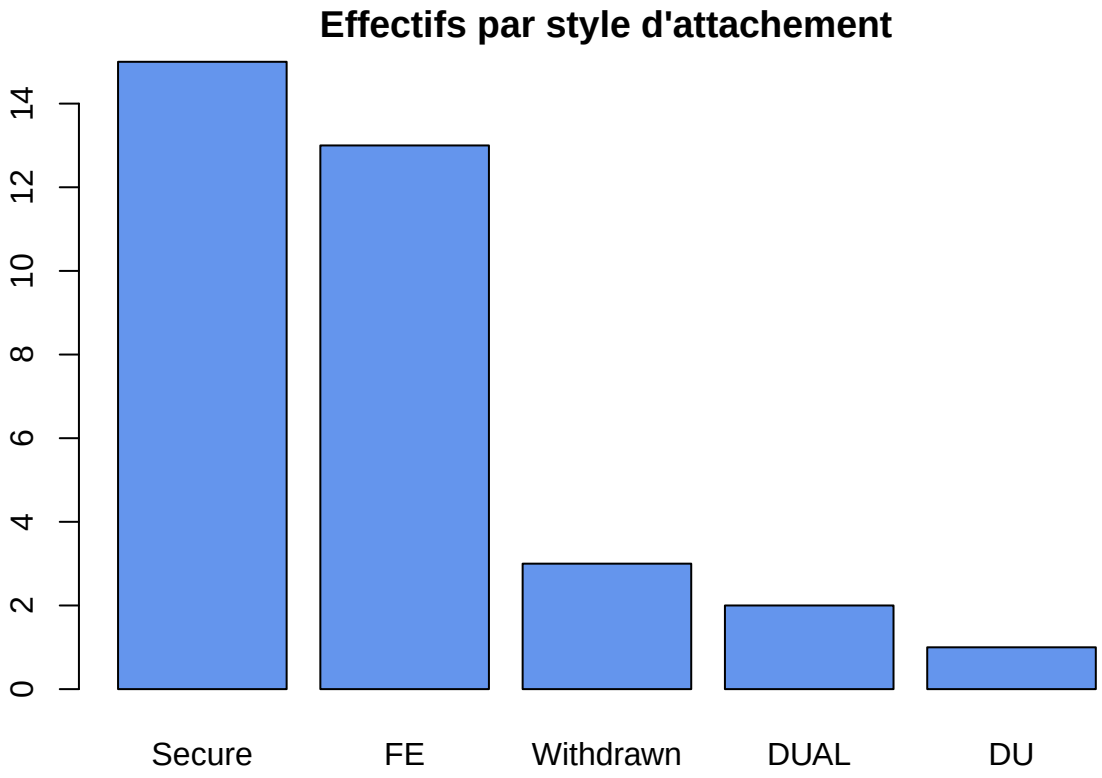

```
cor.test(psycho$BDI_total,psycho$STAIYB_total)
```

```
##
##  Pearson's product-moment correlation
##
## data:  psycho$BDI_total and psycho$STAIYB_total
## t = 8.4354, df = 32, p-value = 1.223e-09
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.6846660 0.9124366
## sample estimates:
##      cor
## 0.830537
```

```
table(psycho$attachement_cluster)
```

```
##
##        DU      DUAL        FE    Secure Withdrawn
##         1         2        13        15         3
```

```
barplot(sort(table(psycho$attachement_cluster), decreasing=TRUE), col="cornflower blue", main="Effectif
```




```
#View(SSInoLog)
SSI <- SSInoLog[-which(SSInoLog$LabelVideo=="Cut"),]
SSIagg <- subset(SSI, select=c(family, SSI, LabelVideo))
#View(SSIagg)

str(SSIagg)
```

```
## 'data.frame':    6040 obs. of  3 variables:
##  $ family    : Factor w/ 35 levels "1606","BAJE059",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ SSI       : num  0.1167 0.1839 0.0893 0.0274 0.1701 ...
##  $ LabelVideo: chr  "No Conflict" "No Conflict" "No Conflict" "No Conflict" ...
```

```
SSIagg$LabelVideo <- as.factor(SSIagg$LabelVideo)

SSIagg <-aggregate(SSIagg, by=list(SSIagg$family), FUN=mean)
```

```
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
```

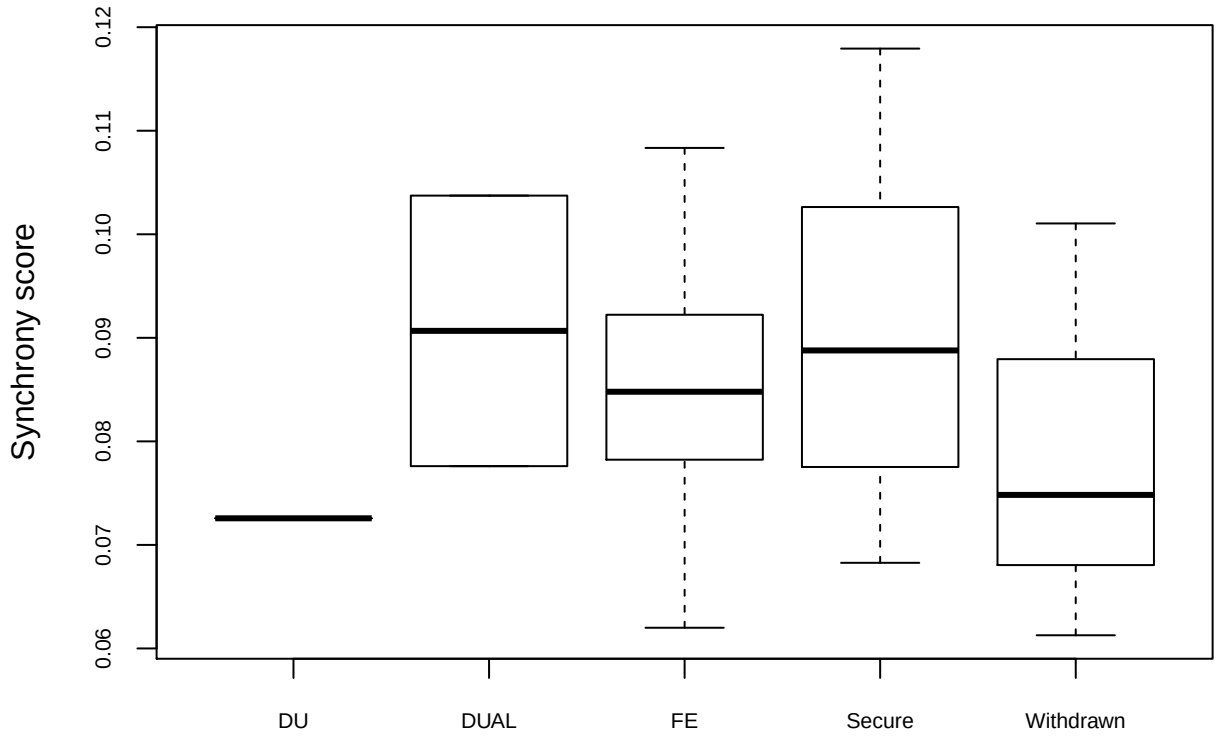
```
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
```

```
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
```

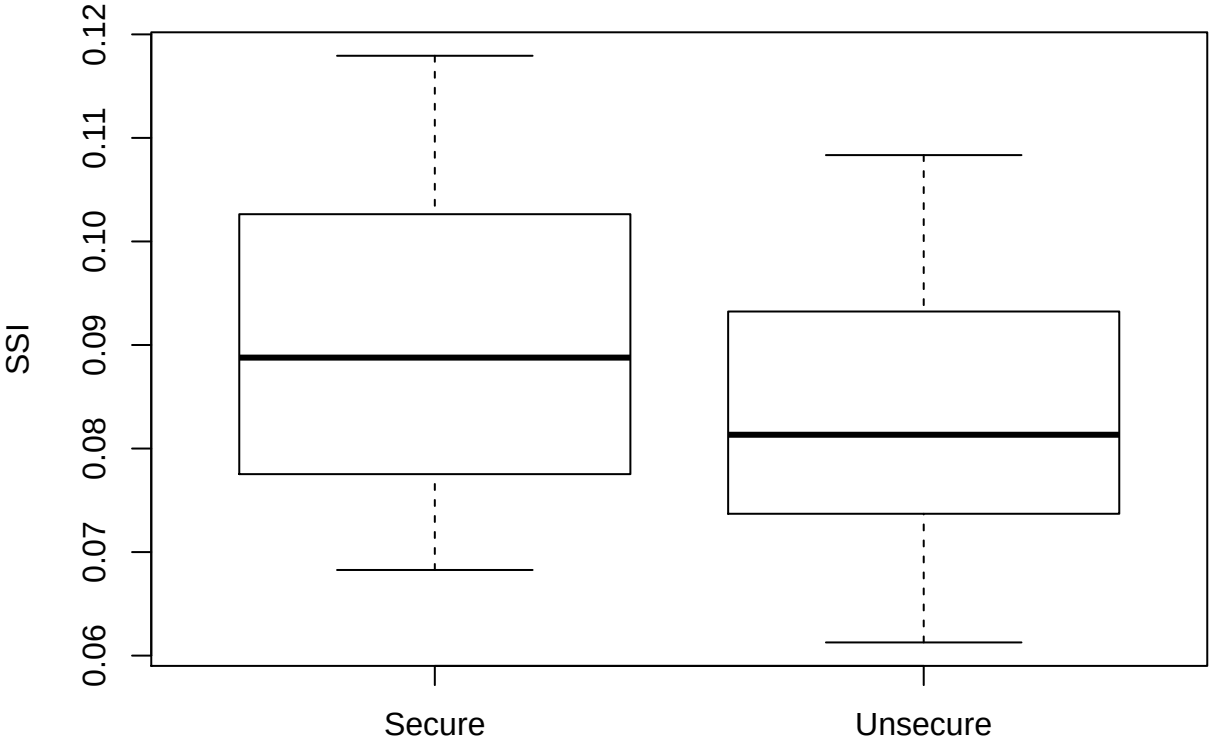
```
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
```

```
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
