RFID Room 2

Data Preprocessing

Function to find the most frequent zone (average wont quite work)

```
MaxTable <- function(x){
     dd <- unique(x)
     dd[which.max(tabulate(match(x,dd)))]
}
```

My to minute function

```
to_minute <- function(xts_object){
  ep_min <- endpoints(na.omit(xts_object), on="minutes",k=60)
  ave_min <- period.apply(na.omit(xts_object), INDEX=ep_min, FUN=MaxTable)
  #this removes the time to allow cbinding
  ave_min <- to.minutes(ave_min,OHLC=F,indexAt="startof")
  return(ave_min)
}
```

My to daily function

```
 to_daily <- function(xts_object){
   ep_day <- endpoints(na.omit(xts_object), on="minutes",k=1440)
   ave_day <- period.apply(na.omit(xts_object), INDEX=ep_day, FUN=MaxTable)
   #this removes the time to allow cbinding
   ave_day <- to.daily(ave_day,OHLC=F,indexAt="startof")
   return(ave_day)
 }
```

My to weekly function

```
to_weekly <- function(xts_object){
  ep_week <- endpoints(na.omit(xts_object), on="minutes",k=10080)
  ave_week <- period.apply(na.omit(xts_object), INDEX=ep_week, FUN=MaxTable)
  #this removes the time to allow cbinding
  ave_week <- to.weekly(ave_week,OHLC=F,indexAt="startof")
  return(ave_week)
}
```

My to monthly function

```
to_monthly <- function(xts_object){
  ep_month <- endpoints(na.omit(xts_object), on="minutes",k=43800)
  ave_month <- period.apply(na.omit(xts_object), INDEX=ep_month, FUN=MaxTable)
  #this removes the time to allow cbinding
  ave_month <- to.monthly(ave_month,OHLC=F,indexAt="startof")
  return(ave_month)
}
```

Function to Calculate Transitions

```
#I don't know when we use this data yet, will be useful when making transition tables for the whole dat
sep_bird_id_xts <- function(samp_name, samp_id,cage_obj){

  raw_sample_tab <- subset(cage_obj, tagname==samp_id)

  xts_object <- xts(raw_sample_tab ,order.by = raw_sample_tab$DateTime)

  result <- list("name" = samp_name, "ID"=samp_id, "xts_obj"=xts_object)
  return(result)
}

sep_bird_id_period <- function(samp_name, samp_id, cage_obj, cutoff){

  raw_sample_tab <- subset(cage_obj, tagname==samp_id)

  xts_object <- xts(raw_sample_tab ,order.by=raw_sample_tab$DateTime)

  xts_object <- xts_object[cutoff]

  daily <- to_daily(xts_object$subzone)#, OHLC=F)
  weekly <- to_weekly(xts_object$subzone)#, OHLC=F)
  monthly <- to_monthly(xts_object$subzone)#, OHLC=F)
  index(monthly) <-as.POSIXct(index(monthly))

  result <- list("name" = samp_name, "ID"=samp_id, "xts_obj"=xts_object, "daily_obj"=daily,
                 "weekly_obj"=weekly, "monthly_obj"=monthly)
  return(result)
}

#input a sample name
#returns a vector labeled with sample name, and bottom, middle, top and total transitions

calc_trans<- function(samp_name,samp_obj, id){
  bottom_trans <-0
  mid_trans<-0
  top_trans<-0
  trans<-0
  count <- 0

  for (i in 2:length(samp_obj$subzone)) {
  previous_state <- as.character(samp_obj$subzone[i-1])
  current_state <- as.character(samp_obj$subzone[i])

  #print(paste(previous_state,":",current_state))
  if (previous_state == current_state) {
    count <- count+1
  }
  else{

    if((current_state == "bottom") || (current_state == "Bottom")){
      bottom_trans <- bottom_trans+1
      trans<- trans+1
    }
    if (current_state=="middle" || current_state == "Middle") {
      mid_trans <- mid_trans+1
      trans <- trans+1
    }
    if(current_state =="top" || current_state=="Top"){
      top_trans <- top_trans+1
      trans <- trans+1
    }
  }
  }

  result <- c(samp_name,id,bottom_trans,mid_trans,top_trans,trans)
  return(result)

}

calc_trans_period<- function(samp_name, daily_indexed, raw_table,freq, day_offset){
  result <- data.frame()

  for (i in 2:(length(daily_indexed))){
    curr_day <- raw_table[paste0(index(daily_indexed)[i-1],"/",index(daily_indexed)[i])]
    curr_day_trans <- calc_trans(paste0(samp_name,".",index(daily_indexed)[i-1]),curr_day, samp_name)
    result <- rbind(result,curr_day_trans)
    }
  colnames(result) <- c("sample","ID","bottom","mid","top","total")
  return(result)
}
```

```
calc_trans_duration<- function(samp_name,samp_obj, id){
  bottom_trans <-0
  mid_trans<-0
  top_trans<-0
  trans<-0
  count <- 0
  bottom_time <- 0
  mid_time <- 0
  top_time <- 0

  for (i in 2:length(samp_obj$subzone)) {
  previous_state <- as.character(samp_obj$subzone[i-1])
  current_state <- as.character(samp_obj$subzone[i])
  previous_time <- index(samp_obj[i-1])
