

# Synchrony in Psychotherapy, example with F1044 patient data

*Thomas Gargot*

*April, 26th 2016*

## Contents

Import data . . . . .	2
<b>Lists</b>	<b>2</b>
Functions list . . . . .	2
Constants generated from data and defining it (data list, ) . . . . .	3
Merge data frame, compute Time in minutes, compute log of motion history dataframe . . . . .	4
<b>Presentation of the data</b>	<b>4</b>
<b>Length of the videos in minutes</b>	<b>6</b>
Length of the videos in number of frames . . . . .	7
Number of Available (True) and Not Available (False) data for each participant . . . . .	7
<b>Global Motion history</b>	<b>8</b>
Mean Motion history by video by participant . . . . .	8
Raw Motion history by video by participant . . . . .	8
Normalized log Motion history by video by participant . . . . .	12
<b>Raw data and mean of Motion History on sliding and non overlapping intervals on 1st video F1044C1 video</b>	<b>16</b>
Raw data . . . . .	16
Sliding interval . . . . .	18
Non overlapping interval . . . . .	19
<b>Focus on the motion history of the first 10 seconds of the first video(C)</b>	<b>20</b>
Sliding interval function on a raw data, 5 frames interval . . . . .	20
Sliding interval function on log data, 5 frames interval . . . . .	22
Motion history of the father during 10-20 seconds of the first video(C) . . . . .	24
<b>Mean motion history by minute plots</b>	<b>26</b>
<b>Mean log motion history by minute plots</b>	<b>39</b>

Motion history by minute for the F1044C video	52
Export no log filtered data in text files	64
Export log filtered data in text files	67
SyncPy utilisation for creating synchrony dataframe	70
Description of SSI data frame . . . . .	70
Synchrony scores for each dyad, triad and for the whole group	70
Evolution of synchrony through time, raw each second . . . . .	71
Evolution of synchrony through time, mean by minute . . . . .	72
Evolution of synchrony through time, mean by 10 minutes . . . . .	72
Models of synchrony . . . . .	72

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

```
setwd("/Users/Ofix/Documents/Fac/internat/Recherche/projets/synchro/synchroData/Git/INCANT/Reports/")
```

## Import data

```
data <- importdata(fullNameList)
```

## Lists

### Functions list

#### MeanMotionByTime

Function that takes raw motion history data and compute the mean on a given interval. Intervals don't overlap, so the frequency of the data change (from 25 frames by seconde to 25 frames/interval by second).

#### Arguments:

- subject : Subject studied (patient, mother, father or therapist)
- indexOfvideos : List of videos studied (element eg 3 or list eg 1:3 or c(1,2,4))
- interval : number of frames in the studied interval
- data : data frame where there is data

```
## Revoir nom des variables : pas clair, faire un schéma
MeanMotionByTime <- function(subject, indexOfvideos=1:NumberOfvideos, interval, data){
  x <- c()
  for (file in indexList[indexOfvideos]){
    dataVector <- data[which(data$indexList==file), subject]
```

```

## with ceiling : superior limit of the round
IntervalNumbersVideo <- ceiling(length(dataVector)/interval)
for (i in 1:IntervalNumbersVideo){
  borneinf<- 1+(i-1)*interval
  bornesup <-i*interval
  dataVectorInterval <- dataVector[borneinf:bornesup]
  mean <- mean(dataVectorInterval, na.rm=TRUE)
  x <- c(x, mean)}}
return (x)}

```

## Slidinginterval

Function that takes raw motion history data and compute the mean on a given interval. The interval overlap, so the frequency of the data don't change. It stays at 25 frames/s.

### Arguments:

- subject : subject studied (patient, mother, father or therapist)
- indexOfvideos : list of videos studied (element eg. 3 or list eg 1:3 or c(1,2,4))
- interval : number of frames in the studied interval
- data : data frame where there is data

```

## faire un schéma
SlidingInterval <- function(subject, indexOfvideos=1:NumberOfvideos, interval, data)
{x <- c()
for (file in indexList[indexOfvideos]){
  dataVector <- data[which(data$indexList==file), subject]
  NBofAnalysedFrames <- length(dataVector)-interval+1
  for (i in 1:NBofAnalysedFrames){
    borneinf<- (i)
    bornesup <-(interval-1+i)
    dataVectorInterval <- dataVector[borneinf:bornesup]
    mean <- mean(dataVectorInterval, na.rm=TRUE)
    x <- c(x, mean)}}
return (x)}

```

## MeanSynchronyByTime (TODO)

### Constants generated from data and defining it (data list, )

```

labelvideolist <-c()
for (i in indexList){
  a <- str_count(i)
  name <- substr(i, 6, a)
  labelvideolist <- c(labelvideolist, name)
}

FileName <- data.frame(unique(data$file), filesList, indexList, labelvideolist)
NumberOfvideos <- length(indexList)

```

```

# blue will refer to father
# red will refer to mother
# green to patient
# orange to therapist

colOrderList <- c("blue","red", "green","orange")

```

## Merge data frame, compute Time in minutes, compute log of motion history dataframe

```

data <- merge(data, FileName, by.x="file", by.y="unique.data.file.", all=TRUE)
data$timeMin <- data$frame/(25*60)

data$father[which (data$father == 0)] <- min(data$father[which (data$father >0)])/2
data$logFather <- log(data$father)

data$mother[which (data$mother == 0)] <- min(data$mother[which (data$mother >0)])/2
data$logMother <- log(data$mother)

data$patient[which (data$patient == 0)] <- min(data$patient[which (data$patient >0)])/2
data$logPatient <- log(data$patient)

data$therapist[which (data$therapist == 0)] <- min(data$therapist[which (data$therapist >0)])/2
data$logTherapist <- log(data$therapist)

data$file <- NULL
data$filesList <- NULL

```

- Time in minutes The timeMin is calculated with a frame rate of 25/sec.
- Motion history distribution The data is not normal at all but with very small movement very frequent and bigger movement much rare with a long tail.

To normalize the distribution to compute synchrony scores on it, we made the Napierian logarithm. It produces negative numbers. SyncPy can't compute negatives scores, they are so shifted to positives values with an arbitrary value of 20 to avoid to keep extreme negative values.

Values equal to 0 can't be logged. They generate a -Inf value. These values are set to NA. We lose the information of no movement at all. If we give a arbitrary value to this data (eg, the minimum value, they are over represented)

## Presentation of the data

```
str(data)
```

```

## 'data.frame':   421797 obs. of  12 variables:
## $ frame      : int   1 2 3 4 5 6 7 8 9 10 ...
## $ father     : num  0.0339 0.0156 0.023 0.0304 0.0299 ...

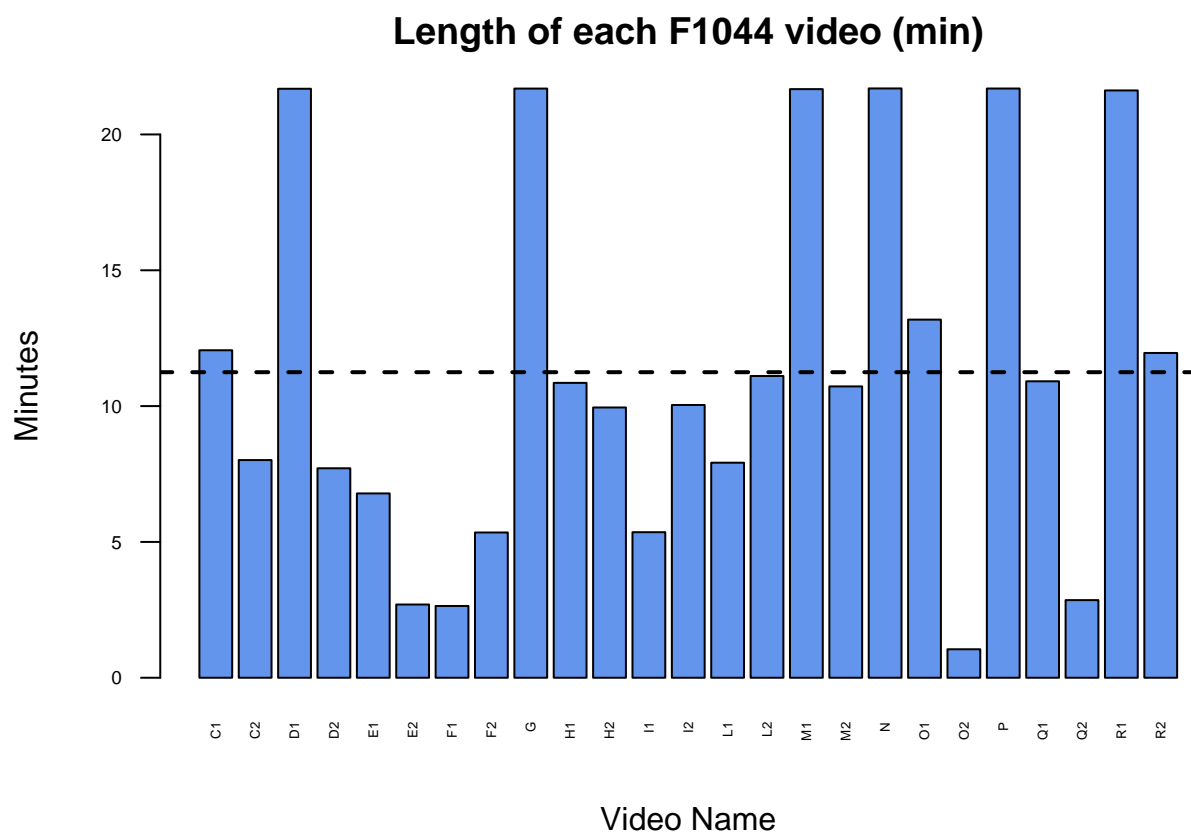
```

```
## $ mother      : num  2.28e-05 2.28e-05 4.56e-05 2.28e-05 1.14e-04 ...
## $ patient     : num  NA NA NA NA NA NA NA NA NA NA NA ...
## $ therapist   : num  0.00172 0.00529 0.00334 0.00223 0.00265 ...
## $ indexList   : Factor w/ 25 levels "F1044C1","F1044C2",...: 1 1 1 1 1 1 1 1 1 1 ...
## $ labelvideolist: Factor w/ 25 levels "C1","C2","D1",...: 1 1 1 1 1 1 1 1 1 1 ...
## $ timeMin     : num  0.000667 0.001333 0.002 0.002667 0.003333 ...
## $ logFather   : num  -3.38 -4.16 -3.77 -3.49 -3.51 ...
## $ logMother   : num  -10.69 -10.69 -10 -10.69 -9.08 ...
## $ logPatient  : num  NA NA NA NA NA NA NA NA NA NA ...
## $ logTherapist: num  -6.37 -5.24 -5.7 -6.11 -5.93 ...
```

```
summary(data)
```

```
##      frame      father      mother      patient
## Min.   :    1  Min.   :0.00  Min.   :0.00  Min.   :0.00
## 1st Qu.: 4362  1st Qu.:0.00  1st Qu.:0.00  1st Qu.:0.00
## Median : 9526  Median :0.00  Median :0.00  Median :0.00
## Mean   :11386  Mean   :0.00  Mean   :0.00  Mean   :0.01
## 3rd Qu.:16227  3rd Qu.:0.00  3rd Qu.:0.00  3rd Qu.:0.01
## Max.   :32540  Max.   :0.16  Max.   :0.21  Max.   :0.26
##      NA's      :217772  NA's   :46817  NA's   :211815
##      therapist  indexList  labelvideolist  timeMin
## Min.   :0.00  F1044N : 32540  N      : 32540  Min.   : 0.000667
## 1st Qu.:0.00  F1044P : 32535  P      : 32535  1st Qu.: 2.908000
## Median :0.00  F1044G : 32532  G      : 32532  Median : 6.350667
## Mean   :0.00  F1044D1: 32523  D1     : 32523  Mean   : 7.590724
## 3rd Qu.:0.00  F1044M1: 32502  M1     : 32502  3rd Qu.:10.818000
## Max.   :0.16  F1044R1: 32430  R1     : 32430  Max.   :21.693333
## NA's      :58310  (Other):226735  (Other):226735
##      logFather  logMother  logPatient  logTherapist
## Min.   : -11.74  Min.   : -11.69  Min.   : -11.80  Min.   : -10.68
## 1st Qu.: -11.74  1st Qu.: -11.69  1st Qu.:  -7.67  1st Qu.:  -6.72
## Median : -10.75  Median :  -9.40  Median :  -5.81  Median :  -5.88
## Mean   :  -9.24  Mean   :  -8.78  Mean   :  -6.38  Mean   :  -6.51
## 3rd Qu.:  -6.72  3rd Qu.:  -6.66  3rd Qu.:  -4.55  3rd Qu.:  -5.49
## Max.   :  -1.85  Max.   :  -1.55  Max.   :  -1.35  Max.   :  -1.84
## NA's      :217772  NA's   :46817  NA's   :211815  NA's   :58310
```

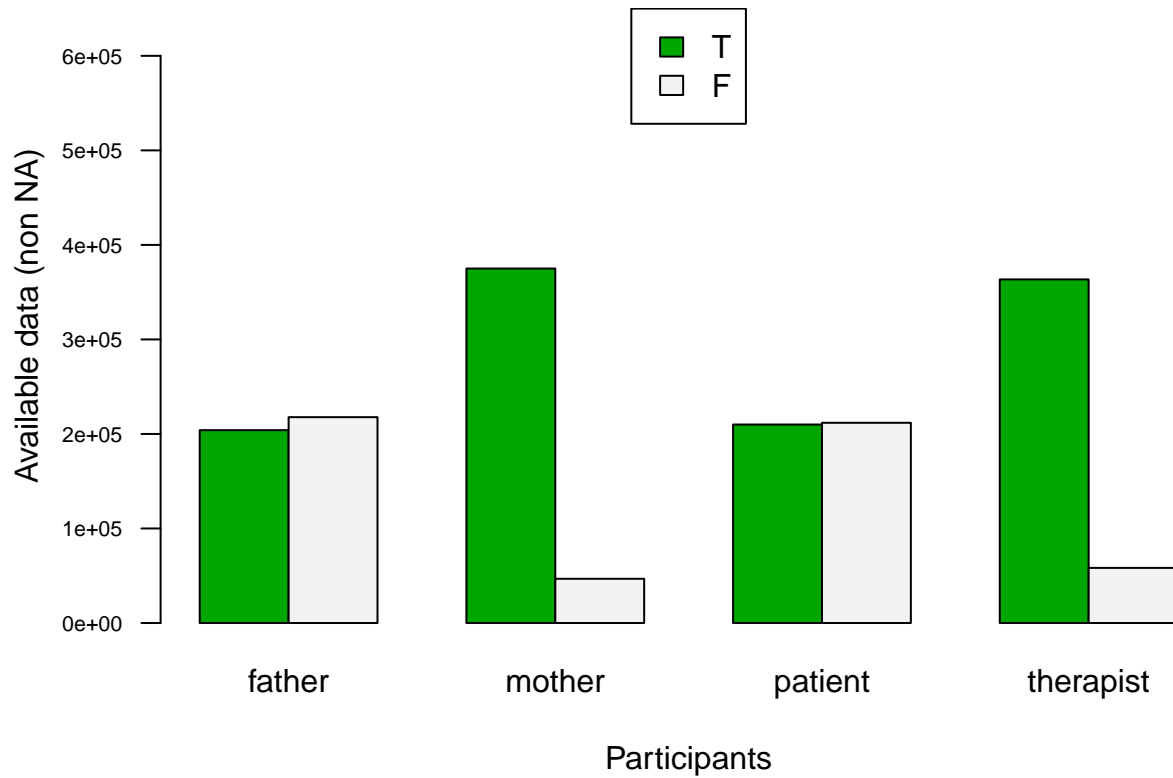
## Length of the videos in minutes



Length of the videos in number of frames

Number of Available (True) and Not Available (False) data for each participant

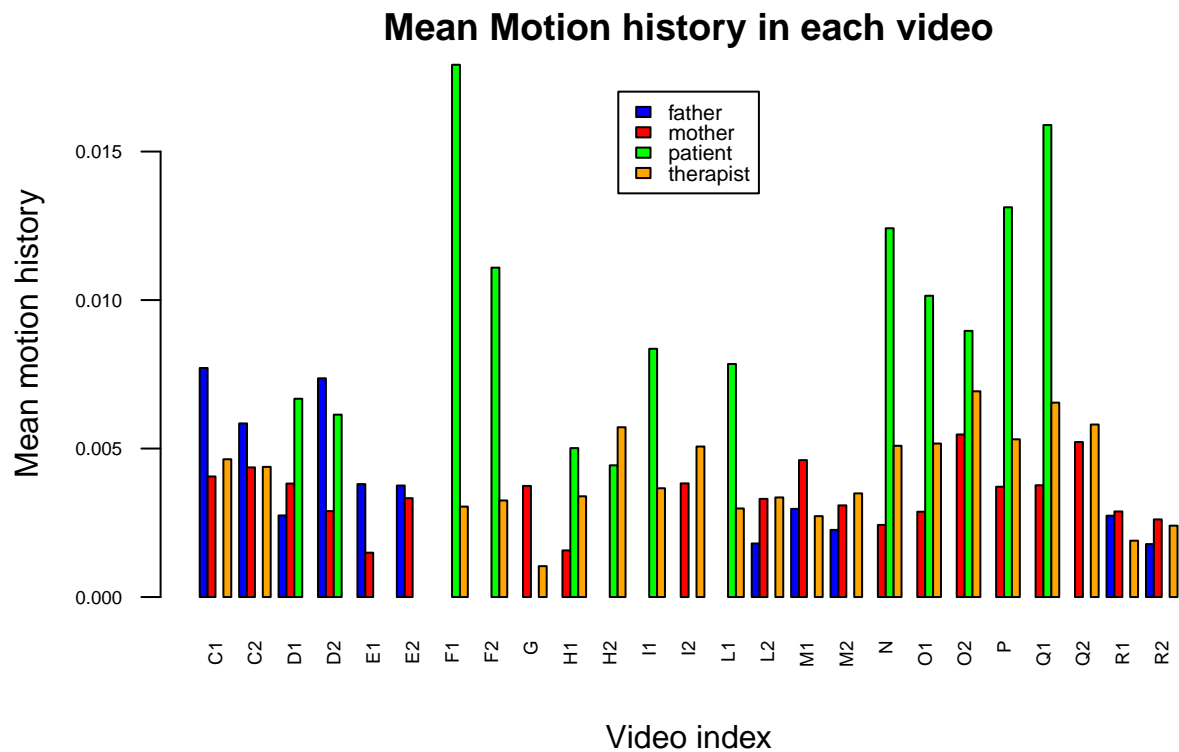
### Number of available data by participant



Some participants are not filmed (eg therapist) or don't come in the sessions (father, patient). Mother and therapist are the more often present participants.

## Global Motion history

### Mean Motion history by video by participant



We can see that configurations of subjects are very different. Consequently, it makes the comparisons of the videos quite complicated. It is not really relevant to compare the synchrony of two persons if the context is different (other people around them).

### Raw Motion history by video by participant

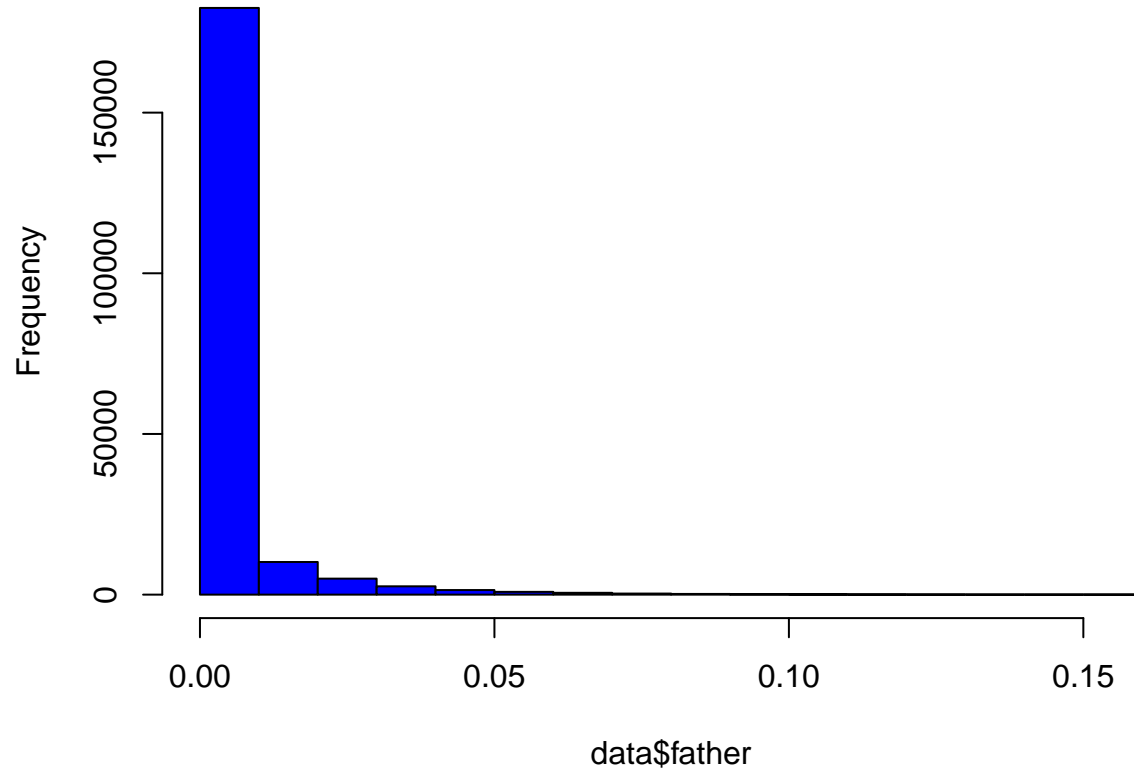
#### Boxplots

#### Histograms

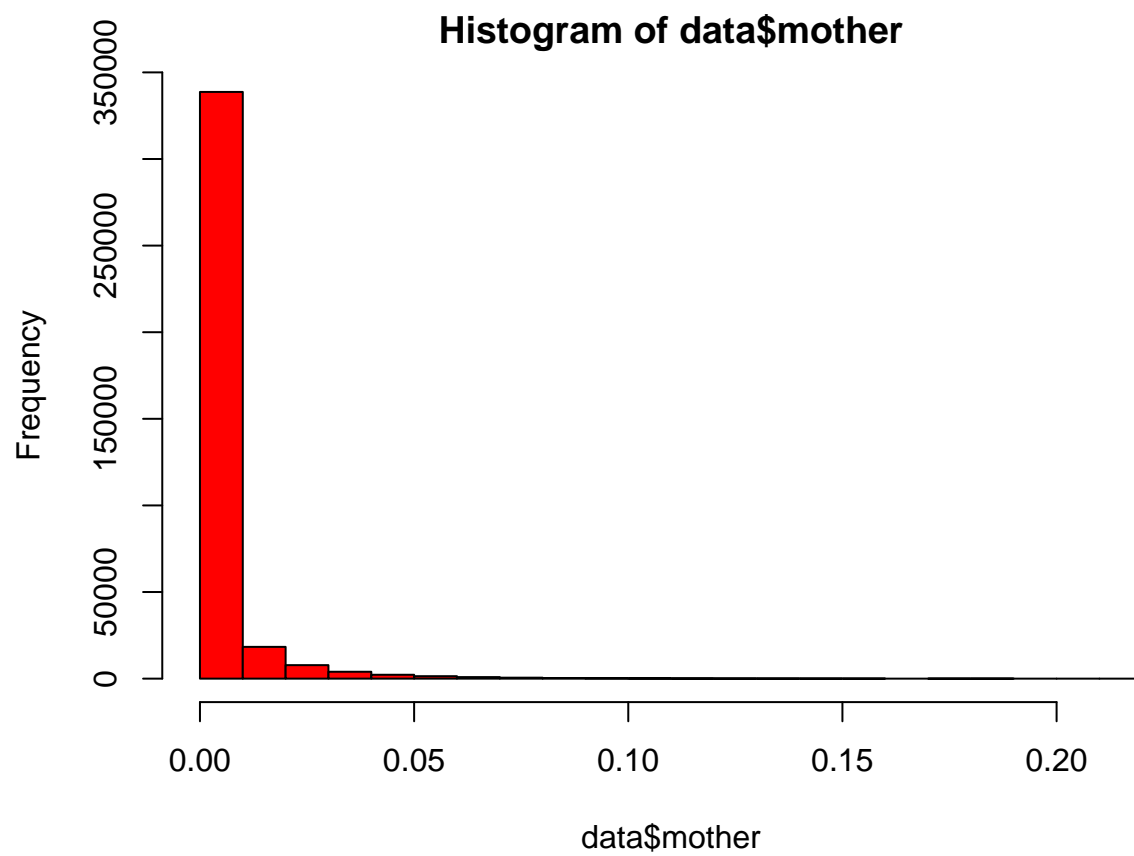
```
par(mar=c(4,4,2,2))  
hist(data$father, col=colOrderList[1])
```



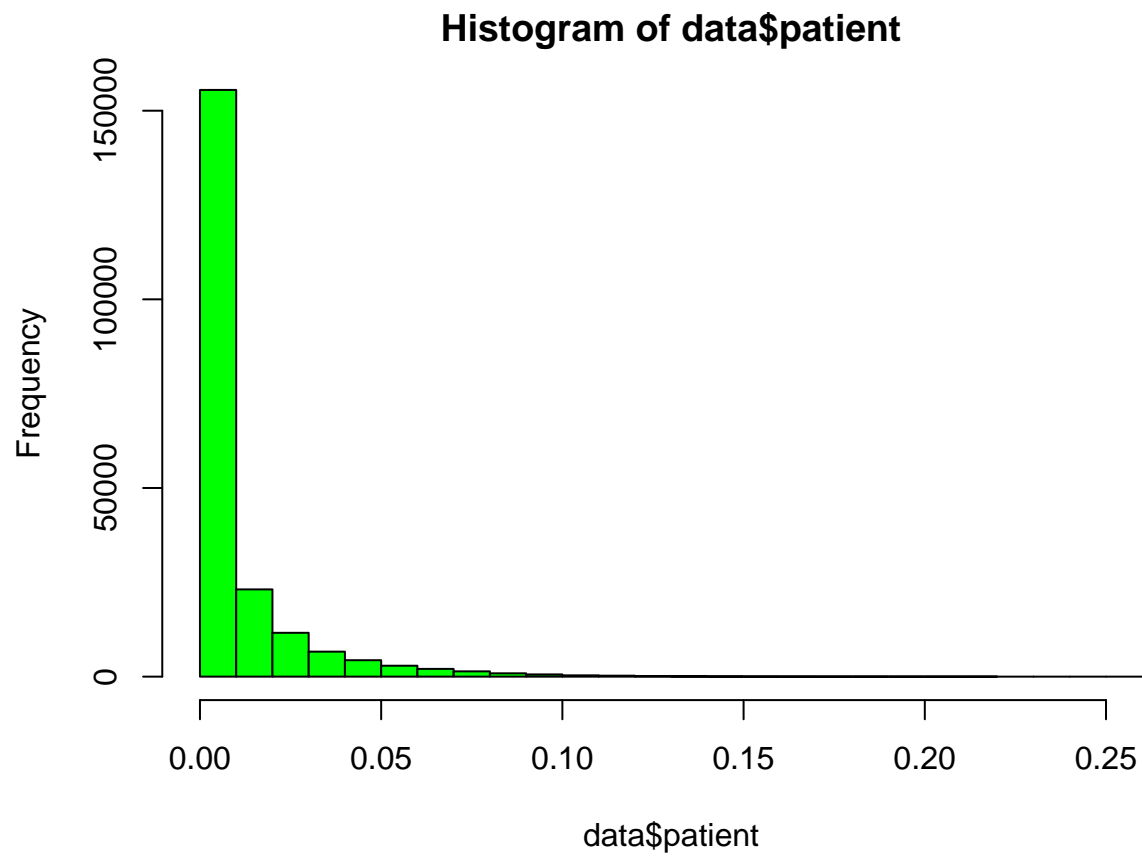
**Histogram of data\$father**



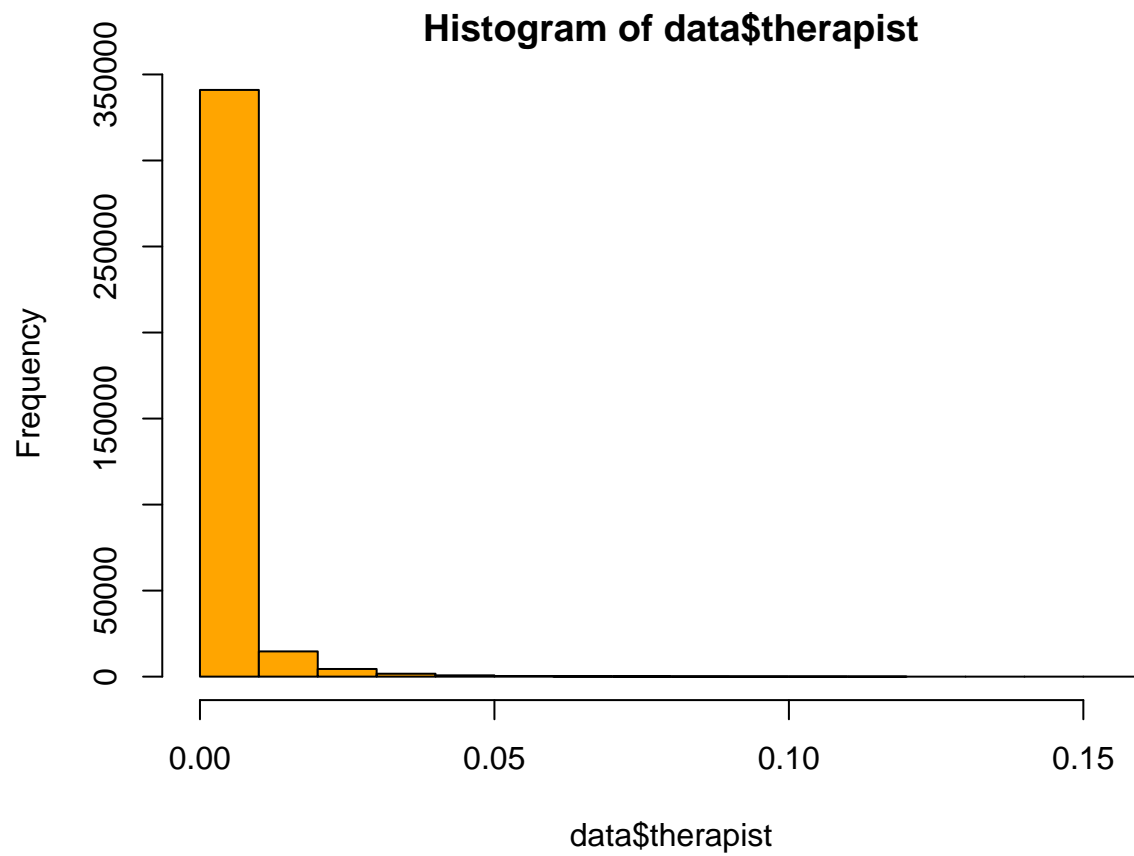
```
hist(data$mother, col=colOrderList[2])
```



```
hist(data$patient, col=colOrderList[3])
```



```
hist(data$therapist, col=colOrderList[4])
```