```

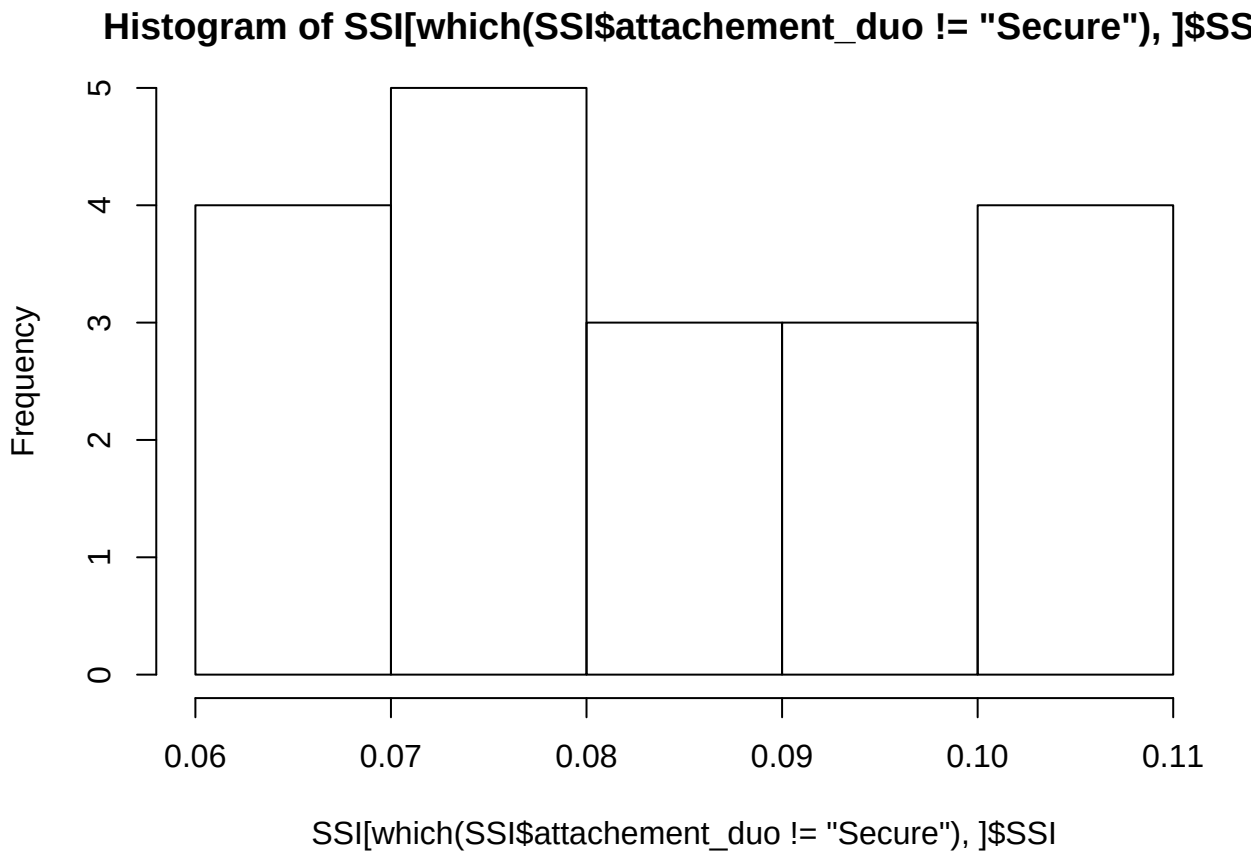
```
#View(SSIagg)
SSIagg$family <-NULL
SSIagg$LabelVideo <-NULL
names(SSIagg) <-c("family", "SSI")
SSI <- merge(SSIagg, psycho, by.x= "family", by.y= "Num._ident_videos")

#View(SSI)
SSI$attachement_duo <- rep(NA)
SSI[which(SSI$attachement_cluster=="Secure"),]$attachement_duo <- "Secure"
SSI[which(SSI$attachement_cluster!="Secure"),]$attachement_duo <- "Unsecure"

boxplot(SSI ~ attachement_cluster, data = SSI, cex.axis=0.7, ylab="Synchrony score")
```




```
boxplot(SSI ~ attachement_duo, data = SSI, ylab="SSI")
```



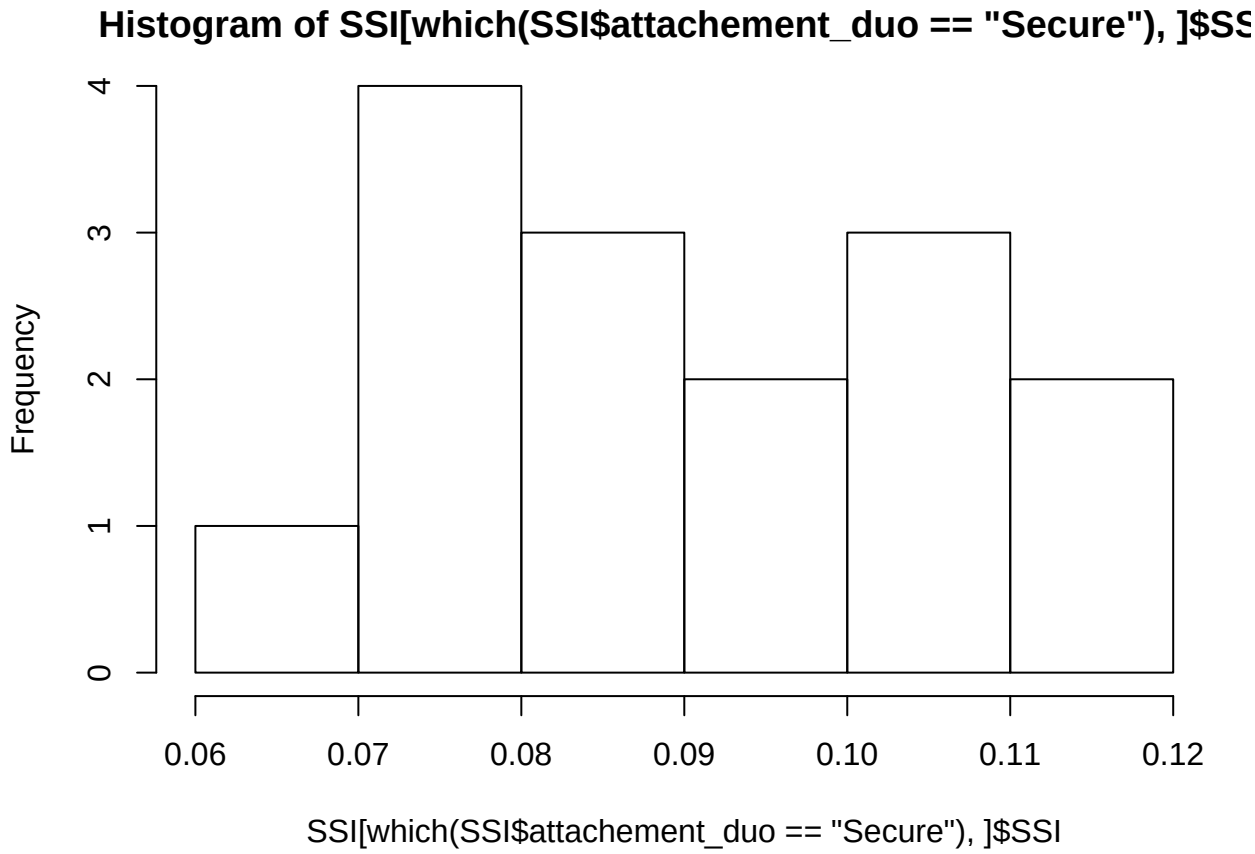

```
hist(SSI[which(SSI$attachement_duo!="Secure"),]$SSI)
```




```
sd(SSI[which(SSI$attachement_duo!="Secure"),]$SSI)
```

```
## [1] 0.01441683
```

```
hist(SSI[which(SSI$attachement_duo=="Secure"),]$SSI)
```



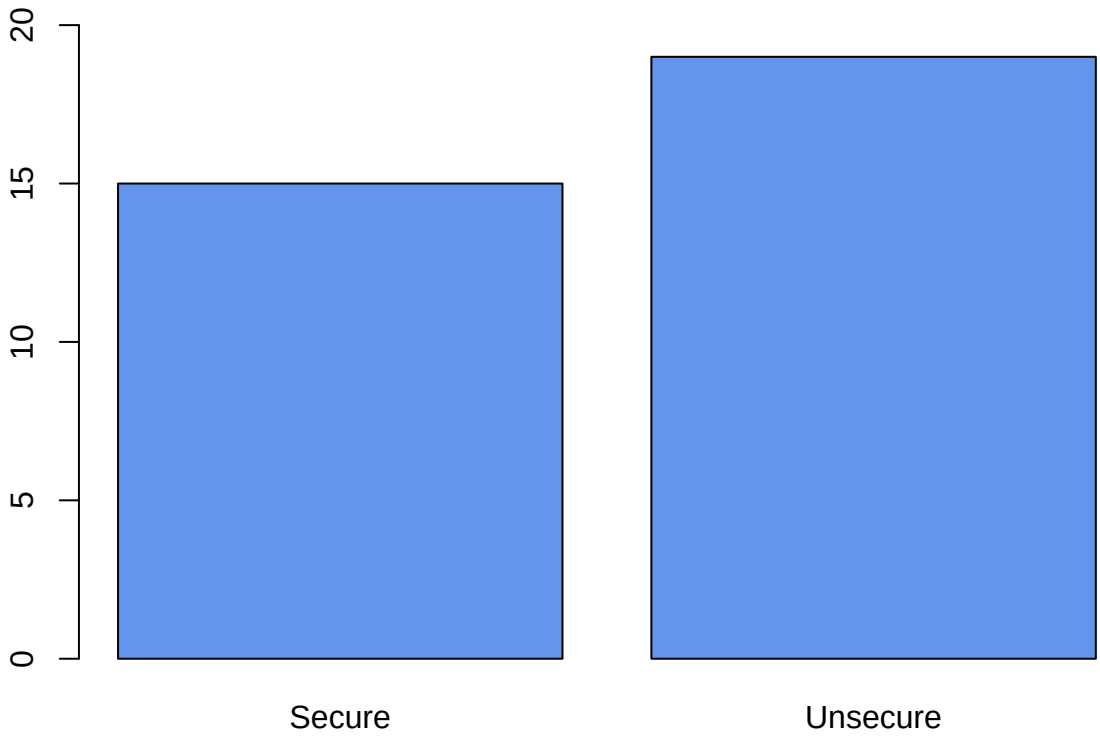

```
sd(SSI[which(SSI$attachement_duo=="Secure"),]$SSI)
```

```
## [1] 0.01607062
```

```
table(SSI$attachement_duo)
```

```
##
##   Secure Unsecure
##       15       19
```

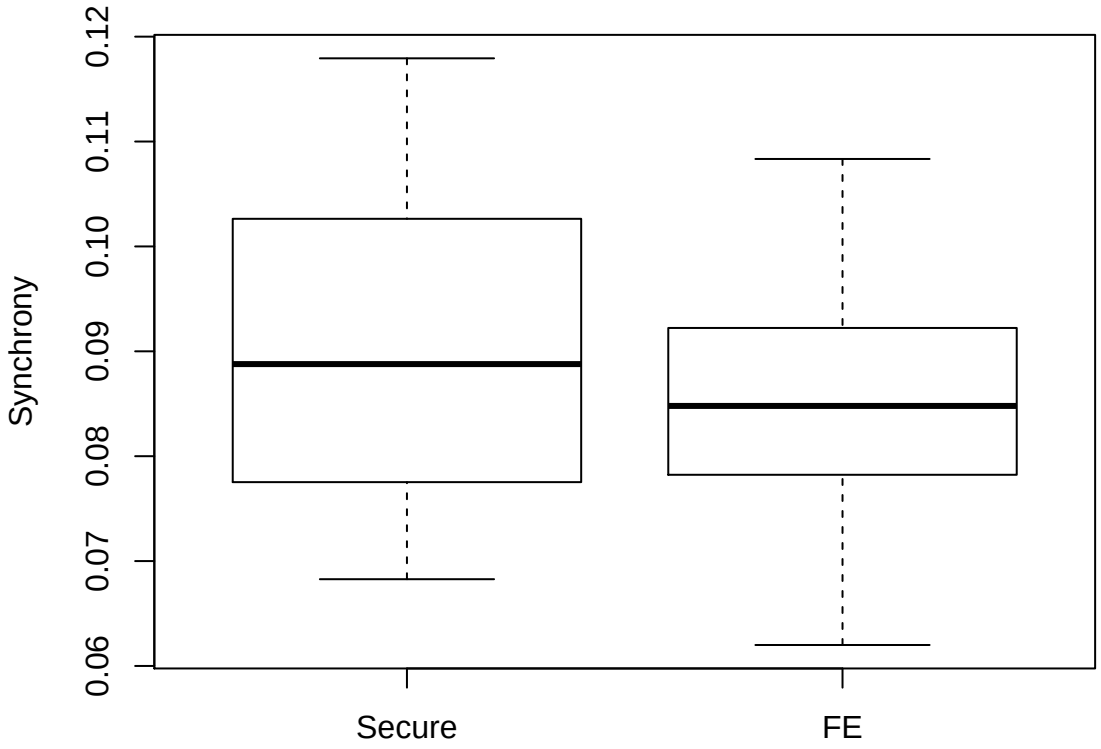
```
barplot(table(SSI$attachement_duo), ylim=c(0,20), col="cornflowerblue")
```




```
#Effectifs inf à 30 donc pas possibles
#t.test(SSI[which(SSI$attachement_duo=="Unsecure"),]$SSI, SSI[which(SSI$attachement_duo=="Secure"),]$SS

wilcox.test(SSI[which(SSI$attachement_duo=="Unsecure"),]$SSI, SSI[which(SSI$attachement_duo=="Secure"),]
```

```
##
##  Wilcoxon rank sum test
##
## data:  SSI[which(SSI$attachement_duo == "Unsecure"), ]$SSI and SSI[which(SSI$attachement_duo == "Secu
## W = 107, p-value = 0.2281
## alternative hypothesis: true location shift is not equal to 0
```

