Canary Care Analysis

P Barber

17 May 2019

Load Data

Load data from the csv file that Canary Care provided.

```
data <- read.csv(file = "Example_HubMessages.csv", stringsAsFactors = F)
```

Look at only Sensor Movements

```
d <- data[data$MessageTypeName == "SensorMovement",]
```

Get a usable time from the text time stamp from the sensor and remove invalid times (TODO Look into why these are here). Reorder the data according to the sensor time as this is the actual event time, messages may not arrive at the server in order.

```
d$time <- as.POSIXct(d$sensorTime, format='%Y-%m-%d %H:%M:%S')
d <- d[!is.na(d$time),]

# timing of events is critical here, must reorder in sensor time.
d <- d[order(d$time),]
```

Exclude data from certain time periods

E.g. Use data up to 11th April 2019, 4 pm (we know there is unusual behaviour after 10:02:50).

```
d <- d[d$time < as.POSIXct("2019-04-11 16:00:00", format='%Y-%m-%d %H:%M:%S'),]
```

Accumulate data day by day to build up a picture of movements

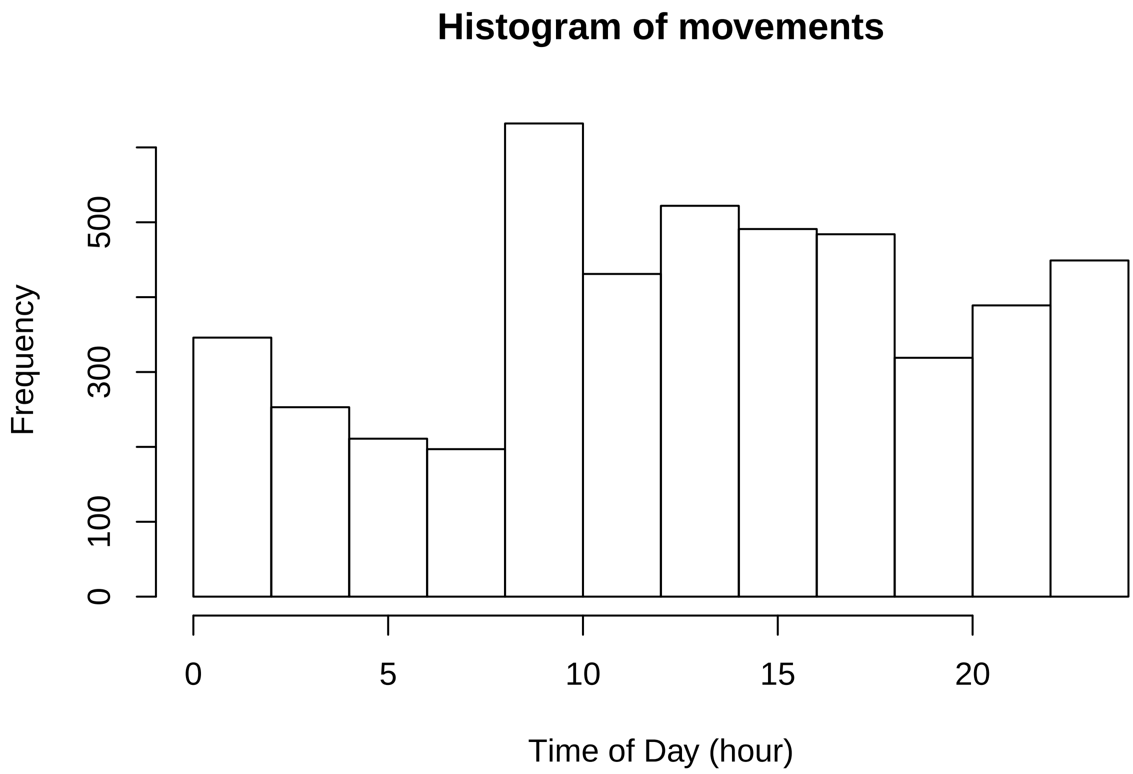
Calculate time of day data and also bin into 15 minute bins for easy plotting.

```
d$hour <- hour(d$time) + minute(d$time)/60 + second(d$time)/3600

#Create Bins
bins=c(paste0(rep(c(paste0(0, 0:9), 10:23), each=4), ".", c("00", 25, 50, 75))[-1], "24:00")

#Divide Data Into Bins
d$timeofday = cut(d$hour, breaks=seq(0, 24, 0.25), labels=bins)

#Reformat to Numeric
d$timeofday <- as.numeric(as.character(d$timeofday))
```

```
## Warning: NAs introduced by coercion
```