  current_time <- index(samp_obj[i])

  #print(paste(previous_state,":",current_state))
```

```r
  if (previous_state == current_state) {
    count <- count+1
  }
  else{

    if((previous_state == "bottom") || (previous_state == "Bottom")){
      bottom_trans <- bottom_trans+1
      trans<- trans+1
      bottom_time <- bottom_time + difftime(current_time, previous_time,units="secs")
    }
    if (previous_state=="middle" || previous_state == "Middle") {
      mid_trans <- mid_trans+1
      trans <- trans+1
      mid_time <- mid_time + difftime(current_time, previous_time, units="secs")
    }
    if(previous_state =="top" || previous_state=="Top"){
      top_trans <- top_trans+1
      trans <- trans+1
      top_time <- top_time + difftime(current_time,previous_time,units="secs")
    }
  }
  }

  result <- c(samp_name,id,bottom_trans,mid_trans,top_trans,trans,bottom_time,mid_time,top_time)
  #colnames(result) <- c("sample_name","ID","bottom_trans","middle_trans","top_trans","total_trans","bo
  return(result)

}

calc_trans_period_duration<- function(samp_name, daily_indexed, raw_table,freq,day_offset ){
  result <- data.frame()

  for (i in 2:(length(daily_indexed))){
    curr_day <- raw_table[paste0(index(daily_indexed)[i-1],"/",index(daily_indexed)[i])]
    curr_day_trans <- calc_trans_duration(paste0(samp_name,".",freq,".",(i+day_offset)),curr_day, samp_n
    result <- rbind(result,curr_day_trans)
    }
  colnames(result) <- c("sample_name","ID","bottom_trans","middle_trans","top_trans","total_trans","bott
  return(result)
}
```

```r
calc_zone_duration <- function(samp_name, id, samp_obj){
  bottom_time <- 0
  mid_time <- 0
  top_time <- 0

  current_zone <- ""
  current_datetime <- ""
  next_datetime <- ""

  for(i in 1:(length(index(samp_obj))-1)){
    current_zone <- as.character(samp_obj[i]$subzone)

    #this is measured in seconds
    sec_in_zone_i <- as.numeric(index(samp_obj[i+1]$accessdate)) - as.numeric(index(samp_obj[i]$accessda

    if(current_zone == "bottom"){
      bottom_time <- bottom_time + sec_in_zone_i
      sec_in_zone_i <- 0
    } else if(current_zone == "middle"){
      mid_time <- mid_time + sec_in_zone_i
      sec_in_zone_i <- 0
    } else if(current_zone == "top"){
      top_time <- top_time + sec_in_zone_i
      sec_in_zone_i <- 0
    }

  }

  return(c(samp_name, id, as.numeric(bottom_time), as.numeric(mid_time), as.numeric(top_time)))

}
```

Import Room 2

```
library(xts)
room_2 <- read.csv("../data/DK20-03-RFID-R2-febmay-080423.csv")

bird_ids_room_2 <- unique(room_2$tagname)
bird_ids_room_2 <- na.trim(sort(bird_ids_room_2))

room_2["DateTime"] <- as.POSIXct(room_2$access, origin="1970-01-01", tz="GMT")

print("what makes up subzone col")
```

```
## [1] "what makes up subzone col"
```

```
unique(room_2$subzone)
```

```
## [1] "Bottom" "Middle" "Top"
```

```
room_2$subzone[room_2$subzone == "Bottom"] <- "bottom"
room_2$subzone[room_2$subzone == "Middle"] <- "middle"
room_2$subzone[room_2$subzone == "Top"] <- "top"

#65475
print("what makes up subzone col")
```

```
## [1] "what makes up subzone col"
```

```
unique(room_2$subzone)
```

```
## [1] "bottom" "middle" "top"
```

```
print("how many NAs in DateTime and Subzone")
```

```
## [1] "how many NAs in DateTime and Subzone"
```

```
sum(is.na(room_2$DateTime))
```

```
## [1] 0
```

```
sum(is.na(room_2$subzone))
```

```
## [1] 0
```

```
#generate list of XTS objects
reslist_room_2 <- list()
for(i in 1:length(bird_ids_room_2)){

  res <- sep_bird_id_xts(samp_name="room 2",cage_obj =room_2, samp_id = bird_ids_room_2[i])
  reslist_room_2[[i]] <- res
}

cutoff_room_2 <- data.frame()
for(i in 1:length(reslist_room_2)){
  id <- reslist_room_2[[i]]$ID
  top <- index(head(reslist_room_2[[i]]$xts_obj))[1]
  bottom <- index(tail(reslist_room_2[[i]]$xts_obj))[6]
  rec <- cbind(id,as.character(top),as.character(bottom))
  cutoff_room_2 <- rbind(cutoff_room_2,rec)

}

#head
cutoff_room_2[order(cutoff_room_2$V2),]
```

```