```
SSIattach <- data.frame(c(sort(SSI[which(SSI$attachement_cluster=="FE"),]$SSI, decreasing = TRUE), rep(m

#View(SSIattach)
par(mar=c(3,5,3,3))

FE <- c(sort(SSI[which(SSI$attachement_cluster=="FE"),]$SSI, decreasing = TRUE), rep(mean(SSI[which(SSI

boxplot(SSI[which(SSI$attachement_cluster=="Secure"),]$SSI, SSI[which(SSI$attachement_cluster=="FE"),]$S
```




```
t.test(SSI[which(SSI$attachement_cluster=="Secure"),]$SSI, SSI[which(SSI$attachement_cluster=="FE"),]$SS
```

```
##
##  Welch Two Sample t-test
##
## data:  SSI[which(SSI$attachement_cluster == "Secure"), ]$SSI and SSI[which(SSI$attachement_cluster ==
## t = 1.2518, df = 26, p-value = 0.2218
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.004540924  0.018684528
## sample estimates:
##  mean of x  mean of y
## 0.09132637 0.08425457
```

```
SSIM <- aggregate(SSI, by=list(SSI$family), FUN=mean)
```

```
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
```

```
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
```

```
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
```

```
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
```

```
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
```

```
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
```

```
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
```

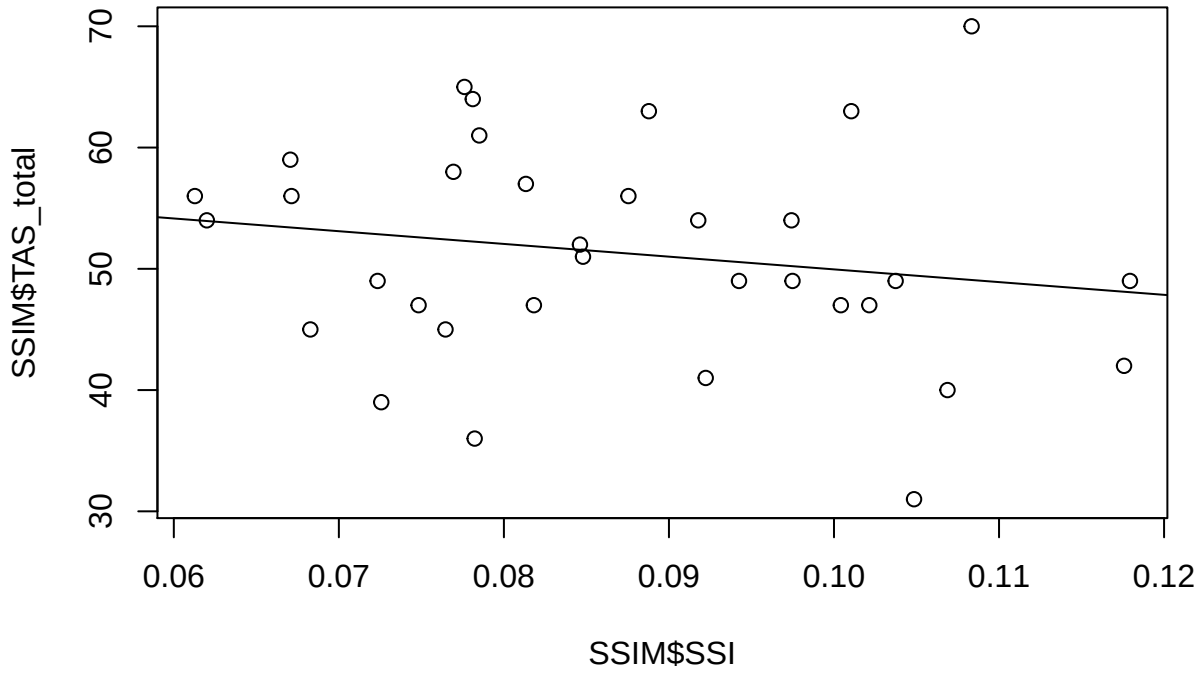
```
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
```

```
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
```

```
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