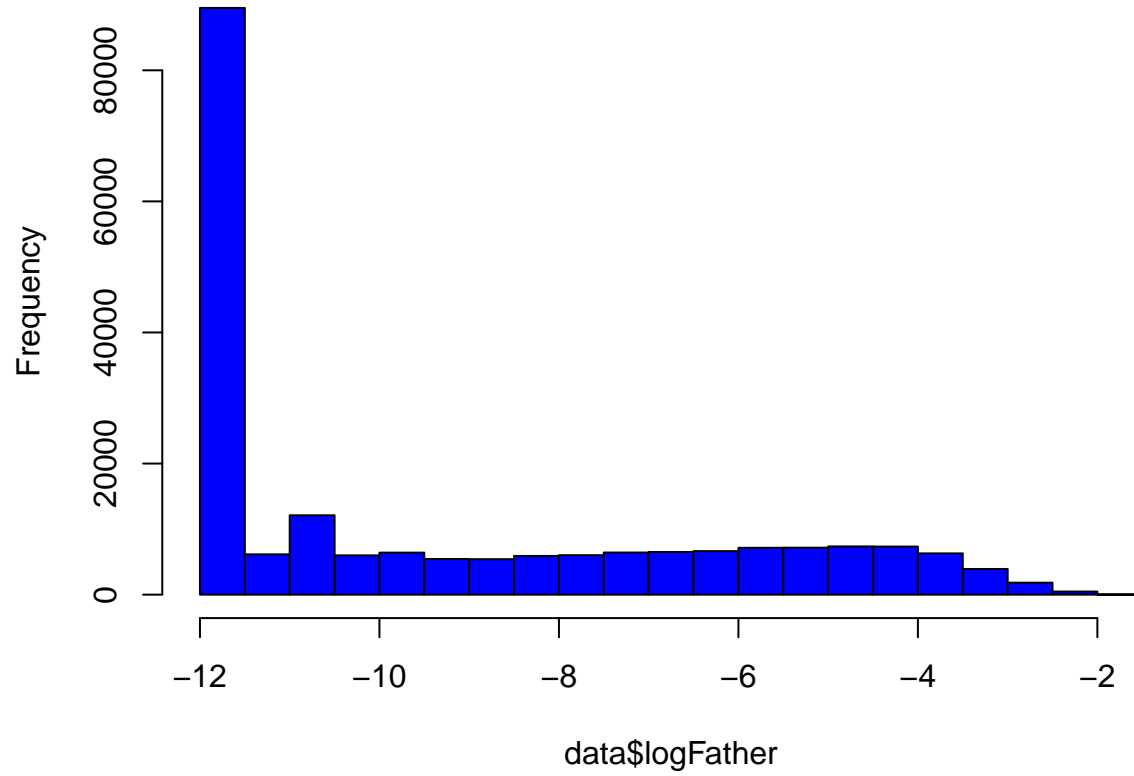
Normalized log Motion history by video by participant

Boxplots

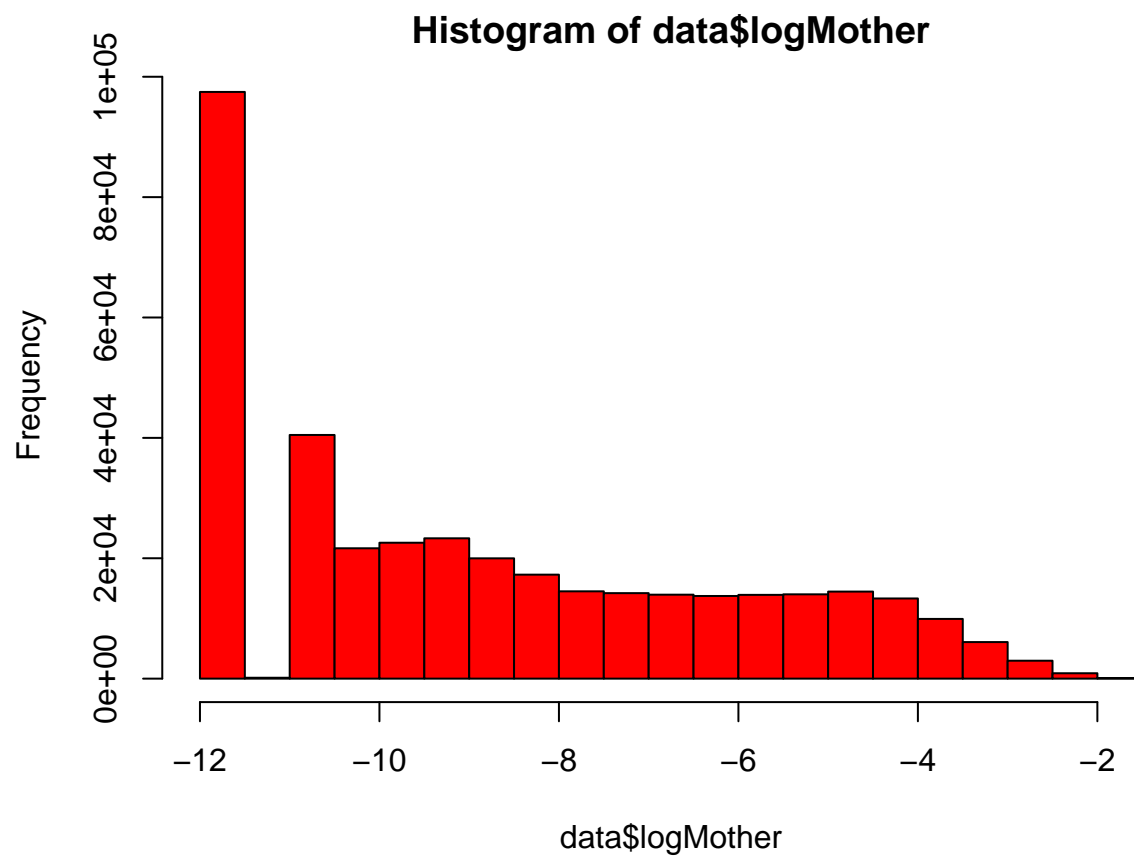
Histograms

```
par(mar=c(4,4,2,2))  
hist(data$logFather, col=colOrderList[1])
```

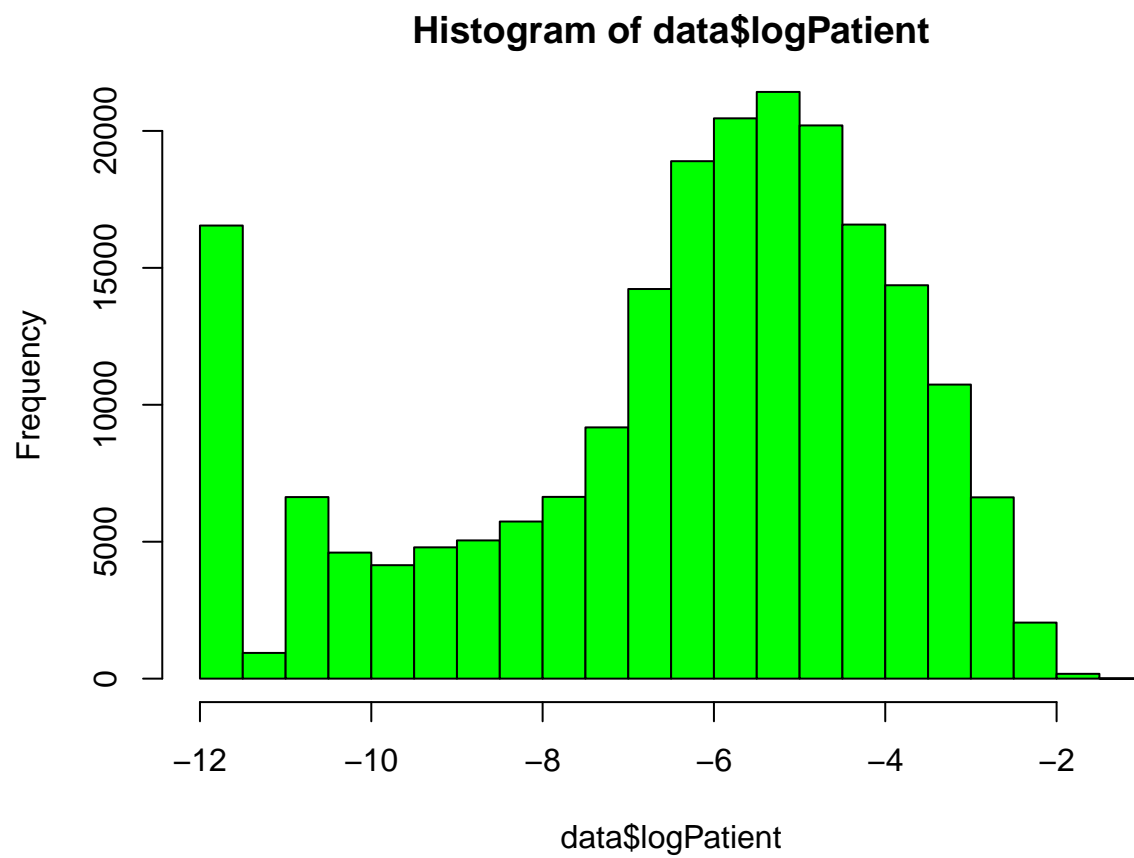
**Histogram of data\$logFather**



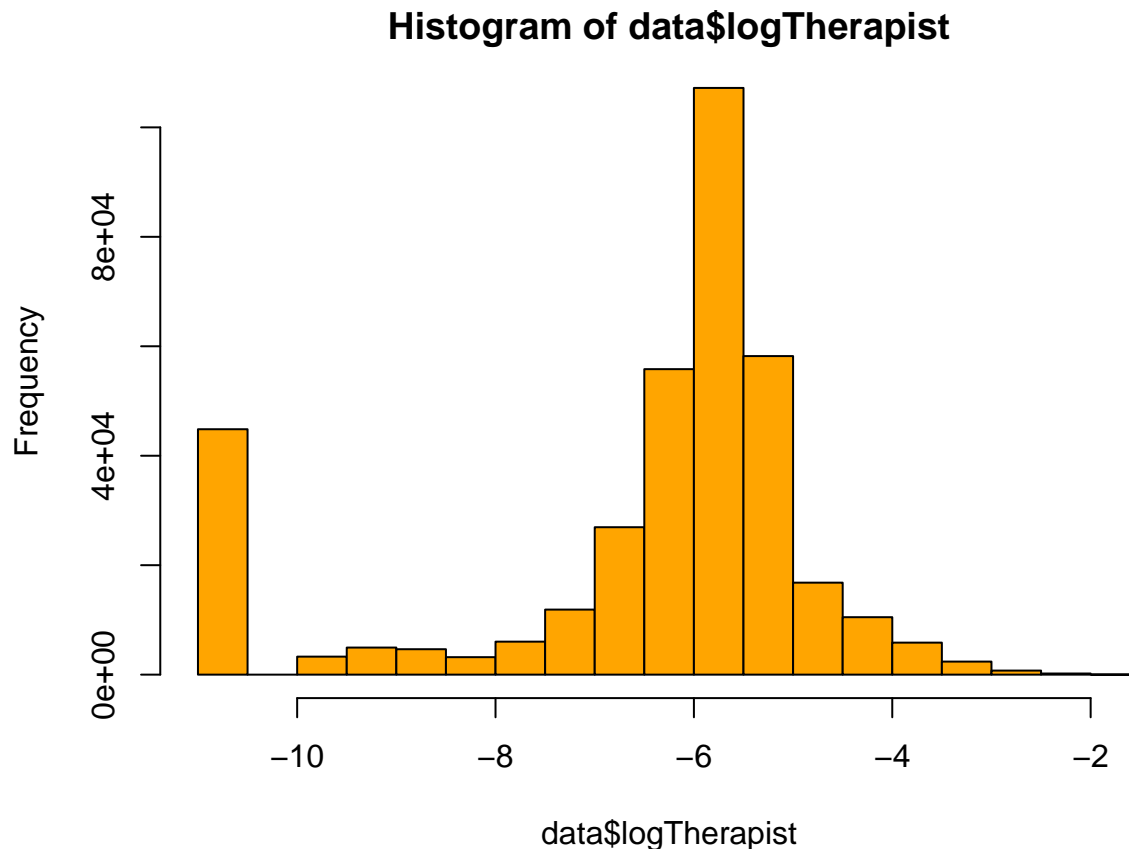
```
hist(data$logMother, col=colOrderList[2])
```



```
hist(data$logPatient, col=colOrderList[3])
```



```
hist(data$logTherapist, col=colOrderList[4])
```



We can see that the father and the mother motion history is very similar. However. The therapist, which is always in a small window of the video, as a very different distribution. We have less signal on it. In some videos the patient is in this window, it explains, it intermediates position.

## Raw data and mean of Motion History on sliding and non overlapping intervals on 1st video F1044C1 video

It is the first video of F1044C. The father, mother and therapist are present. The patient is absent.

### Raw data

```
rawFatherF1044C1 <- data[which(data$indexList=="F1044C1"),]$father
rawMotherF1044C1 <- data[which(data$indexList=="F1044C1"),]$mother
rawTherapistF1044C1 <- data[which(data$indexList=="F1044C1"),]$therapist

logFatherF1044C1 <- data[which(data$indexList=="F1044C1"),]$logFather
logMotherF1044C1 <- data[which(data$indexList=="F1044C1"),]$logMother
logTherapistF1044C1 <- data[which(data$indexList=="F1044C1"),]$logTherapist

summary(rawFatherF1044C1)
```

```
##      Min.   1st Qu.   Median     Mean  3rd Qu.    Max.
## 7.950e-06 2.079e-05 6.862e-04 7.711e-03 7.278e-03 1.568e-01
```



```
summary(rawMotherF1044C1)
```

```
##      Min.   1st Qu.   Median     Mean   3rd Qu.     Max.
## 8.350e-06 4.563e-05 1.825e-04 4.060e-03 2.806e-03 1.116e-01
```

```
summary(rawTherapistF1044C1)
```

```
##      Min.   1st Qu.   Median     Mean   3rd Qu.     Max.
## 0.001393 0.002832 0.003482 0.004640 0.004411 0.083240
```

```
summary(logFatherF1044C1)
```

```
##      Min.   1st Qu.   Median     Mean   3rd Qu.     Max.
## -11.740 -10.780  -7.284  -7.656  -4.923  -1.853
```

```
summary(logMotherF1044C1)
```

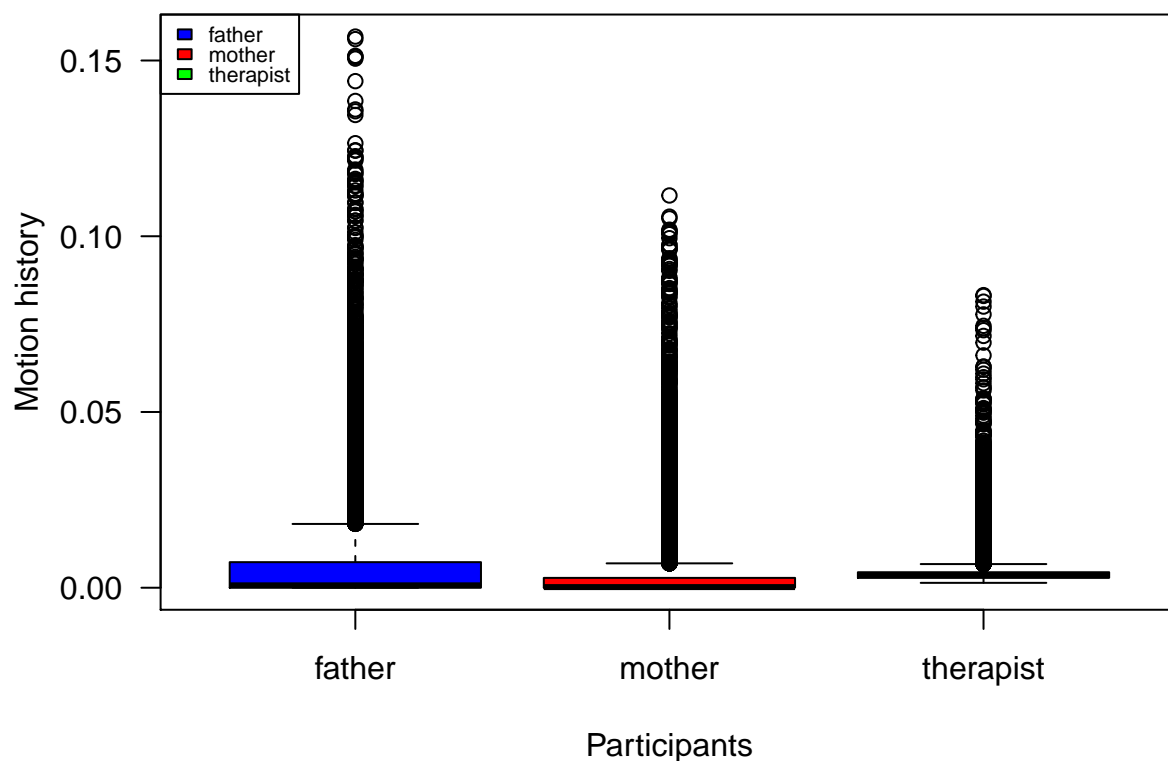
```
##      Min.   1st Qu.   Median     Mean   3rd Qu.     Max.
## -11.690  -9.995  -8.609  -8.048  -5.876  -2.193
```

```
summary(logTherapistF1044C1)
```

```
##      Min.   1st Qu.   Median     Mean   3rd Qu.     Max.
##  -6.576  -5.867  -5.660  -5.562  -5.424  -2.486
```

```
par(mar=c(4,4,3,2))
boxplot(rawFatherF1044C1, rawMotherF1044C1, rawTherapistF1044C1,
        col=colOrderList[c(1,2,4)],
        names=ParticipantsList[c(1,2,4)],
        main= "Box plots of motion history raw data on F1044C1 video", las=1, ylab = "Motion history", xlab = "Participant")
par(mar=c(1,0.5,0.5,1))
legend("topleft", ParticipantsList[c(1,2,4)], fill=colOrderList, cex=0.7)
```

## Box plots of motion history raw data on F1044C1 video



## Sliding interval

```
## REMINDER:
# SlidingInterval <- function(subject, indexOfvideos=1:NumberOfvideos, interval, data) with :
# subject : subject studied (patient, mother, father or therapist)
# indexOfvideos : list of videos studied (element eg. 3 or list eg 1:3 or c(1,2,4))
# interval : number of frames in the studied interval
# data : data frame where there is data

slidedFatherF1044C1 <- SlidingInterval("father", 1 , 5, data)
slidedMotherF1044C1 <- SlidingInterval("mother", 1 , 5, data)
slidedPatientF1044C1 <- SlidingInterval("patient", 1 , 5, data)
slidedTherapistF1044C1 <- SlidingInterval("therapist", 1 , 5, data)

slidedLogFatherF1044C1 <- SlidingInterval("logFather", 1 , 5, data)
slidedLogMotherF1044C1 <- SlidingInterval("logMother", 1 , 5, data)
slidedLogPatientF1044C1 <- SlidingInterval("logPatient", 1 , 5, data)
slidedLogTherapistF1044C1 <- SlidingInterval("logTherapist", 1 , 5, data)

summary(slidedFatherF1044C1)
```

```
##      Min.   1st Qu.   Median     Mean   3rd Qu.     Max.
## 7.950e-06 2.397e-05 1.011e-03 7.710e-03 7.678e-03 1.506e-01
```

```
summary(slidedMotherF1044C1)
```

```
##      Min.   1st Qu.   Median     Mean  3rd Qu.     Max.
## 8.350e-06 6.098e-05 2.464e-04 4.060e-03 3.185e-03 1.013e-01
```

```
summary(slidedPatientF1044C1)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.   Max.   NA's
##      NA      NA      NA     NaN    NA      NA   18080
```

```
summary(slidedTherapistF1044C1)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.   Max.
## 0.002126 0.003101 0.003436 0.004640 0.004067 0.077910
```

```
summary(slidedLogFatherF1044C1)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.   Max.
## -11.740 -10.950  -7.352  -7.656  -5.039  -1.894
```

```
summary(slidedLogMotherF1044C1)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.   Max.
## -11.690 -10.050  -8.713  -8.048  -6.001  -2.291
```

```
summary(slidedLogPatientF1044C1)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.   Max.   NA's
##      NA      NA      NA     NaN    NA      NA   18080
```

```
summary(slidedLogTherapistF1044C1)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.   Max.
##  -6.162  -5.800  -5.697  -5.562  -5.535  -2.554
```

## Non overlapping interval

```
fatherFiveF1044C1<- MeanMotionByTime("father", indexOfvideos=1, interval=5, data)
motherFiveF1044C1 <- MeanMotionByTime("mother", indexOfvideos=1, interval=5, data)
therapistFiveF1044C1 <- MeanMotionByTime("therapist", indexOfvideos=1, interval=5, data)
fatherLogFiveF1044C1<- MeanMotionByTime("logFather", indexOfvideos=1, interval=5, data)
motherLogFiveF1044C1 <- MeanMotionByTime("logMother", indexOfvideos=1, interval=5, data)
therapistLogFiveF1044C1 <- MeanMotionByTime("logTherapist", indexOfvideos=1, interval=5, data)
summary(fatherFiveF1044C1)
```

```
##      Min.   1st Qu.   Median     Mean   3rd Qu.     Max.
## 7.950e-06 2.397e-05 1.033e-03 7.711e-03 7.794e-03 1.500e-01
```

```
summary(motherFiveF1044C1)
```

```
##      Min.   1st Qu.   Median     Mean   3rd Qu.     Max.
## 8.350e-06 5.931e-05 2.464e-04 4.060e-03 3.194e-03 1.010e-01
```

```
summary(therapistFiveF1044C1)
```

```
##      Min.   1st Qu.   Median     Mean   3rd Qu.     Max.
## 0.002126 0.003101 0.003426 0.004640 0.004058 0.075780
```

```
summary(fatherLogFiveF1044C1)
```

```
##      Min.   1st Qu.   Median     Mean   3rd Qu.     Max.
## -11.740 -10.970  -7.360  -7.656  -5.025  -1.899
```

```
summary(motherLogFiveF1044C1)
```

```
##      Min.   1st Qu.   Median     Mean   3rd Qu.     Max.
## -11.690 -10.050  -8.738  -8.048  -5.989  -2.294
```

```
summary(therapistLogFiveF1044C1)
```

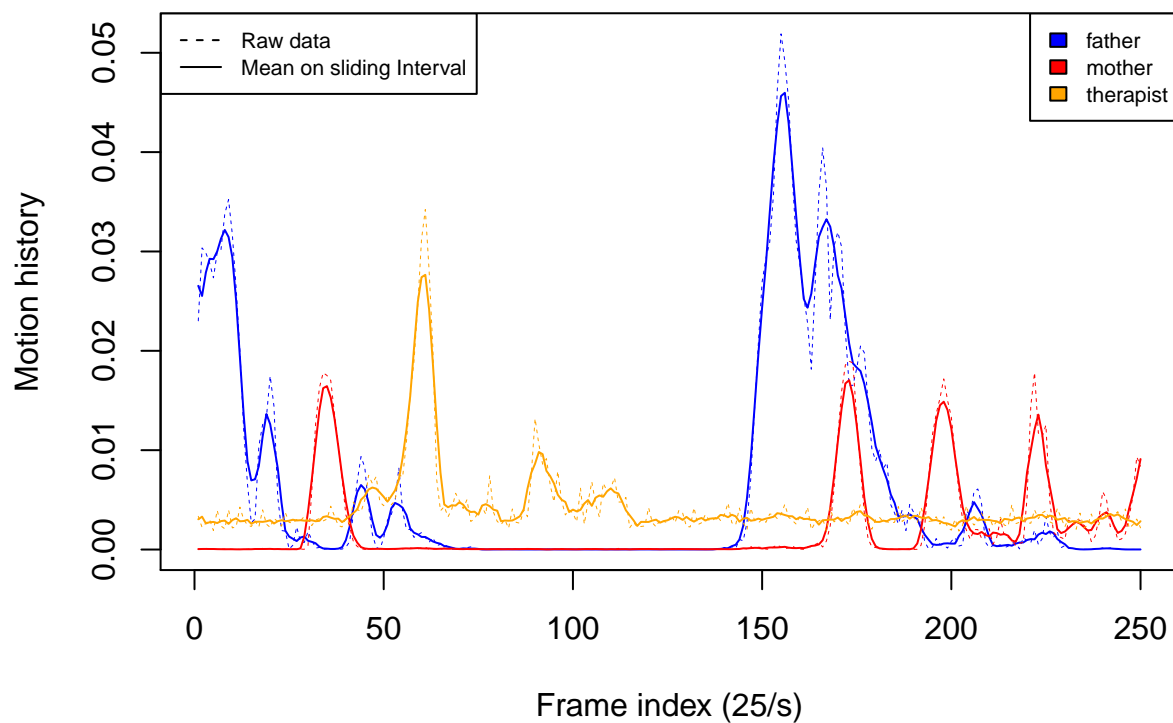
```
##      Min.   1st Qu.   Median     Mean   3rd Qu.     Max.
##  -6.162  -5.799  -5.699  -5.562  -5.534  -2.582
```

Focus on the motion history of the first 10 seconds of the first video(C)

Sliding interval function on a raw data, 5 frames interval

```
par(mar=c(4,4,4,2))
plot(1:250, data$father[3:252], main="Mean motion history, (Sliding 5 frames interval raw data)
      on F1044C1 video, first 10 seconds ", xlab="Frame index (25/s)",
      ylab="Motion history",
      col="blue", type="l", lty=2, lwd=0.5)
lines(slidedFatherF1044C1[1:250], col="blue", lty=1)
lines(data$mother[3:252], col="red", lty=2, lwd=0.5)
lines(slidedMotherF1044C1[1:250], col="red", lty=1)
lines(data$therapist[3:252], col="orange", lty=2, lwd=0.5)
lines(slidedTherapistF1044C1[1:250], col="orange", lty=1)
legend("topleft", c("Raw data", "Mean on sliding Interval"), lty=c(2, 1), cex=0.7)
legend("topright", ParticipantsList[c(1,2,4)], fill=colOrderList[c(1,2,4)], cex=0.7)
```

# **Mean motion history, (Sliding 5 frames interval raw data) on F1044C1 video, first 10 seconds**

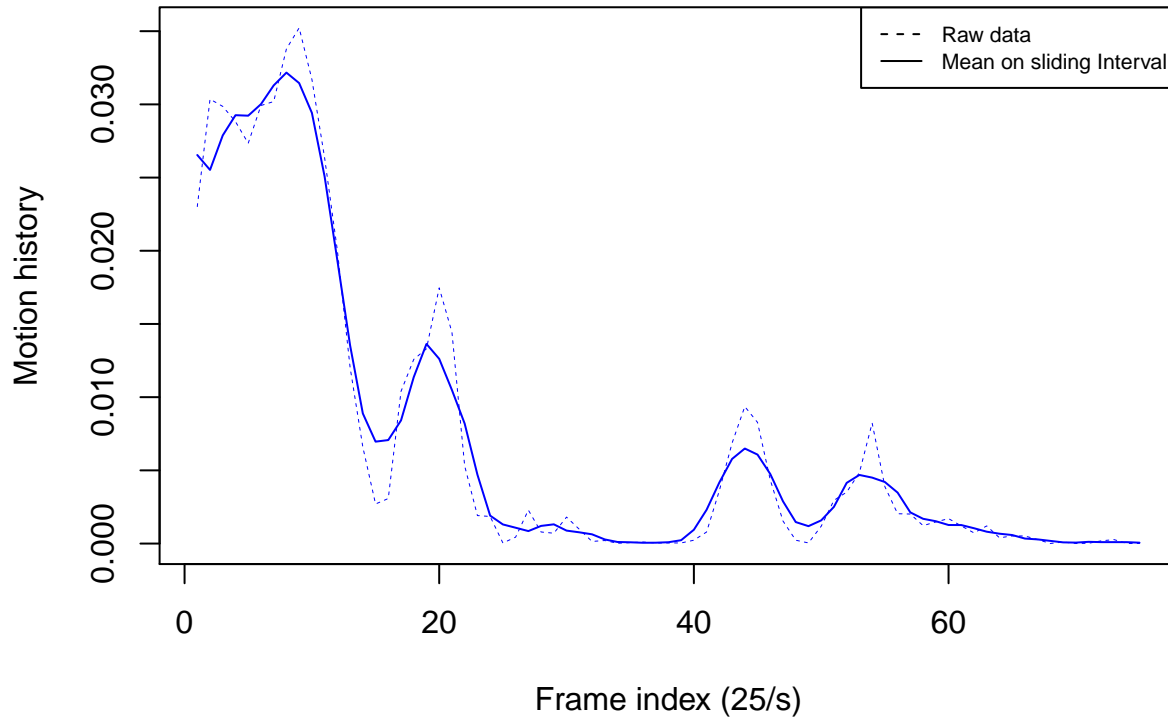


```

par(mar=c(4,4,4,2))
plot(1:75, data$father[3:77], main="Mean motion history, (Sliding 5 frames interval raw data)
    on F1044C1 video, first 10 seconds ", xlab="Frame index (25/s)",
    ylab="Motion history",
    col="blue", type="l", lty=2, lwd=0.5)
lines(slidedFatherF1044C1[1:75], col="blue", lty=1)
legend("topright", c("Raw data", "Mean on sliding Interval") , lty=c(2, 1), cex=0.7)

```

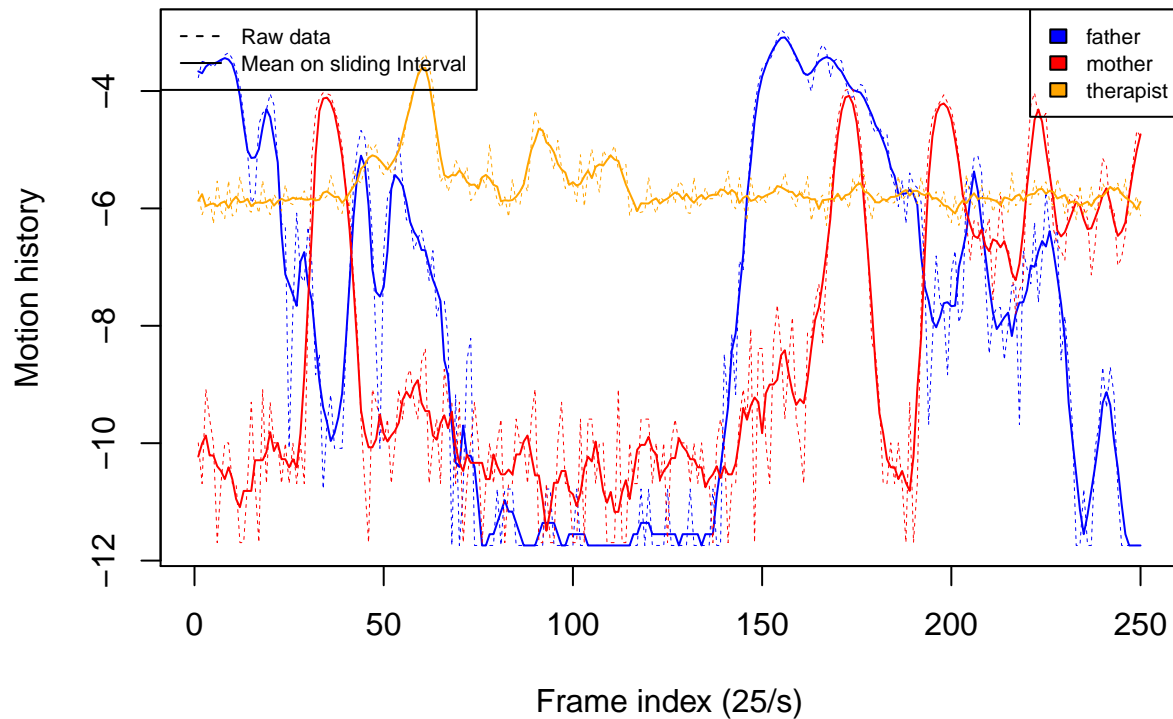
### Mean motion history, (Sliding 5 frames interval raw data) on F1044C1 video, first 10 seconds



Sliding interval function on log data, 5 frames interval

```
par(mar=c(4,4,4,2))
plot(1:250, data$logFather[3:252], main="Mean motion history, (Sliding 5 frames interval log data)
on F1044C1 video, first 10 seconds", xlab="Frame index (25/s)",
     ylab="Motion history",
     col="blue", type="l", lty=2, lwd=0.5)
lines(slidedLogFatherF1044C1[1:250], col="blue", lty=1)
lines(data$logMother[3:252], col="red", lty=2, lwd=0.5)
lines(slidedLogMotherF1044C1[1:250], col="red", lty=1)
lines(data$logTherapist[3:252], col="orange", lty=2, lwd=0.5)
lines(slidedLogTherapistF1044C1[1:250], col="orange", lty=1)
legend("topleft", c("Raw data", "Mean on sliding Interval"), lty=c(2, 1), cex=0.7)
legend("topright", ParticipantsList[c(1,2,4)], fill=colOrderList[c(1,2,4)], cex=0.7)
```