##      id                  V2                  V3
## 37 9019 2021-02-01 16:49:19 2021-05-07 15:59:25
## 10 6910 2021-02-01 16:50:10 2021-03-04 10:53:16
## 40 9026 2021-02-01 16:52:22 2021-05-07 15:45:36
## 1  6855 2021-02-01 16:54:44 2021-05-07 16:45:34
## 6  6901 2021-02-01 16:55:06 2021-05-07 16:29:21
## 16 6935 2021-02-01 16:57:13 2021-05-07 16:22:02
## 17 6937 2021-02-01 16:57:33 2021-05-07 16:14:25
## 2  6860 2021-02-01 16:57:46 2021-05-07 14:58:10
## 3  6872 2021-02-01 16:59:54 2021-05-07 16:37:17
## 24 6960 2021-02-01 17:00:23 2021-05-07 16:25:57
## 14 6925 2021-02-01 17:01:17 2021-05-07 16:45:30
## 12 6914 2021-02-01 17:03:41 2021-04-08 13:05:36
## 25 6962 2021-02-01 17:04:35 2021-05-07 16:30:25
## 23 6958 2021-02-01 17:04:40 2021-05-07 16:16:47
## 30 6981 2021-02-01 17:06:27 2021-05-07 16:34:46
## 34 6989 2021-02-01 17:11:36 2021-05-07 16:33:17
## 15 6926 2021-02-01 17:16:05 2021-05-07 16:41:12
## 4  6877 2021-02-01 17:20:28 2021-05-07 16:40:03
## 22 6956 2021-02-01 17:25:22 2021-05-07 14:54:01
## 31 6983 2021-02-01 17:27:52 2021-05-07 13:14:16
## 32 6986 2021-02-01 17:29:10 2021-05-07 16:25:00
## 8  6908 2021-02-01 17:31:40 2021-05-07 16:18:13
## 18 6942 2021-02-01 17:41:27 2021-05-07 15:53:40
## 39 9024 2021-02-01 17:42:56 2021-05-07 15:48:26
## 20 6952 2021-02-01 17:49:45 2021-05-07 16:31:01
## 9  6909 2021-02-01 17:51:31 2021-05-07 16:17:43
## 38 9021 2021-02-01 18:04:12 2021-05-07 16:43:25
## 28 6975 2021-02-01 18:06:10 2021-05-07 16:03:34
## 29 6978 2021-02-01 18:09:02 2021-05-07 16:41:44
## 26 6966 2021-02-01 18:13:12 2021-03-31 21:25:58
## 11 6911 2021-02-01 18:14:42 2021-05-07 16:40:05
## 36 9005 2021-02-01 18:15:23 2021-05-07 16:23:26
## 7  6903 2021-02-01 18:17:54 2021-05-07 16:43:30
## 13 6919 2021-02-01 18:22:33 2021-05-07 16:42:28
## 19 6946 2021-02-01 18:32:47 2021-05-07 16:36:09
## 35 9001 2021-02-01 18:34:49 2021-05-07 15:18:17
## 5  6890 2021-02-01 18:42:01 2021-05-07 16:23:49
## 33 6988 2021-02-01 19:09:11 2021-05-07 16:26:00
## 21 6955 2021-02-01 19:22:09 2021-05-07 16:45:20
## 27 6971 2021-02-01 20:08:57 2021-05-07 16:03:12
```

```
#tail
cutoff_room_2[order(cutoff_room_2$V3),]
```

```
##      id                  V2                  V3
## 10 6910 2021-02-01 16:50:10 2021-03-04 10:53:16
## 26 6966 2021-02-01 18:13:12 2021-03-31 21:25:58
## 12 6914 2021-02-01 17:03:41 2021-04-08 13:05:36
## 31 6983 2021-02-01 17:27:52 2021-05-07 13:14:16
## 22 6956 2021-02-01 17:25:22 2021-05-07 14:54:01
## 2  6860 2021-02-01 16:57:46 2021-05-07 14:58:10
## 35 9001 2021-02-01 18:34:49 2021-05-07 15:18:17
## 40 9026 2021-02-01 16:52:22 2021-05-07 15:45:36
## 39 9024 2021-02-01 17:42:56 2021-05-07 15:48:26
## 18 6942 2021-02-01 17:41:27 2021-05-07 15:53:40
## 37 9019 2021-02-01 16:49:19 2021-05-07 15:59:25
## 27 6971 2021-02-01 20:08:57 2021-05-07 16:03:12
## 28 6975 2021-02-01 18:06:10 2021-05-07 16:03:34
## 17 6937 2021-02-01 16:57:33 2021-05-07 16:14:25
## 23 6958 2021-02-01 17:04:40 2021-05-07 16:16:47
## 9  6909 2021-02-01 17:51:31 2021-05-07 16:17:43
## 8  6908 2021-02-01 17:31:40 2021-05-07 16:18:13
## 16 6935 2021-02-01 16:57:13 2021-05-07 16:22:02
## 36 9005 2021-02-01 18:15:23 2021-05-07 16:23:26
## 5  6890 2021-02-01 18:42:01 2021-05-07 16:23:49
## 32 6986 2021-02-01 17:29:10 2021-05-07 16:25:00
## 24 6960 2021-02-01 17:00:23 2021-05-07 16:25:57
## 33 6988 2021-02-01 19:09:11 2021-05-07 16:26:00
## 6  6901 2021-02-01 16:55:06 2021-05-07 16:29:21
## 25 6962 2021-02-01 17:04:35 2021-05-07 16:30:25
## 20 6952 2021-02-01 17:49:45 2021-05-07 16:31:01
## 34 6989 2021-02-01 17:11:36 2021-05-07 16:33:17
## 30 6981 2021-02-01 17:06:27 2021-05-07 16:34:46
## 19 6946 2021-02-01 18:32:47 2021-05-07 16:36:09
## 3  6872 2021-02-01 16:59:54 2021-05-07 16:37:17
## 4  6877 2021-02-01 17:20:28 2021-05-07 16:40:03
## 11 6911 2021-02-01 18:14:42 2021-05-07 16:40:05
## 15 6926 2021-02-01 17:16:05 2021-05-07 16:41:12
## 29 6978 2021-02-01 18:09:02 2021-05-07 16:41:44
## 13 6919 2021-02-01 18:22:33 2021-05-07 16:42:28
## 38 9021 2021-02-01 18:04:12 2021-05-07 16:43:25
## 7  6903 2021-02-01 18:17:54 2021-05-07 16:43:30
## 21 6955 2021-02-01 19:22:09 2021-05-07 16:45:20
## 14 6925 2021-02-01 17:01:17 2021-05-07 16:45:30
## 1  6855 2021-02-01 16:54:44 2021-05-07 16:45:34
```

```
#remove <- c("6910","6914","9005")
remove <- c("6910","6966","6914")
bird_ids_room_2_new <- bird_ids_room_2 [! bird_ids_room_2  %in% remove ]
#generate list of the first 10 truncated XTS objects
reslist2_room_2 <- list()
for(i in 1:length(bird_ids_room_2_new)){
  res <- sep_bird_id_period(samp_name="room 2",cage_obj =room_2, samp_id = bird_ids_room_2_new[i],cutof
  reslist2_room_2[[i]] <- res
}

#find wholly non-na daily timeset, to be able to feed into the transcalc
big_table_room_2 <- reslist2_room_2[[1]]$daily_obj