```

```
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
```

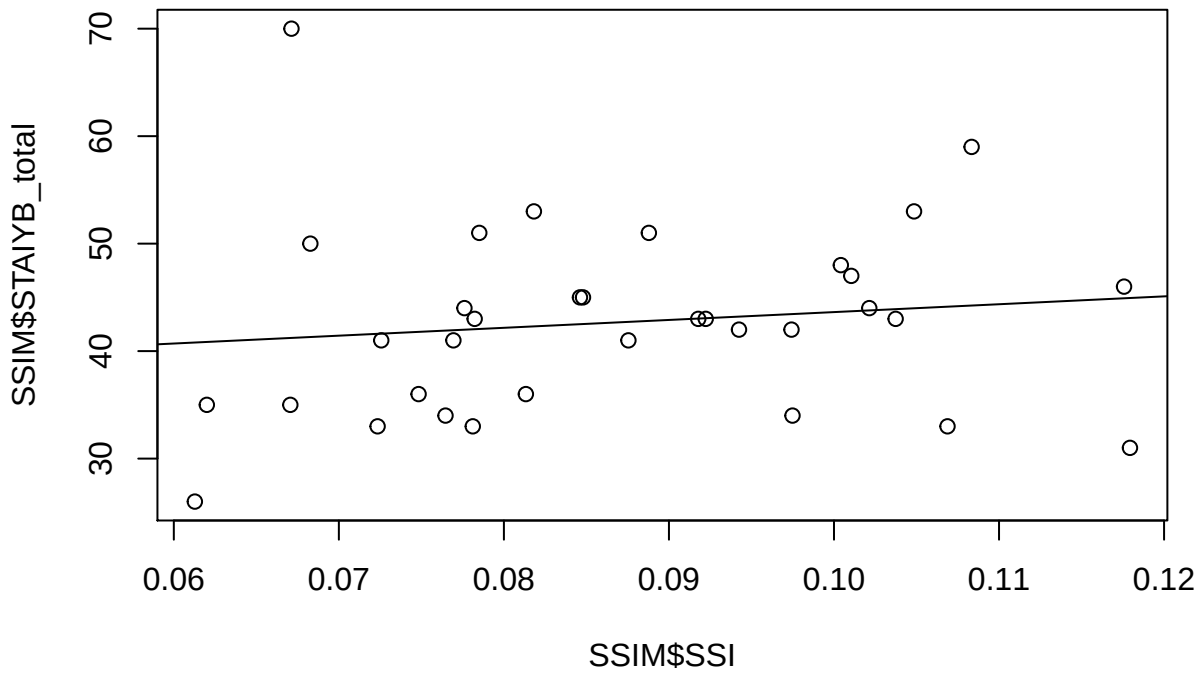
```
plot(SSIM$SSI, SSIM$TAS_total)
model <- lm(SSIM$TAS_total~SSIM$SSI)
abline(model)
```




```
cor.test(SSIM$SSI, SSIM$TAS_total)
```

```
##
##  Pearson's product-moment correlation
##
## data:  SSIM$SSI and SSIM$TAS_total
## t = -1.0528, df = 32, p-value = 0.3003
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.4907738  0.1654264
## sample estimates:
##        cor
## -0.1829752
```

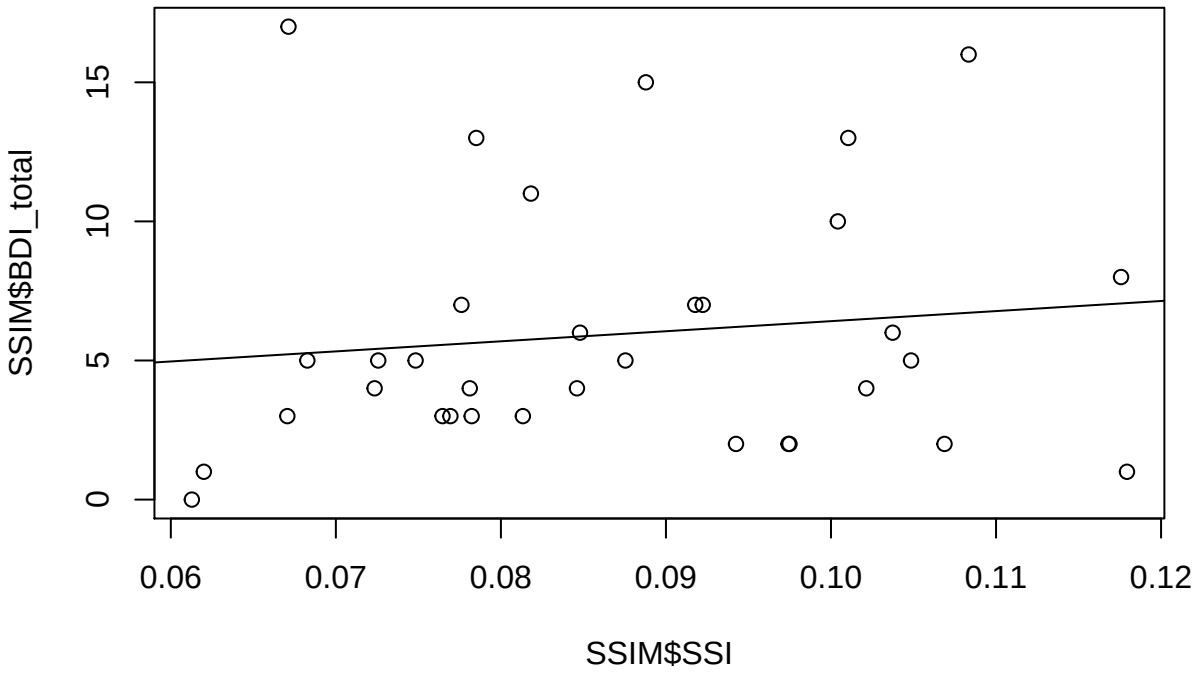
```
plot(SSIM$SSI, SSIM$STAIYB_total)
model <- lm(SSIM$STAIYB_total~SSIM$SSI)
abline(model)
```




```
cor.test(SSIM$SSI, SSIM$STAIYB_total)
```

```
##
##  Pearson's product-moment correlation
##
## data:  SSIM$SSI and SSIM$STAIYB_total
## t = 0.73236, df = 32, p-value = 0.4693
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.2192940  0.4471445
## sample estimates:
##      cor
## 0.128393
```

```
plot(SSIM$SSI, SSIM$BDI_total)
model <- lm(SSIM$BDI_total~SSIM$SSI)
abline(model)
```




```
cor.test(SSIM$SSI, SSIM$BDI_total)
```

```
##
##  Pearson's product-moment correlation
##
## data:  SSIM$SSI and SSIM$BDI_total
## t = 0.7112, df = 32, p-value = 0.4821
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.2228239  0.4441704
## sample estimates:
##       cor
## 0.1247412
```

```
#View(SSInoLog)
SSI <- SSInoLog[-which(SSInoLog$LabelVideo=="Cut"),]
SSIagg <- subset(SSI, select=c(family, SSI, LabelVideo))
#View(SSIagg)
names(SSIagg)
```

```
## [1] "family"     "SSI"        "LabelVideo"
```

```
str(SSIagg)
```