Plot a histogram of movements throughout the day.

```r
hist(d$timeofday, xlab="Time of Day (hour)", main="Histogram of movements")
```



We see there can be movement at any time of the day or night. Wakeup is usually after 8am, most movement is in the morning and bedtime is probably after 10pm.

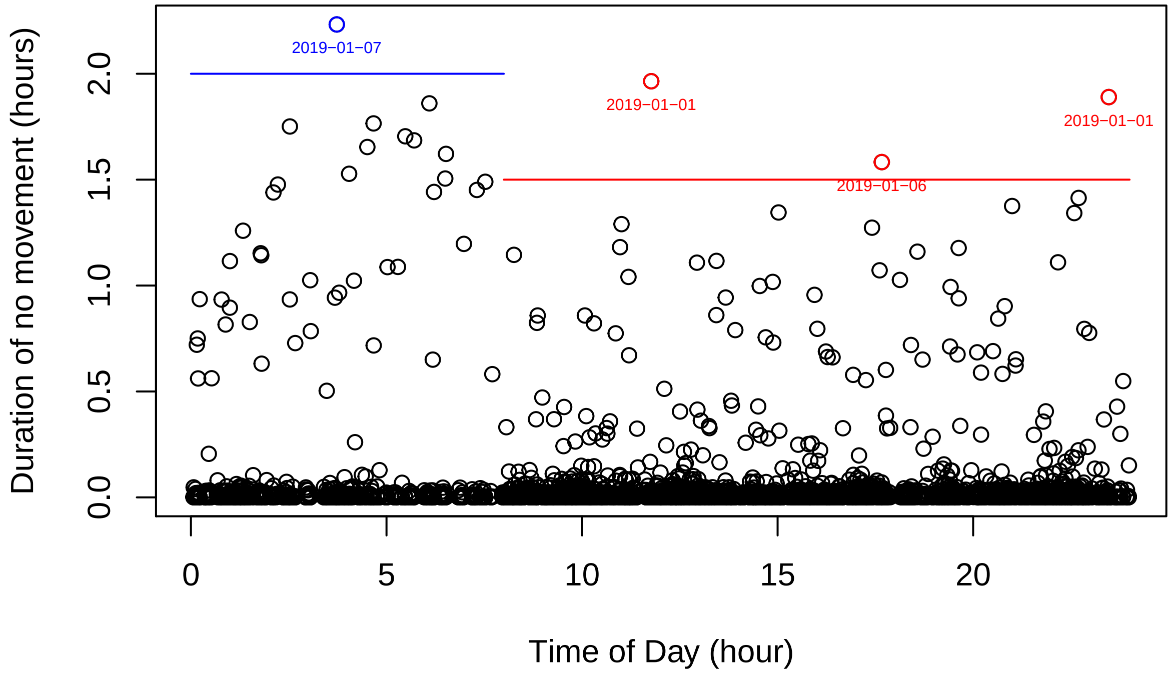
Let's now get the difference between consequetive movements.

```r
diff <- d$hour[-1] - d$hour[1:length(d$hour)-1]
diff <- c(diff, 0)
diff <- ifelse(diff<0, diff+24, diff)

d$time_diff <- diff
```