for(i in 2:length(reslist2_room_2)){
  current <- reslist2_room_2[[i]]$daily_obj
  colnames(current) <- as.character(reslist2_room_2[[i]]$ID)
  big_table_room_2 <- cbind(big_table_room_2,current)
}
print("How many NAs in big_table_room_2")
```

```
## [1] "How many NAs in big_table_room_2"
```

```
print(sum(is.na(big_table_room_2)))
```

```
## [1] 0
```

Import Room 3

```
library(xts)
library(tidyverse)

room_3 <- read.csv("../data/DK20-03-RFID-R3-febmay-080423.csv") %>% na.exclude()

bird_ids_room_3 <- unique(room_3$tagname)
bird_ids_room_3 <- na.trim(sort(bird_ids_room_3))

room_3["DateTime"] <- as.POSIXct(room_3$access, origin="1970-01-01", tz="GMT")

print("what makes up subzone col")
```

```
## [1] "what makes up subzone col"
```

```
unique(room_3$subzone)
```

```
## [1] "Middle" "Top"    "Bottom"
```

```
room_3$subzone[room_3$subzone == "Bottom"] <- "bottom"
room_3$subzone[room_3$subzone == "Middle"] <- "middle"
room_3$subzone[room_3$subzone == "Top"] <- "top"
```

```r
print("what makes up subzone col")
```

```
## [1] "what makes up subzone col"
```

```r
unique(room_3$subzone)
```

```
## [1] "middle" "top"    "bottom"
```

```r
print("how many NA's are in Datetime and Subzone")
```

```
## [1] "how many NA’s are in Datetime and Subzone"
```

```r
sum(is.na(room_3$DateTime))
```

```
## [1] 0
```

```r
sum(is.na(room_3$subzone))
```

```
## [1] 0
```

```r
room_3 <- room_3[!is.na(room_3$DateTime),]

print("how many NA's are in Datetime and Subzone")
```

```
## [1] "how many NA’s are in Datetime and Subzone"
```

```r
sum(is.na(room_3$DateTime))
```

```
## [1] 0
```

```r
sum(is.na(room_3$subzone))
```

```
## [1] 0
```

```r
#generate list of XTS objects
reslist_room_3 <- list()
for(i in 1:length(bird_ids_room_3)){

  res <- sep_bird_id_xts(samp_name="room_3",cage_obj =room_3, samp_id = bird_ids_room_3[i])
  reslist_room_3[[i]] <- res
}



cutoff_room_3 <- data.frame()
for(i in 1:length(reslist_room_3)){
  id <- reslist_room_3[[i]]$ID
  top <- index(head(reslist_room_3[[i]]$xts_obj))[1]
  bottom <- index(tail(reslist_room_3[[i]]$xts_obj))[6]
  rec <- cbind(id,as.character(top),as.character(bottom))
  cutoff_room_3 <- rbind(cutoff_room_3,rec)

}

#head
cutoff_room_3[order(cutoff_room_3$V2),]
```

```
##      id                  V2                  V3
## 5  6905 2021-03-09 17:36:43 2021-05-07 18:32:55
## 37 9013 2021-03-09 17:37:22 2021-05-07 17:43:02
## 28 6976 2021-03-09 17:38:09 2021-05-07 18:26:58
## 24 6967 2021-03-09 17:39:03 2021-05-07 18:23:46
## 16 6938 2021-03-09 17:39:44 2021-05-07 15:07:33
## 36 9010 2021-03-09 17:41:07 2021-05-07 17:48:48
## 7  6912 2021-03-09 17:41:11 2021-03-22 17:17:22
## 14 6933 2021-03-09 17:41:23 2021-05-07 18:29:38
## 32 6992 2021-03-09 17:41:59 2021-05-07 18:30:30
## 34 9004 2021-03-09 17:41:59 2021-05-07 18:03:26
## 39 9018 2021-03-09 17:42:16 2021-05-07 16:52:31
## 10 6921 2021-03-09 17:44:40 2021-05-07 18:27:20
## 13 6927 2021-03-09 17:45:10 2021-05-07 17:58:07
## 30 6980 2021-03-09 17:45:59 2021-05-07 18:28:57
## 29 6977 2021-03-09 17:46:06 2021-05-07 18:32:59
## 38 9016 2021-03-09 17:46:07 2021-05-07 17:59:10
## 6  6907 2021-03-09 17:46:14 2021-05-07 18:26:24
## 3  6883 2021-03-09 17:46:24 2021-05-07 17:56:49
## 1  6736 2021-03-09 17:47:01 2021-05-07 18:23:11
## 33 6998 2021-03-09 17:47:13 2021-05-07 18:30:45
## 35 9008 2021-03-09 17:47:34 2021-03-16 17:07:36
## 4  6902 2021-03-09 17:49:09 2021-05-07 17:52:55
## 20 6948 2021-03-09 17:51:23 2021-04-20 19:19:49
## 15 6934 2021-03-09 17:51:36 2021-05-07 18:04:36
## 12 6924 2021-03-09 17:53:31 2021-05-07 18:30:39
## 31 6987 2021-03-09 17:54:42 2021-04-08 00:39:55
## 19 6947 2021-03-09 17:54:59 2021-05-07 18:29:42
## 21 6953 2021-03-09 18:01:56 2021-04-21 16:54:50
## 9  6918 2021-03-09 18:06:23 2021-05-07 18:23:51
## 2  6879 2021-03-09 18:08:33 2021-04-25 15:49:17
## 11 6922 2021-03-09 18:11:28 2021-05-07 18:31:55
## 17 6943 2021-03-09 18:28:23 2021-05-07 18:07:49
## 27 6972 2021-03-09 18:30:08 2021-05-07 17:49:16
## 26 6969 2021-03-09 18:30:21 2021-05-07 18:18:57
## 8  6915 2021-03-09 18:47:36 2021-03-11 23:48:40
## 23 6963 2021-03-09 19:10:20 2021-05-07 17:58:03
## 22 6959 2021-03-09 19:27:40 2021-05-07 17:59:52
## 18 6944 2021-03-09 19:58:52 2021-05-07 16:44:52
## 25 6968 2021-03-25 19:03:00 2021-05-07 18:32:22
```

```
#tail
cutoff_room_3[order(cutoff_room_3$V3),]
```

```