```
## 'data.frame':    6040 obs. of  3 variables:
##  $ family    : Factor w/ 35 levels "1606","BAJE059",..: 1 1 1 1 1 1 1 1 1 1 ...
##  $ SSI       : num  0.1167 0.1839 0.0893 0.0274 0.1701 ...
##  $ LabelVideo: chr  "No Conflict" "No Conflict" "No Conflict" "No Conflict" ...
```

```
SSIagg$LabelVideo <- as.factor(SSIagg$LabelVideo)
names(SSIagg)
```

```
## [1] "family"     "SSI"        "LabelVideo"
```

```r
SSIagg <-aggregate(SSIagg, by=list(SSIagg$family, SSIagg$LabelVideo), FUN=mean)
```

```
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
```

```
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
```

```
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
```

```
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
```

```
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
```

```
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
```

```
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
```

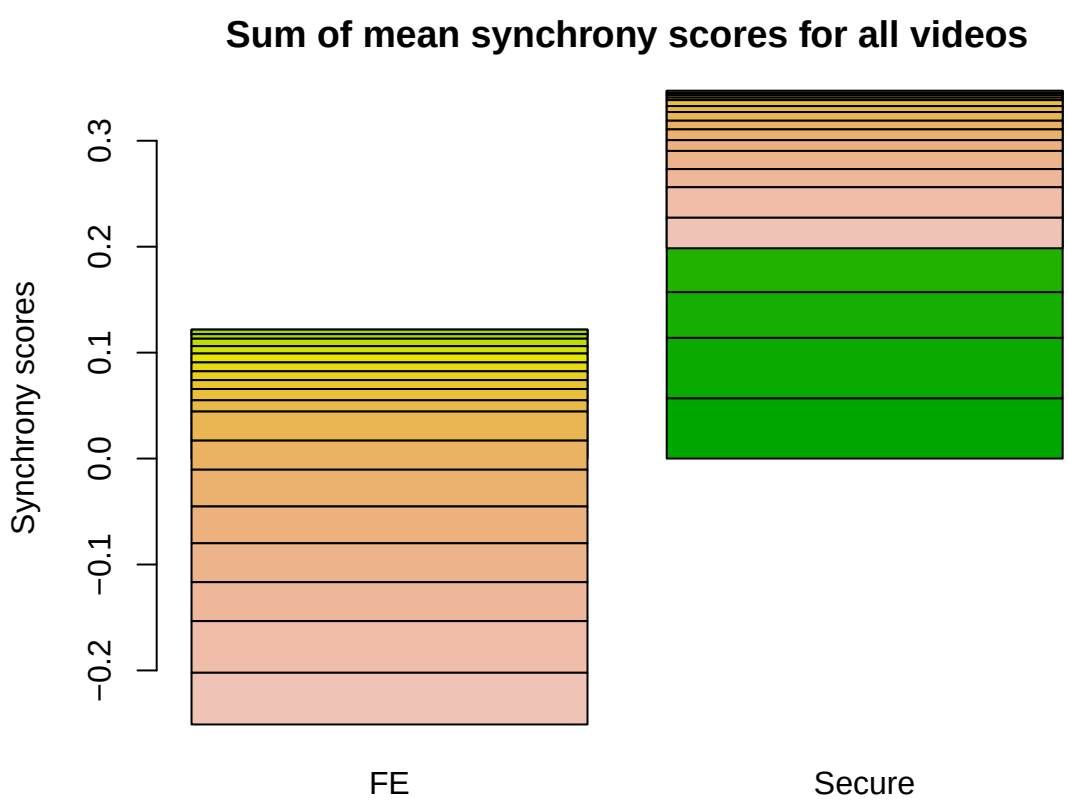
```
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
```

```r
#View(SSIagg)
SSIagg$family <-NULL
SSIagg$LabelVideo <-NULL
names(SSIagg) <-c("family", "LabelVideo", "SSI")
SSI <- merge(SSIagg, psycho, by.x= "family", by.y= "Num._ident_videos")
#View(SSI)
SSI$diff <-rep(NA)
for (i in unique(SSI$family)){
SSI[which(SSI$family==i),]$diff <-
    a <-
SSI[which(SSI$family==i & SSI$LabelVideo=="Conflict"),]$SSI - SSI[which(SSI$family==i & SSI$LabelVideo==
  print (a)
}
```

```
## [1] 0.01120112
## [1] -0.004316818
## [1] -0.008206679
## [1] 0.02058735
## [1] 0.04319279
## [1] 0.008910005
## [1] 0.05697003
## [1] -0.00699422
## [1] -0.01064564
```

```
## [1] 0.001173828
## [1] -0.002096179
## [1] -0.005612297
## [1] -0.002542164
## [1] 0.01571619
## [1] -0.0237813
## [1] -0.02880027
## [1] 0.001811938
## [1] -0.03673722
## [1] 0.01275727
## [1] 0.0006123615
## [1] -0.01714405
## [1] 0.02585718
## [1] -0.007329623
## [1] -0.01014714
## [1] 0.01877874
## [1] -0.02744715
## [1] 0.00543184
## [1] 0.007484757
## [1] 0.01165041
## [1] -0.002303032
## [1] 0.002393419
## [1] -0.04878173
## [1] -0.03472463
## [1] 0.002624708
```

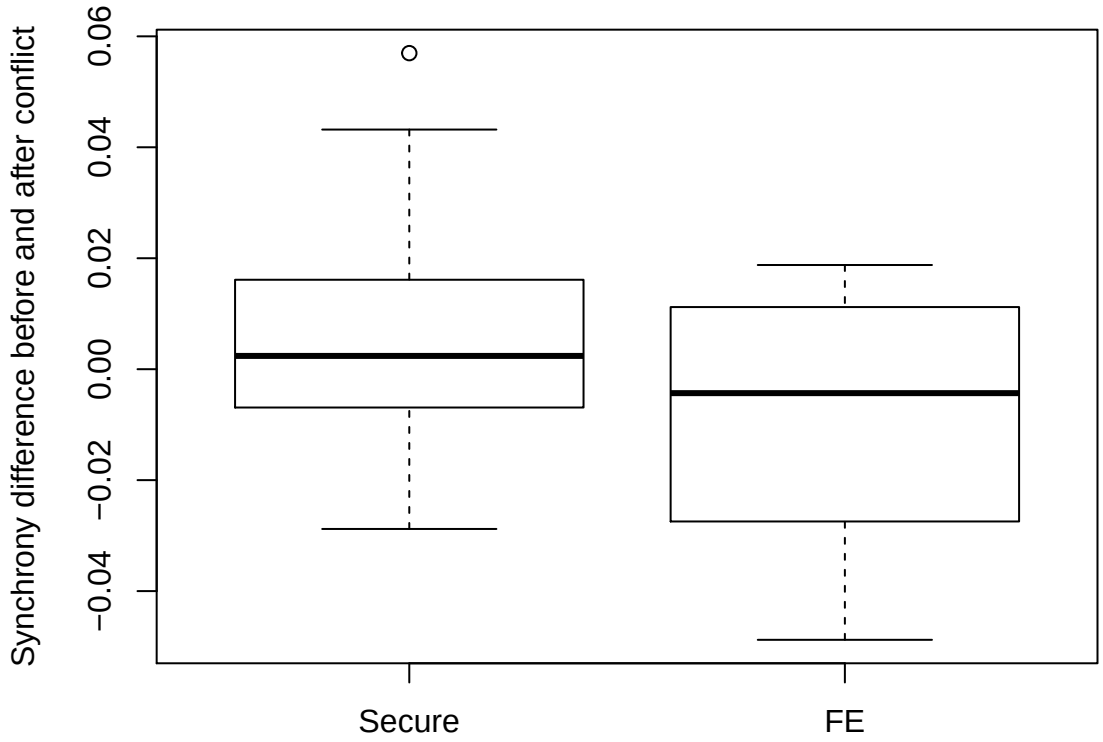
```
SSIdiff <- data.frame(sort(c(SSI[which(SSI$attachement_cluster=="FE"),]$diff, rep(mean(SSI[which(SSI$at
par(mar=c(3,5,3,3))
barplot(as.matrix(SSIdiff), col=terrain.colors(34), ylab="Synchrony scores", main="Sum of mean synchron
```



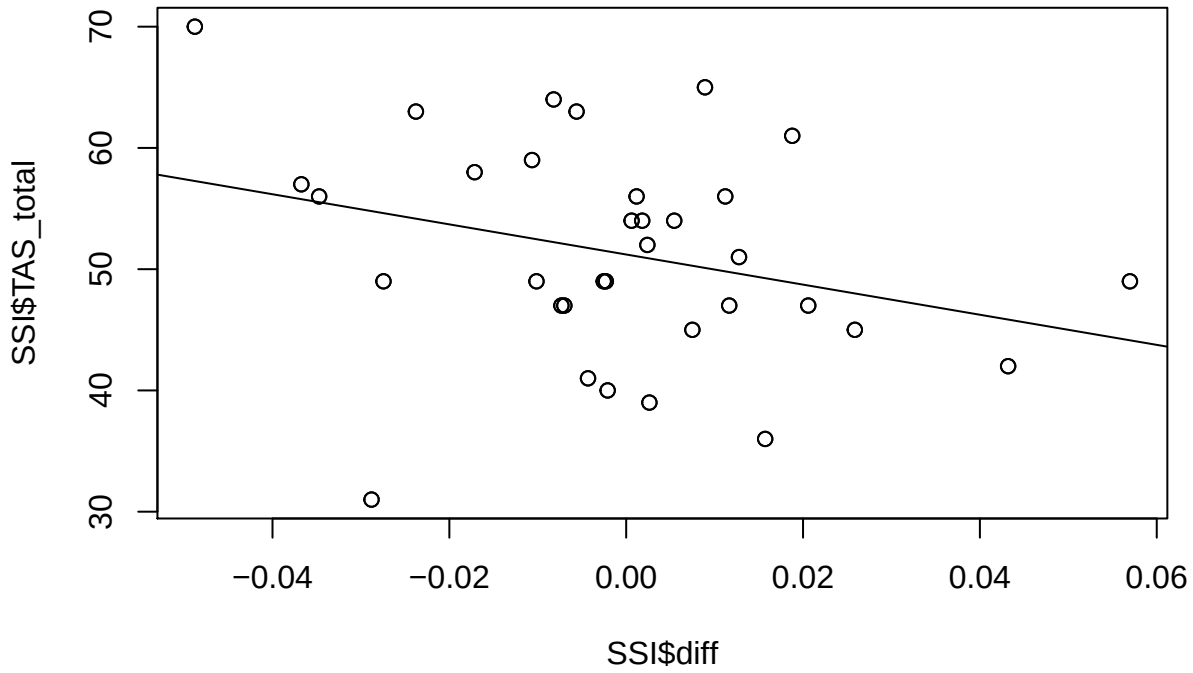
```
FE <- c(sort(SSI[which(SSI$attachement_cluster=="FE"),]$SSI, decreasing = TRUE), rep(mean(SSI[which(SSI

t.test(SSI[which(SSI$attachement_cluster=="Secure" & SSI$LabelVideo=="Conflict"),]$diff, SSI[which(SSI$a
```

```
##
##  Welch Two Sample t-test
##
## data:  SSI[which(SSI$attachement_cluster == "Secure" & SSI$LabelVideo ==  and SSI[which(SSI$attachem
## t = 1.7738, df = 25.578, p-value = 0.088
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.002393389  0.032361620
## sample estimates:
##    mean of x    mean of y
##  0.006617209 -0.008366906
```

```
boxplot(SSI[which(SSI$attachement_cluster=="Secure" & SSI$LabelVideo=="Conflict"),]$diff, SSI[which(SSI
```




```
#View(SSI)
plot(SSI$diff, SSI$TAS_total)
model <- lm(SSI$TAS_total~SSI$diff)
abline(model)
```