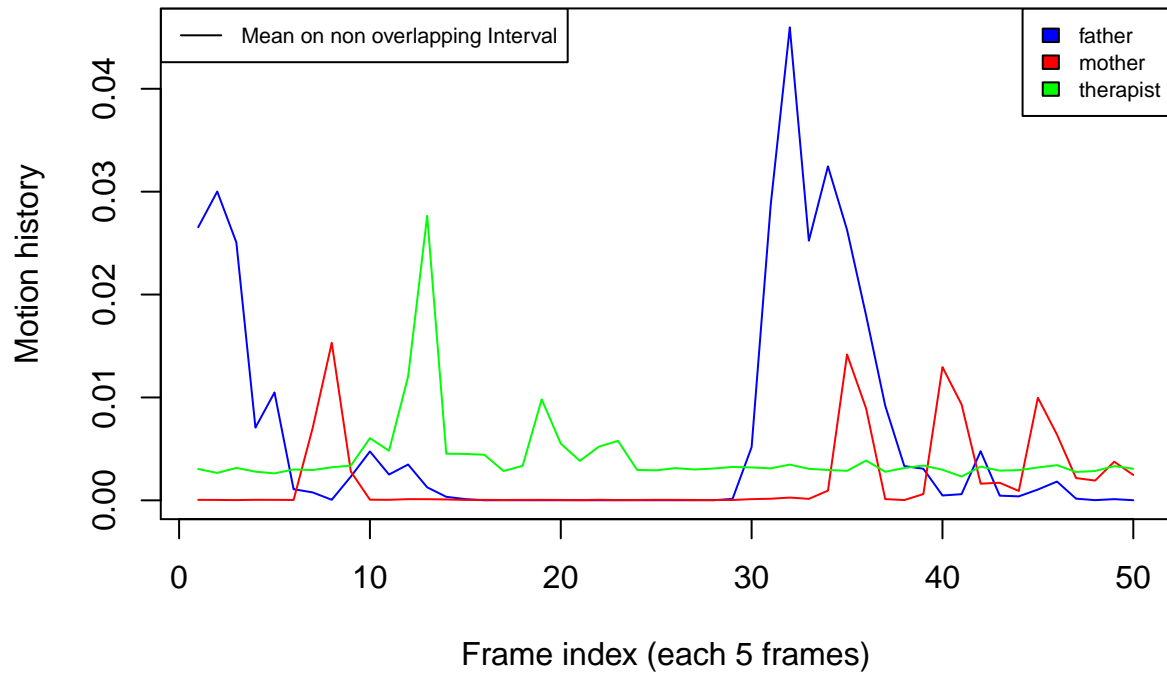
## Mean motion history, (Sliding 5 frames interval log data) on F1044C1 video, first 10 seconds



Non overlapping interval function on a 5 frames interval

```
plot (1:50, fatherFiveF1044C1[1:50], type="l", col="blue",
main="Mean Motion history (non overlapping 5 frames
      intervals) for father on F1044C video, first 10 seconds",
ylab="Motion history", xlab="Frame index (each 5 frames)" )
lines(motherFiveF1044C1[1:50], col="red", lty=1)
lines(therapistFiveF1044C1[1:50], col="green", lty=1)
legend("topleft", "Mean on non overlapping Interval" , lty=1, cex=0.7)
legend("topright", ParticipantsList[c(1,2,4)], fill=colOrderList, cex=0.7)
```

## Mean Motion history (non overlapping 5 frames intervals) for father on F1044C video, first 10 seconds



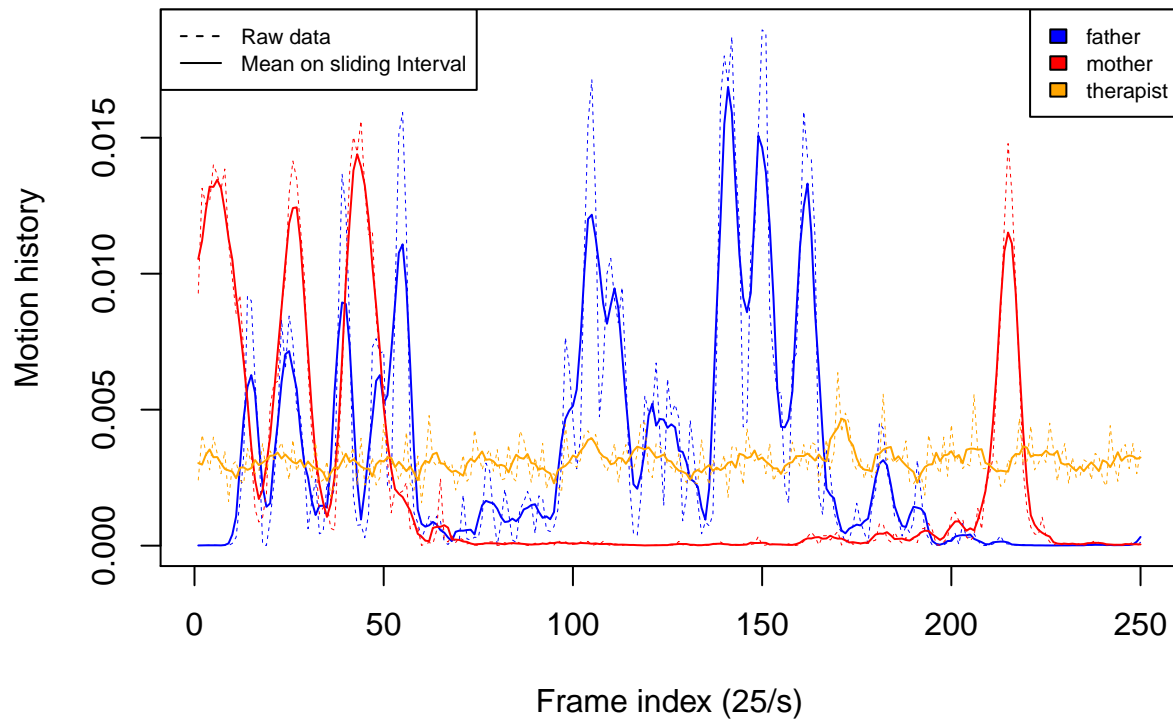
Motion history of the father during 10-20 seconds of the first video(C)

Non overlapping interval function on a 5 frames interval

```
par(mar=c(4,4,4,2))
plot(1:250, data$father[253:502], main="Mean motion history (Sliding 5 frames
interval) for father on F1044C video, 10-20 seconds", xlab="Frame index (25/s)",
ylab="Motion history", col="blue", type="l", lty=2, lwd=0.5)
lines(slidedFatherF1044C1[251:500], col="blue", lty=1)
lines(data$mother[253:502], col="red", lty=2, lwd=0.5)
lines(slidedMotherF1044C1[251:500], col="red", lty=1)
lines(data$therapist[253:502], col="orange", lty=2, lwd=0.5)
lines(slidedTherapistF1044C1[251:500], col="orange", lty=1)
legend("topleft", c("Raw data", "Mean on sliding Interval"), lty=c(2, 1), cex=0.7)
legend("topright", ParticipantsList[c(1,2,4)], fill=colOrderList[c(1,2,4)], cex=0.7)
```



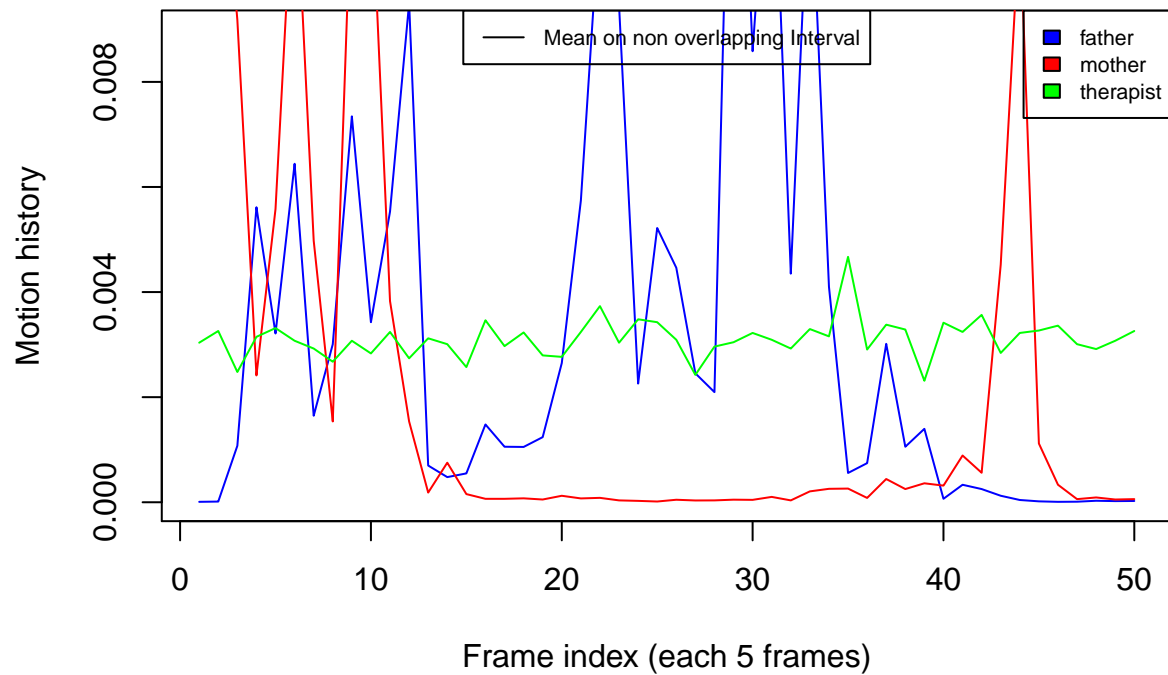
### Mean motion history (Sliding 5 frames interval) for father on F1044C video, 10–20 seconds



Non overlapping interval function on a 5 frames interval

```
plot (1:50, fatherFiveF1044C1[51:100], type="l", col="blue",
main="Mean motion history (non overlapping 5 frames intervals) for
father on F1044C video, between 10-20 seconds",
ylab="Motion history", xlab="Frame index (each 5 frames)", ylim=c(0, 0.009))
lines(motherFiveF1044C1[51:100], col="red", lty=1)
lines(therapistFiveF1044C1[51:100], col="green", lty=1)
legend("top", "Mean on non overlapping Interval" , lty=1, cex=0.7)
legend("topright", ParticipantsList[c(1,2,4)], fill=colOrderList, cex=0.7)
```

## Mean motion history (non overlapping 5 frames intervals) for father on F1044C video, between 10–20 seconds



## Mean motion history by minute plots

```
for (i in 1:NumberOfvideos){
  fatherMinute<- MeanMotionByTime("father", indexOfvideos=i, interval=1500, data)

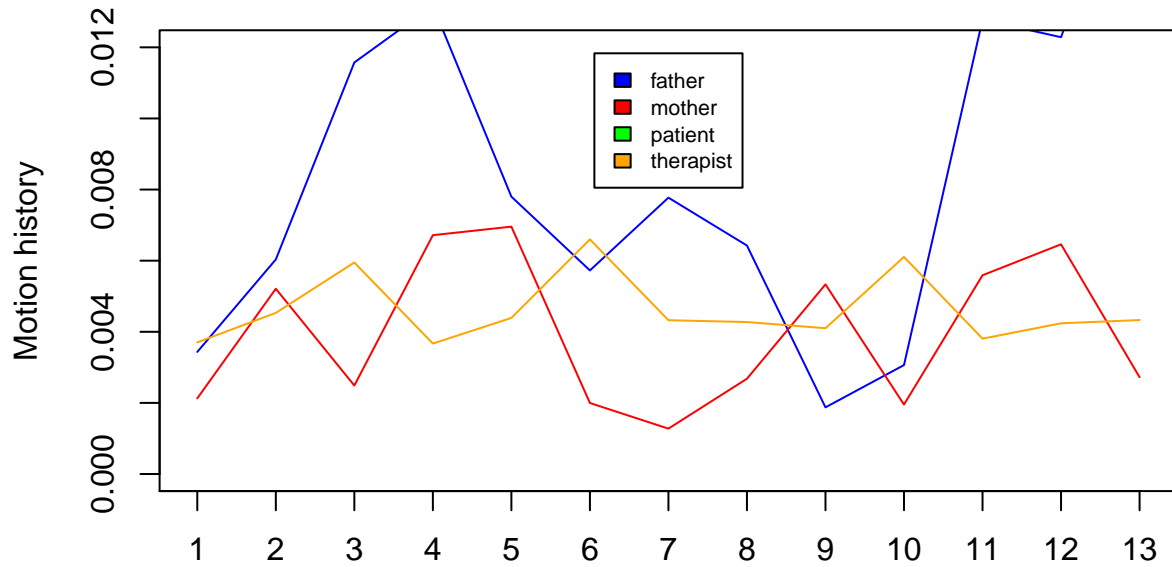
  MotherMinute<- MeanMotionByTime("mother", indexOfvideos=i, interval=1500, data)

  TherapistMinute<- MeanMotionByTime("therapist", indexOfvideos=i, interval=1500, data)

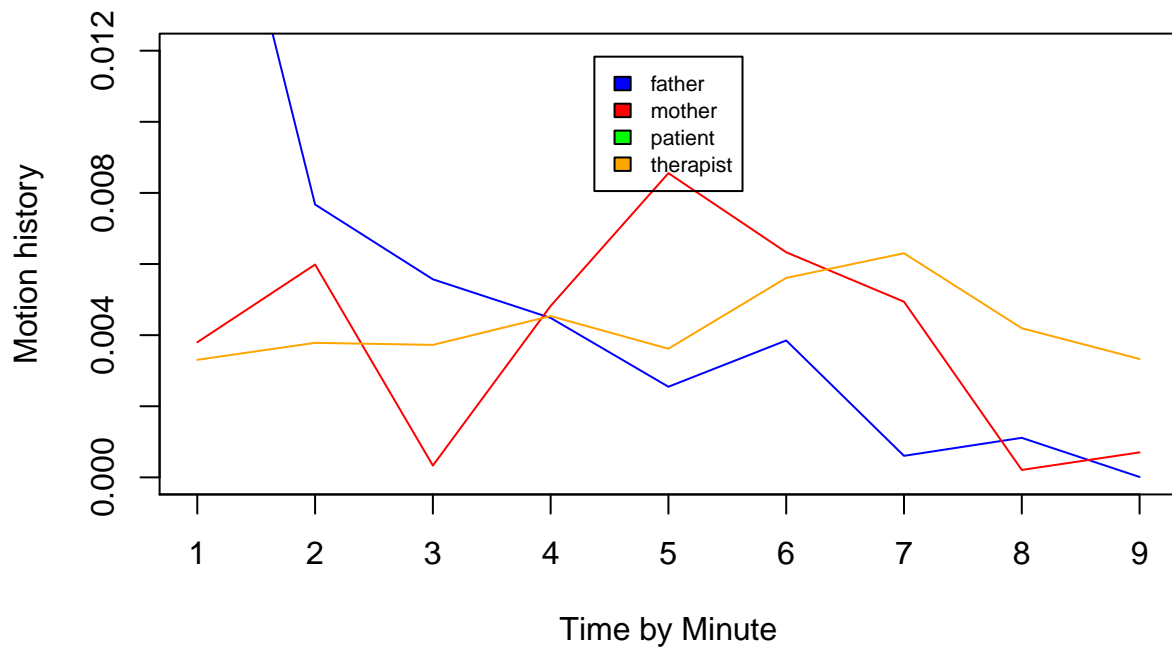
  PatientMinute<- MeanMotionByTime("patient", indexOfvideos=i, interval=1500, data)

  par(mar=c(4,4,4,2))
  plot (1:length(fatherMinute), fatherMinute, type="l", col="blue",
        main=paste("Mean motion history (non overlapping minute intervals)
on F1044", labelvideolist[i], " video" , sep=""),
        ylab="Motion history", xlab="Time by Minute", ylim=c(0, 12E-03),
        xaxp=c(0, length(fatherMinute), length(fatherMinute)))
  lines(MotherMinute, col="red")
  lines(TherapistMinute, col="orange")
  lines(PatientMinute, col="green")
  legend("top", inset=.05, ParticipantsList,
        fill=colOrderList, cex=0.7)}
```

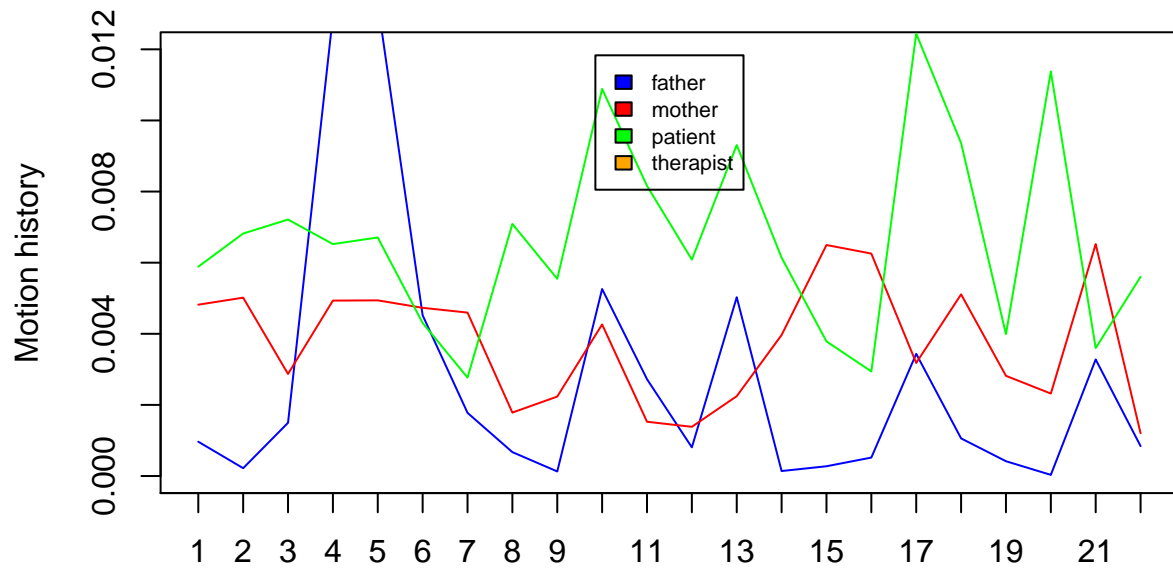
**Mean motion history (non overlapping minute intervals)  
on F1044C1 video**



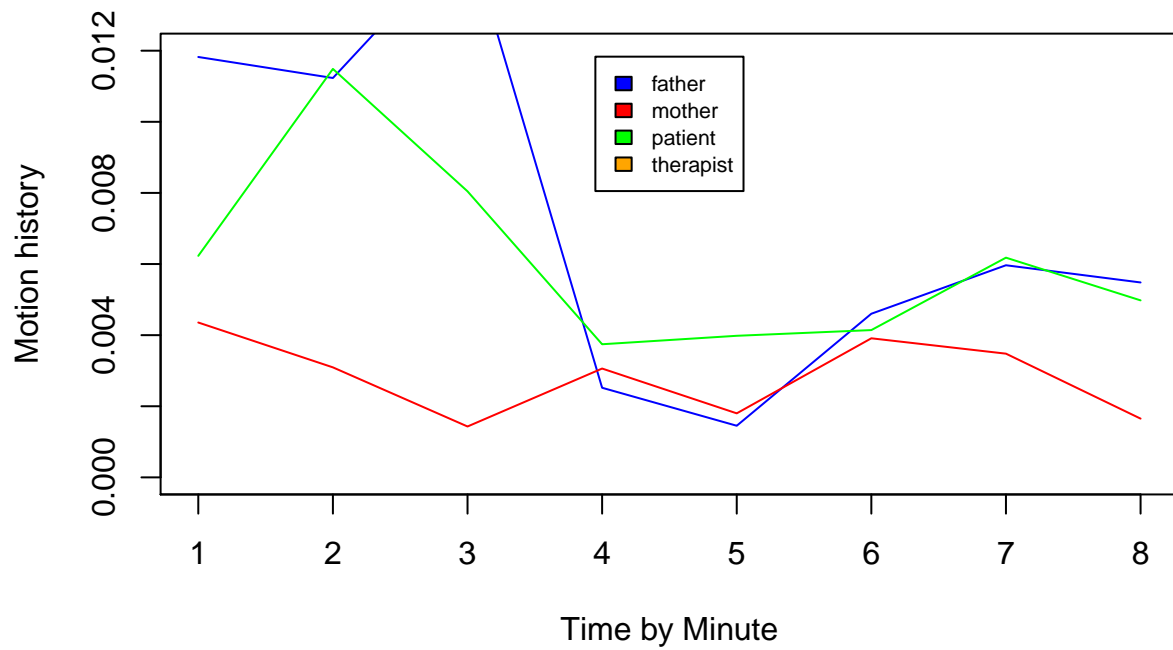
Time by Minute  
**Mean motion history (non overlapping minute intervals)  
on F1044C2 video**



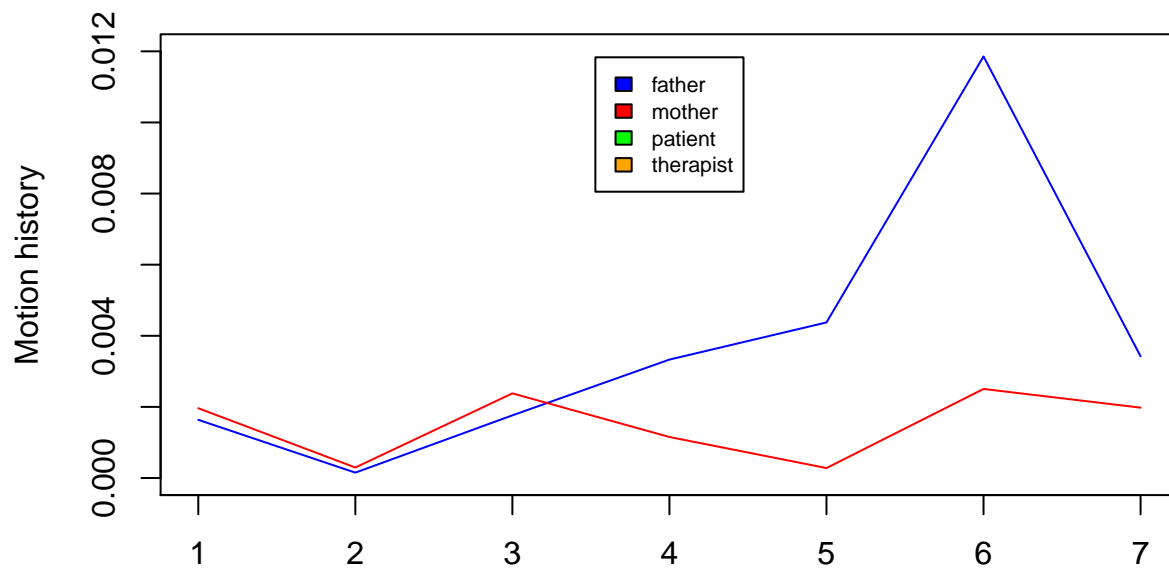
**Mean motion history (non overlapping minute intervals)  
on F1044D1 video**



**Mean motion history (non overlapping minute intervals)  
on F1044D2 video**

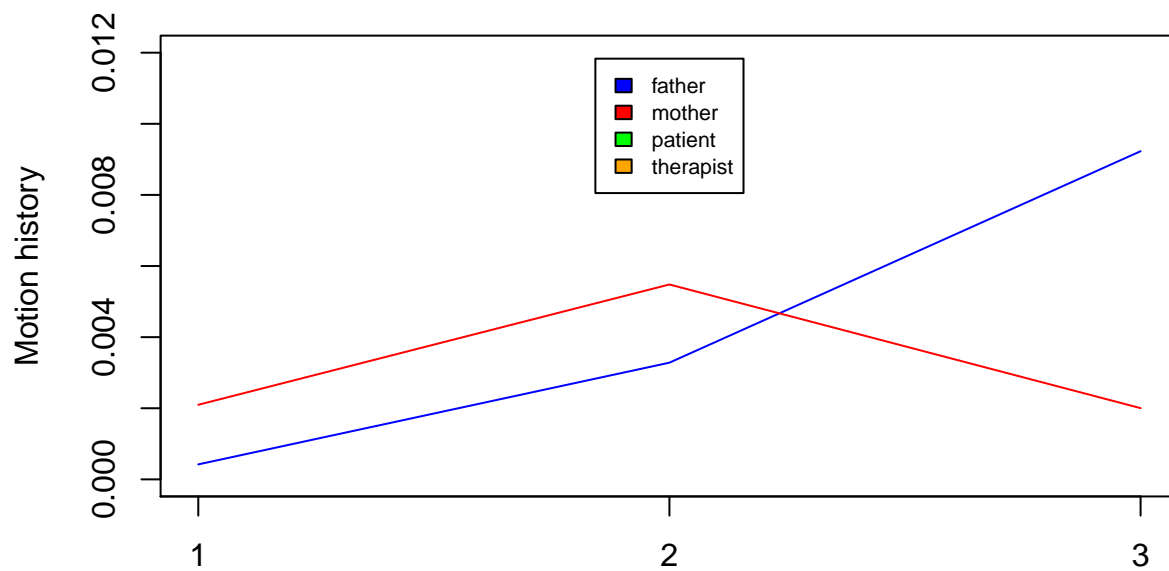


**Mean motion history (non overlapping minute intervals)  
on F1044E1 video**



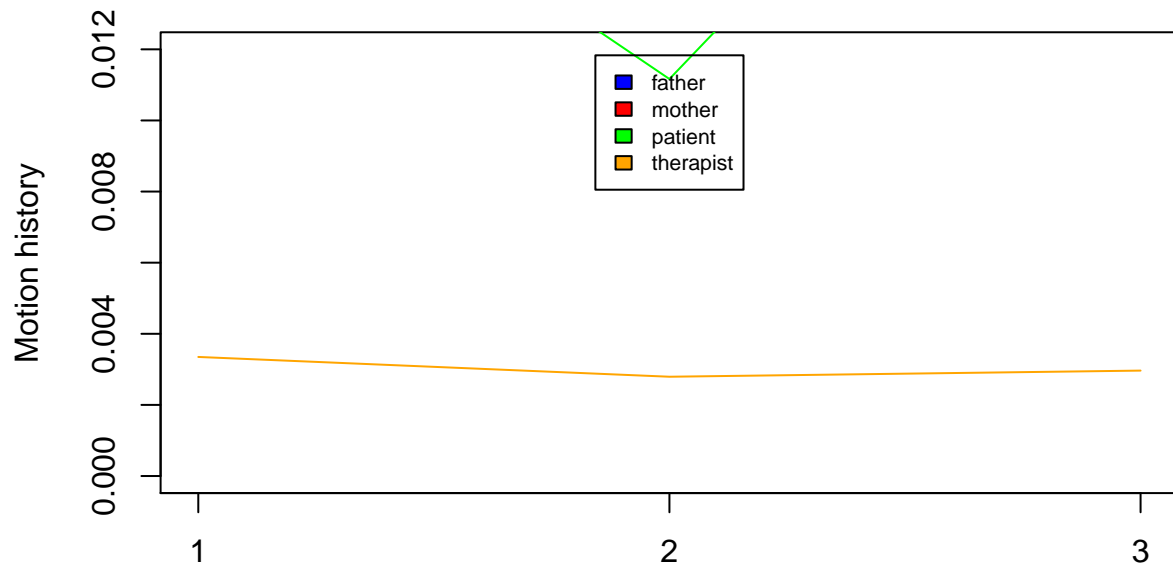
Time by Minute

**Mean motion history (non overlapping minute intervals)  
on F1044E2 video**

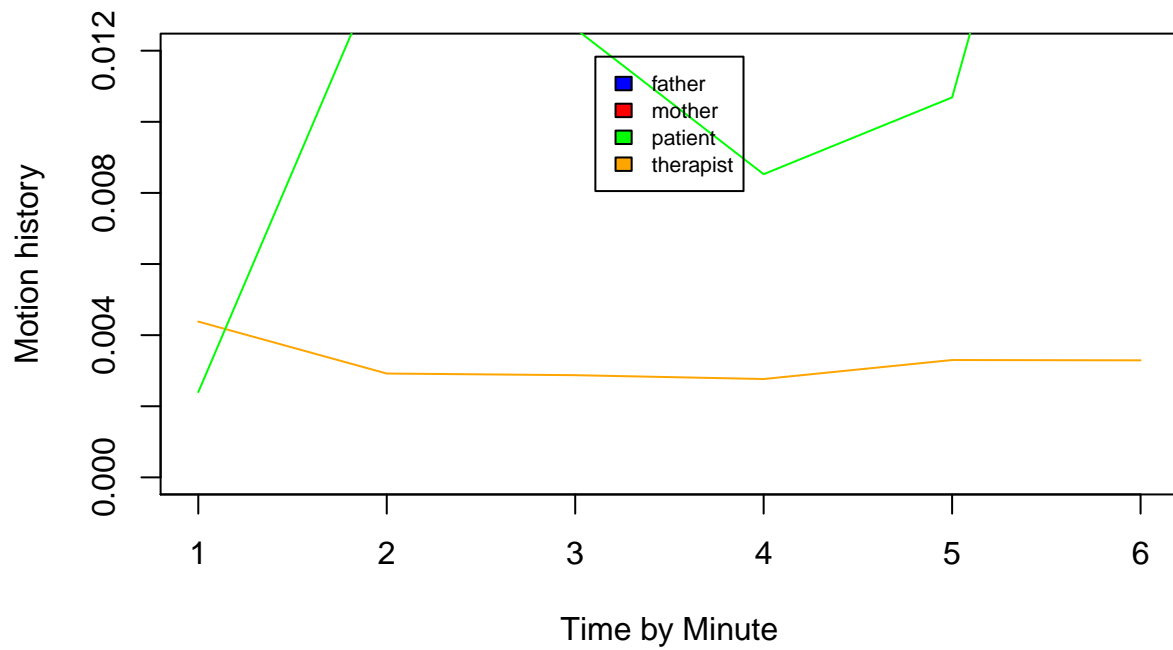


Time by Minute

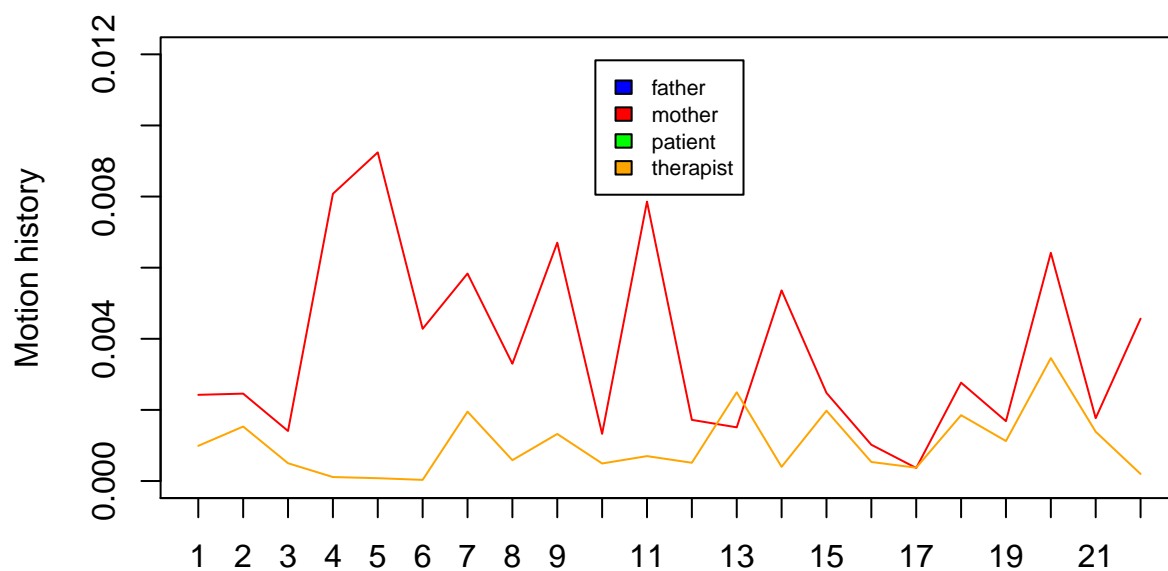
**Mean motion history (non overlapping minute intervals)  
on F1044F1 video**