##      id                  V2                  V3
## 8  6915 2021-03-09 18:47:36 2021-03-11 23:48:40
## 35 9008 2021-03-09 17:47:34 2021-03-16 17:07:36
## 7  6912 2021-03-09 17:41:11 2021-03-22 17:17:22
## 31 6987 2021-03-09 17:54:42 2021-04-08 00:39:55
## 20 6948 2021-03-09 17:51:23 2021-04-20 19:19:49
## 21 6953 2021-03-09 18:01:56 2021-04-21 16:54:50
## 2  6879 2021-03-09 18:08:33 2021-04-25 15:49:17
## 16 6938 2021-03-09 17:39:44 2021-05-07 15:07:33
## 18 6944 2021-03-09 19:58:52 2021-05-07 16:44:52
## 39 9018 2021-03-09 17:42:16 2021-05-07 16:52:31
## 37 9013 2021-03-09 17:37:22 2021-05-07 17:43:02
## 36 9010 2021-03-09 17:41:07 2021-05-07 17:48:48
## 27 6972 2021-03-09 18:30:08 2021-05-07 17:49:16
## 4  6902 2021-03-09 17:49:09 2021-05-07 17:52:55
## 3  6883 2021-03-09 17:46:24 2021-05-07 17:56:49
## 23 6963 2021-03-09 19:10:20 2021-05-07 17:58:03
## 13 6927 2021-03-09 17:45:10 2021-05-07 17:58:07
## 38 9016 2021-03-09 17:46:07 2021-05-07 17:59:10
## 22 6959 2021-03-09 19:27:40 2021-05-07 17:59:52
## 34 9004 2021-03-09 17:41:59 2021-05-07 18:03:26
## 15 6934 2021-03-09 17:51:36 2021-05-07 18:04:36
## 17 6943 2021-03-09 18:28:23 2021-05-07 18:07:49
## 26 6969 2021-03-09 18:30:21 2021-05-07 18:18:57
## 1  6736 2021-03-09 17:47:01 2021-05-07 18:23:11
## 24 6967 2021-03-09 17:39:03 2021-05-07 18:23:46
## 9  6918 2021-03-09 18:06:23 2021-05-07 18:23:51
## 6  6907 2021-03-09 17:46:14 2021-05-07 18:26:24
## 28 6976 2021-03-09 17:38:09 2021-05-07 18:26:58
## 10 6921 2021-03-09 17:44:40 2021-05-07 18:27:20
## 30 6980 2021-03-09 17:45:59 2021-05-07 18:28:57
## 14 6933 2021-03-09 17:41:23 2021-05-07 18:29:38
## 19 6947 2021-03-09 17:54:59 2021-05-07 18:29:42
## 32 6992 2021-03-09 17:41:59 2021-05-07 18:30:30
## 12 6924 2021-03-09 17:53:31 2021-05-07 18:30:39
## 33 6998 2021-03-09 17:47:13 2021-05-07 18:30:45
## 11 6922 2021-03-09 18:11:28 2021-05-07 18:31:55
## 25 6968 2021-03-25 19:03:00 2021-05-07 18:32:22
## 5  6905 2021-03-09 17:36:43 2021-05-07 18:32:55
## 29 6977 2021-03-09 17:46:06 2021-05-07 18:32:59
```

```r
#remove <- c("6998")
remove <- c("6915","9008","6912","6987","6948","6953","6879","6968","9029")
bird_ids_room_3_new <- bird_ids_room_3 [! bird_ids_room_3  %in% remove ]
#generate list of the first 10 truncated XTS objects
reslist2_room_3 <- list()
for(i in 1:length(bird_ids_room_3_new)){
  res <- sep_bird_id_period(samp_name="room 3",cage_obj=room_3, samp_id =bird_ids_room_3_new[i],cutoff =
  #print(res)

  reslist2_room_3[[i]] <- res
}



#find wholly non-na daily timeset, to be able to feed into the transcalc
big_table_room_3 <- reslist2_room_3[[1]]$daily_obj
for(i in 2:length(reslist2_room_3)){
  current <- reslist2_room_3[[i]]$daily_obj
  big_table_room_3 <- cbind(big_table_room_3,current)
}
print("how many NA's are in big_table_room_3")
```

```
## [1] "how many NA’s are in big_table_room_3"
```

```r
print(sum(is.na(big_table_room_3)))
```

```
## [1] 0
```

Import Room 8

```
library(xts)

room_8 <- read.csv("../data/DK20-03-RFID-r8-febmay-080423.csv")

bird_ids_room_8 <- na.trim(unique(room_8$tagname))
bird_ids_room_8 <- sort(bird_ids_room_8)

room_8["DateTime"] <- as.POSIXct(room_8$access, origin="1970-01-01", tz="GMT")

print("what makes up subzone col")
```

```
## [1] "what makes up subzone col"
```

```
unique(room_8$subzone)
```

```
## [1] "Bottom" "Top"    "middle" "test"
```

```
room_8$subzone[room_8$subzone == "Bottom"] <- "bottom"
room_8$subzone[room_8$subzone == "middle"] <- "middle"
room_8$subzone[room_8$subzone == "Top"] <- "top"
room_8 <-subset(room_8, subzone!="test")

print("what makes up subzone col")
```

```
## [1] "what makes up subzone col"
```

```
unique(room_8$subzone)
```

```
## [1] "bottom" "top"    "middle"
```

```
print("how many NA's in Datetime and Subzone")
```

```
## [1] "how many NA’s in Datetime and Subzone"
```

```
sum(is.na(room_8$DateTime))
```

```
## [1] 0
```

```
sum(is.na(room_8$subzone))
```

```
## [1] 0
```

```
#generate list of XTS objects
reslist_room_8 <- list()
for(i in 1:length(bird_ids_room_8)){

  res <- sep_bird_id_xts(samp_name="room 8",cage_obj =room_8, samp_id = bird_ids_room_8[i])
  reslist_room_8[[i]] <- res
}

cutoff_room_8 <- data.frame()
for(i in 1:length(reslist_room_8)){
  id <- reslist_room_8[[i]]$ID
  top <- index(head(reslist_room_8[[i]]$xts_obj))[1]
  bottom <- index(tail(reslist_room_8[[i]]$xts_obj))[6]
  rec <- cbind(id,as.character(top),as.character(bottom))