```
cor.test(SSI[which(SSI$LabelVideo=="Conflict"),]$diff, SSI[which(SSI$LabelVideo=="Conflict"),]$TAS_total
```

```
##
##  Pearson's product-moment correlation
##
## data:  SSI[which(SSI$LabelVideo == "Conflict"), ]$diff and SSI[which(SSI$LabelVideo == "Conflict"), ]
## t = -1.7862, df = 32, p-value = 0.08355
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.58018843  0.04126969
## sample estimates:
##        cor
## -0.3010981
```

```
table(SSI$insecurite_level)
```

```
##
##                         Clearly        Markedly           Midly
##               4              28               8               2
##          Mildly Mildly insecure Mildly Insecure      Moderately
##              10               2               2               4
##       Moderatly
##               8
```

```
#Quantite de mvt et BDI
str(MH)
```

```
## 'data.frame':    802834 obs. of  18 variables:
##  $ family       : chr  "1606" "1606" "1606" "1606" ...
##  $ slidedFather : num  0.002084 0.001758 0.000729 0.000693 0.000537 ...
##  $ slidedMother : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ slidedChild  : num  2.57e-03 1.55e-03 2.54e-04 2.09e-04 5.28e-05 ...
##  $ frame_index  : int  1 2 3 4 5 6 7 8 9 10 ...
##  $ slidedParent : num  0.002084 0.001758 0.000729 0.000693 0.000537 ...
```

```
##  $ CutBefore    : chr  "00:11" "00:11" "00:11" "00:11" ...
##  $ CutMiddle1   : chr  "05:30" "05:30" "05:30" "05:30" ...
##  $ CutMiddle2   : chr  "06:16" "06:16" "06:16" "06:16" ...
##  $ CutFinal     : chr  "16:27" "16:27" "16:27" "16:27" ...
##  $ ChildSex     : Factor w/ 2 levels "Female","Male": 1 1 1 1 1 1 1 1 1 1 ...
##  $ ParentSex    : Factor w/ 2 levels "Female","Male": 2 2 2 2 2 2 2 2 2 2 ...
##  $ CutBeforeMin : num  0.183 0.183 0.183 0.183 0.183 ...
##  $ CutMiddle1Min: num  5.5 5.5 5.5 5.5 5.5 5.5 5.5 5.5 5.5 5.5 ...
##  $ CutMiddle2Min: num  6.27 6.27 6.27 6.27 6.27 ...
##  $ CutFinalMin  : num  16.4 16.4 16.4 16.4 16.4 ...
##  $ LabelVideo   : chr  "Cut" "Cut" "Cut" "Cut" ...
##  $ Time_min     : num  0.000667 0.001333 0.002 0.002667 0.003333 ...
```

```
str(psycho)
```

```
## 'data.frame':    34 obs. of  7 variables:
##  $ Num._ident_videos  : Factor w/ 34 levels "1606","BAJE059",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ attachement_cluster: Factor w/ 5 levels "DU","DUAL","FE",..: 3 3 4 4 4 2 4 3 3 5 ...
##  $ insecurite_level   : Factor w/ 9 levels "","Clearly","Markedly",..: 3 5 2 2 2 3 2 7 4 5 ...
##  $ TAS_total          : int  56 41 64 47 42 65 49 47 59 56 ...
##  $ global_score       : Factor w/ 14 levels "","10 Midly Fearful",..: 6 10 5 5 5 13 12 11 2 3 ...
##  $ STAIYB_total       : int  70 43 33 48 46 44 34 44 35 26 ...
##  $ BDI_total          : int  17 7 4 10 8 7 2 4 3 0 ...
```

```
MH1 <- merge(MH, psycho, by.x="family", by.y="Num._ident_videos")
MH1 <- MH1[-which(MH1$LabelVideo=="Cut"),]
str(MH1)
```

```
## 'data.frame':    755257 obs. of  24 variables:
##  $ family             : chr  "1606" "1606" "1606" "1606" ...
##  $ slidedFather       : num  0.000116 0.000327 0.000807 0.001484 0.002285 ...
##  $ slidedMother       : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ slidedChild        : num  0.00671 0.00784 0.00867 0.00907 0.00888 ...