Plot time differences versus time of day. Also, think about setting rules based on no movement for a length of time. Here using 2 thresholds (t) one for before the hour h and one for after. The horizontal line show these thresholds. Any time between movements that exceed these thresholds are coloured and labelled with their date.

```r
plot(d$hour, d$time_diff, xlab = "Time of Day (hour)",
     ylab = "Duration of no movement (hours)")

h = 8
t = 1.5
extreme <- d[d$time_diff>t & d$hour>h, c("MessageID", "sensorTime", "SensorName", "timeofday", "time",
if(dim(extreme)[1]>0){
  points(extreme$hour, extreme$time_diff, col="red")
  text(extreme$hour, extreme$time_diff, col="red", labels=date(extreme$time),
     cex=0.5, pos=1)
```

```
}
lines(c(h,24), c(t,t), col="red")

t = 2.0
extreme <- d[d$time_diff>t & d$hour<h, c("MessageID", "sensorTime", "SensorName", "timeofday", "time",
if(dim(extreme)[1]>0){
  points(extreme$hour, extreme$time_diff, col="blue")
  text(extreme$hour, extreme$time_diff, col="blue", labels=date(extreme$time),
     cex=0.5, pos=1)
}
lines(c(0,h), c(t,t), col="blue")
```



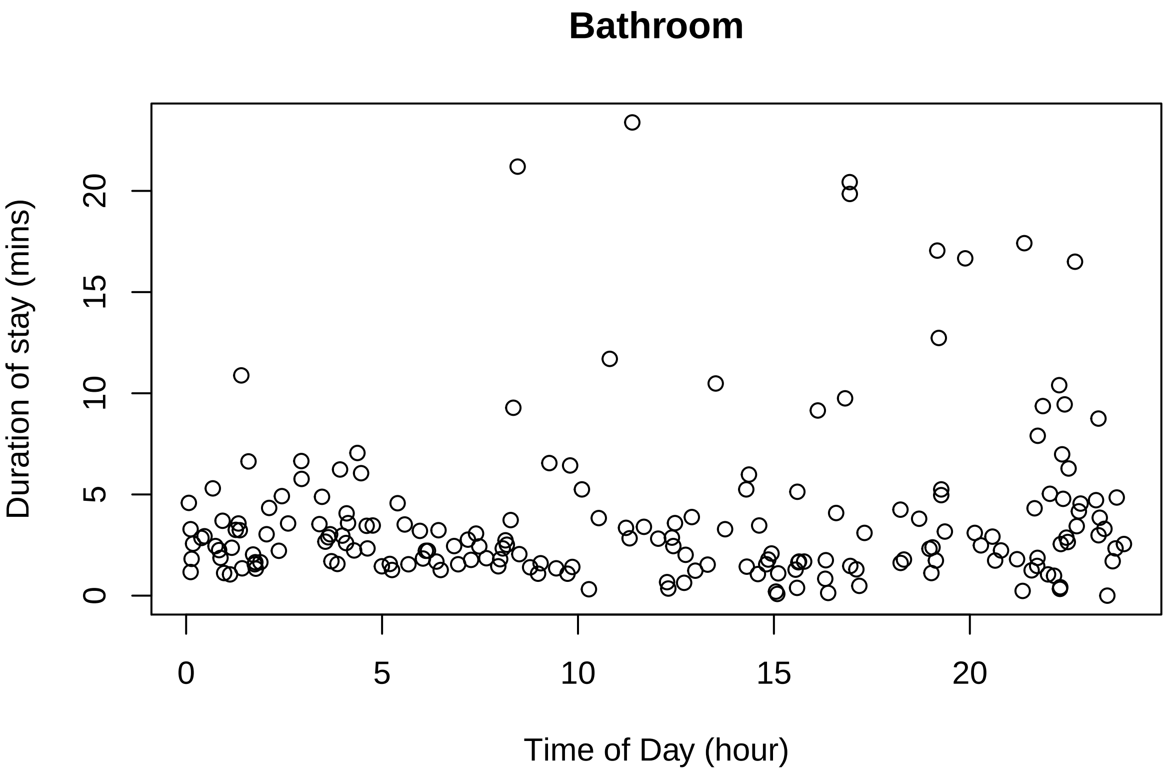
We can see how many alarms we would have got for a given threshold. The threshold is a trade off between catching a real event and getting too many flase alarms.