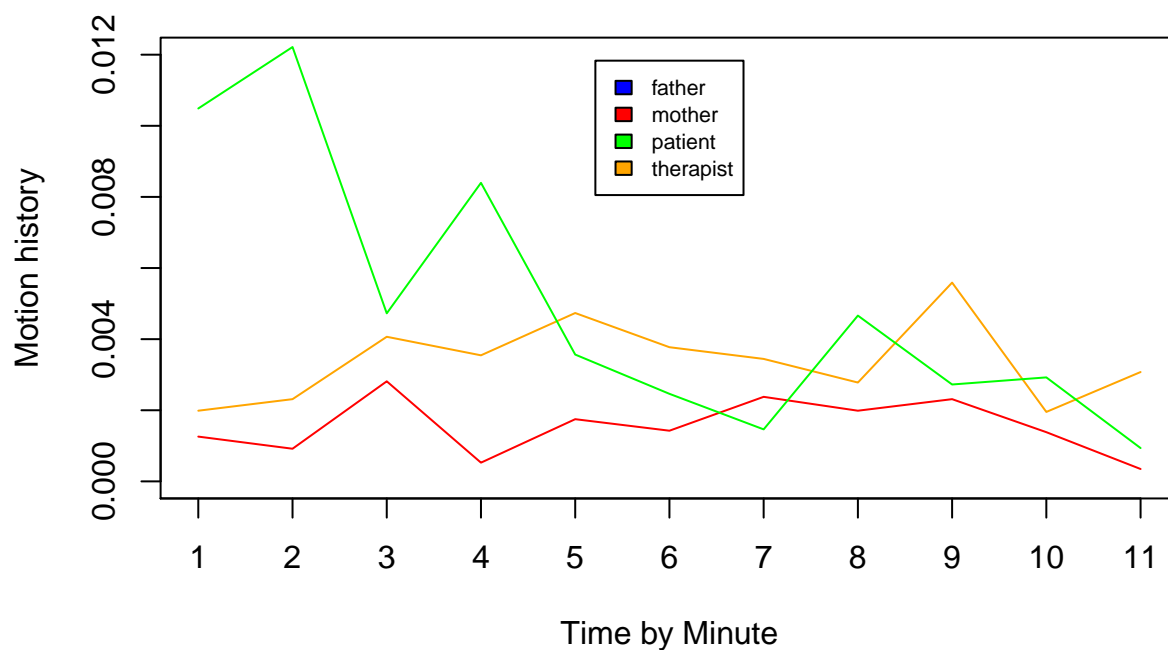
**Mean motion history (non overlapping minute intervals)  
on F1044F2 video**



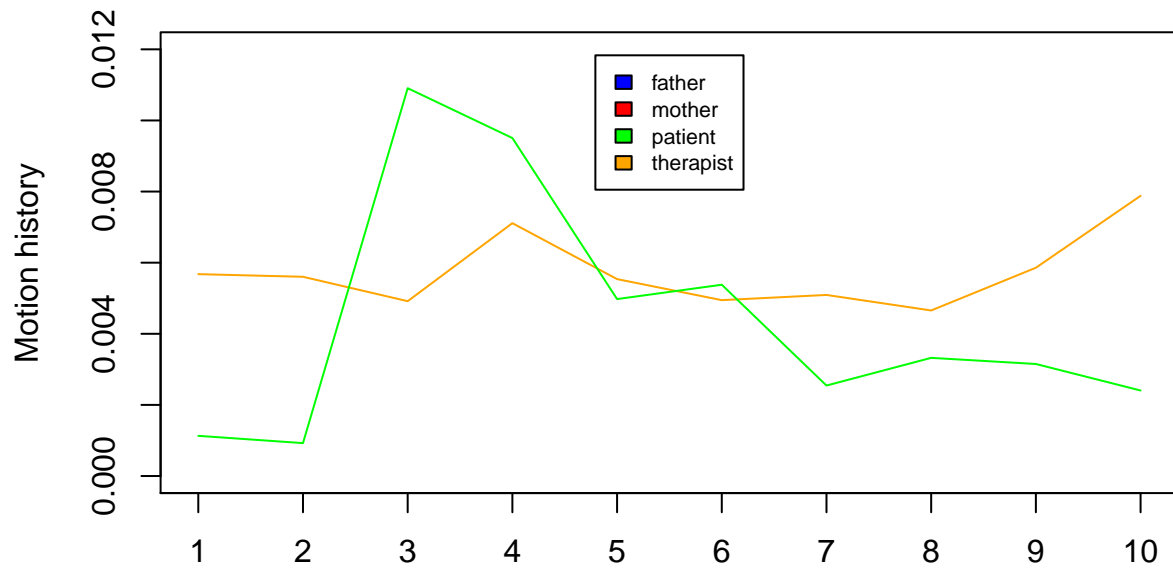
**Mean motion history (non overlapping minute intervals)  
on F1044G video**



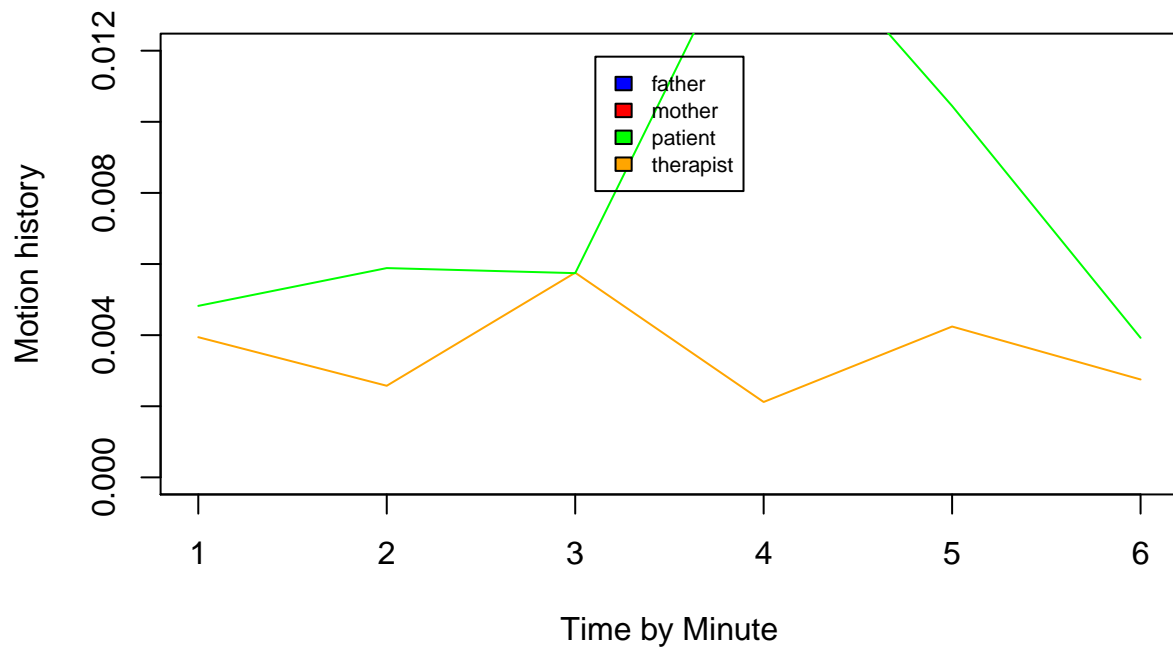
**Mean motion history (non overlapping minute intervals)  
on F1044H1 video**



**Mean motion history (non overlapping minute intervals)  
on F1044H2 video**

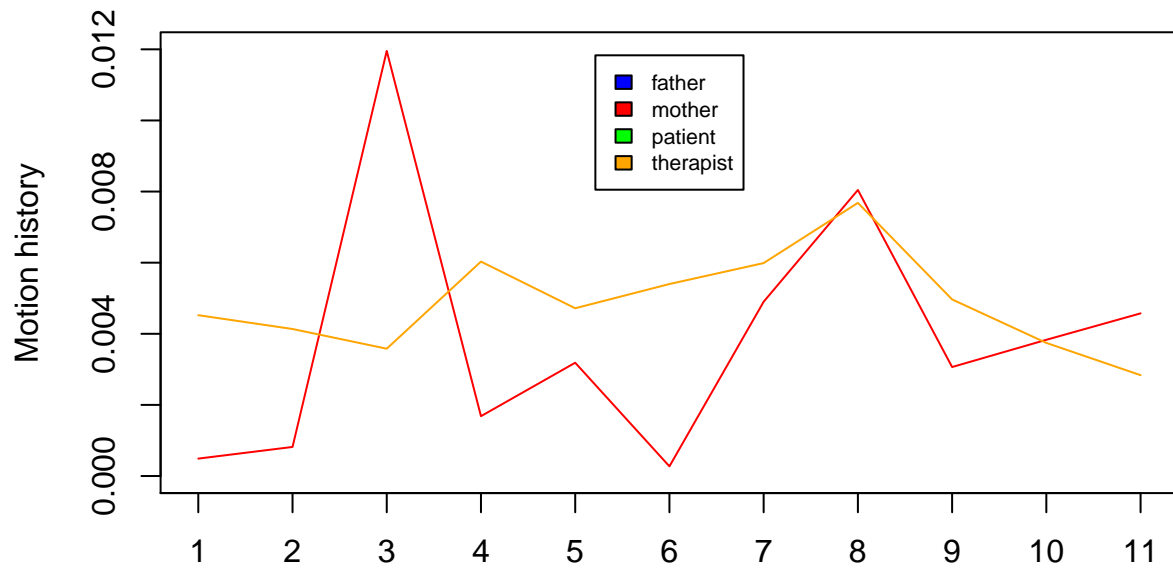


Time by Minute  
**Mean motion history (non overlapping minute intervals)  
on F1044I1 video**

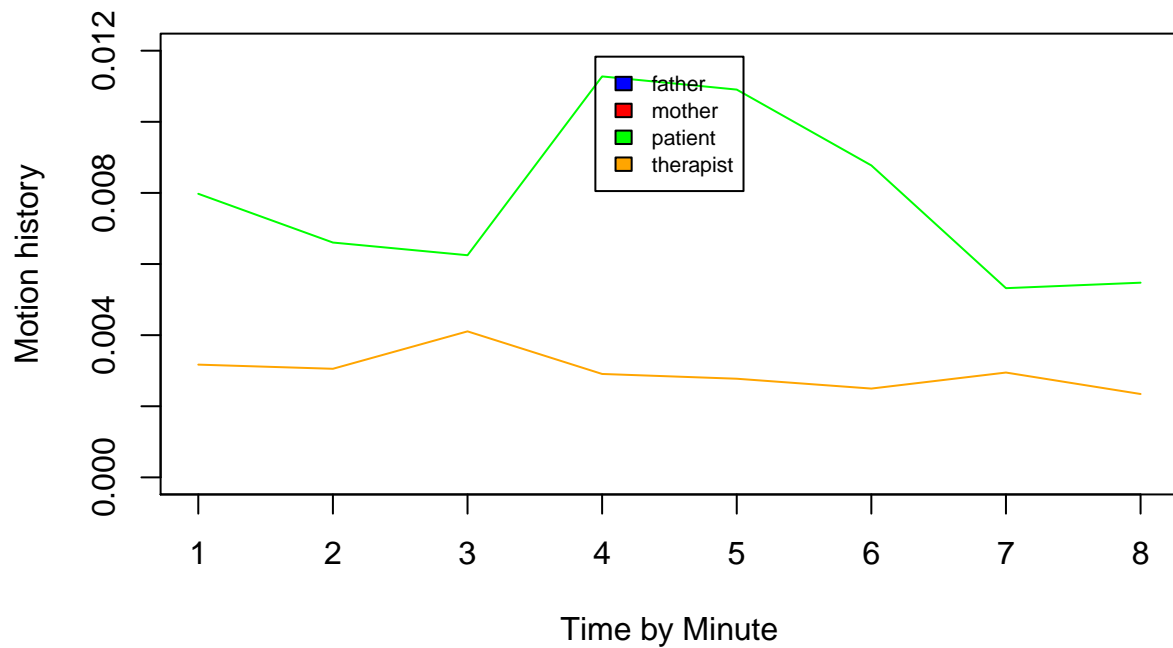




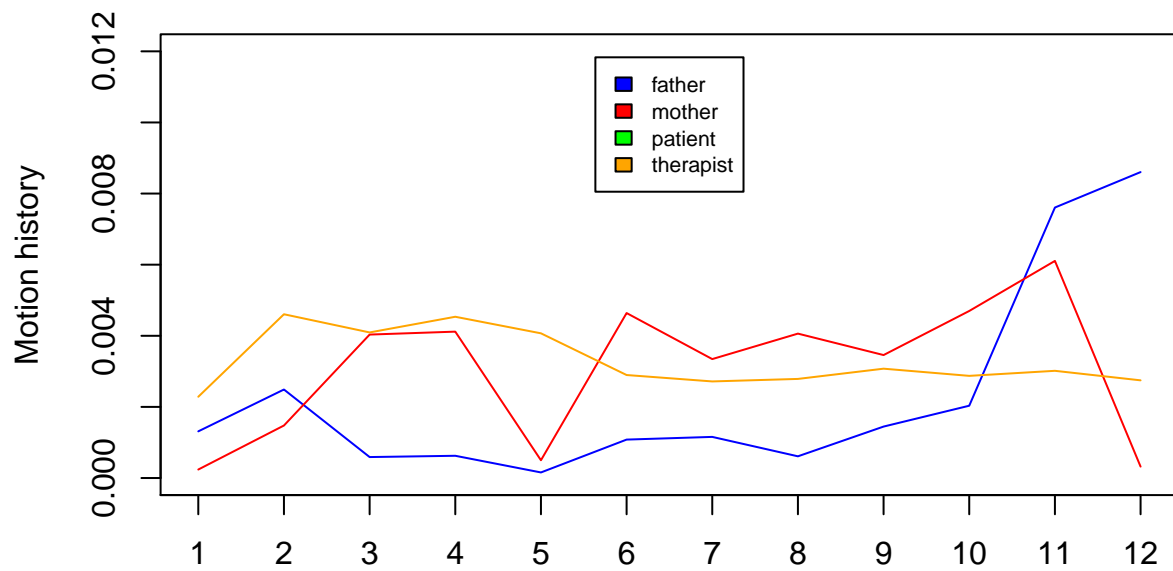
**Mean motion history (non overlapping minute intervals)  
on F1044I2 video**



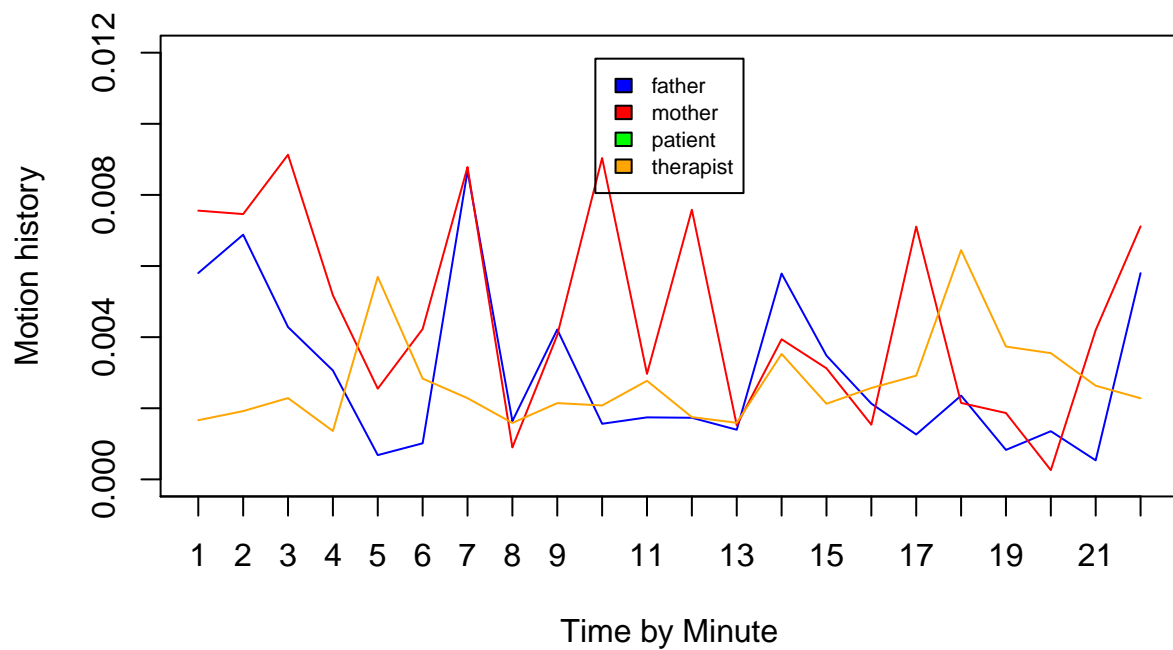
**Mean motion history (non overlapping minute intervals)  
on F1044L1 video**



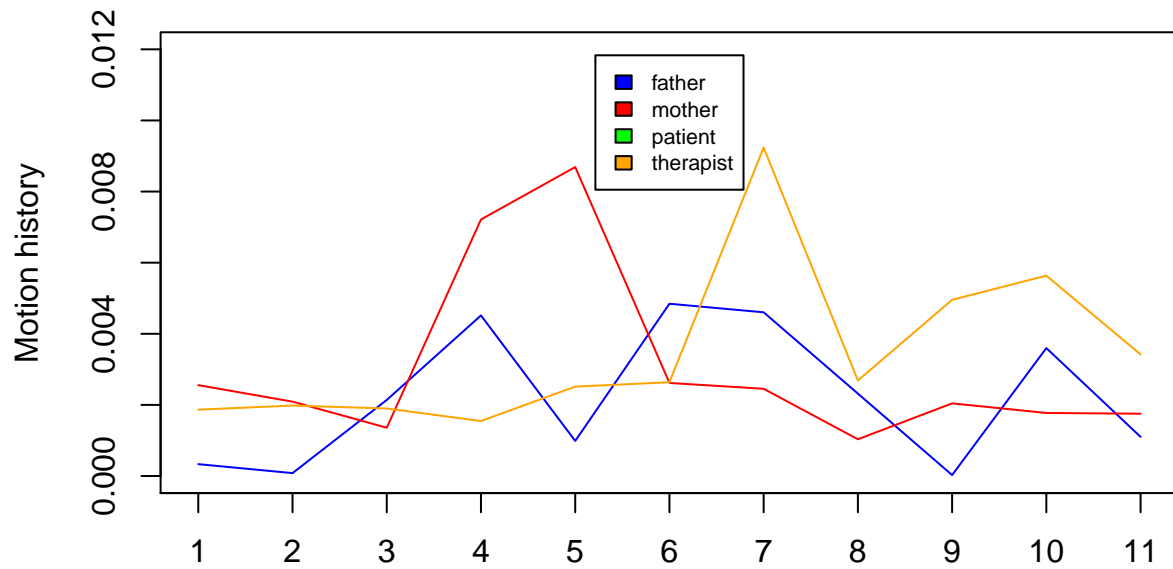
**Mean motion history (non overlapping minute intervals)  
on F1044L2 video**



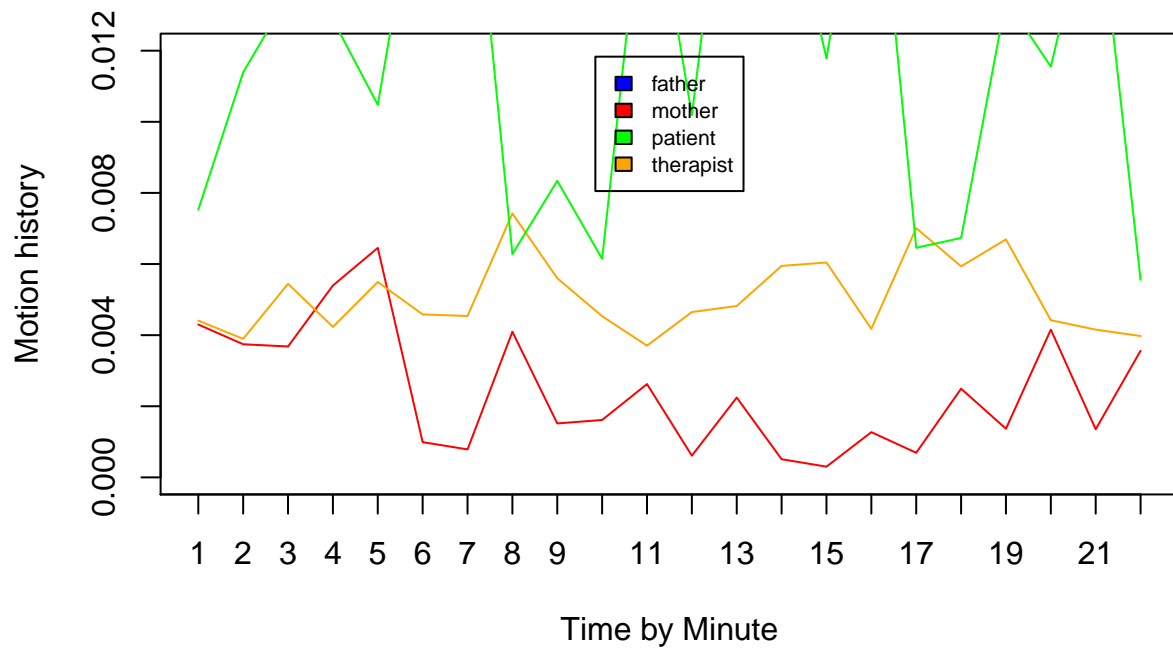
**Mean motion history (non overlapping minute intervals)  
on F1044M1 video**



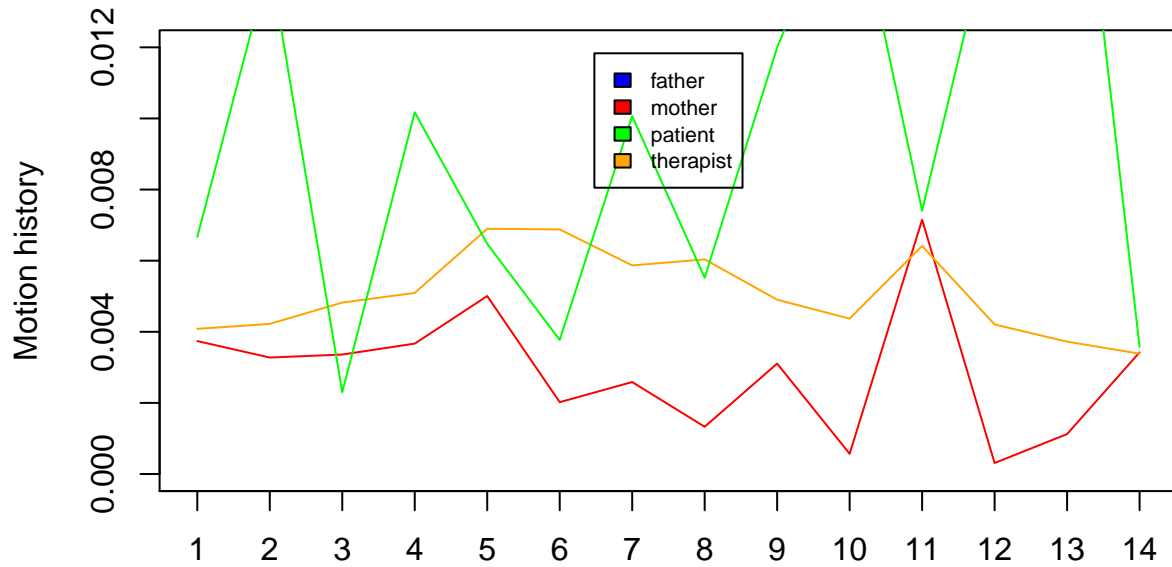
**Mean motion history (non overlapping minute intervals)  
on F1044M2 video**



**Mean motion history (non overlapping minute intervals)  
on F1044N video**

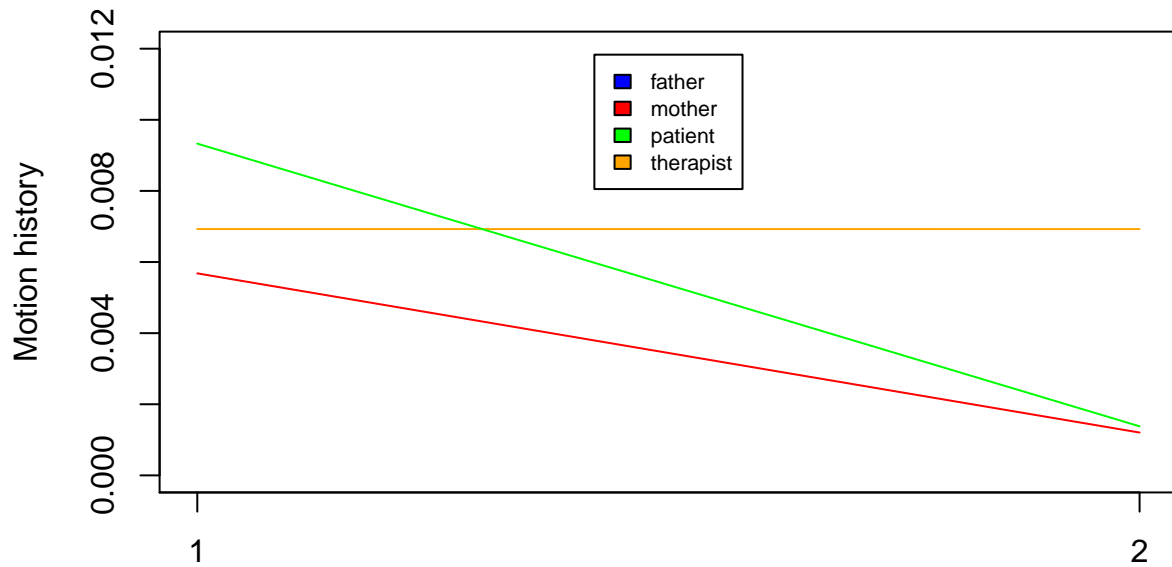


**Mean motion history (non overlapping minute intervals)  
on F1044O1 video**



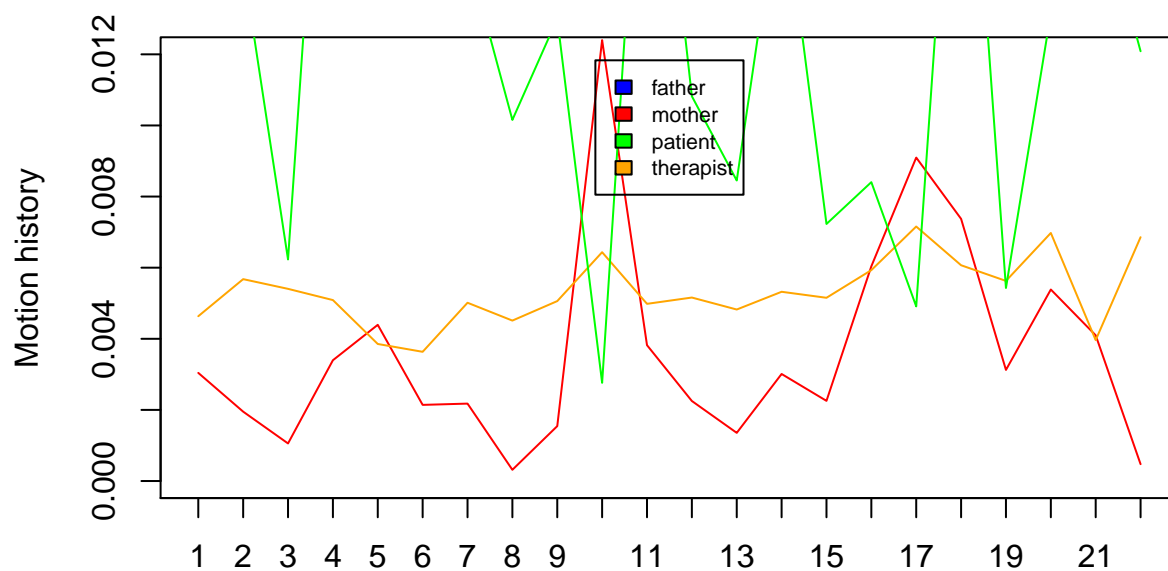
Time by Minute

**Mean motion history (non overlapping minute intervals)  
on F1044O2 video**

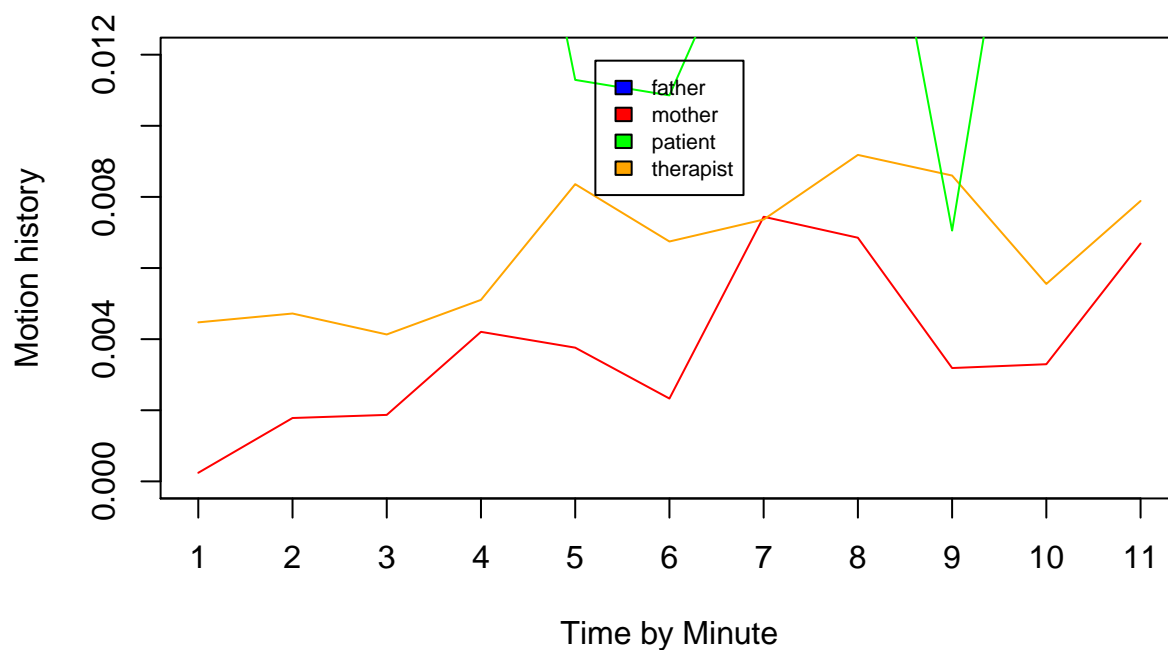


Time by Minute

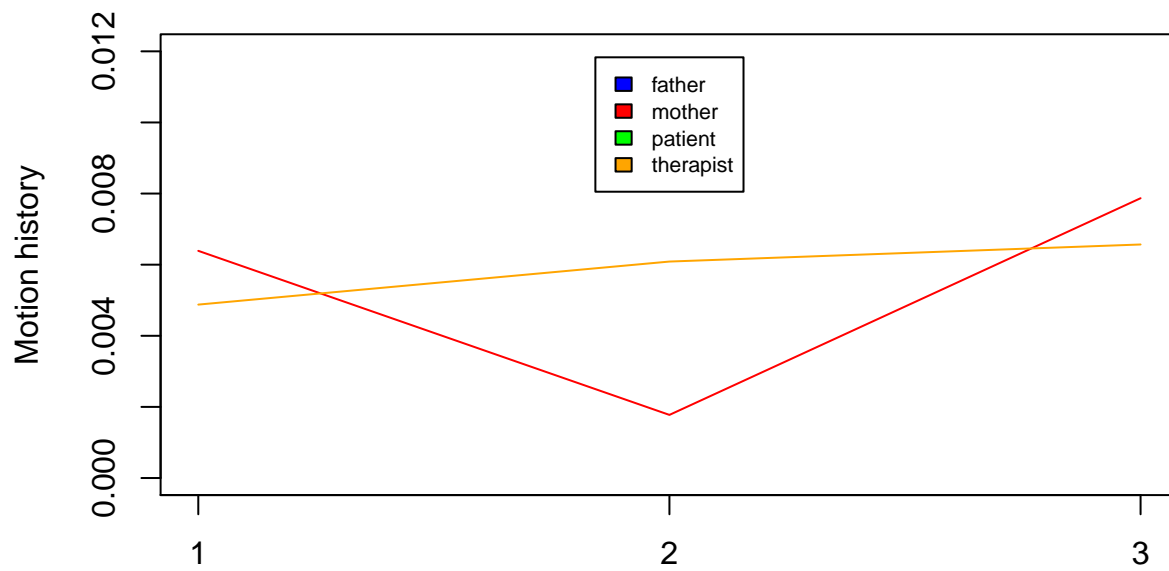
**Mean motion history (non overlapping minute intervals)  
on F1044P video**