##  $ frame_index        : int  276 277 278 279 280 281 282 283 284 285 ...
##  $ slidedParent       : num  0.000116 0.000327 0.000807 0.001484 0.002285 ...
##  $ CutBefore          : chr  "00:11" "00:11" "00:11" "00:11" ...
##  $ CutMiddle1         : chr  "05:30" "05:30" "05:30" "05:30" ...
##  $ CutMiddle2         : chr  "06:16" "06:16" "06:16" "06:16" ...
##  $ CutFinal           : chr  "16:27" "16:27" "16:27" "16:27" ...
##  $ ChildSex           : Factor w/ 2 levels "Female","Male": 1 1 1 1 1 1 1 1 1 1 ...
##  $ ParentSex          : Factor w/ 2 levels "Female","Male": 2 2 2 2 2 2 2 2 2 2 ...
##  $ CutBeforeMin       : num  0.183 0.183 0.183 0.183 0.183 ...
##  $ CutMiddle1Min      : num  5.5 5.5 5.5 5.5 5.5 5.5 5.5 5.5 5.5 5.5 ...
##  $ CutMiddle2Min      : num  6.27 6.27 6.27 6.27 6.27 ...
##  $ CutFinalMin        : num  16.4 16.4 16.4 16.4 16.4 ...
##  $ LabelVideo         : chr  "No Conflict" "No Conflict" "No Conflict" "No Conflict" ...
##  $ Time_min           : num  0.184 0.185 0.185 0.186 0.187 ...
##  $ attachement_cluster: Factor w/ 5 levels "DU","DUAL","FE",..: 3 3 3 3 3 3 3 3 3 3 ...
##  $ insecurite_level   : Factor w/ 9 levels "","Clearly","Markedly",..: 3 3 3 3 3 3 3 3 3 3 ...
##  $ TAS_total          : int  56 56 56 56 56 56 56 56 56 56 ...
##  $ global_score       : Factor w/ 14 levels "","10 Midly Fearful",..: 6 6 6 6 6 6 6 6 6 6 ...
##  $ STAIYB_total       : int  70 70 70 70 70 70 70 70 70 70 ...
##  $ BDI_total          : int  17 17 17 17 17 17 17 17 17 17 ...
```

```r
MHagg <- aggregate(MH1, by=list(MH1$family, MH1$LabelVideo), FUN=mean)
```

```
## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
```

```
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
```

```
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
```

```
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
```

```
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
```

```
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
```

```
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
```

```
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
```

```
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
```

```
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
```

```
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
```

```
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
```

```
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
```

```
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
```

```
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
```

```
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
```

```
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
```

```
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
```

```
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
```

```
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
```

```
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
```

```
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
```

```
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
```

```
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
```

```
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
```

```
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
```

```
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
```

```
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
```

```
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
```

```
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
```

```
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
```

```
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
```

```
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
```

```
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
```

```
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
```

```
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
```

```
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
```

```
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
```

```
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
```

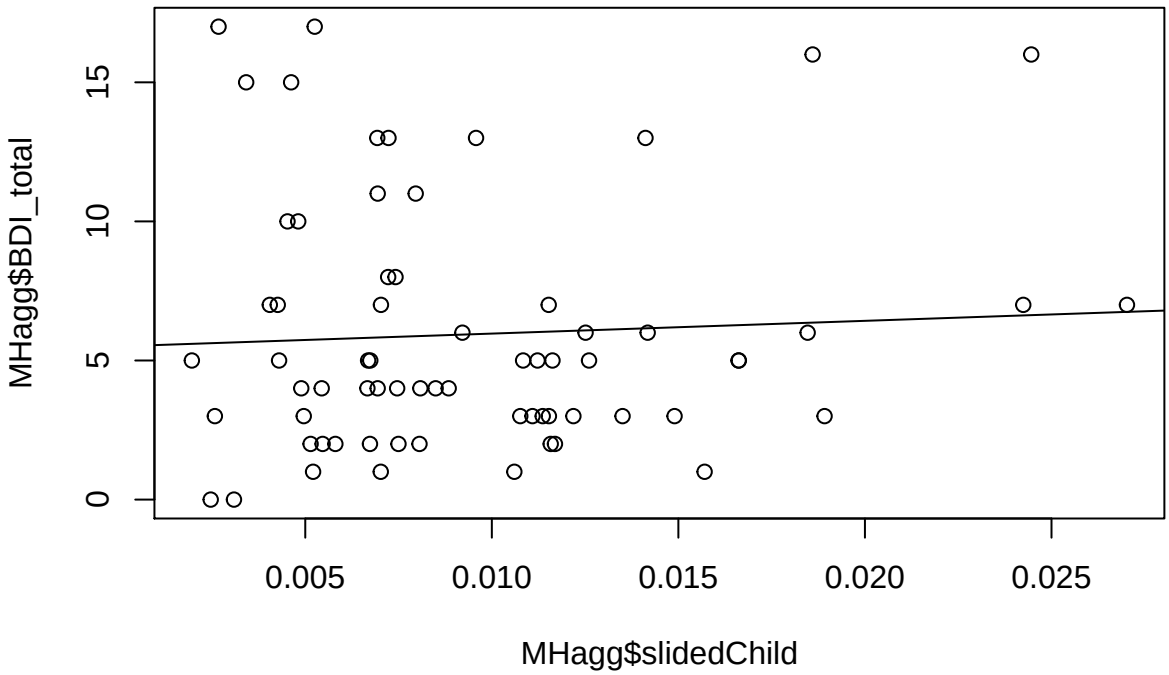
```
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
```

```
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
```

```
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA
```

```r
#View(MHagg)
```

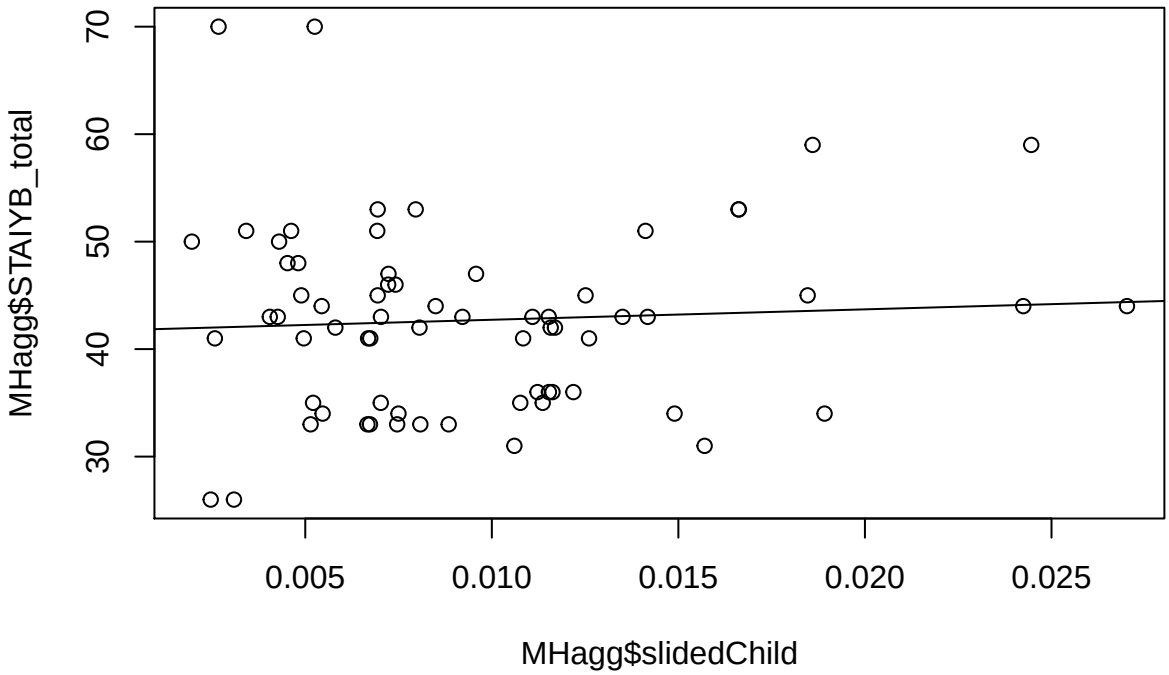
```r
plot(MHagg$slidedChild, MHagg$BDI_total)
lm <- lm(MHagg$BDI_total~MHagg$slidedChild)
abline(lm)
```




```
cor.test(MHagg$slidedChild,MHagg$BDI_total)
```

```
##
##  Pearson's product-moment correlation
##
## data:  MHagg$slidedChild and MHagg$BDI_total
## t = 0.45377, df = 66, p-value = 0.6515
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.1851177  0.2903334
## sample estimates:
##        cor
## 0.05576882
```

```
#View(MH1)
plot(MHagg$slidedChild, MHagg$STAIYB_total)
lm <- lm(MHagg$STAIYB_total~MHagg$slidedChild)
abline(lm)
```




```
cor.test(MHagg$slidedChild,MHagg$STAIYB_total)
```

```
##
##  Pearson's product-moment correlation
##
## data:  MHagg$slidedChild and MHagg$STAIYB_total
## t = 0.48863, df = 66, p-value = 0.6267
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.1809776  0.2942512
## sample estimates:
##        cor
## 0.06003815
```

```
library(reshape2)
SSIPsycho <- merge(SSInoLog, psycho, by.x="family", by.y="Num._ident_videos")
data_long <- melt(SSInoLog, id.vars = c('family', 'ChildSex', "ParentSex"),  measure.vars = c(1,2,3,4))
```

```
## Warning: attributes are not identical across measure variables; they will
## be dropped
```

```
SSIPsycho$CutBefore <-NULL
SSIPsycho$SSI_fa_ch <-NULL
SSIPsycho$SSI_mo_ch <-NULL
SSIPsycho$CutMiddle1 <-NULL
SSIPsycho$CutMiddle2 <-NULL
SSIPsycho$CutFinal <-NULL
SSIPsycho$CutBeforeMin <-NULL
SSIPsycho$CutMiddle1Min <-NULL
SSIPsycho$CutMiddle2Min <-NULL
SSIPsycho$CutFinalMin <-NULL
SSIPsycho$global_score <-NULL
SSIPsycho <- SSIPsycho[-which(SSIPsycho$LabelVideo=="Cut"),]
#View(data_long)
```

```r
write.csv(SSIPsycho, file="SSIPsycho.csv")

SSIPsycho$LabelVideo <- as.factor(SSIPsycho$LabelVideo)

mod1 <- glmer(LabelVideo ~ SSI + (1|family), data=SSIPsycho, family=binomial, nAGQ=20)
summary(mod1)
```

```
## Generalized linear mixed model fit by maximum likelihood (Adaptive
##   Gauss-Hermite Quadrature, nAGQ = 20) [glmerMod]
##  Family: binomial  ( logit )
## Formula: LabelVideo ~ SSI + (1 | family)
##    Data: SSIPsycho
##
##      AIC      BIC   logLik deviance df.resid
##   7784.2   7804.3  -3889.1   7778.2     6037
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -0.7363 -0.7245 -0.7236  1.3793  1.3823