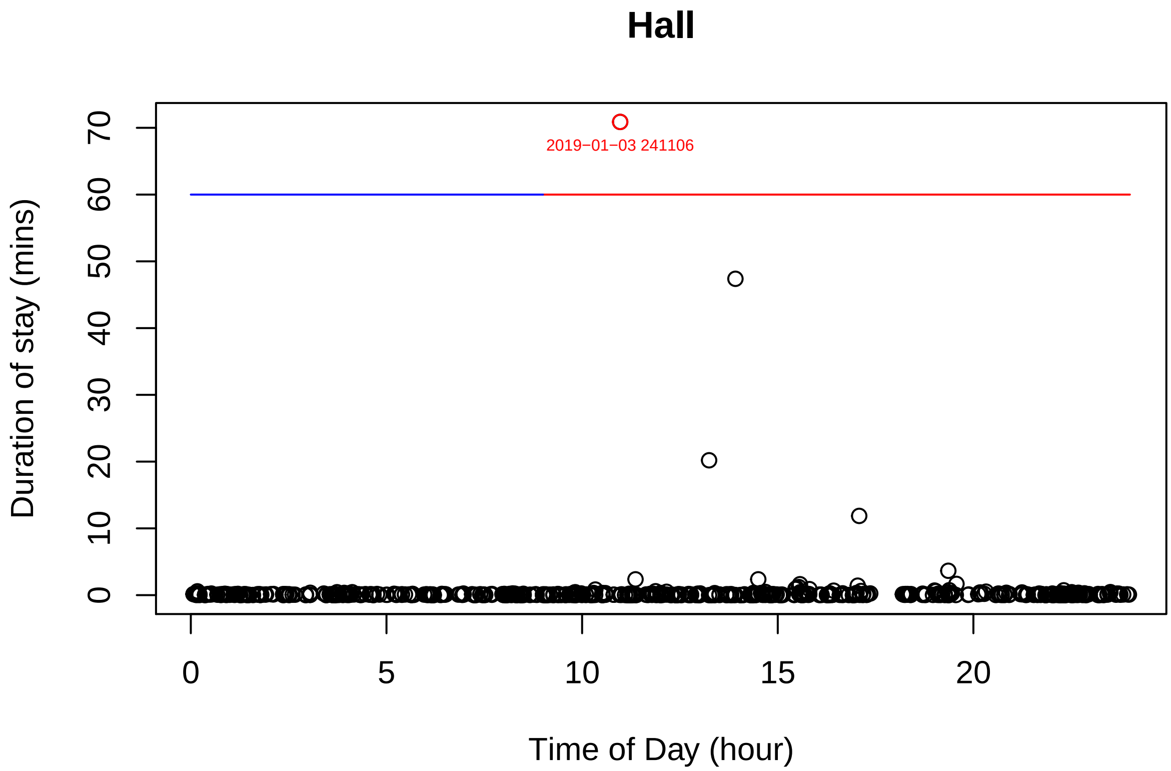
Let's now look for how long are the stays in one single place. We look for the first event in the specified place and then measure the time to the next movement in a new place.