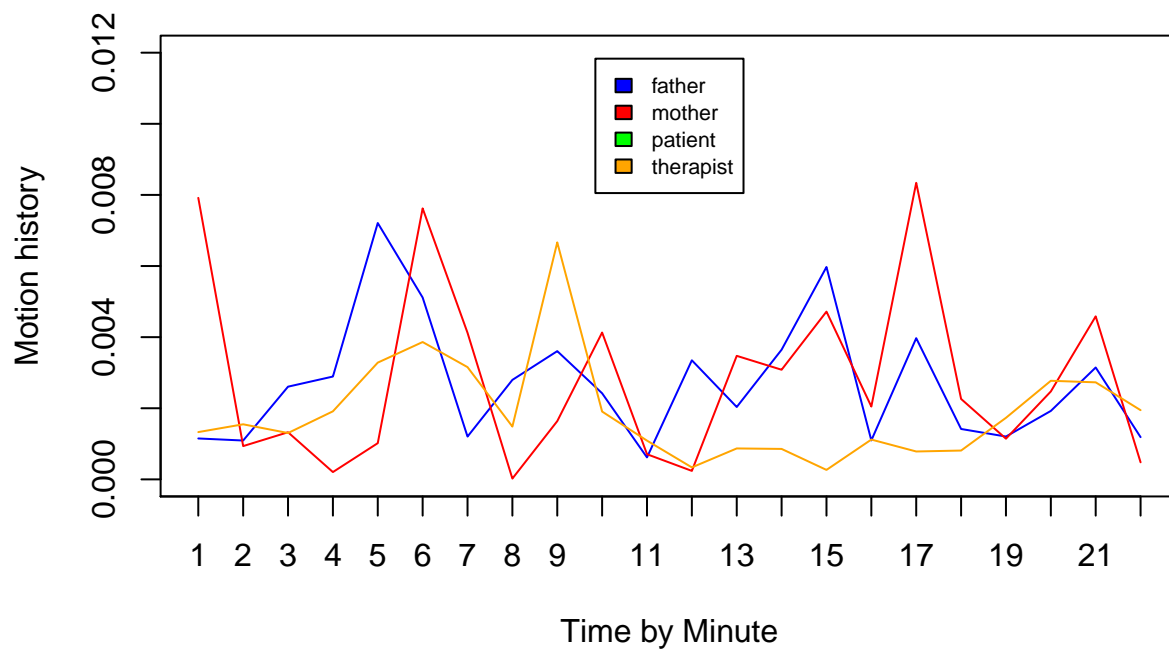
Time by Minute  
**Mean motion history (non overlapping minute intervals)  
on F1044Q1 video**



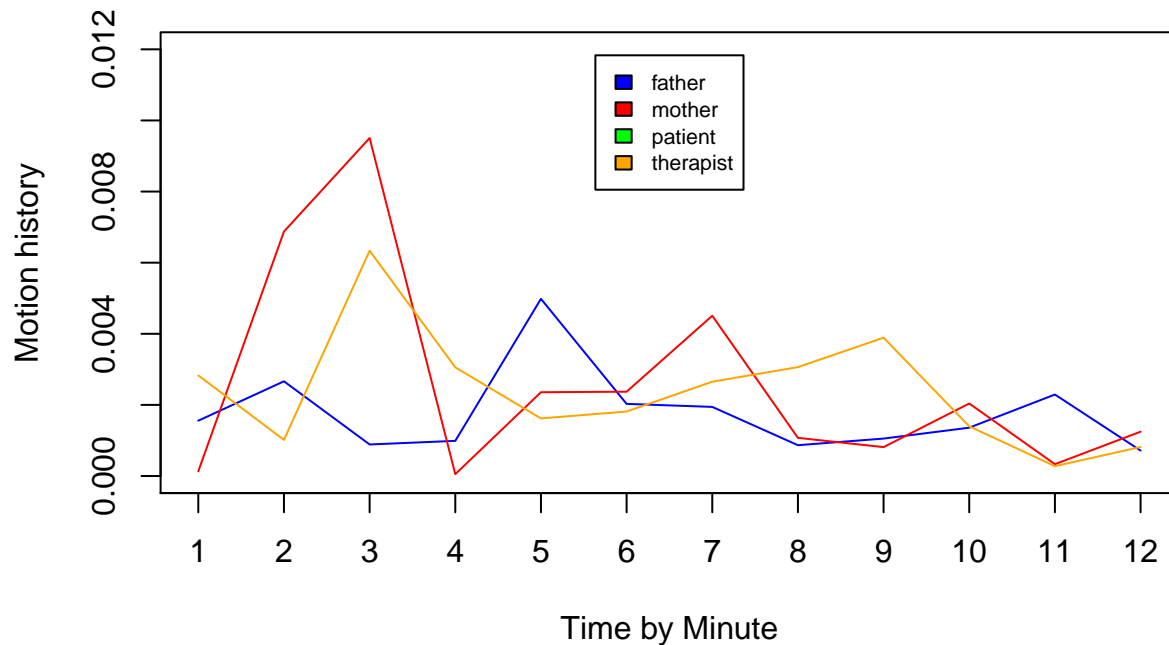
**Mean motion history (non overlapping minute intervals)  
on F1044Q2 video**



Time by Minute  
**Mean motion history (non overlapping minute intervals)  
on F1044R1 video**



## Mean motion history (non overlapping minute intervals) on F1044R2 video



## Mean log motion history by minute plots

```
for (i in 1:NumberOfvideos){
  fatherMinute<- MeanMotionByTime("logFather", indexOfvideos=i, interval=1500, data)

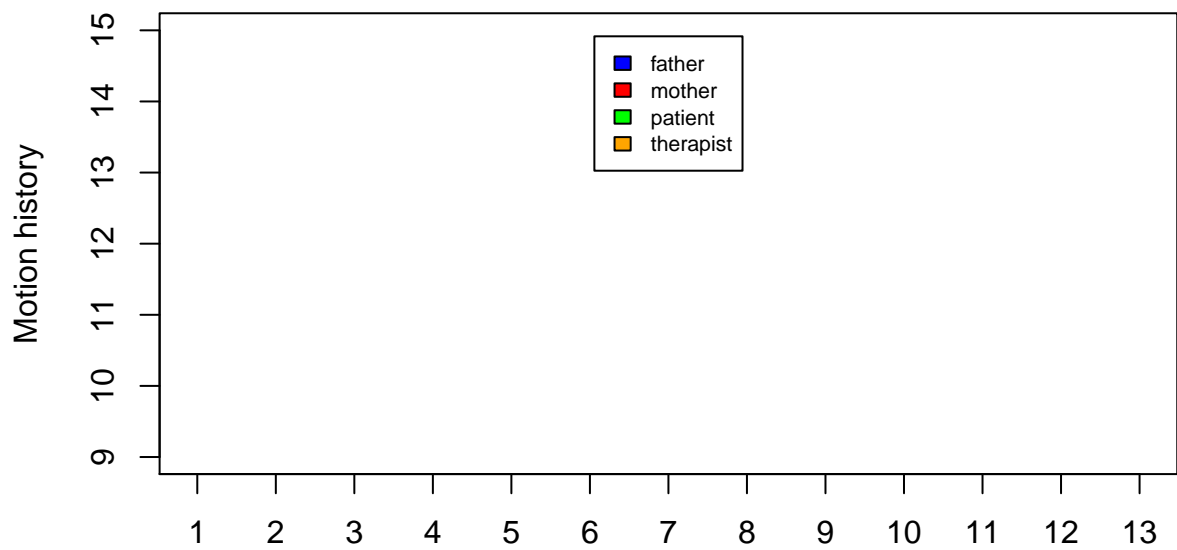
  MotherMinute<- MeanMotionByTime("logMother", indexOfvideos=i, interval=1500, data)

  TherapistMinute<- MeanMotionByTime("logTherapist", indexOfvideos=i, interval=1500, data)

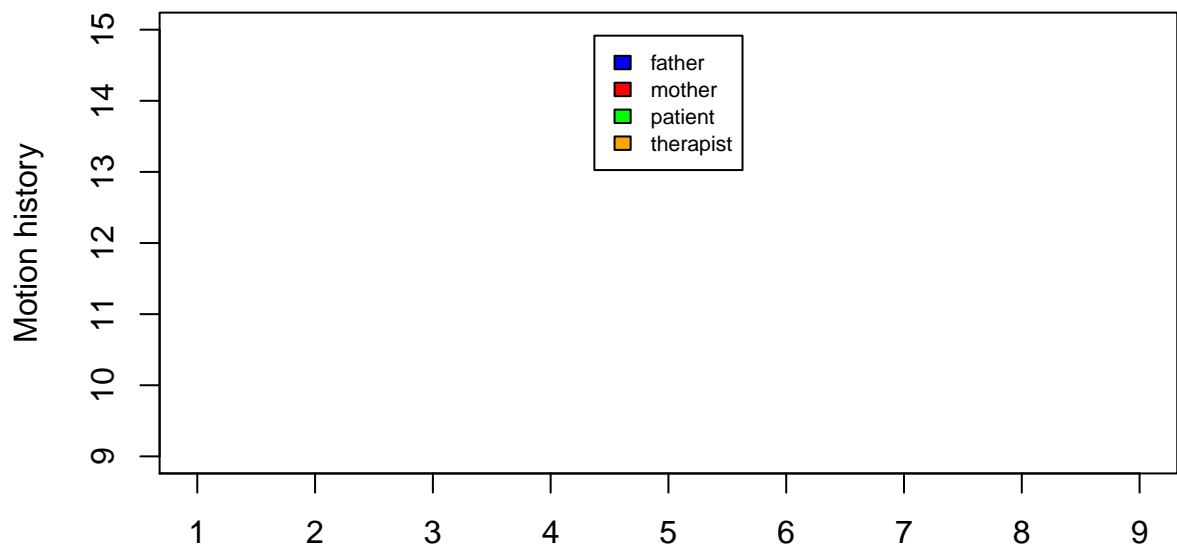
  PatientMinute<- MeanMotionByTime("logPatient", indexOfvideos=i, interval=1500, data)

  par(mar=c(4,4,4,2))
  plot (1:length(fatherMinute), fatherMinute, type="l", col="blue",
        main=paste("Mean motion history (non overlapping minute intervals)
on F1044", labelvideolist[i], " video" , sep=""),
        ylab="Motion history", xlab="Time by Minute", ylim=c(9, 15),
        xaxp=c(0, length(fatherMinute), length(fatherMinute)))
  lines(MotherMinute, col="red")
  lines(TherapistMinute, col="orange")
  lines(PatientMinute, col="green")
  legend("top", inset=.05, ParticipantsList,
        fill=colOrderList, cex=0.7)}
```

**Mean motion history (non overlapping minute intervals)  
on F1044C1 video**

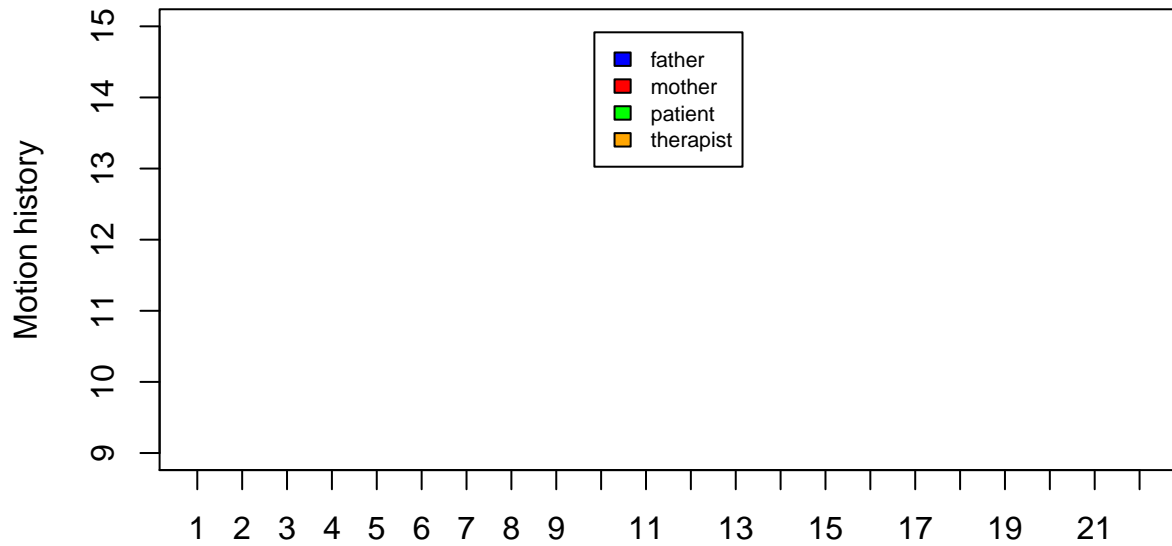


Time by Minute  
**Mean motion history (non overlapping minute intervals)  
on F1044C2 video**

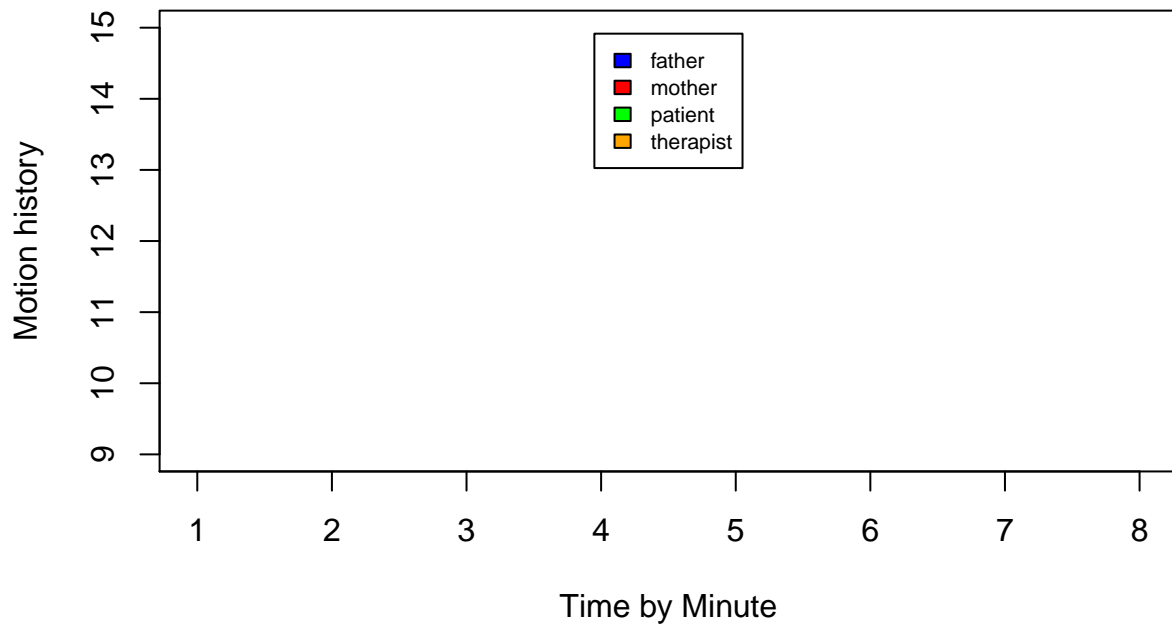




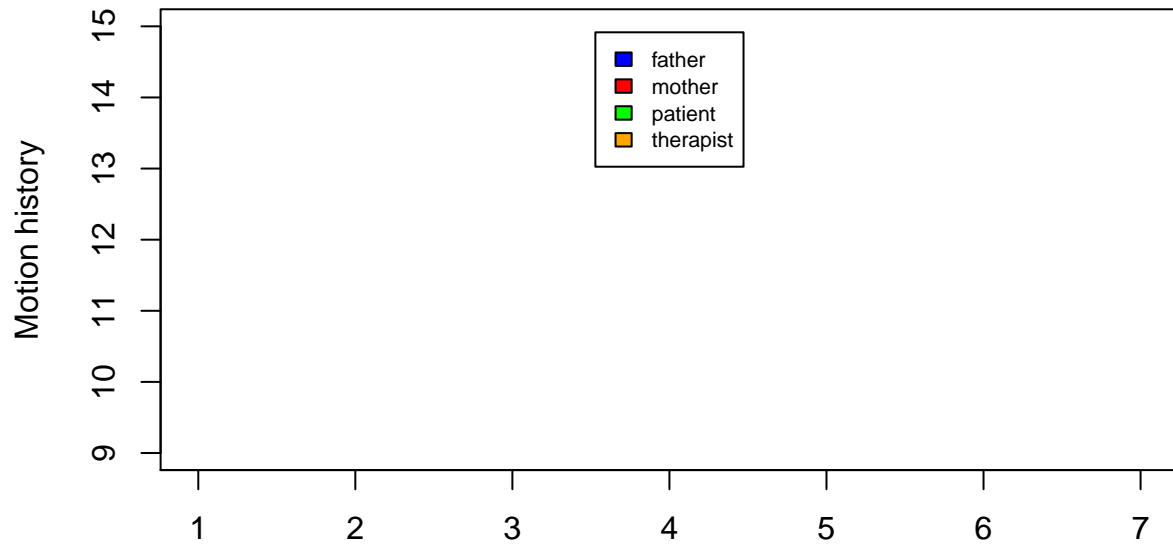
**Mean motion history (non overlapping minute intervals)  
on F1044D1 video**



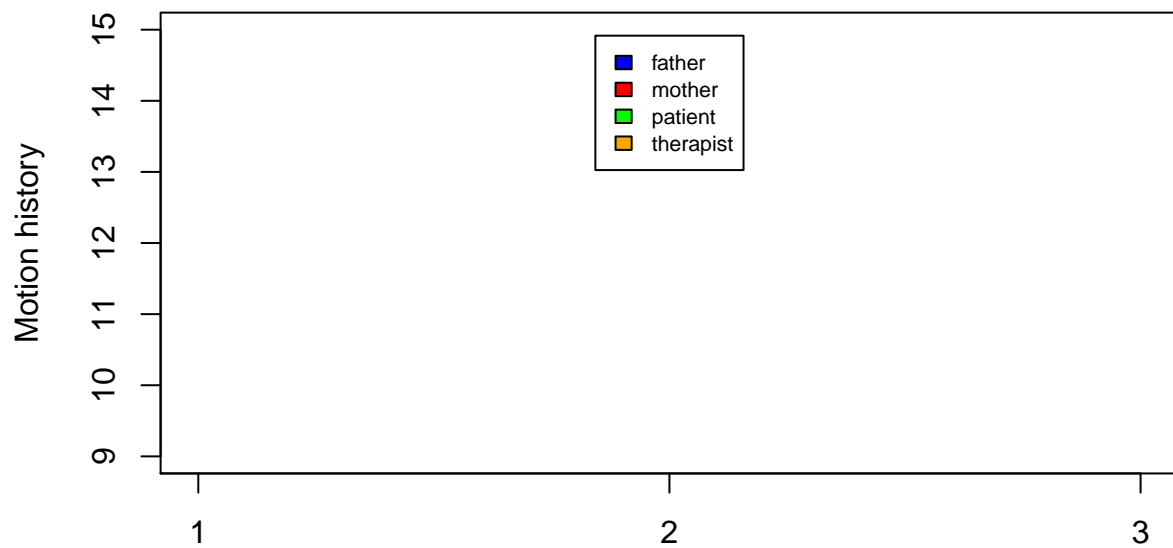
**Mean motion history (non overlapping minute intervals)  
on F1044D2 video**



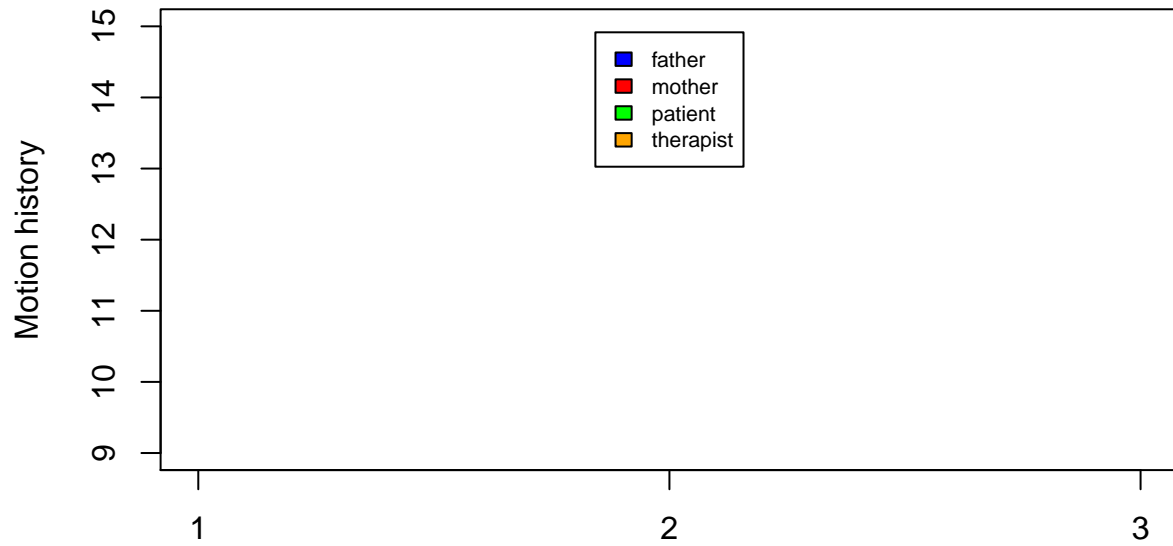
**Mean motion history (non overlapping minute intervals)  
on F1044E1 video**



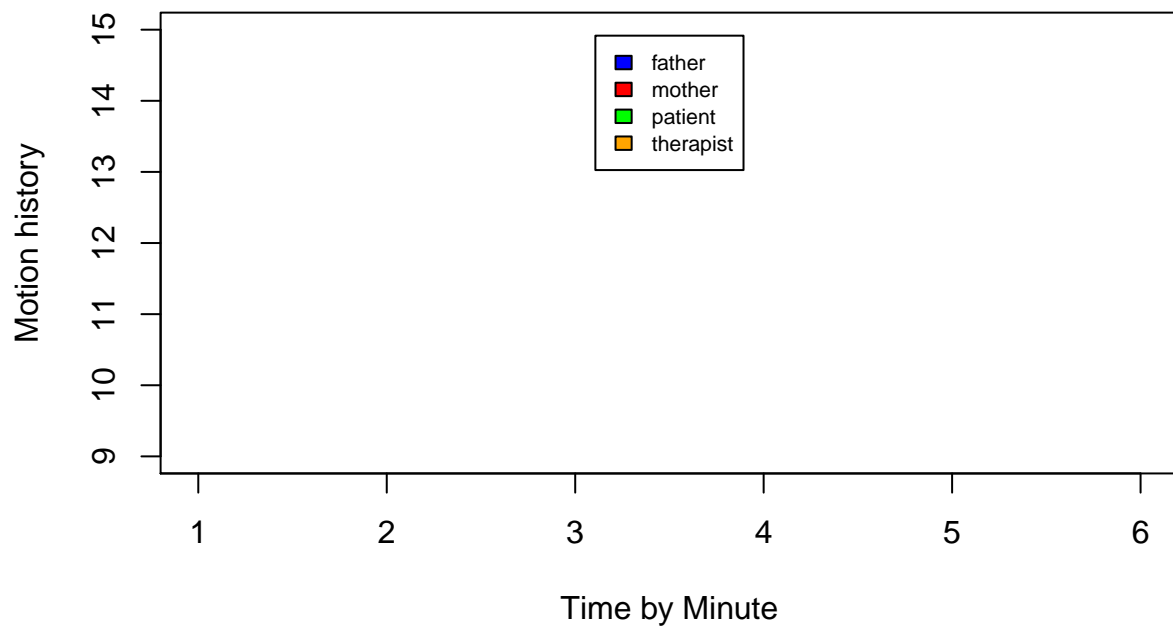
Time by Minute  
**Mean motion history (non overlapping minute intervals)  
on F1044E2 video**



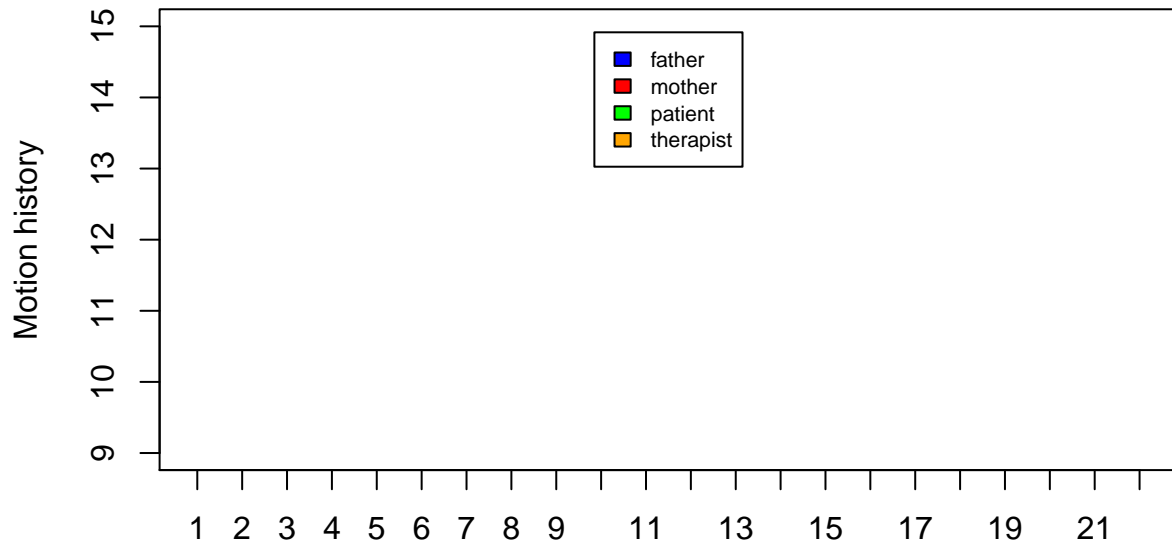
**Mean motion history (non overlapping minute intervals)  
on F1044F1 video**



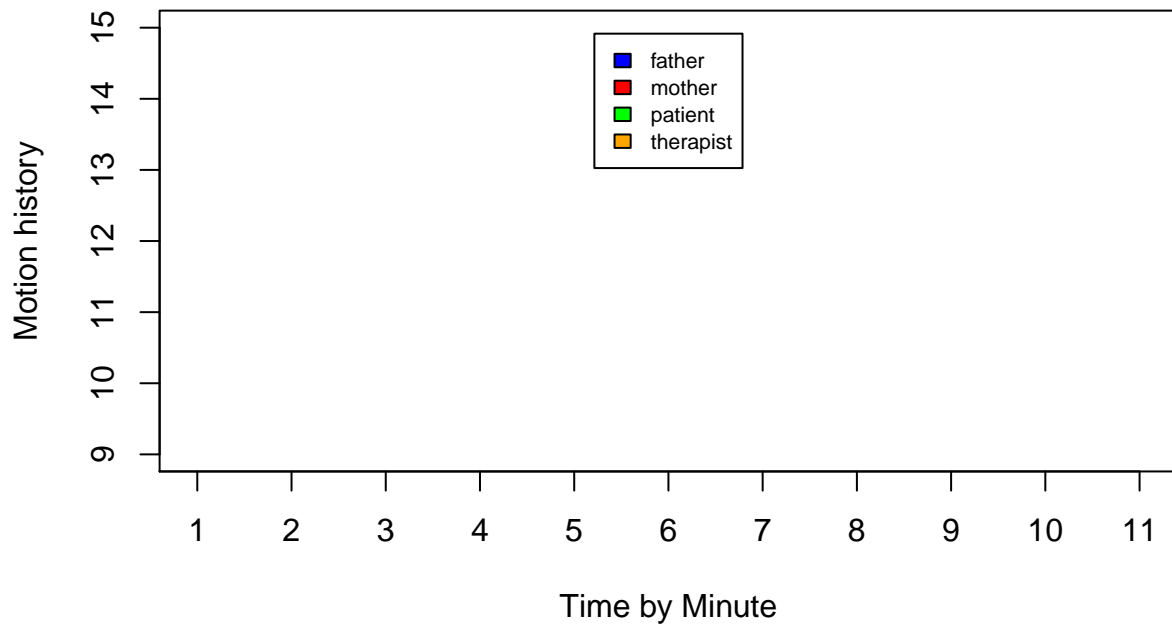
Time by Minute  
**Mean motion history (non overlapping minute intervals)  
on F1044F2 video**



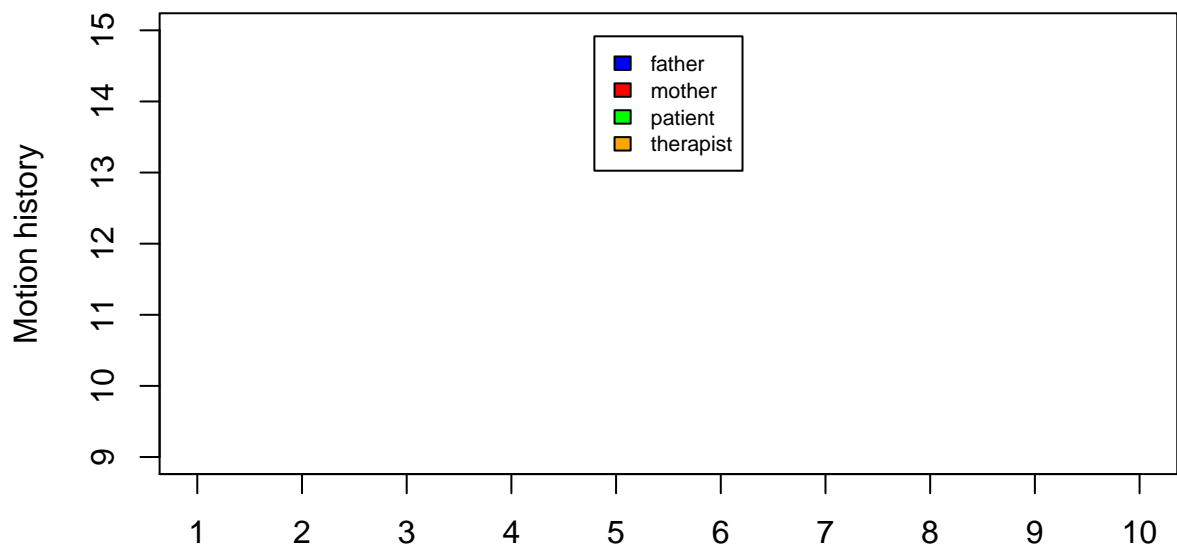
**Mean motion history (non overlapping minute intervals)  
on F1044G video**



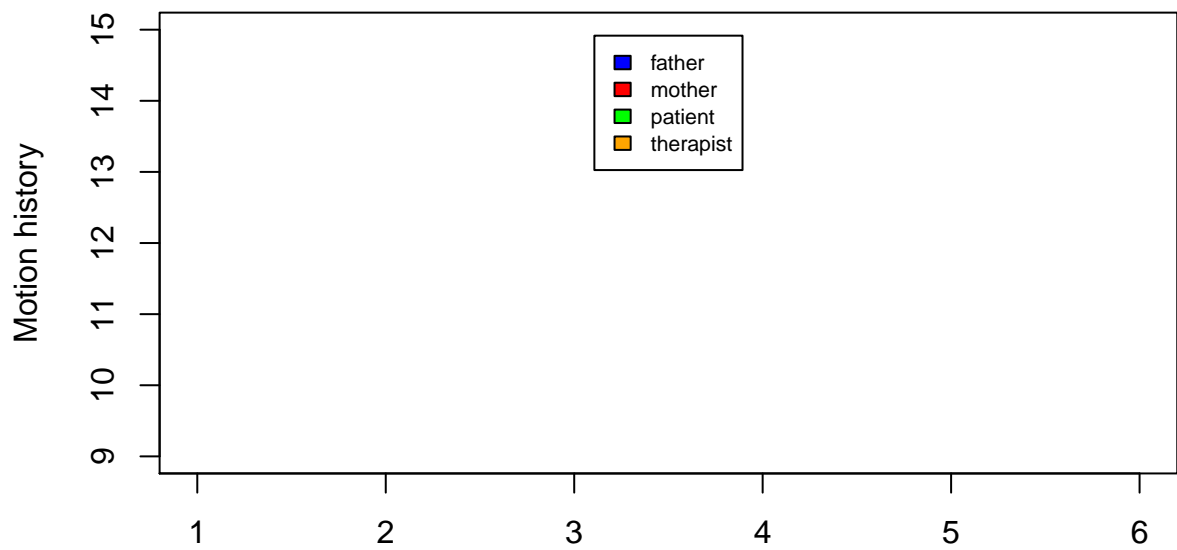
Time by Minute  
**Mean motion history (non overlapping minute intervals)  
on F1044H1 video**