##
## Random effects:
##  Groups Name        Variance Std.Dev.
##  family (Intercept) 0        0
## Number of obs: 6040, groups:  family, 34
##
## Fixed effects:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.64745    0.03400 -19.043   <2e-16 ***
## SSI          0.04105    0.23601   0.174    0.862
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##     (Intr)
## SSI -0.605
```

```r
mod <- glmer(LabelVideo ~ SSI + (1|family) + ChildSex + ParentSex +  attachement_cluster, data=SSIPsycho
summary(mod)
```

```
## Generalized linear mixed model fit by maximum likelihood (Adaptive
##   Gauss-Hermite Quadrature, nAGQ = 20) [glmerMod]
##  Family: binomial  ( logit )
## Formula:
## LabelVideo ~ SSI + (1 | family) + ChildSex + ParentSex + attachement_cluster
##    Data: SSIPsycho
##
##      AIC      BIC   logLik deviance df.resid
##   7794.6   7855.0  -3888.3   7776.6     6031
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -0.7553 -0.7292 -0.7181  1.3571  1.4347
##
## Random effects:
```

```
##  Groups Name        Variance Std.Dev.
##  family (Intercept) 0        0
## Number of obs: 6040, groups:  family, 34
##
## Fixed effects:
##                             Estimate Std. Error z value Pr(>|z|)
## (Intercept)                -0.72196    0.16244  -4.445 8.81e-06 ***
## SSI                         0.03907    0.23641   0.165    0.869
## ChildSexMale                0.02778    0.07215   0.385    0.700
## ParentSexMale              -0.05291    0.07342  -0.721    0.471
## attachement_clusterDUAL     0.09810    0.19702   0.498    0.619
## attachement_clusterFE       0.11124    0.16917   0.658    0.511
## attachement_clusterSecure   0.06600    0.16770   0.394    0.694
## attachement_clusterWithdrawn 0.01896   0.18736   0.101    0.919
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) SSI    ChldSM PrntSM a_DUAL att_FE attc_S
## SSI         -0.106
## ChildSexMal -0.004  0.034
## ParentSexMl -0.002  0.020  0.013
## attchm_DUAL -0.813 -0.022 -0.001  0.000
## attchmnt_FE -0.947 -0.021 -0.068 -0.134  0.783
## attchmnt_cS -0.955 -0.031 -0.089 -0.064  0.790  0.934
## attchmnt_cW -0.856 -0.012 -0.133 -0.002  0.707  0.832  0.842
```

```
mod <- glmer(attachement_cluster ~ SSI + (1|family) + LabelVideo, data=SSIPsycho, family=binomial , nAG
summary(mod)
```

```
## Generalized linear mixed model fit by maximum likelihood (Adaptive
##   Gauss-Hermite Quadrature, nAGQ = 20) [glmerMod]
##  Family: binomial  ( logit )
## Formula: attachement_cluster ~ SSI + (1 | family) + LabelVideo
##    Data: SSIPsycho
##
##      AIC      BIC   logLik deviance df.resid
##     28.8     55.6    -10.4     20.8     6036
##
## Scaled residuals:
##       Min        1Q    Median        3Q       Max
## -0.007148  0.000000  0.000000  0.000000  0.000000
##
## Random effects:
##  Groups Name        Variance Std.Dev.
##  family (Intercept) 9036     95.06
## Number of obs: 6040, groups:  family, 34
##
## Fixed effects:
##                        Estimate Std. Error z value Pr(>|z|)
## (Intercept)             44.8007    49.3331   0.908    0.364
## SSI                      0.7120    59.2376   0.012    0.990
## LabelVideoNo Conflict    0.0394    12.7557   0.003    0.998
##
## Correlation of Fixed Effects:
```

```
##             (Intr) SSI
## SSI         -0.094
## LblVdNCnflc -0.086  0.002
```

```
mod <- glmer(attachement_cluster ~ SSI + (1|family) + LabelVideo, data=SSIPsycho, family=binomial , nAG
summary(mod)
```

```
## Generalized linear mixed model fit by maximum likelihood (Adaptive
##   Gauss-Hermite Quadrature, nAGQ = 20) [glmerMod]
##  Family: binomial  ( logit )
## Formula: attachement_cluster ~ SSI + (1 | family) + LabelVideo
##    Data: SSIPsycho
##
##      AIC      BIC   logLik deviance df.resid
##     28.8     55.6    -10.4     20.8     6036
##
## Scaled residuals:
##       Min        1Q    Median        3Q       Max
## -0.007148  0.000000  0.000000  0.000000  0.000000
##
## Random effects:
##  Groups Name        Variance Std.Dev.
##  family (Intercept) 9036     95.06
## Number of obs: 6040, groups:  family, 34
##
## Fixed effects:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)            44.8007    49.3331   0.908    0.364
## SSI                     0.7120    59.2376   0.012    0.990
## LabelVideoNo Conflict   0.0394    12.7557   0.003    0.998
##
## Correlation of Fixed Effects:
##             (Intr) SSI
## SSI         -0.094
## LblVdNCnflc -0.086  0.002
```

```
mod <- glmer(attachement_cluster ~ SSI + (1|family) + LabelVideo, data=SSIPsycho, family=binomial , nAG
summary(mod)
```

```
## Generalized linear mixed model fit by maximum likelihood (Adaptive
##   Gauss-Hermite Quadrature, nAGQ = 20) [glmerMod]
##  Family: binomial  ( logit )
## Formula: attachement_cluster ~ SSI + (1 | family) + LabelVideo
##    Data: SSIPsycho
##
##      AIC      BIC   logLik deviance df.resid
##     28.8     55.6    -10.4     20.8     6036
##
## Scaled residuals:
##       Min        1Q    Median        3Q       Max
## -0.007148  0.000000  0.000000  0.000000  0.000000
##
## Random effects:
##  Groups Name        Variance Std.Dev.
##  family (Intercept) 9036     95.06
```

```
## Number of obs: 6040, groups:  family, 34
##
## Fixed effects:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)            44.8007    49.3331   0.908    0.364
## SSI                     0.7120    59.2376   0.012    0.990
## LabelVideoNo Conflict   0.0394    12.7557   0.003    0.998
##
## Correlation of Fixed Effects:
##             (Intr) SSI
## SSI         -0.094
## LblVdNCnflc -0.086  0.002
```

```
SSIPsycho$attachement_duo <- rep(NA)
SSIPsycho[which(SSIPsycho$attachement_cluster=="Secure"),]$attachement_duo <- "Secure"
SSIPsycho[which(SSIPsycho$attachement_cluster!="Secure"),]$attachement_duo <- "Unsecure"
SSIPsycho$attachement_duo <- as.factor(SSIPsycho$attachement_duo)

mod <- glmer(attachement_duo ~ SSI + (1|family) + LabelVideo, data=SSIPsycho, family=binomial , nAGQ=20)
summary(mod)
```

```
## Generalized linear mixed model fit by maximum likelihood (Adaptive
##   Gauss-Hermite Quadrature, nAGQ = 20) [glmerMod]
##  Family: binomial  ( logit )
## Formula: attachement_duo ~ SSI + (1 | family) + LabelVideo
##    Data: SSIPsycho
##
##      AIC      BIC   logLik deviance df.resid
##     61.7     88.6    -26.9     53.7     6036
##
## Scaled residuals:
##       Min        1Q    Median        3Q       Max
## -0.006484 -0.005592  0.004850  0.005008  0.005612
##
## Random effects:
##  Groups Name        Variance Std.Dev.
##  family (Intercept) 2090     45.72
## Number of obs: 6040, groups:  family, 34
##
## Fixed effects:
##                        Estimate Std. Error z value Pr(>|z|)
## (Intercept)            1.313203   4.507116   0.291    0.771
## SSI                   -0.305572  11.498227  -0.027    0.979
## LabelVideoNo Conflict  0.006044   2.761918   0.002    0.998
##
## Correlation of Fixed Effects:
##             (Intr) SSI
## SSI         -0.222
## LblVdNCnflc -0.211 -0.003
```

```
mod1 <- glmer(LabelVideo ~ SSI + (1|family) + attachement_duo, data=SSIPsycho, family=binomial, nAGQ=20)
summary(mod1)
```

```
## Generalized linear mixed model fit by maximum likelihood (Adaptive
##   Gauss-Hermite Quadrature, nAGQ = 20) [glmerMod]
```

```
##  Family: binomial  ( logit )
## Formula: LabelVideo ~ SSI + (1 | family) + attachement_duo
##    Data: SSIPsycho
##
##      AIC      BIC   logLik deviance df.resid
##   7786.1   7812.9  -3889.0   7778.1     6036
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -0.7401 -0.7267 -0.7204  1.3753  1.3898
##
## Random effects:
##  Groups Name        Variance Std.Dev.
##  family (Intercept) 0        0
## Number of obs: 6040, groups:  family, 34
##
## Fixed effects:
##                          Estimate Std. Error z value Pr(>|z|)
## (Intercept)             -0.65834    0.04632 -14.212   <2e-16 ***
## SSI                      0.04390    0.23615   0.186    0.853
## attachement_duoUnsecure  0.01894    0.05463   0.347    0.729
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) SSI
## SSI         -0.467
## attchmnt_dU -0.679  0.035
```