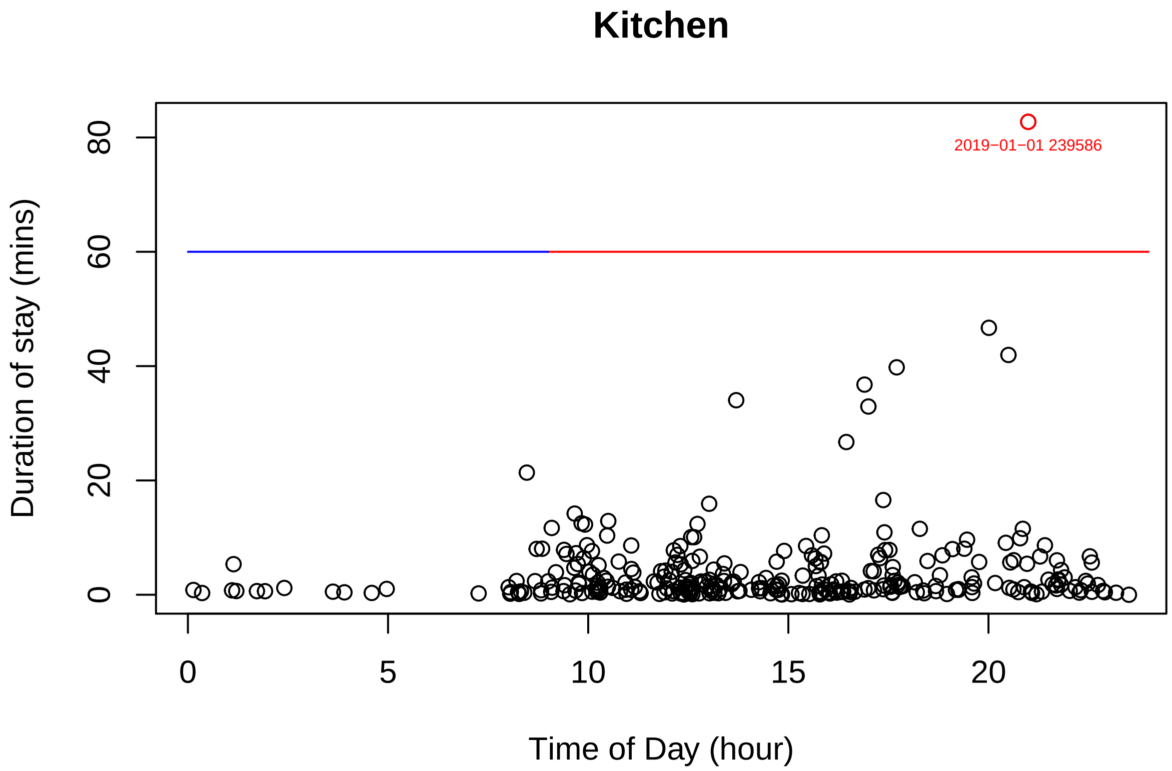
```
# d should be a dataframe of all movements

places <- c("Bathroom", "Hall", "Kitchen", "Lounge", "Bedroom")


for (place in places){
  i = 1

  start <- vector()
```