  cutoff_room_8 <- rbind(cutoff_room_8,rec)
}

#head
cutoff_room_8[order(cutoff_room_8$V2),]
```

```
##      id                  V2                  V3
## 3  6917 2021-02-01 18:40:31 2021-05-07 17:34:09
## 21 6974 2021-02-01 18:40:53 2021-04-22 17:46:37
## 14 6950 2021-02-01 18:40:56 2021-05-07 17:16:49
## 32 9009 2021-02-01 18:41:02 2021-03-25 16:35:40
## 23 6982 2021-02-01 19:34:11 2021-05-07 18:25:43
## 25 6985 2021-02-01 19:34:40 2021-05-07 18:37:40
## 11 6941 2021-02-01 19:37:18 2021-05-07 18:18:18
## 15 6951 2021-02-01 19:42:48 2021-05-07 17:58:57
## 19 6970 2021-02-01 19:44:55 2021-05-07 17:32:35
## 36 9017 2021-02-01 19:47:21 2021-05-07 18:08:11
## 37 9020 2021-02-01 19:51:24 2021-05-07 18:20:07
## 28 6995 2021-02-01 19:57:41 2021-03-10 18:03:22
## 31 7000 2021-02-01 20:23:46 2021-05-07 18:20:54
## 1  6886 2021-02-01 20:24:45 2021-05-07 18:35:21
## 20 6973 2021-02-01 20:26:57 2021-05-07 18:39:44
## 7  6931 2021-02-01 20:42:48 2021-05-07 18:03:13
## 9  6939 2021-02-01 20:57:04 2021-05-07 15:43:27
## 16 6957 2021-02-01 20:58:46 2021-05-07 18:03:16
## 22 6979 2021-02-01 20:58:50 2021-05-07 18:36:25
## 29 6996 2021-02-01 21:03:58 2021-05-06 23:14:55
## 2  6913 2021-02-01 21:06:14 2021-05-07 17:50:11
## 6  6929 2021-02-01 22:51:08 2021-05-07 18:20:32
## 33 9011 2021-02-01 23:18:34 2021-03-24 17:15:07
## 17 6961 2021-02-02 00:27:26 2021-05-07 17:30:56
## 34 9012 2021-02-02 00:38:28 2021-05-07 18:25:42
## 18 6964 2021-02-02 10:09:29 2021-05-07 17:30:38
## 26 6993 2021-02-02 10:37:37 2021-05-07 18:18:15
## 40 9025 2021-02-02 11:24:04 2021-05-07 16:27:15
## 27 6994 2021-02-04 19:04:25 2021-05-07 16:46:27
## 35 9014 2021-02-04 19:09:15 2021-05-07 18:43:18
## 10 6940 2021-02-04 19:15:57 2021-05-07 17:32:58
## 12 6945 2021-02-04 19:32:22 2021-05-07 16:12:00
## 30 6997 2021-02-04 19:55:02 2021-05-07 18:09:32
## 4  6923 2021-02-04 20:12:30 2021-05-07 18:38:05
## 39 9023 2021-02-04 20:45:34 2021-05-07 18:01:57
## 38 9022 2021-02-04 20:57:44 2021-05-07 18:21:35
## 5  6928 2021-02-04 21:08:19 2021-05-07 14:15:41
## 24 6984 2021-02-04 21:34:56 2021-05-07 18:33:57
## 13 6949 2021-02-05 03:10:07 2021-05-07 17:32:09
## 8  6936 2021-02-05 11:39:55 2021-05-07 18:10:48
```

```
#tail
cutoff_room_8[order(cutoff_room_8$V3),]
```

```
##      id                  V2                  V3
## 28 6995 2021-02-01 19:57:41 2021-03-10 18:03:22
## 33 9011 2021-02-01 23:18:34 2021-03-24 17:15:07
## 32 9009 2021-02-01 18:41:02 2021-03-25 16:35:40
## 21 6974 2021-02-01 18:40:53 2021-04-22 17:46:37
## 29 6996 2021-02-01 21:03:58 2021-05-06 23:14:55
## 5  6928 2021-02-04 21:08:19 2021-05-07 14:15:41
## 9  6939 2021-02-01 20:57:04 2021-05-07 15:43:27
## 12 6945 2021-02-04 19:32:22 2021-05-07 16:12:00
## 40 9025 2021-02-02 11:24:04 2021-05-07 16:27:15
## 27 6994 2021-02-04 19:04:25 2021-05-07 16:46:27
## 14 6950 2021-02-01 18:40:56 2021-05-07 17:16:49
## 18 6964 2021-02-02 10:09:29 2021-05-07 17:30:38
## 17 6961 2021-02-02 00:27:26 2021-05-07 17:30:56
## 13 6949 2021-02-05 03:10:07 2021-05-07 17:32:09
## 19 6970 2021-02-01 19:44:55 2021-05-07 17:32:35
## 10 6940 2021-02-04 19:15:57 2021-05-07 17:32:58
## 3  6917 2021-02-01 18:40:31 2021-05-07 17:34:09
## 2  6913 2021-02-01 21:06:14 2021-05-07 17:50:11
## 15 6951 2021-02-01 19:42:48 2021-05-07 17:58:57
## 39 9023 2021-02-04 20:45:34 2021-05-07 18:01:57
## 7  6931 2021-02-01 20:42:48 2021-05-07 18:03:13
## 16 6957 2021-02-01 20:58:46 2021-05-07 18:03:16
## 36 9017 2021-02-01 19:47:21 2021-05-07 18:08:11
## 30 6997 2021-02-04 19:55:02 2021-05-07 18:09:32
## 8  6936 2021-02-05 11:39:55 2021-05-07 18:10:48
## 26 6993 2021-02-02 10:37:37 2021-05-07 18:18:15
## 11 6941 2021-02-01 19:37:18 2021-05-07 18:18:18
## 37 9020 2021-02-01 19:51:24 2021-05-07 18:20:07
## 6  6929 2021-02-01 22:51:08 2021-05-07 18:20:32
## 31 7000 2021-02-01 20:23:46 2021-05-07 18:20:54
## 38 9022 2021-02-04 20:57:44 2021-05-07 18:21:35
## 34 9012 2021-02-02 00:38:28 2021-05-07 18:25:42
## 23 6982 2021-02-01 19:34:11 2021-05-07 18:25:43
## 24 6984 2021-02-04 21:34:56 2021-05-07 18:33:57
## 1  6886 2021-02-01 20:24:45 2021-05-07 18:35:21
## 22 6979 2021-02-01 20:58:50 2021-05-07 18:36:25
## 25 6985 2021-02-01 19:34:40 2021-05-07 18:37:40
## 4  6923 2021-02-04 20:12:30 2021-05-07 18:38:05
## 20 6973 2021-02-01 20:26:57 2021-05-07 18:39:44
## 35 9014 2021-02-04 19:09:15 2021-05-07 18:43:18
```

```
remove <-c("6949","6936","6974","9009","9011","6995")