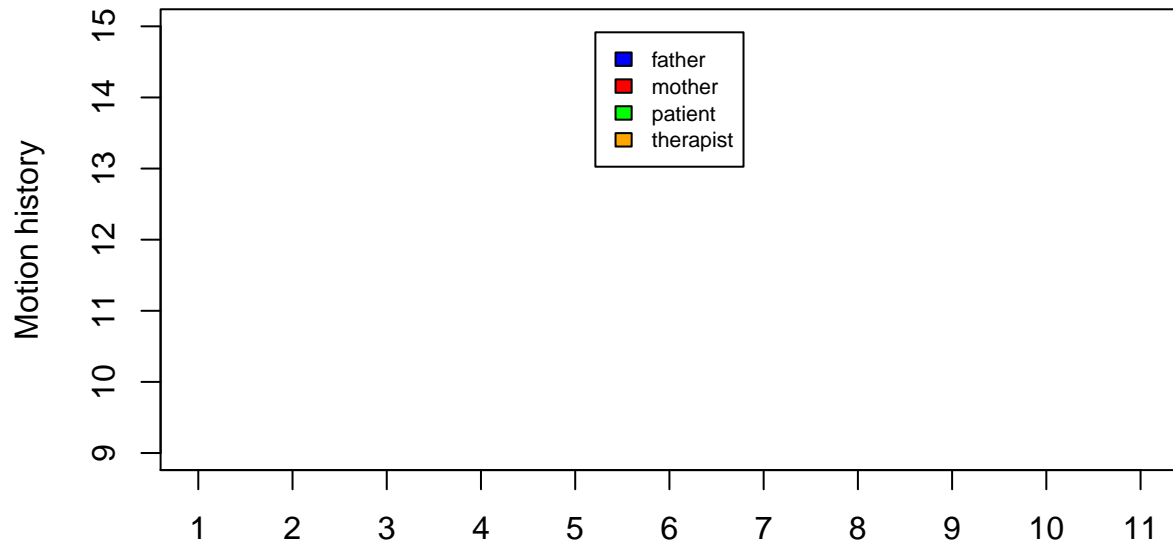
**Mean motion history (non overlapping minute intervals)  
on F1044H2 video**



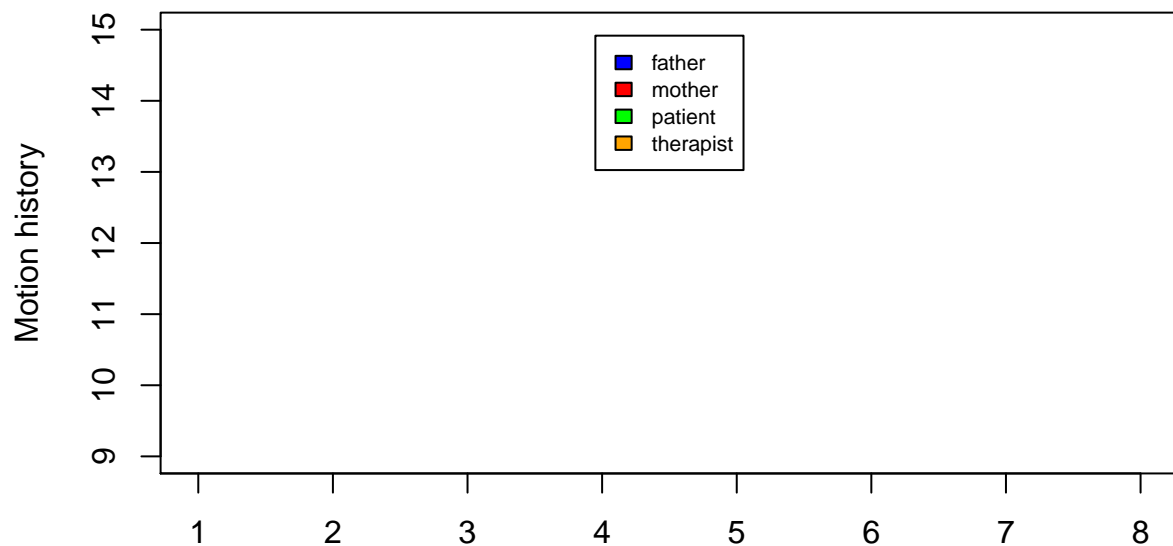
Time by Minute  
**Mean motion history (non overlapping minute intervals)  
on F1044I1 video**



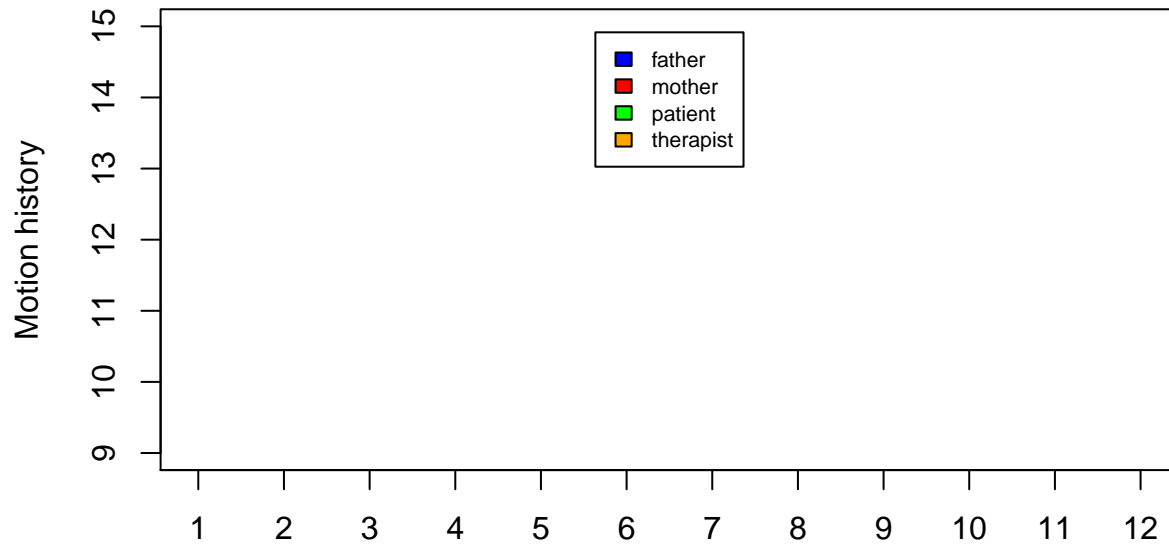
**Mean motion history (non overlapping minute intervals)  
on F1044I2 video**



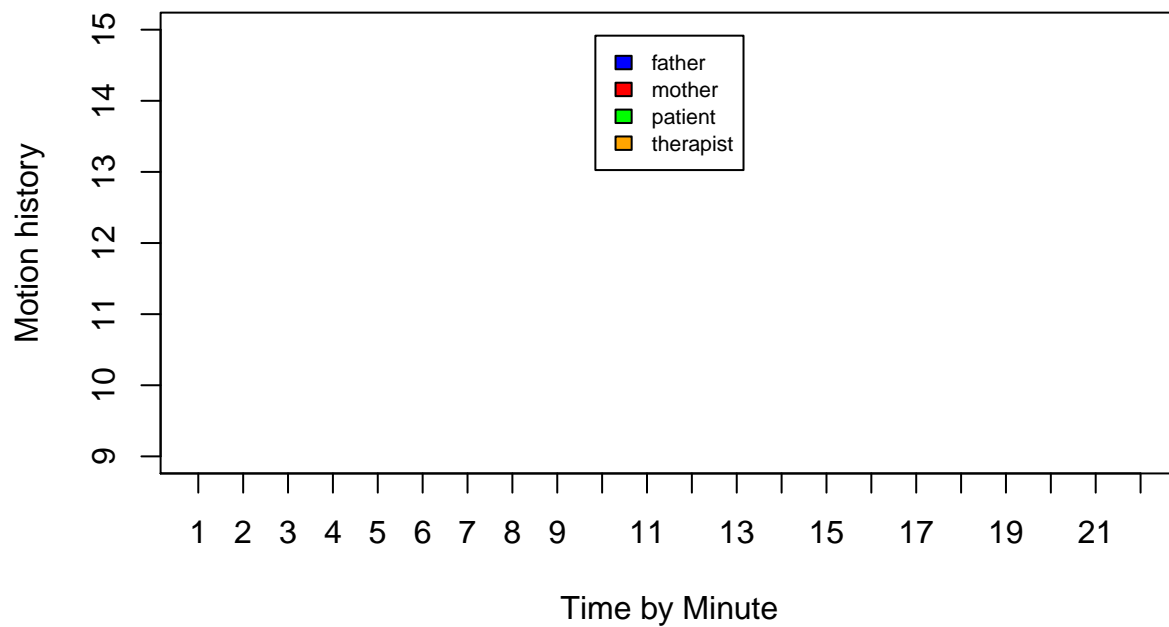
Time by Minute  
**Mean motion history (non overlapping minute intervals)  
on F1044L1 video**



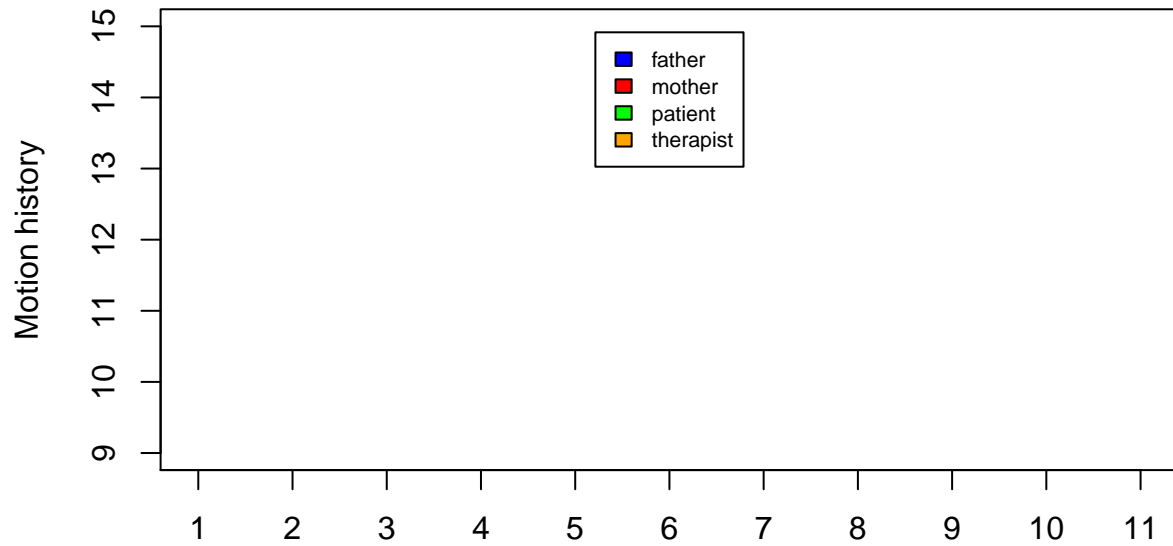
**Mean motion history (non overlapping minute intervals)  
on F1044L2 video**



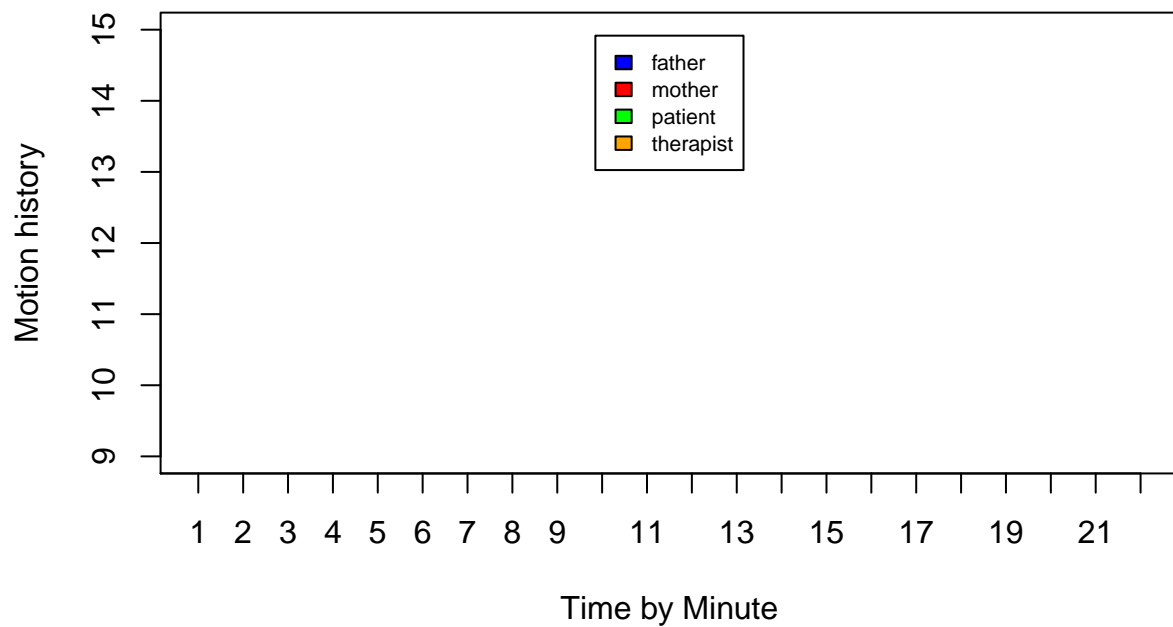
Time by Minute  
**Mean motion history (non overlapping minute intervals)  
on F1044M1 video**



**Mean motion history (non overlapping minute intervals)  
on F1044M2 video**

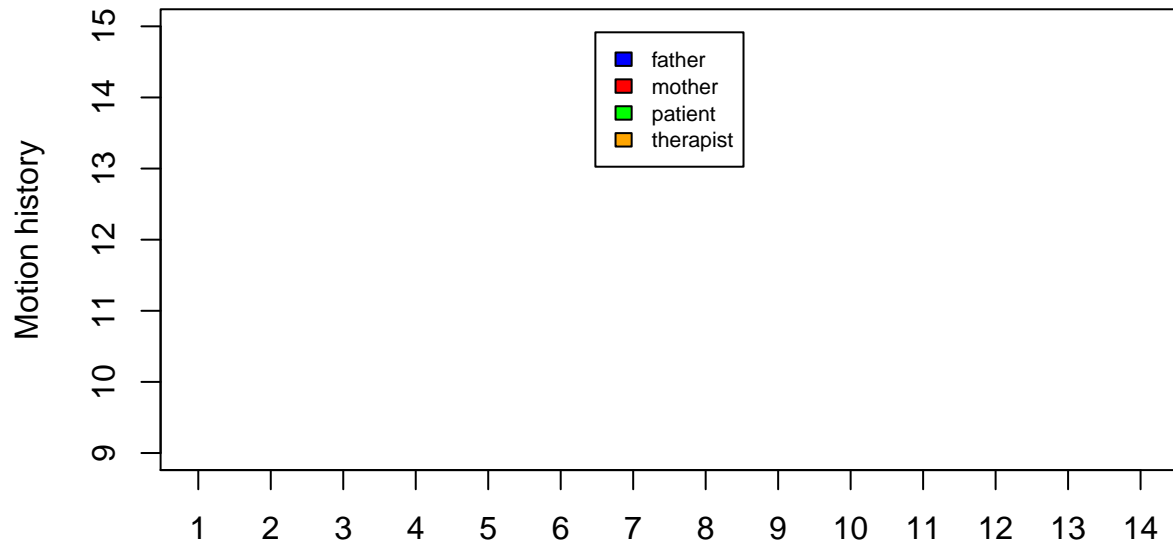


Time by Minute  
**Mean motion history (non overlapping minute intervals)  
on F1044N video**

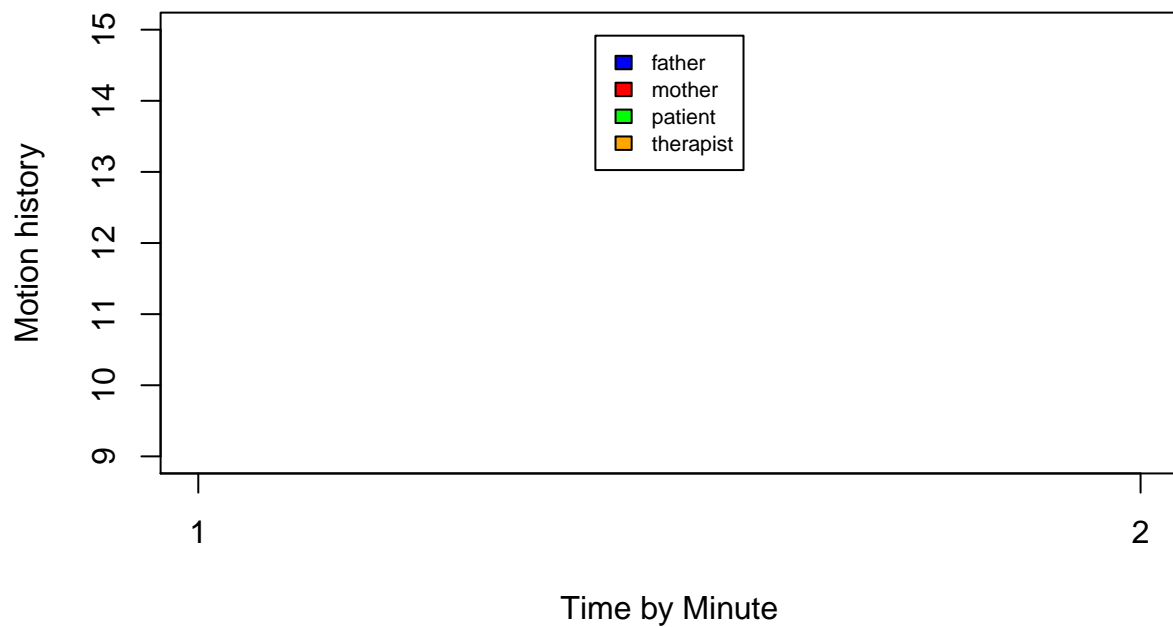




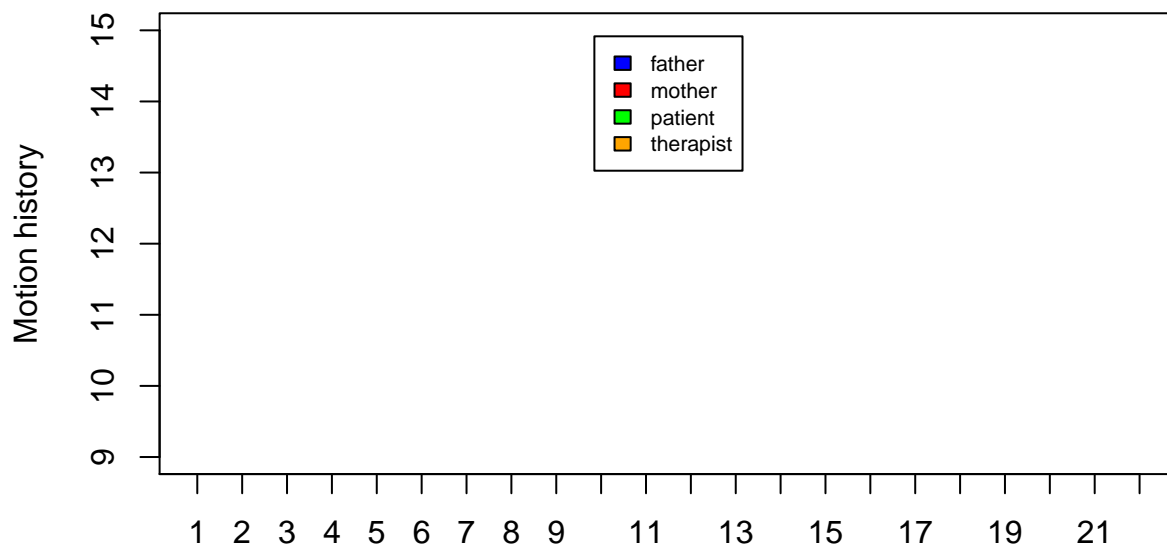
**Mean motion history (non overlapping minute intervals)  
on F1044O1 video**



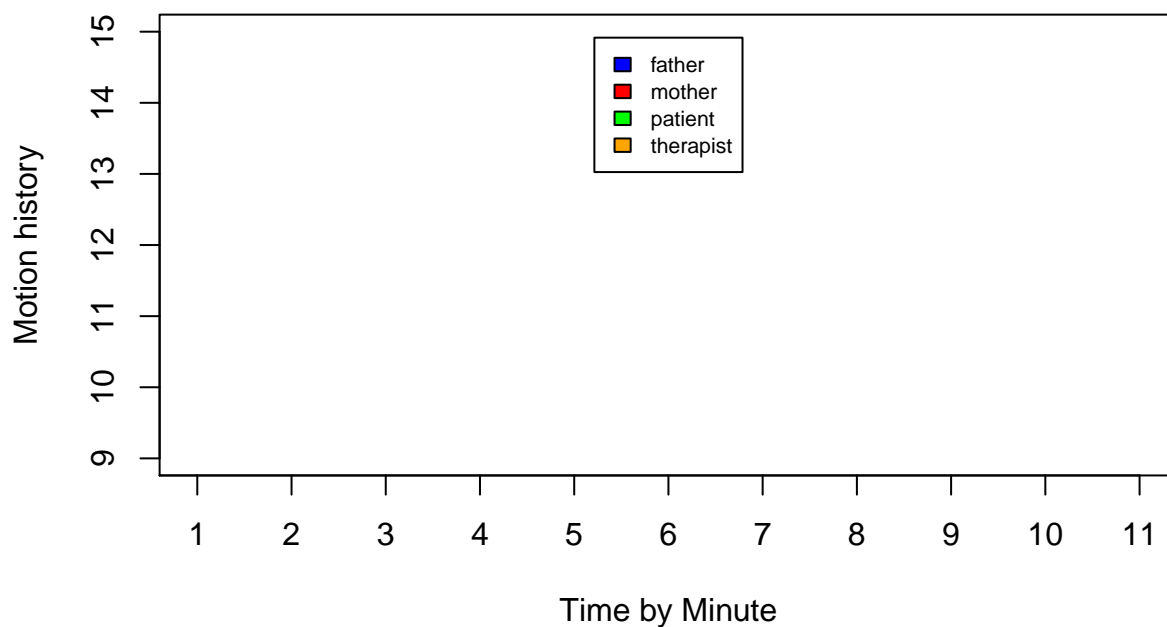
**Mean motion history (non overlapping minute intervals)  
on F1044O2 video**



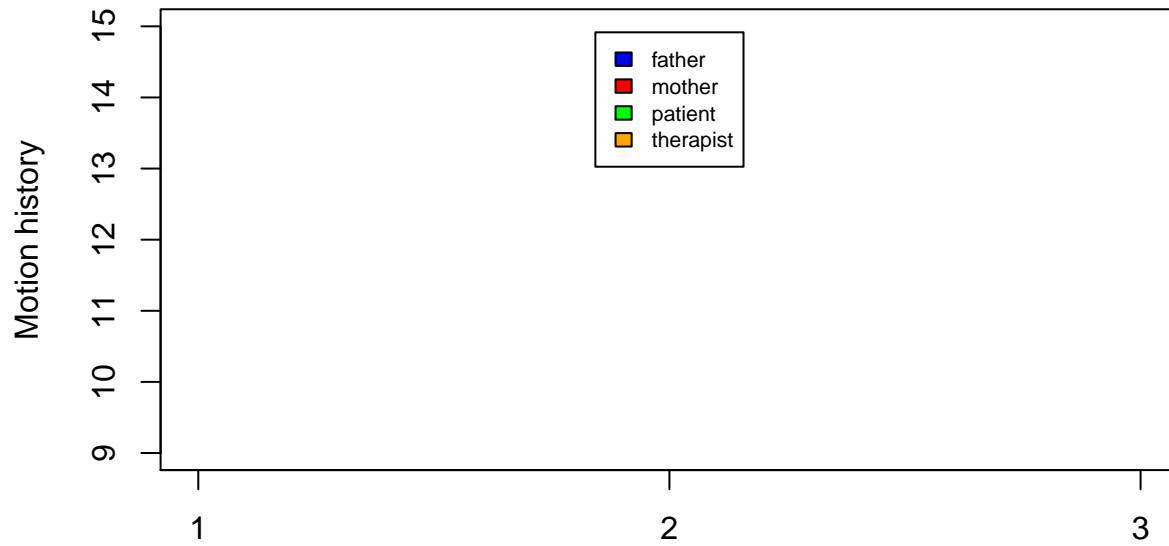
**Mean motion history (non overlapping minute intervals)  
on F1044P video**



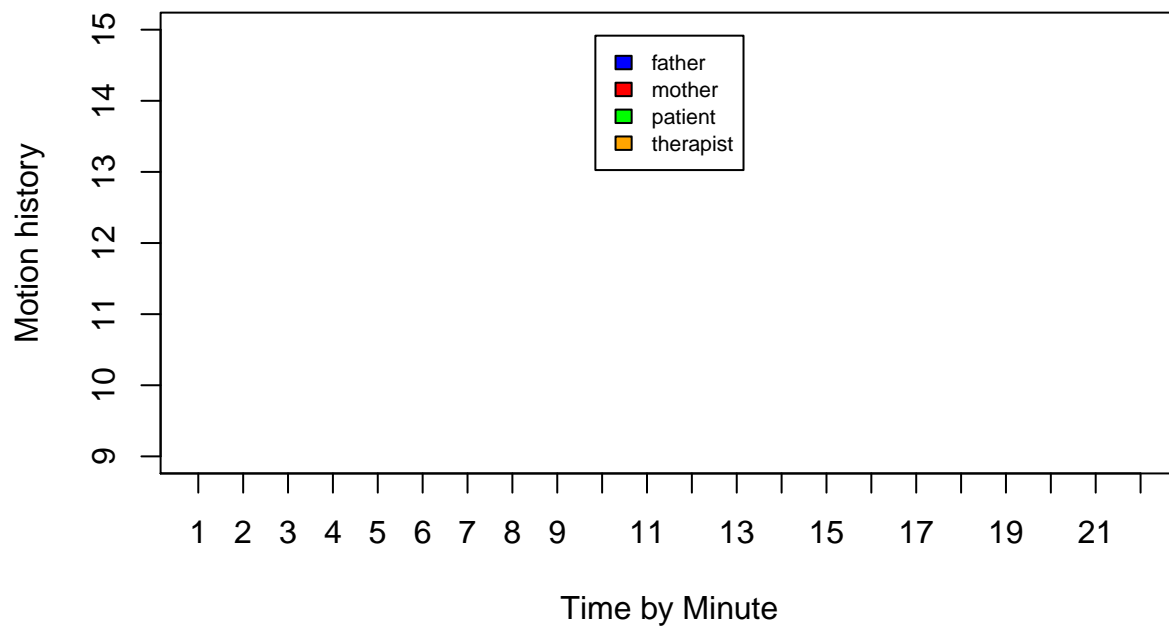
**Mean motion history (non overlapping minute intervals)  
on F1044Q1 video**



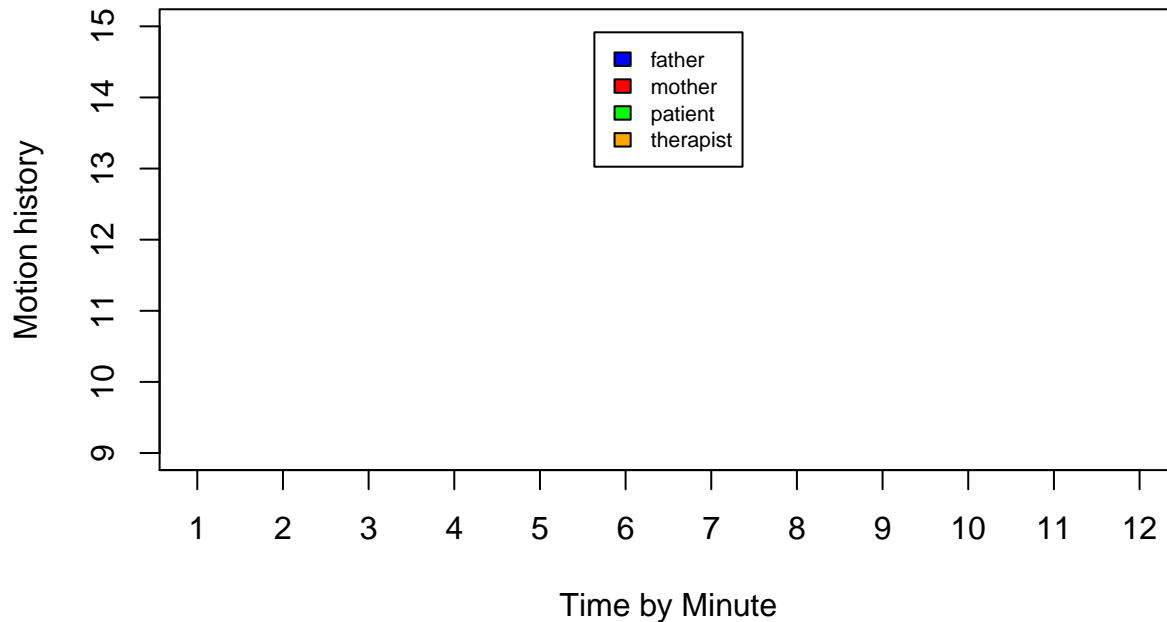
**Mean motion history (non overlapping minute intervals)  
on F1044Q2 video**



Time by Minute  
**Mean motion history (non overlapping minute intervals)  
on F1044R1 video**



## Mean motion history (non overlapping minute intervals) on F1044R2 video



## Motion history by minute for the F1044C video

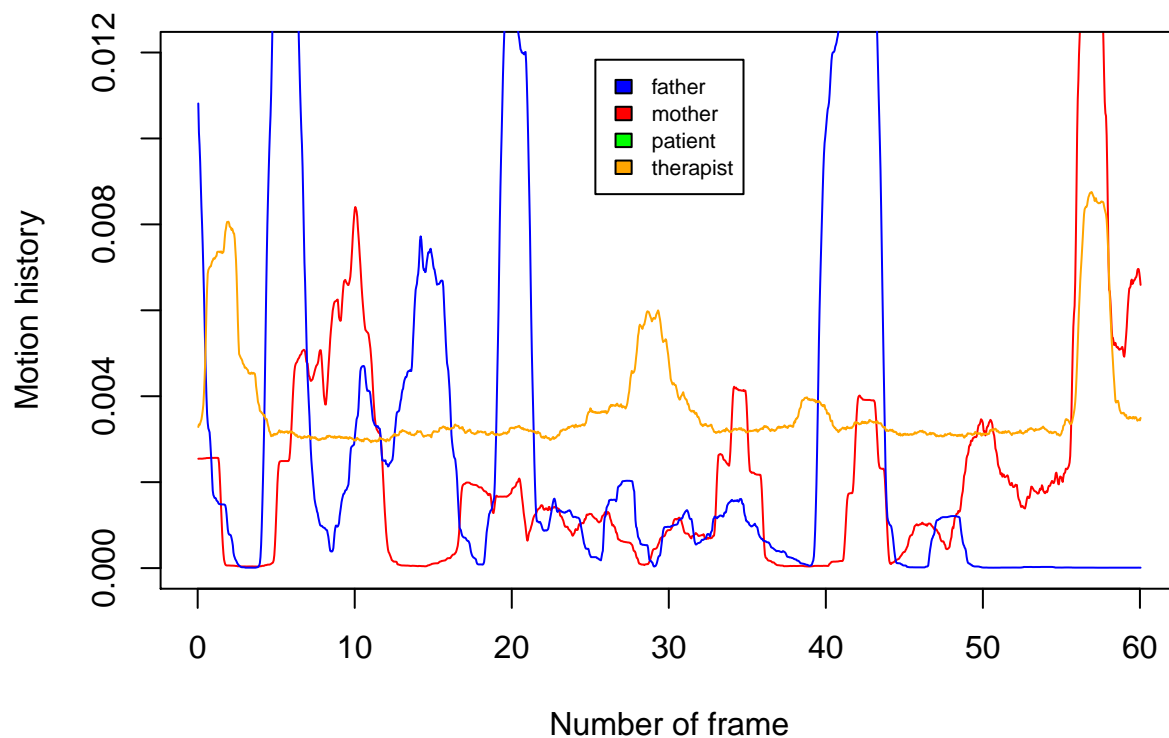
```
slidedFather <- SlidingInterval("father", 1, 50, data)
slidedMother <- SlidingInterval("mother", 1, 50, data)
slidedTherapist <- SlidingInterval("therapist", 1, 50, data)
slidedPatient <- SlidingInterval("patient", 1, 50, data)
```

```
framesByMinute <- 60*25
F1044C_Minutes <- ceiling(length(slidedFather)/framesByMinute)
for (i in 1:(F1044C_Minutes-1)){
  par(mar=c(4,4,4,2))
  borneInf <- i+framesByMinute*(i-1)
  borneSup <- i+i*framesByMinute
  slidedFatherMinute<-slidedFather[borneInf:borneSup]
  slidedMotherMinute<-slidedMother[borneInf:borneSup]
  slidedTherapistMinute<-slidedTherapist[borneInf:borneSup]
  slidedPatientMinute<-slidedPatient[borneInf:borneSup]
  slidedVideoDF <- data.frame(slidedFatherMinute, slidedMotherMinute, slidedTherapistMinute, slidedPatientMinute)
  str(slidedVideoDF)
  plot(slidedVideoDF$minute, slidedVideoDF$slidedMotherMinute, type="l", col="red",
  main=paste("Motion history with Sliding interval function during",
  minute, i, " in F1044C video", sep=""),
  ylab="Motion history", xlab="Number of frame", ylim=c(0, 12E-03))
  # xaxp=c(0, length(slidedFatherMinute), length(slidedFatherMinute)))
  lines(slidedVideoDF$minute, slidedVideoDF$slidedFatherMinute, col="blue")
  lines(slidedVideoDF$minute, slidedVideoDF$slidedTherapistMinute, col="orange")
  lines(slidedVideoDF$minute, slidedVideoDF$slidedPatientMinute, col="green")
}
```

```
legend("top", inset=.05, ParticipantsList,
      fill=colOrderList, cex=0.7)}
```