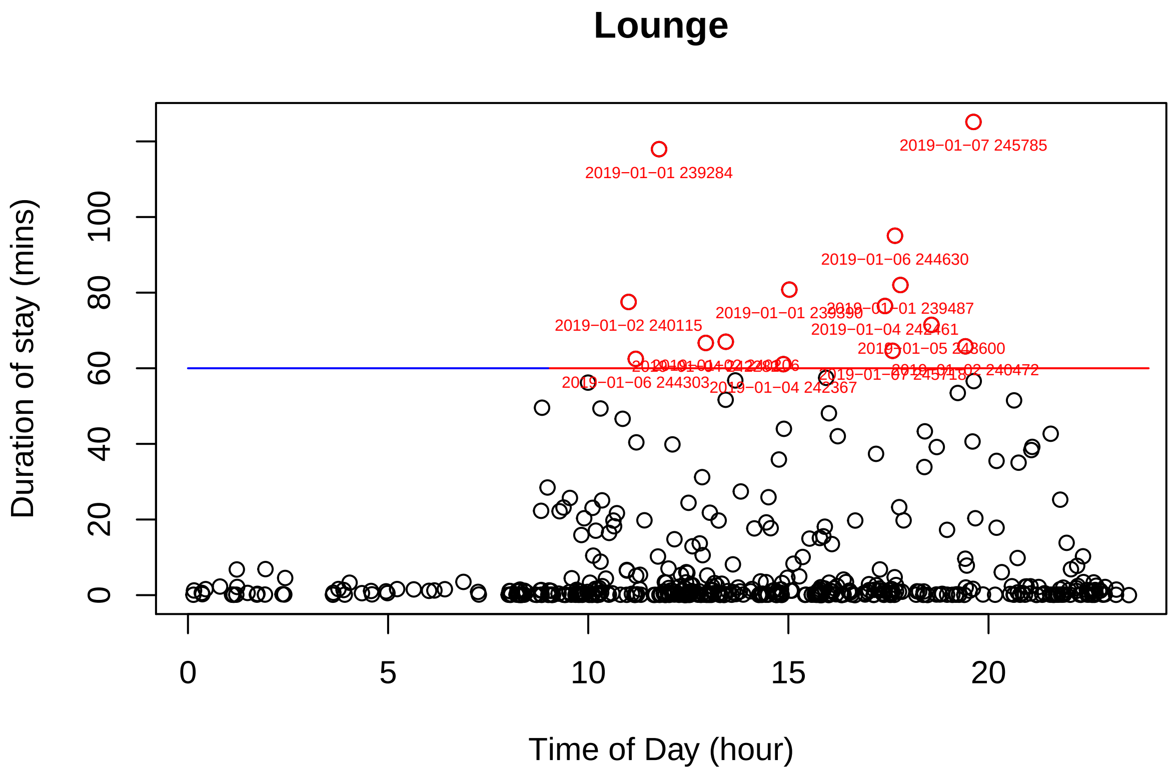
```
stop <- vector()
duration <- vector()
start_hour <- vector()
delta_i <- vector()

while (i<dim(d)[1]){

  # Find next mention of place
  while (d[i,"SensorName"]!=place && i<dim(d)[1]){
    i = i+1
  }

  this_i = d[i,"MessageID"]
  this_start <- d[i,"time"]
  start <- c(start, this_start)
  start_hour <- c(start_hour, d[i,"hour"])

  # Find next mention of another place
  while (d[i,"SensorName"]==place && i<dim(d)[1]){
    i = i+1
  }

  delta_i <- c(delta_i, this_i)
  this_stop <- d[i,"time"]
  stop <- c(stop, this_stop)

  this_duration <- this_stop-this_start
  units(this_duration) <- "mins"

  duration <- c(duration, this_duration)

}

start <- as.POSIXct(start, origin = "1970-01-01")
stop <- as.POSIXct(stop, origin = "1970-01-01")

stays <- data.frame(start, stop, start_hour, duration, delta_i)
colnames(stays) <- c("start", "stop", "hour", "duration", "delta_i")


plot(stays$hour, stays$duration, xlab = "Time of Day (hour)",
     ylab = "Duration of stay (mins)", main = place)

h = 9

# DAY
t = 60
extreme <- stays[stays$duration>t & stays$hour>h, c("start", "hour", "duration", "delta_i")]
if(dim(extreme)[1]>0){
  points(extreme$hour, extreme$duration, col="red")
  text(extreme$hour, extreme$duration, col="red", labels=paste(date(extreme$start), extreme$delta_i),
       cex=0.5, pos=1)
}
```