bird_ids_room_8_new <- bird_ids_room_8 [! bird_ids_room_8  %in% remove ]
#generate list of the first 10 truncated XTS objects
reslist2_room_8 <- list()
for(i in 1:length(bird_ids_room_8_new)){
  res <- sep_bird_id_period(samp_name="room 8",cage_obj =room_8, samp_id = bird_ids_room_8_new[i],cutof
  reslist2_room_8[[i]] <- res
}

#find wholly non-na daily timeset, to be able to feed into the transcalc
big_table_room_8 <- reslist2_room_8[[1]]$daily_obj
for(i in 2:length(reslist2_room_8)){
  current <- reslist2_room_8[[i]]$daily_obj
  big_table_room_8 <- cbind(big_table_room_8,current)
}
print("how many NA's in big_table_room_8")
```

```
## [1] "how many NA's in big_table_room_8"
```

```
print(sum(is.na(big_table_room_8)))
```

```
## [1] 0
```

Import Room 11

```
library(xts)

room_11 <- read.csv("../data/DK20-03-RFID-R11-febmay-080423.csv")

bird_ids_room_11 <- na.trim(unique(room_11$tagname))
bird_ids_room_11 <- sort(bird_ids_room_11)

room_11 <- room_11[order(room_11$access),]

room_11["DateTime"] <- as.POSIXct(room_11$access, origin="1970-01-01", tz="GMT")

print("what composes the subzone column")
```

```
## [1] "what composes the subzone column"
```

```
unique(room_11$subzone)
```

```
## [1] "B"      "M"      "T"      "bottom" "middle" "top"    ""
```

```
room_11$subzone[room_11$subzone == "M"] <- "middle"
room_11$subzone[room_11$subzone == "B"] <- "bottom"
room_11$subzone[room_11$subzone == "T"] <- "top"

print("what composes the subzone column")
```

```
## [1] "what composes the subzone column"
```

```
unique(room_11$subzone)
```

```
## [1] "bottom" "middle" "top"    ""
```

```
print("how many NA's in the DateTime col")
```

```
## [1] "how many NA's in the DateTime col"
```

```
sum(is.na(room_11$DateTime))
```

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

```
room_11 <- room_11[!is.na(room_11$DateTime),]

print("how many NA's in the Datetime col")
```

```
## [1] "how many NA's in the Datetime col"
```

```
sum(is.na(room_11$DateTime))
```

```
## [1] 0
```

```
#generate list of XTS objects
reslist_room_11 <- list()
for(i in 1:length(bird_ids_room_11)){
  res <- sep_bird_id_xts(samp_name="room 11",cage_obj =room_11, samp_id = bird_ids_room_11[i])
  reslist_room_11[[i]] <- res
}

# index 10 is an error right now, how can we fix na value

remove <- c("6888")

cutoff_room_11 <- data.frame()
for(i in 1:length(reslist_room_11)){
  print(i)
  id <- reslist_room_11[[i]]$ID
  if(id %in% remove){
  } else{
    top <- index(head(reslist_room_11[[i]]$xts_obj))[1]
    bottom <- index(tail(reslist_room_11[[i]]$xts_obj))[6]
    rec <- cbind(id,as.character(top),as.character(bottom))
    cutoff_room_11 <- rbind(cutoff_room_11,rec)
  }
}
```

```
## [1] 1
## [1] 2
## [1] 3
## [1] 4
## [1] 5
## [1] 6
## [1] 7
## [1] 8
## [1] 9
## [1] 10
## [1] 11
## [1] 12
## [1] 13
## [1] 14
## [1] 15
## [1] 16
## [1] 17
## [1] 18
## [1] 19
## [1] 20
## [1] 21
## [1] 22
## [1] 23
## [1] 24
## [1] 25
## [1] 26
## [1] 27
## [1] 28
## [1] 29
## [1] 30
## [1] 31
## [1] 32
## [1] 33
## [1] 34
## [1] 35
## [1] 36
## [1] 37
## [1] 38
## [1] 39
## [1] 40
## [1] 41
## [1] 42
## [1] 43
## [1] 44
## [1] 45
## [1] 46
## [1] 47
## [1] 48
## [1] 49
## [1] 50
## [1] 51
## [1] 52
## [1] 53
## [1] 54
## [1] 55
## [1] 56
## [1] 57
## [1] 58
## [1] 59
## [1] 60
## [1] 61
## [1] 62
## [1] 63
```

```
#head
cutoff_room_11[order(cutoff_room_11$V2),]
```

```
##      id                  V2                  V3
## 4  6880 2020-11-16 20:39:27 2021-01-31 16:29:25
## 5  6881 2020-11-16 20:39:39 2021-02-06 00:19:43
## 13 6893 2020-12-01 22:27:35 2021-02-18 17:20:46
## 12 6892 2020-12-01 22:27:39 2021-02-18 17:02:03
## 14 6894 2020-12-01 22:27:44 2021-02-18 17:05:02
## 2  6862 2020-12-02 20:27:52 2021-01-22 10:26:50
## 18 6898 2020-12-02 20:29:06 2021-01-25 19:25:53
## 6  6882 2020-12-02 20:32:22 2021-02-18 17:32:55
## 19 6899 2020-12-02 20:32:59 2021-01-07 22:17:59
## 9  6887 2020-12-02 20:33:59 2021-02-18 16:33:33
## 16 6896 2020-12-02 20:34:35 2021-01-11 17:10:19
## 10 6889 2020-12-02 20:35:57 2021-02-07 00:31:34
## 15 6895 2020-12-02 20:36:49 2021-02-18 17:00:35
## 8  6885 2020-12-02 20:39:20 2021-02-04 15:20:14
## 20 6900 2020-12-02 20:40:06 2021-02-18 00:47:25
## 17 6897 2020-12-02 20:40:42 2021-02-18 16:45:33
## 11 6891 2020-12-02 20:42:56 2021-02-18 13:38:37
## 7  6884 2020-12-02 20:43:38 2021-01-18 19:17:24