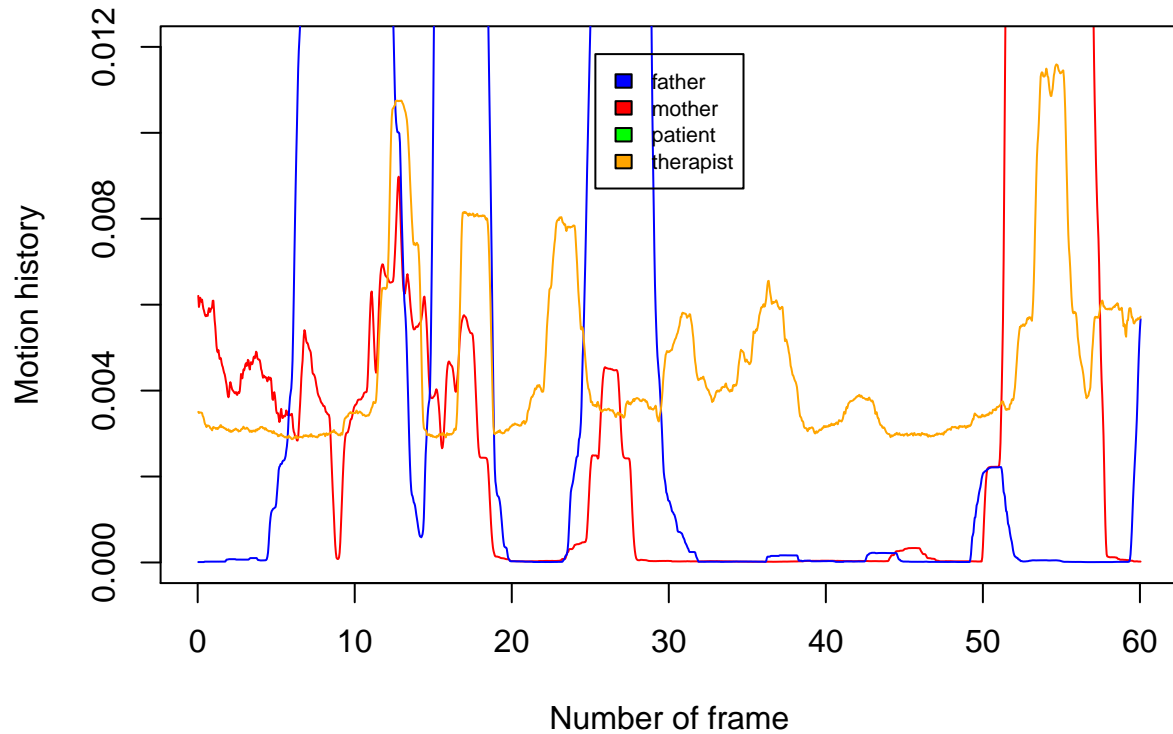
```
## 'data.frame': 1501 obs. of 6 variables:
## $ slidedFatherMinute : num 0.01081 0.01014 0.00985 0.00945 0.00891 ...
## $ slidedMotherMinute : num 0.00255 0.00255 0.00255 0.00255 0.00255 ...
## $ slidedTherapistMinute: num 0.00328 0.00334 0.00333 0.00337 0.00342 ...
## $ slidedPatientMinute : num NaN NaN NaN NaN NaN NaN NaN NaN NaN ...
## $ frames : int 1 2 3 4 5 6 7 8 9 10 ...
## $ minute : num 0.04 0.08 0.12 0.16 0.2 0.24 0.28 0.32 0.36 0.4 ...
```

### Motion history with Sliding interval function during minute 1 in F1044C video



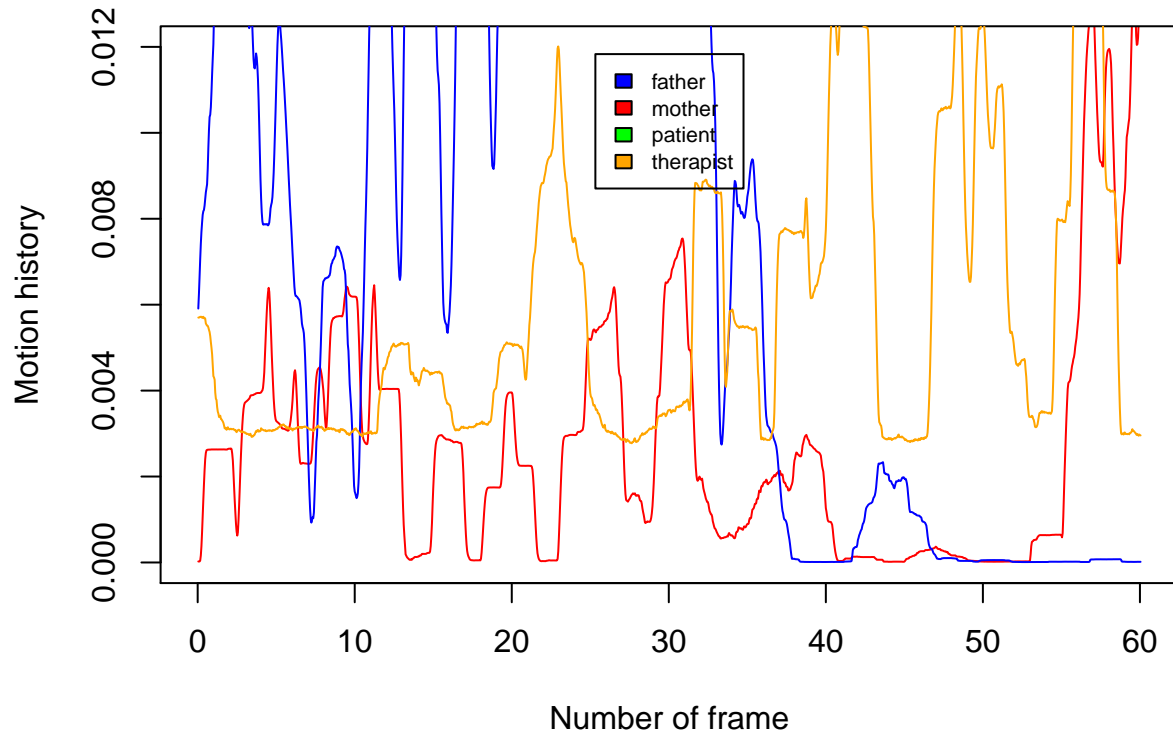
```
## 'data.frame': 1501 obs. of 6 variables:
## $ slidedFatherMinute : num 9.75e-06 9.75e-06 9.75e-06 9.75e-06 9.49e-06 ...
## $ slidedMotherMinute : num 0.0062 0.00595 0.00603 0.00616 0.00611 ...
## $ slidedTherapistMinute: num 0.00351 0.00348 0.00349 0.00349 0.00349 ...
## $ slidedPatientMinute : num NaN NaN NaN NaN NaN NaN NaN NaN NaN ...
## $ frames : int 1 2 3 4 5 6 7 8 9 10 ...
## $ minute : num 0.04 0.08 0.12 0.16 0.2 0.24 0.28 0.32 0.36 0.4 ...
```

## Motion history with Sliding interval function during minute 2 in F1044C video



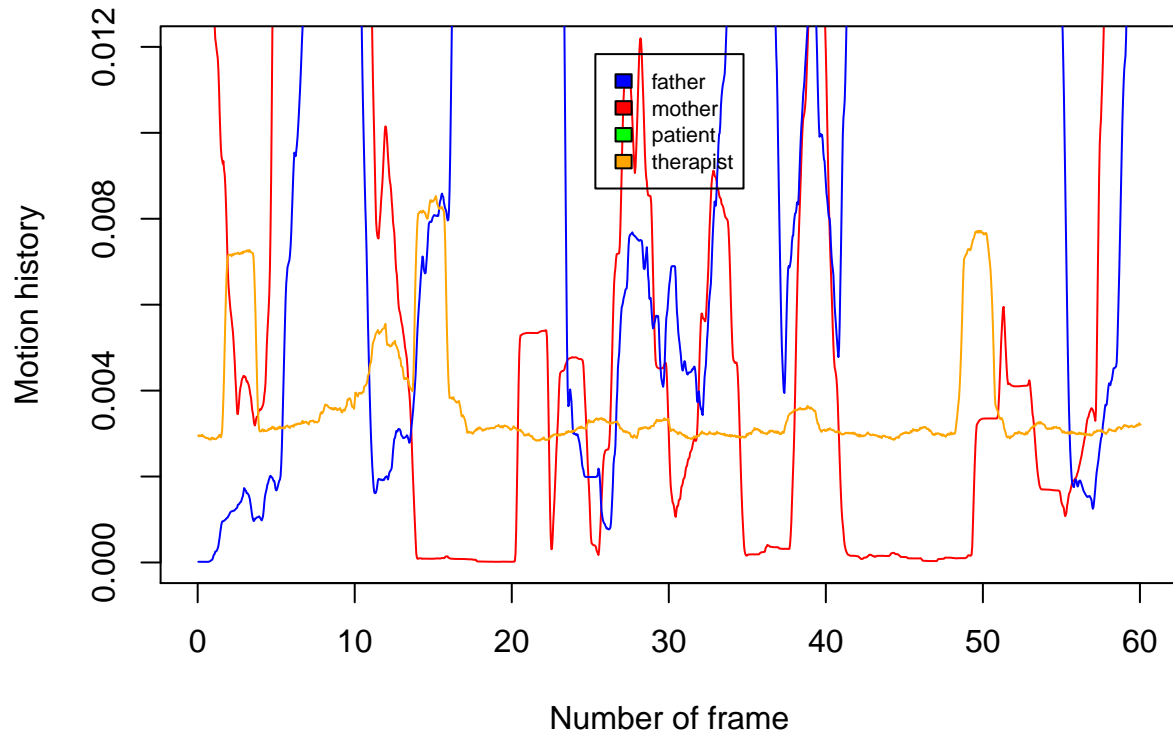
```
## 'data.frame': 1501 obs. of 6 variables:
## $ slidedFatherMinute : num 0.00591 0.00625 0.00663 0.00705 0.00742 ...
## $ slidedMotherMinute : num 2.35e-05 2.35e-05 2.25e-05 4.29e-05 1.61e-04 ...
## $ slidedTherapistMinute: num 0.0057 0.0057 0.00571 0.00572 0.00572 ...
## $ slidedPatientMinute : num NaN NaN NaN NaN NaN NaN NaN NaN NaN ...
## $ frames : int 1 2 3 4 5 6 7 8 9 10 ...
## $ minute : num 0.04 0.08 0.12 0.16 0.2 0.24 0.28 0.32 0.36 0.4 ...
```

## Motion history with Sliding interval function during minute 3 in F1044C video



```
## 'data.frame':   1501 obs. of  6 variables:
## $ slidedFatherMinute   : num  1.56e-05 1.56e-05 1.56e-05 1.59e-05 1.59e-05 ...
## $ slidedMotherMinute   : num  0.0135 0.0136 0.0137 0.0137 0.0138 ...
## $ slidedTherapistMinute: num  0.00295 0.00296 0.00294 0.00295 0.00296 ...
## $ slidedPatientMinute  : num  NaN NaN NaN NaN NaN NaN NaN NaN NaN ...
## $ frames               : int   1  2  3  4  5  6  7  8  9 10 ...
## $ minute               : num   0.04 0.08 0.12 0.16 0.2 0.24 0.28 0.32 0.36 0.4 ...
```

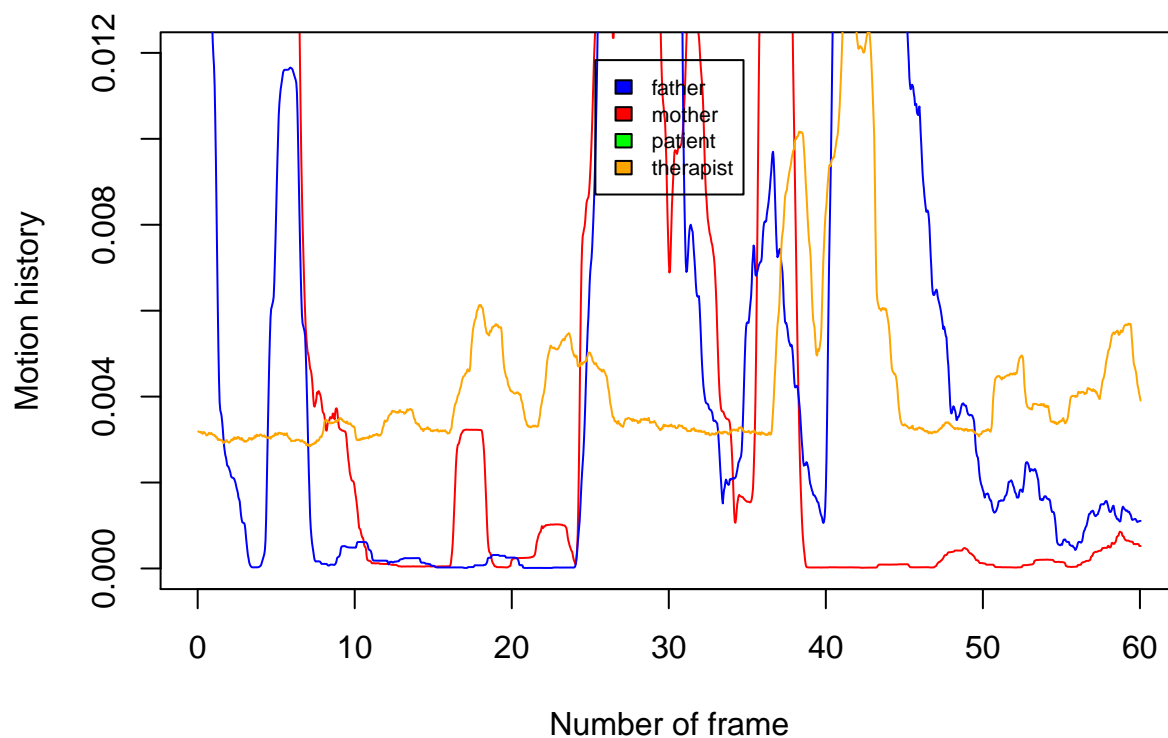
## Motion history with Sliding interval function during minute 4 in F1044C video



```
## 'data.frame':  1501 obs. of  6 variables:
## $ slidedFatherMinute   : num  0.0189 0.0188 0.0188 0.0187 0.0187 ...
## $ slidedMotherMinute   : num  0.0181 0.0185 0.0188 0.0191 0.0193 ...
## $ slidedTherapistMinute: num  0.00319 0.00318 0.00318 0.00318 0.00316 ...
## $ slidedPatientMinute  : num  NaN NaN NaN NaN NaN NaN NaN NaN NaN ...
## $ frames                : int   1  2  3  4  5  6  7  8  9 10 ...
## $ minute                : num  0.04 0.08 0.12 0.16 0.2 0.24 0.28 0.32 0.36 0.4 ...
```

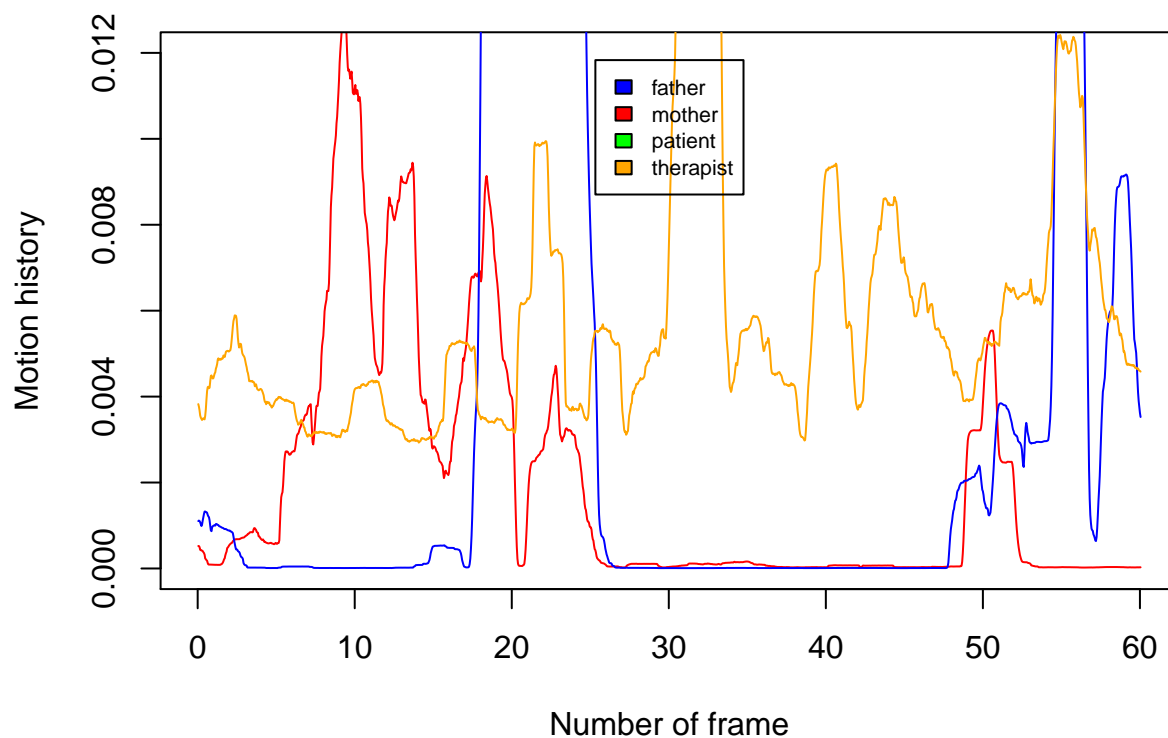


## Motion history with Sliding interval function during minute 5 in F1044C video



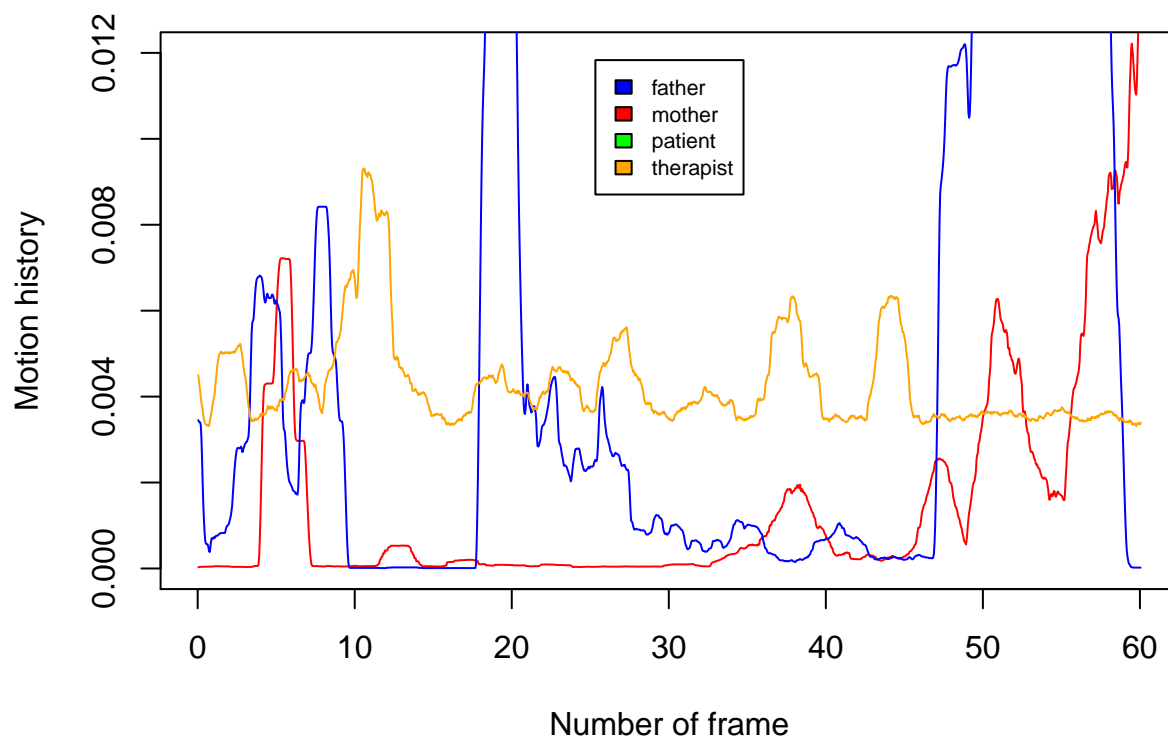
```
## 'data.frame':   1501 obs. of  6 variables:
## $ slidedFatherMinute   : num  0.0011 0.00112 0.0011 0.00108 0.00103 ...
## $ slidedMotherMinute   : num  0.000523 0.000522 0.000498 0.000433 0.000433 ...
## $ slidedTherapistMinute: num  0.00383 0.00377 0.00368 0.00356 0.00351 ...
## $ slidedPatientMinute  : num  NaN NaN NaN NaN NaN NaN NaN NaN NaN ...
## $ frames                : int   1  2  3  4  5  6  7  8  9 10 ...
## $ minute                : num  0.04 0.08 0.12 0.16 0.2 0.24 0.28 0.32 0.36 0.4 ...
```

## Motion history with Sliding interval function during minute 6 in F1044C video



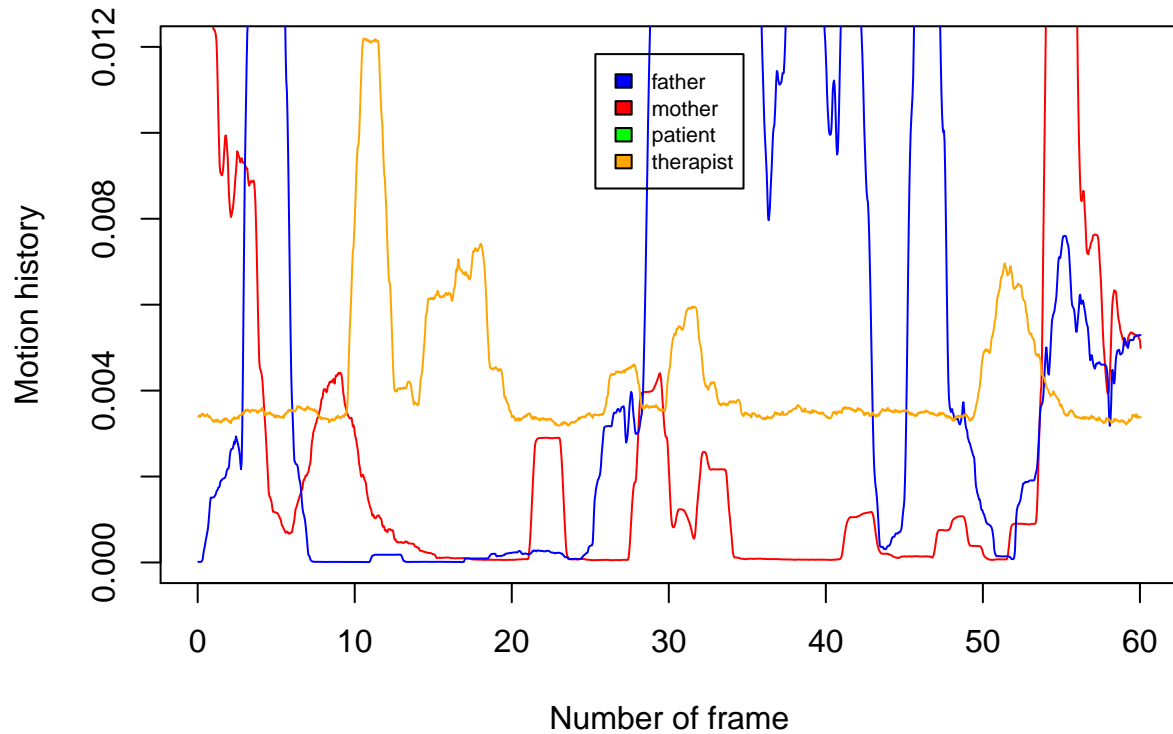
```
## 'data.frame': 1501 obs. of 6 variables:
## $ slidedFatherMinute : num 0.00345 0.00341 0.00341 0.0034 0.00333 ...
## $ slidedMotherMinute : num 2.90e-05 2.95e-05 3.45e-05 3.71e-05 3.87e-05 ...
## $ slidedTherapistMinute: num 0.0045 0.00438 0.00424 0.00412 0.004 ...
## $ slidedPatientMinute : num NaN NaN NaN NaN NaN NaN NaN NaN NaN ...
## $ frames : int 1 2 3 4 5 6 7 8 9 10 ...
## $ minute : num 0.04 0.08 0.12 0.16 0.2 0.24 0.28 0.32 0.36 0.4 ...
```

## Motion history with Sliding interval function during minute 7 in F1044C video



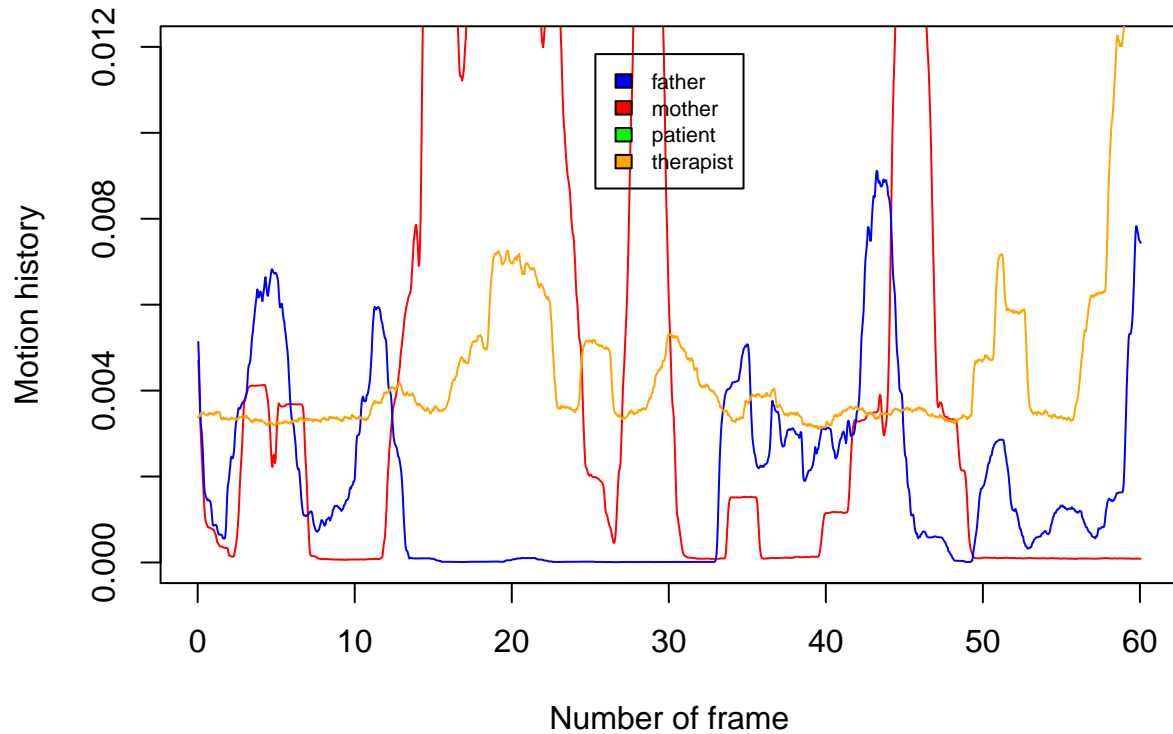
```
## 'data.frame':   1501 obs. of  6 variables:
## $ slidedFatherMinute   : num  1.70e-05 1.42e-05 1.35e-05 1.33e-05 1.33e-05 ...
## $ slidedMotherMinute   : num  0.0143 0.0144 0.0146 0.0147 0.0146 ...
## $ slidedTherapistMinute: num  0.0034 0.00338 0.00342 0.00342 0.00339 ...
## $ slidedPatientMinute  : num  NaN NaN NaN NaN NaN NaN NaN NaN NaN ...
## $ frames                : int   1  2  3  4  5  6  7  8  9 10 ...
## $ minute                : num   0.04 0.08 0.12 0.16 0.2 0.24 0.28 0.32 0.36 0.4 ...
```

# **Motion history with Sliding interval function during minute 8 in F1044C video**



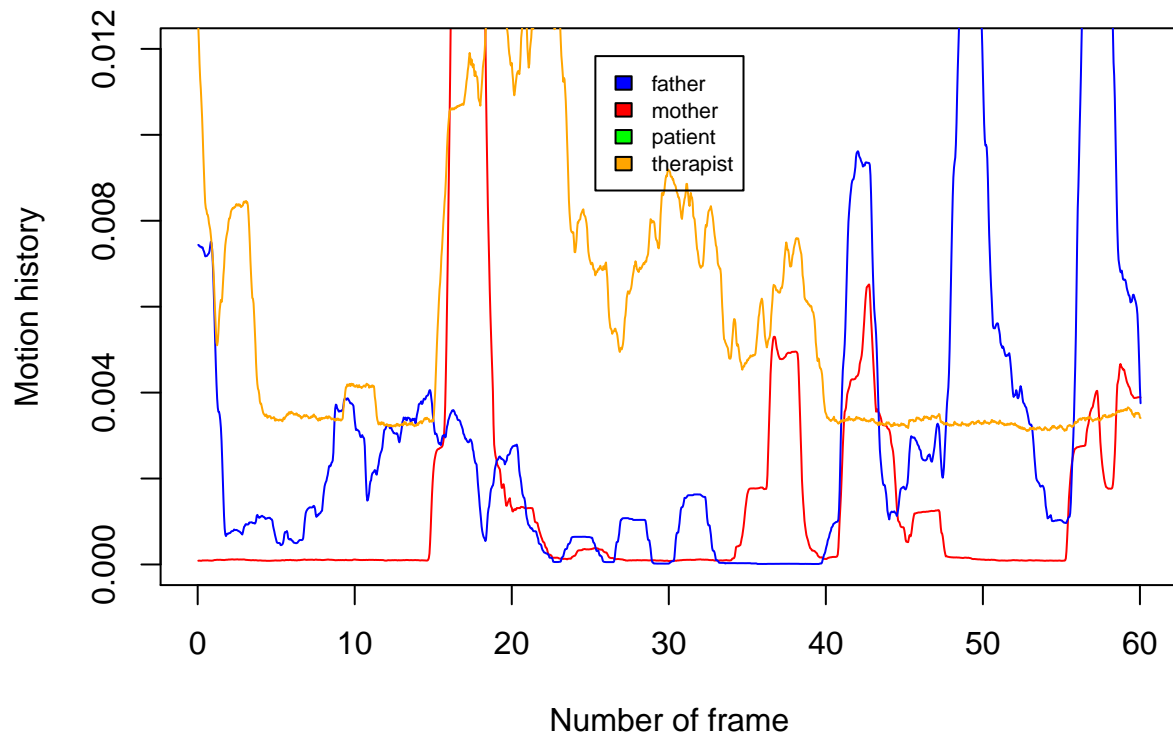
```
## 'data.frame':  1501 obs. of  6 variables:
## $ slidedFatherMinute   : num  0.00513 0.00473 0.00412 0.00355 0.00319 ...
## $ slidedMotherMinute   : num  0.0047 0.00437 0.0039 0.00338 0.00296 ...
## $ slidedTherapistMinute: num  0.00338 0.00341 0.00342 0.00341 0.00348 ...
## $ slidedPatientMinute  : num  NaN NaN NaN NaN NaN NaN NaN NaN NaN ...
## $ frames               : int   1  2  3  4  5  6  7  8  9 10 ...
## $ minute               : num   0.04 0.08 0.12 0.16 0.2 0.24 0.28 0.32 0.36 0.4 ...
```