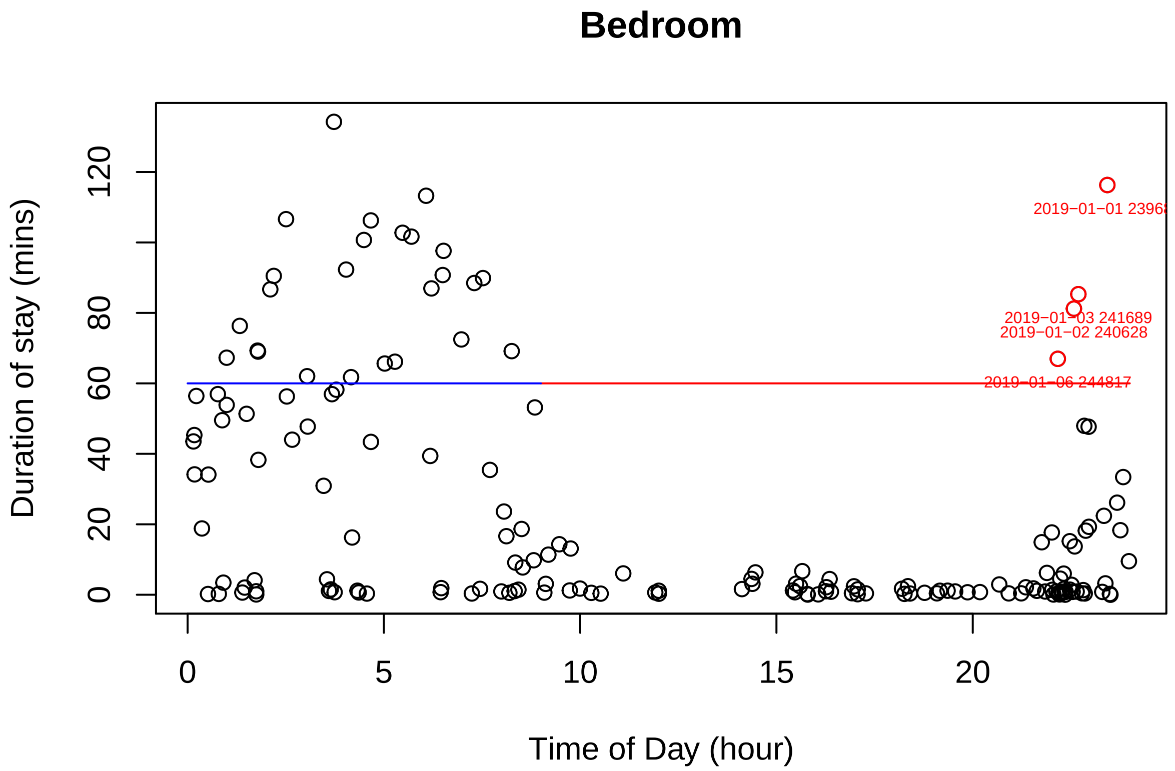
```
  lines(c(h,24), c(t,t), col="red")

  # NIGHT
  t = 60
  extreme <- stays[stays$duration>t & stays$hour<h, c("start", "hour", "duration", "delta_i")]
  if(dim(extreme)[1]>0 && place!="Bedroom"){
    points(extreme$hour, extreme$duration, col="blue")
    text(extreme$hour, extreme$duration, col="blue", labels=paste(date(extreme$start), extreme$delta_i)
         cex=0.5, pos=1)
  }
  lines(c(0,h), c(t,t), col="blue")
}
```




Hall
Duration of stay (mins)
0
10
20
30
40
50
60
70
2019−01−03 241106
0
5
10
15
20
Time of Day (hour)


Kitchen
2019−01−01 239586
Duration of stay (mins)
0
20
40
60
80
Time of Day (hour)
0
5
10
15
20


Lounge
Duration of stay (mins)
Time of Day (hour)
0
20
40
60
80
100
0
5
10
15
20
2019−01−07 245785
2019−01−01 239284
2019−01−06 244630
2019−01−01 239390
2019−01−01 239487
2019−01−02 240115
2019−01−04 242461
2019−01−05 243600
2019−01−02 240472
2019−01−07 245718
2019−01−06 244303
2019−01−04 242367



To investigate strange things create a smaller dataset of all events, not just movements, for a given day. You can also see when lights were turned on and off.

```
d <- data
#d <- data[data$MessageTypeName == "SensorMovement",]

d$time <- as.POSIXct(d$sensorTime, format='%Y-%m-%d %H:%M:%S')
d$hour <- hour(d$time) + minute(d$time)/60 + second(d$time)/3600

d_less <- d[date(d$time)=="2019-02-14", c("MessageID", "sensorTime", "SensorName", "MessageTypeName", "

d_less <- d_less[order(d_less$time),]
```

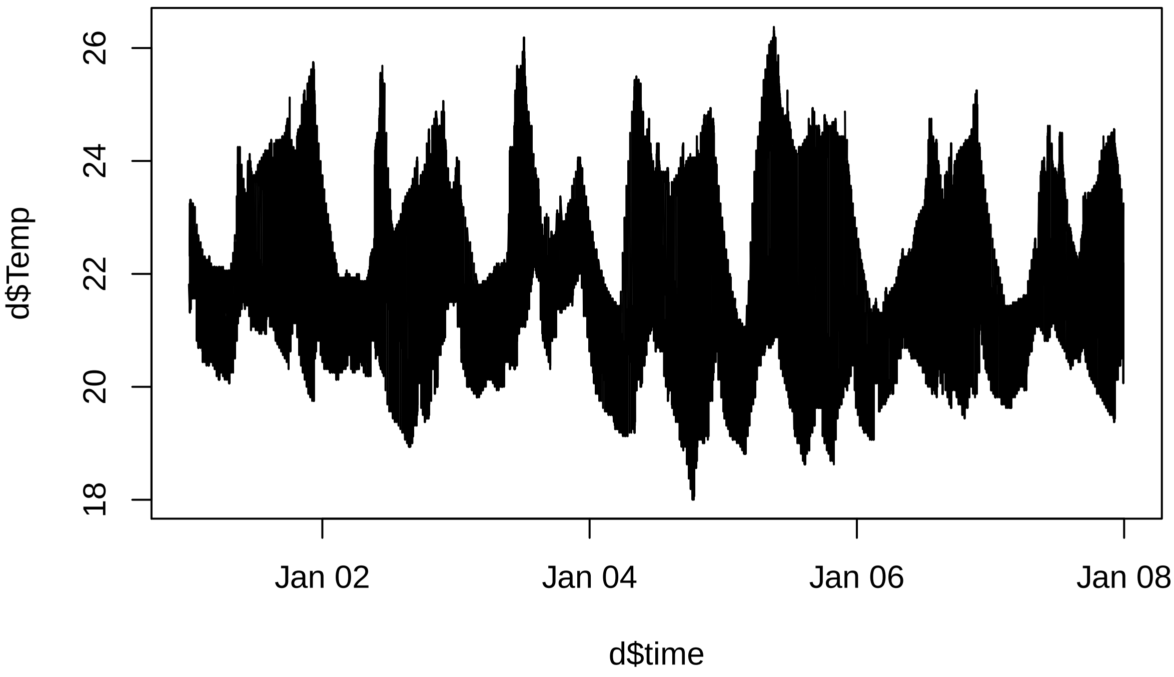
Look at Temperature

Get a fresh dataset as before.

```
d <- data

d$time <- as.POSIXct(d$receivedTime, format='%Y-%m-%d %H:%M:%S')

d <- d[d$time < as.POSIXct("2019-04-11 16:00:00", format='%Y-%m-%d %H:%M:%S'),]

d <- d[!is.na(d$time),]
```

Plot temperature versus time.

```
plot(d$time, d$Temp, type = "l")
```


d$Temp
Jan 02
Jan 04
Jan 06
Jan 08
d$time