## Motion history with Sliding interval function during minute 9 in F1044C video



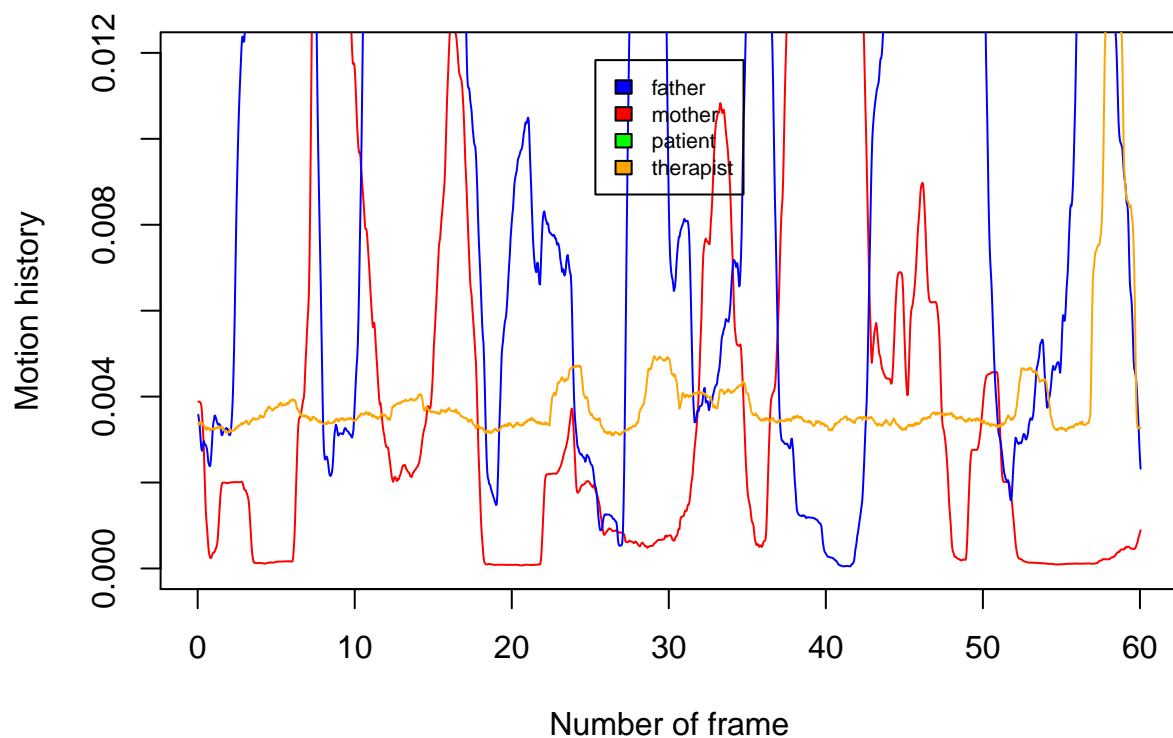
```
## 'data.frame':   1501 obs. of  6 variables:
## $ slidedFatherMinute   : num  0.00744 0.00743 0.0074 0.0074 0.0074 ...
## $ slidedMotherMinute   : num  8.73e-05 8.82e-05 8.91e-05 8.78e-05 8.55e-05 ...
## $ slidedTherapistMinute: num  0.0125 0.0121 0.0117 0.0113 0.0111 ...
## $ slidedPatientMinute  : num  NaN NaN NaN NaN NaN NaN NaN NaN NaN ...
## $ frames               : int   1  2  3  4  5  6  7  8  9 10 ...
## $ minute               : num   0.04 0.08 0.12 0.16 0.2 0.24 0.28 0.32 0.36 0.4 ...
```

## Motion history with Sliding interval function during minute 10 in F1044C video



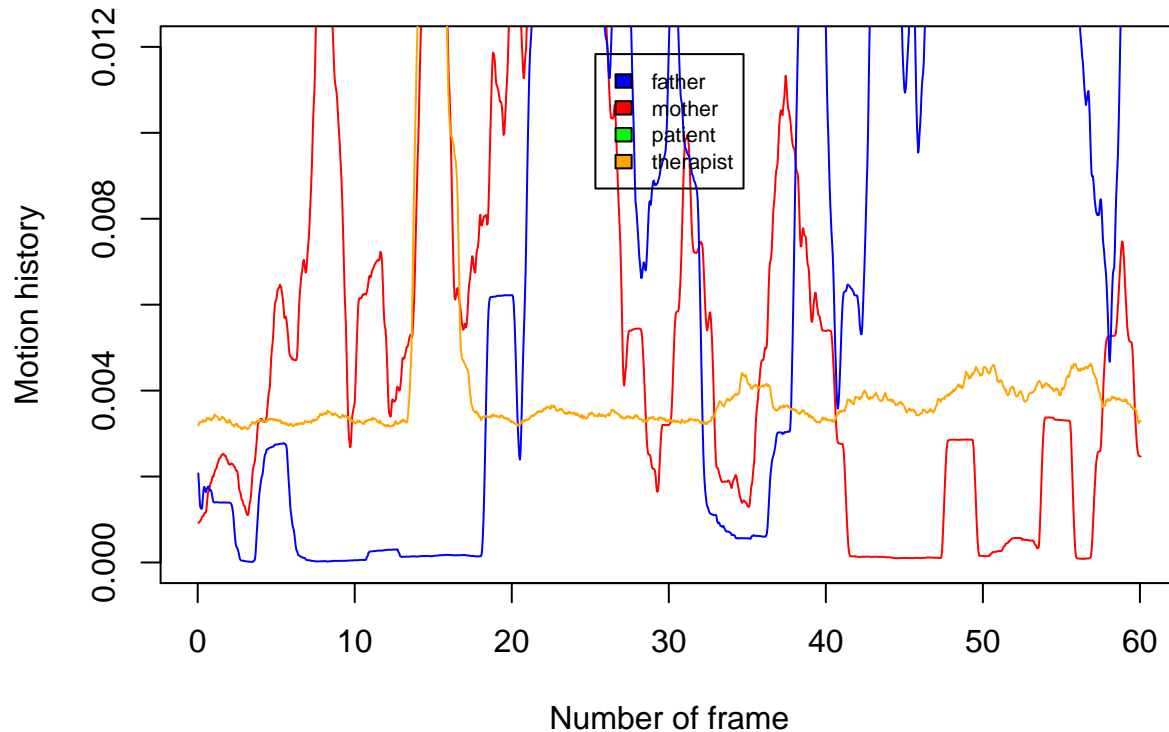
```
## 'data.frame':   1501 obs. of  6 variables:
## $ slidedFatherMinute   : num  0.00358 0.00346 0.00336 0.00316 0.00293 ...
## $ slidedMotherMinute   : num  0.00388 0.00388 0.00388 0.00387 0.00385 ...
## $ slidedTherapistMinute: num  0.00338 0.00335 0.00339 0.00341 0.00339 ...
## $ slidedPatientMinute  : num  NaN NaN NaN NaN NaN NaN NaN NaN NaN ...
## $ frames                : int   1  2  3  4  5  6  7  8  9 10 ...
## $ minute                : num   0.04 0.08 0.12 0.16 0.2  0.24 0.28 0.32 0.36 0.4  ...
```

## Motion history with Sliding interval function during minute 11 in F1044C video



```
## 'data.frame':   1501 obs. of  6 variables:
## $ slidedFatherMinute   : num  0.00207 0.00183 0.00151 0.0013 0.00127 ...
## $ slidedMotherMinute   : num  0.000923 0.000929 0.000934 0.000964 0.000987 ...
## $ slidedTherapistMinute: num  0.00319 0.00322 0.00326 0.00326 0.00326 ...
## $ slidedPatientMinute  : num  NaN NaN NaN NaN NaN NaN NaN NaN NaN ...
## $ frames                : int   1  2  3  4  5  6  7  8  9 10 ...
## $ minute                : num   0.04 0.08 0.12 0.16 0.2 0.24 0.28 0.32 0.36 0.4 ...
```

## Motion history with Sliding interval function during minute 12 in F1044C video



## Export no log filtered data in text files

```
## REMINDER:
#SlidingInterval <- function(subject, indexOfvideos=1:NumberOfvideos, interval, data) with :
# subject : subject studied (patient, mother, father or therapist)
# indexOfvideos : list of videos studied (element eg. 3 or list eg 1:3 or c(1,2,4))
# interval : number of frames in the studied interval
# data : data frame where there is data

#index de la vid?eo de 1ere a la length de indexvideo

videoIndex <- 1
# videoName est le nom de la video actuelle
for (videoName in indexList){
# Compute slinding interval for each participant
print(paste("Computing slidedFather", videoName))
slidedFather <- SlidingInterval("father", videoIndex, 5, data)
logSlidedFather <- SlidingInterval("logFather", videoIndex, 5, data)

print(paste("Computing slidedMother", videoName))
slidedMother <- SlidingInterval("mother", videoIndex, 5, data)
logSlidedMother <- SlidingInterval("logMother", videoIndex, 5, data)

print(paste("Computing slidedPatient", videoName))
slidedPatient <- SlidingInterval("patient", videoIndex, 5, data)
```



```

logSlidedPatient <- SlidingInterval("logPatient", videoIndex, 5, data)

print(paste("Computing slidedTherapist", videoName))
slidedTherapist <- SlidingInterval("therapist", videoIndex, 5, data)
logSlidedTherapist <- SlidingInterval("logTherapist", videoIndex, 5, data)

# create a data frame to store temporarily this data with NA
slidedVideo <- data.frame(
  slidedFather, slidedMother, slidedPatient, slidedTherapist)

dataFrame <- FALSE
dfSliding <- data.frame()
for (participant in 1:4){
  # If the column is not empty, takes its length and begin a data frame with it
  if (dataFrame == FALSE & (length(slidedVideo[participant][!is.na(slidedVideo[participant])]) > 0)) {
    dfSliding <- data.frame(
      "video"=rep(indexList[videoIndex], length(slidedVideo[participant])),
      frame_index = (1:dim(slidedVideo[1])[1]),
      slidedVideo[participant])
    dataFrame <- TRUE}
  else if (dataFrame == FALSE){}
  else{
    dfSliding <- cbind(dfSliding, slidedVideo[participant])
  }
  write.csv(dfSliding, paste("/Users/Ofix/Documents/Fac/internat/Recherche/projets/synchro/synchroD",
    videoIndex <-(videoIndex+1)
}

```

```

## [1] "Computing slidedFather F1044C1"
## [1] "Computing slidedMother F1044C1"
## [1] "Computing slidedPatient F1044C1"
## [1] "Computing slidedTherapist F1044C1"
## [1] "Computing slidedFather F1044C2"
## [1] "Computing slidedMother F1044C2"
## [1] "Computing slidedPatient F1044C2"
## [1] "Computing slidedTherapist F1044C2"
## [1] "Computing slidedFather F1044D1"
## [1] "Computing slidedMother F1044D1"
## [1] "Computing slidedPatient F1044D1"
## [1] "Computing slidedTherapist F1044D1"
## [1] "Computing slidedFather F1044D2"
## [1] "Computing slidedMother F1044D2"
## [1] "Computing slidedPatient F1044D2"
## [1] "Computing slidedTherapist F1044D2"
## [1] "Computing slidedFather F1044E1"
## [1] "Computing slidedMother F1044E1"
## [1] "Computing slidedPatient F1044E1"
## [1] "Computing slidedTherapist F1044E1"
## [1] "Computing slidedFather F1044E2"
## [1] "Computing slidedMother F1044E2"
## [1] "Computing slidedPatient F1044E2"
## [1] "Computing slidedTherapist F1044E2"
## [1] "Computing slidedFather F1044F1"

```

```

## [1] "Computing slidedMother F1044F1"
## [1] "Computing slidedPatient F1044F1"
## [1] "Computing slidedTherapist F1044F1"
## [1] "Computing slidedFather F1044F2"
## [1] "Computing slidedMother F1044F2"
## [1] "Computing slidedPatient F1044F2"
## [1] "Computing slidedTherapist F1044F2"
## [1] "Computing slidedFather F1044G"
## [1] "Computing slidedMother F1044G"
## [1] "Computing slidedPatient F1044G"
## [1] "Computing slidedTherapist F1044G"
## [1] "Computing slidedFather F1044H1"
## [1] "Computing slidedMother F1044H1"
## [1] "Computing slidedPatient F1044H1"
## [1] "Computing slidedTherapist F1044H1"
## [1] "Computing slidedFather F1044H2"
## [1] "Computing slidedMother F1044H2"
## [1] "Computing slidedPatient F1044H2"
## [1] "Computing slidedTherapist F1044H2"
## [1] "Computing slidedFather F1044I1"
## [1] "Computing slidedMother F1044I1"
## [1] "Computing slidedPatient F1044I1"
## [1] "Computing slidedTherapist F1044I1"
## [1] "Computing slidedFather F1044I2"
## [1] "Computing slidedMother F1044I2"
## [1] "Computing slidedPatient F1044I2"
## [1] "Computing slidedTherapist F1044I2"
## [1] "Computing slidedFather F1044L1"
## [1] "Computing slidedMother F1044L1"
## [1] "Computing slidedPatient F1044L1"
## [1] "Computing slidedTherapist F1044L1"
## [1] "Computing slidedFather F1044L2"
## [1] "Computing slidedMother F1044L2"
## [1] "Computing slidedPatient F1044L2"
## [1] "Computing slidedTherapist F1044L2"
## [1] "Computing slidedFather F1044M1"
## [1] "Computing slidedMother F1044M1"
## [1] "Computing slidedPatient F1044M1"
## [1] "Computing slidedTherapist F1044M1"
## [1] "Computing slidedFather F1044M2"
## [1] "Computing slidedMother F1044M2"
## [1] "Computing slidedPatient F1044M2"
## [1] "Computing slidedTherapist F1044M2"
## [1] "Computing slidedFather F1044N"
## [1] "Computing slidedMother F1044N"
## [1] "Computing slidedPatient F1044N"
## [1] "Computing slidedTherapist F1044N"
## [1] "Computing slidedFather F1044O1"
## [1] "Computing slidedMother F1044O1"
## [1] "Computing slidedPatient F1044O1"
## [1] "Computing slidedTherapist F1044O1"
## [1] "Computing slidedFather F1044O2"
## [1] "Computing slidedMother F1044O2"
## [1] "Computing slidedPatient F1044O2"

```

```
## [1] "Computing slidedTherapist F104402"
## [1] "Computing slidedFather F1044P"
## [1] "Computing slidedMother F1044P"
## [1] "Computing slidedPatient F1044P"
## [1] "Computing slidedTherapist F1044P"
## [1] "Computing slidedFather F1044Q1"
## [1] "Computing slidedMother F1044Q1"
## [1] "Computing slidedPatient F1044Q1"
## [1] "Computing slidedTherapist F1044Q1"
## [1] "Computing slidedFather F1044Q2"
## [1] "Computing slidedMother F1044Q2"
## [1] "Computing slidedPatient F1044Q2"
## [1] "Computing slidedTherapist F1044Q2"
## [1] "Computing slidedFather F1044R1"
## [1] "Computing slidedMother F1044R1"
## [1] "Computing slidedPatient F1044R1"
## [1] "Computing slidedTherapist F1044R1"
## [1] "Computing slidedFather F1044R2"
## [1] "Computing slidedMother F1044R2"
## [1] "Computing slidedPatient F1044R2"
## [1] "Computing slidedTherapist F1044R2"
```

## Export log filtere data in text files

```
videoIndex <- 1
# videoName est le nom de la video actuelle
for (videoName in indexList){
  # Compute slinding interval for each participant
  print(paste("Computing slidedFather", videoName))
  slidedFather <- SlidingInterval("logFather", videoIndex, 5, data)

  print(paste("Computing slidedMother", videoName))
  slidedMother <- SlidingInterval("logMother", videoIndex, 5, data)

  print(paste("Computing slidedPatient", videoName))
  slidedPatient <- SlidingInterval("logPatient", videoIndex, 5, data)

  print(paste("Computing slidedTherapist", videoName))
  slidedTherapist <- SlidingInterval("logTherapist", videoIndex, 5, data)

  # create a data frame to store temporarily this data with NA
  slidedVideo <- data.frame(
    slidedFather, slidedMother, slidedPatient, slidedTherapist)

  dataFrame <- FALSE
  dfSliding <- data.frame()
  for (participant in 1:4){
    # If the colum is not empty, takes its length and begin a data frame with it
    if (dataFrame == FALSE & (length(slidedVideo[participant][!is.na(slidedVideo[participant])])) > 0){
      dfSliding <- data.frame(
        "video"=rep(indexList[videoIndex], length(slidedVideo[participant])),
```

```

        frame_index = (1:dim(slidedVideo[1])[1]),
        slidedVideo[participant])
        dataFrame <- TRUE}
    else if (dataFrame == FALSE){}
    else{
        dfSliding <- cbind(dfSliding, slidedVideo[participant])}
}

write.csv(dfSliding, paste("/Users/Ofix/Documents/Fac/internat/Recherche/projets/synchro/synchroD",
videoIndex <-(videoIndex+1)
}

```

```

## [1] "Computing slidedFather F1044C1"
## [1] "Computing slidedMother F1044C1"
## [1] "Computing slidedPatient F1044C1"
## [1] "Computing slidedTherapist F1044C1"
## [1] "Computing slidedFather F1044C2"
## [1] "Computing slidedMother F1044C2"
## [1] "Computing slidedPatient F1044C2"
## [1] "Computing slidedTherapist F1044C2"
## [1] "Computing slidedFather F1044D1"
## [1] "Computing slidedMother F1044D1"
## [1] "Computing slidedPatient F1044D1"
## [1] "Computing slidedTherapist F1044D1"
## [1] "Computing slidedFather F1044D2"
## [1] "Computing slidedMother F1044D2"
## [1] "Computing slidedPatient F1044D2"
## [1] "Computing slidedTherapist F1044D2"
## [1] "Computing slidedFather F1044E1"
## [1] "Computing slidedMother F1044E1"
## [1] "Computing slidedPatient F1044E1"
## [1] "Computing slidedTherapist F1044E1"
## [1] "Computing slidedFather F1044E2"
## [1] "Computing slidedMother F1044E2"
## [1] "Computing slidedPatient F1044E2"
## [1] "Computing slidedTherapist F1044E2"
## [1] "Computing slidedFather F1044F1"
## [1] "Computing slidedMother F1044F1"
## [1] "Computing slidedPatient F1044F1"
## [1] "Computing slidedTherapist F1044F1"
## [1] "Computing slidedFather F1044F2"
## [1] "Computing slidedMother F1044F2"
## [1] "Computing slidedPatient F1044F2"
## [1] "Computing slidedTherapist F1044F2"
## [1] "Computing slidedFather F1044G"
## [1] "Computing slidedMother F1044G"
## [1] "Computing slidedPatient F1044G"
## [1] "Computing slidedTherapist F1044G"
## [1] "Computing slidedFather F1044H1"
## [1] "Computing slidedMother F1044H1"
## [1] "Computing slidedPatient F1044H1"
## [1] "Computing slidedTherapist F1044H1"
## [1] "Computing slidedFather F1044H2"

```

```

## [1] "Computing slidedMother F1044H2"
## [1] "Computing slidedPatient F1044H2"
## [1] "Computing slidedTherapist F1044H2"
## [1] "Computing slidedFather F1044I1"
## [1] "Computing slidedMother F1044I1"
## [1] "Computing slidedPatient F1044I1"
## [1] "Computing slidedTherapist F1044I1"
## [1] "Computing slidedFather F1044I2"
## [1] "Computing slidedMother F1044I2"
## [1] "Computing slidedPatient F1044I2"
## [1] "Computing slidedTherapist F1044I2"
## [1] "Computing slidedFather F1044L1"
## [1] "Computing slidedMother F1044L1"
## [1] "Computing slidedPatient F1044L1"
## [1] "Computing slidedTherapist F1044L1"
## [1] "Computing slidedFather F1044L2"
## [1] "Computing slidedMother F1044L2"
## [1] "Computing slidedPatient F1044L2"
## [1] "Computing slidedTherapist F1044L2"
## [1] "Computing slidedFather F1044M1"
## [1] "Computing slidedMother F1044M1"
## [1] "Computing slidedPatient F1044M1"
## [1] "Computing slidedTherapist F1044M1"
## [1] "Computing slidedFather F1044M2"
## [1] "Computing slidedMother F1044M2"
## [1] "Computing slidedPatient F1044M2"
## [1] "Computing slidedTherapist F1044M2"
## [1] "Computing slidedFather F1044N"
## [1] "Computing slidedMother F1044N"
## [1] "Computing slidedPatient F1044N"
## [1] "Computing slidedTherapist F1044N"
## [1] "Computing slidedFather F1044O1"
## [1] "Computing slidedMother F1044O1"
## [1] "Computing slidedPatient F1044O1"
## [1] "Computing slidedTherapist F1044O1"
## [1] "Computing slidedFather F1044O2"
## [1] "Computing slidedMother F1044O2"
## [1] "Computing slidedPatient F1044O2"
## [1] "Computing slidedTherapist F1044O2"
## [1] "Computing slidedFather F1044P"
## [1] "Computing slidedMother F1044P"
## [1] "Computing slidedPatient F1044P"
## [1] "Computing slidedTherapist F1044P"
## [1] "Computing slidedFather F1044Q1"
## [1] "Computing slidedMother F1044Q1"
## [1] "Computing slidedPatient F1044Q1"
## [1] "Computing slidedTherapist F1044Q1"
## [1] "Computing slidedFather F1044Q2"
## [1] "Computing slidedMother F1044Q2"
## [1] "Computing slidedPatient F1044Q2"
## [1] "Computing slidedTherapist F1044Q2"
## [1] "Computing slidedFather F1044R1"
## [1] "Computing slidedMother F1044R1"
## [1] "Computing slidedPatient F1044R1"

```

```
## [1] "Computing slidedTherapist F1044R1"
## [1] "Computing slidedFather F1044R2"
## [1] "Computing slidedMother F1044R2"
## [1] "Computing slidedPatient F1044R2"
## [1] "Computing slidedTherapist F1044R2"
```

## SyncPy utilisation for creating synchrony dataframe

After extracting filtered motion motion history with mean on sliding interval (overlapping interval) of 5 frames

And after putting 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 (F1044C, F1044D1, etc) that we can import with R with

this following function

```
getwd()
```

## Description of SSI data frame

```
str(SSIdataFrame)
```

## Synchrony scores for each dyad, triad and for the whole group

In legend, mean for all the video.

```
for (i in unique(SSIdataFrame$video))
{par(mar=c(4,4,4,3), mfrow=c(1,1))
plot(SSIdataFrame[which(SSIdataFrame$video==i),]$Time_min,
SSIdataFrame[which(SSIdataFrame$video==i),]$SSI_fa_mo,
type="l", ylim=c(0, 0.3), col=rainbow(4)[1],
main=paste("Synchrony scores for each dyad and for \n the whole group in", i, "video"),
xlab = "Time (minute)", ylab="Synchrony score", lwd=2,
xaxp=c(0,length(SSIdataFrame$Time_min), length(SSIdataFrame$Time_min)))
abline(h=mean(SSIdataFrame$SSI_fa_mo, na.rm=TRUE), col=rainbow(11)[1], lwd=2, lty=2)

lines(SSIdataFrame[which(SSIdataFrame$video==i),]$SSI_fa_mo_pa, col=rainbow(11)[2], lwd=2)
abline(h= mean(SSIdataFrame$SSI_fa_mo_pa, na.rm=TRUE), col=rainbow(11)[2], lwd=2, lty=2)

# lines(SSIdataFrame[which(SSIdataFrame$video==i),]$SSI_fa_mo_pa_th, col=rainbow(11)[3], lwd=2)
# abline(h= mean(SSIdataFrame$SSI_fa_mo_pa_th, na.rm=TRUE), col=rainbow(11)[3], lwd=2, lty=2)
```

```

lines(SSIdataFrame[which(SSIdataFrame$video==i),]$SSI_fa_mo_th, col=rainbow(11)[4], lwd=2)
abline(h= mean(SSIdataFrame$SSI_fa_mo_th, na.rm=TRUE), col=rainbow(11)[4], lwd=2, lty=2)

lines(SSIdataFrame[which(SSIdataFrame$video==i),]$SSI_fa_pa, col=rainbow(11)[5], lwd=2)
abline(h= mean(SSIdataFrame$SSI_fa_pa, na.rm=TRUE), col=rainbow(11)[5], lwd=2, lty=2)

# lines(SSIdataFrame[which(SSIdataFrame$video==i),]$SSI_fa_pa_th, col=rainbow(11)[6], lwd=2)
# abline(h= mean(SSIdataFrame$SSI_fa_pa_th, na.rm=TRUE), col=rainbow(11)[6], lwd=2, lty=2)

lines(SSIdataFrame[which(SSIdataFrame$video==i),]$SSI_fa_th, col=rainbow(11)[7], lwd=2)
abline(h= mean(SSIdataFrame$SSI_fa_th, na.rm=TRUE), col=rainbow(11)[7], lwd=2, lty=2)

lines(SSIdataFrame[which(SSIdataFrame$video==i),]$SSI_mo_pa, col=rainbow(11)[8], lwd=2)
abline(h= mean(SSIdataFrame$SSI_mo_pa, na.rm=TRUE), col=rainbow(11)[8], lwd=2, lty=2)

lines(SSIdataFrame[which(SSIdataFrame$video==i),]$SSI_mo_pa_th, col=rainbow(11)[9], lwd=2)
abline(h= mean(SSIdataFrame$SSI_mo_pa_th, na.rm=TRUE), col=rainbow(11)[9], lwd=2, lty=2)
lines(SSIdataFrame[which(SSIdataFrame$video==i),]$SSI_mo_th, col=rainbow(11)[10], lwd=2)
abline(h= mean(SSIdataFrame$SSI_mo_th, na.rm=TRUE), col=rainbow(11)[10], lwd=2, lty=2)

lines(SSIdataFrame[which(SSIdataFrame$video==i),]$SSI_pa_th, col=rainbow(11)[11], lwd=2)
abline(h= mean(SSIdataFrame$SSI_pa_th, na.rm=TRUE), col=rainbow(11)[11], lwd=2, lty=2)

legend("topleft", inset=.05, c("fa_mo", "fa_mo_pa", "fa_mo_pa_th",
"fa_mo_th", "fa_pa", "fa_pa_th", "fa_th",
"mo_pa", "mo_pa_th", "mo_th", "pa_th"),
col=rainbow(11), cex=0.6, lwd=2)

legend("topright", inset=.05, c(paste ("Mean fa_mo :",
round(mean(SSIdataFrame$SSI_fa_mo, na.rm=TRUE),3)),
paste ("Mean fa_mo_pa :", round(mean(SSIdataFrame$SSI_fa_mo_pa, na.rm=TRUE),3)),
# paste ("Mean fa_mo_pa_th :", #round(mean(SSIdataFrame$SSI_fa_mo_pa_th),3)),
paste ("Mean fa_mo_th :", round(mean(SSIdataFrame$SSI_fa_mo_th, na.rm=TRUE),3)),
paste ("Mean fa_pa :", round(mean(SSIdataFrame$SSI_fa_pa, na.rm=TRUE),3)),
# paste ("Mean fa_pa_th :", round(mean(SSIdataFrame$SSI_fa_pa_th, na.rm=TRUE),3)),
paste ("Mean fa_th :", round(mean(SSIdataFrame$SSI_fa_th, na.rm=TRUE),3)),
paste ("Mean mo_pa :", round(mean(SSIdataFrame$SSI_mo_pa, na.rm=TRUE),3)),
paste ("Mean mo_pa_th :", round(mean(SSIdataFrame$SSI_mo_pa_th, na.rm=TRUE),3)),
paste ("Mean mo_th :", round(mean(SSIdataFrame$SSI_mo_th, na.rm=TRUE),3)),
paste ("Mean pa_th :", round(mean(SSIdataFrame$SSI_pa_th, na.rm=TRUE),3))),
col=rainbow(11), cex=0.5, lty=2, lwd=1)})

```

## Evolution of synchrony through time, raw each second

```

par(mar=c(4,4,4,4))
col <- 1
for (i in 6:length(SSIdataFrame)){
  plot(1:length(SSIdataFrame[,i]), SSIdataFrame[,i], type="l",
  col=rainbow(11)[col], main = names(SSIdataFrame)[i])
  col <- col+1}

```

## Evolution of synchrony through time, mean by minute

```
par(mar=c(4,4,4,4))
col = 1
for (indexSSI in 6:length(SSIdataFrame)){
  IntervalNumbersVideo <- ceiling(length(SSIdataFrame[,indexSSI])/6)
  SSIColumn <- SSIdataFrame[,indexSSI]
  SSIdataFrameMinute <- c()
  for (i in 1:IntervalNumbersVideo){
    borneInf <- 1+(i-1)*6
    borneSup <- i * 6
    SSIVectorInterval <- SSIColumn[borneInf:borneSup]
    mean <- mean(SSIVectorInterval, na.rm=TRUE)
    SSIdataFrameMinute <- c(SSIdataFrameMinute, mean)}
  plot(1:length(SSIdataFrameMinute), SSIdataFrameMinute, type="l", col=rainbow(11)[col], main = names
  col <- col+1}
```

## Evolution of synchrony through time, mean by 10 minutes

```
par(mar=c(4,4,4,4))
col = 1
for (indexSSI in 6:length(SSIdataFrame)){
  IntervalNumbersVideo <- ceiling(length(SSIdataFrame[,indexSSI])/60)
  SSIColumn <- SSIdataFrame[,indexSSI]
  SSIdataFrameTenMinute <- c()
  for (i in 1:IntervalNumbersVideo){
    borneInf <- 1+(i-1)*60
    borneSup <- i * 60
    SSIVectorInterval <- SSIColumn[borneInf:borneSup]
    mean <- mean(SSIVectorInterval, na.rm=TRUE)
    SSIdataFrameTenMinute <- c(SSIdataFrameTenMinute, mean)}
  plot(1:length(SSIdataFrameTenMinute), SSIdataFrameTenMinute, type="l", col=rainbow(11)[col], main =
  col <- col+1}
```

## Models of synchrony

```
SSI_fa_th_lme <- lmer(SSI_fa_th ~ Time_min + (1|video), data=SSIdataFrame)
summary(SSI_fa_th_lme)
#plot(SSI_fa_th_lme)
res <- residuals(SSI_fa_th_lme)
hist(SSIdataFrame$SSI_fa_th)
qqnorm(res)
SSI_fa_th_List <- c()
for (i in indexList){
  SSI_fa_th_List <- c(SSI_fa_th_List, mean(SSIdataFrame[which(SSIdataFrame$video==i),]$SSI_fa_th, na.rm=
})
print(SSI_fa_th_List)
#plot(SSI_fa_th_List, type="b")
```



```

# log of the data
log_SSI_fa_th <- hist(log(SSIdataFrame$SSI_fa_th))
SSI_fa_th_log_lme <- lmer(log(SSI_fa_th) ~ Time_min + (1|video), data=SSIdataFrame)
res_log <- residuals(SSI_fa_th_log_lme)
qqnorm(res_log)
summary(SSI_fa_th_log_lme)

```

```

# root square of the data
sq_SSI_fa_th <- hist(sqrt(SSIdataFrame$SSI_fa_th))
SSI_fa_th_sq_lme <- lmer(sqrt(SSI_fa_th) ~ Time_min + (1|video), data=SSIdataFrame)
res_sq <- residuals(SSI_fa_th_sq_lme)
qqnorm(res_sq)
summary(SSI_fa_th_sq_lme)

```