## 35 9007 2021-02-01 17:46:07 2021-05-07 17:21:55
## 23 6916 2021-02-01 17:47:23 2021-05-07 17:55:43
## 33 9003 2021-02-01 17:48:13 2021-05-07 16:44:55
## 22 6906 2021-02-01 17:48:52 2021-02-18 17:05:36
## 24 6920 2021-02-01 17:49:34 2021-02-18 16:58:55
## 1  6853 2021-02-01 17:57:17 2021-05-07 17:08:45
## 28 6965 2021-02-01 17:57:24 2021-05-07 18:36:29
## 27 6954 2021-02-01 17:57:48 2021-04-29 18:03:39
## 21 6904 2021-02-01 17:58:13 2021-05-07 18:45:20
## 38 9027 2021-02-01 17:58:17 2021-05-07 17:19:44
## 34 9006 2021-02-01 18:01:13 2021-05-07 18:35:46
## 36 9015 2021-02-01 18:01:34 2021-05-07 18:44:14
## 30 6991 2021-02-01 18:05:25 2021-04-08 18:20:58
## 31 6999 2021-02-01 18:06:15 2021-05-07 16:36:01
## 32 9002 2021-02-01 18:06:40 2021-05-07 18:44:58
## 39 9030 2021-02-01 18:08:18 2021-05-07 18:38:28
## 37 9023 2021-02-01 18:20:14 2021-05-07 18:32:54
## 29 6990 2021-02-01 18:45:59 2021-05-07 18:04:17
## 26 6932 2021-02-01 19:08:26 2021-05-07 17:43:18
## 25 6930 2021-02-01 19:39:37 2021-02-18 16:45:31
## 3  6868 2021-02-06 00:34:22                <NA>
## 41 9033 2021-02-18 18:36:30 2021-05-07 17:18:29
## 50 9047 2021-02-18 18:36:54 2021-05-07 18:29:39
## 47 9044 2021-02-18 18:37:29 2021-05-07 18:33:27
## 45 9042 2021-02-18 18:38:02 2021-05-07 18:27:14
## 52 9049 2021-02-18 18:38:17 2021-05-07 16:46:17
## 49 9046 2021-02-18 18:39:38 2021-05-07 16:25:36
## 58 9056 2021-02-18 18:40:19 2021-05-07 18:41:36
## 48 9045 2021-02-18 18:40:38 2021-05-07 17:46:32
## 46 9043 2021-02-18 18:40:44 2021-05-07 18:23:28
## 59 9058 2021-02-18 18:42:14 2021-05-07 16:36:28
## 43 9038 2021-02-18 18:42:37 2021-05-07 17:32:46
## 60 9059 2021-02-18 18:43:17 2021-04-19 17:20:44
## 53 9050 2021-02-18 18:46:36 2021-05-07 15:47:23
## 44 9041 2021-02-18 18:54:54 2021-05-07 18:46:13
## 51 9048 2021-02-18 18:57:28 2021-05-07 17:49:59
## 61 9060 2021-02-18 19:01:04 2021-04-19 17:13:42
## 56 9054 2021-02-18 19:05:49 2021-05-07 18:44:58
## 42 9035 2021-02-18 19:09:52 2021-05-07 18:12:04
## 40 9032 2021-02-18 19:11:12 2021-05-07 18:34:14
## 57 9055 2021-02-18 19:28:25 2021-05-07 18:10:18
## 62 9061 2021-02-18 21:19:01 2021-05-07 18:02:34
## 55 9053 2021-02-18 21:38:39 2021-05-07 17:47:59
## 54 9052 2021-02-18 23:15:10 2021-05-07 16:23:47
```

```
#tail
cutoff_room_11[order(cutoff_room_11$V3),]
```

```
##      id                  V2                  V3
## 19 6899 2020-12-02 20:32:59 2021-01-07 22:17:59
## 16 6896 2020-12-02 20:34:35 2021-01-11 17:10:19
## 7  6884 2020-12-02 20:43:38 2021-01-18 19:17:24
## 2  6862 2020-12-02 20:27:52 2021-01-22 10:26:50
## 18 6898 2020-12-02 20:29:06 2021-01-25 19:25:53
## 4  6880 2020-11-16 20:39:27 2021-01-31 16:29:25
## 8  6885 2020-12-02 20:39:20 2021-02-04 15:20:14
## 5  6881 2020-11-16 20:39:39 2021-02-06 00:19:43
## 10 6889 2020-12-02 20:35:57 2021-02-07 00:31:34
## 20 6900 2020-12-02 20:40:06 2021-02-18 00:47:25
## 11 6891 2020-12-02 20:42:56 2021-02-18 13:38:37
## 9  6887 2020-12-02 20:33:59 2021-02-18 16:33:33
## 25 6930 2021-02-01 19:39:37 2021-02-18 16:45:31
## 17 6897 2020-12-02 20:40:42 2021-02-18 16:45:33
## 24 6920 2021-02-01 17:49:34 2021-02-18 16:58:55
## 15 6895 2020-12-02 20:36:49 2021-02-18 17:00:35
## 12 6892 2020-12-01 22:27:39 2021-02-18 17:02:03
## 14 6894 2020-12-01 22:27:44 2021-02-18 17:05:02
## 22 6906 2021-02-01 17:48:52 2021-02-18 17:05:36
## 13 6893 2020-12-01 22:27:35 2021-02-18 17:20:46
## 6  6882 2020-12-02 20:32:22 2021-02-18 17:32:55
## 30 6991 2021-02-01 18:05:25 2021-04-08 18:20:58
## 61 9060 2021-02-18 19:01:04 2021-04-19 17:13:42
## 60 9059 2021-02-18 18:43:17 2021-04-19 17:20:44
## 27 6954 2021-02-01 17:57:48 2021-04-29 18:03:39
## 53 9050 2021-02-18 18:46:36 2021-05-07 15:47:23
## 54 9052 2021-02-18 23:15:10 2021-05-07 16:23:47
## 49 9046 2021-02-18 18:39:38 2021-05-07 16:25:36
## 31 6999 2021-02-01 18:06:15 2021-05-07 16:36:01
## 59 9058 2021-02-18 18:42:14 2021-05-07 16:36:28
## 33 9003 2021-02-01 17:48:13 2021-05-07 16:44:55
## 52 9049 2021-02-18 18:38:17 2021-05-07 16:46:17
## 1  6853 2021-02-01 17:57:17 2021-05-07 17:08:45
## 41 9033 2021-02-18 18:36:30 2021-05-07 17:18:29
## 38 9027 2021-02-01 17:58:17 2021-05-07 17:19:44
## 35 9007 2021-02-01 17:46:07 2021-05-07 17:21:55
## 43 9038 2021-02-18 18:42:37 2021-05-07 17:32:46
## 26 6932 2021-02-01 19:08:26 2021-05-07 17:43:18
## 48 9045 2021-02-18 18:40:38 2021-05-07 17:46:32
## 55 9053 2021-02-18 21:38:39 2021-05-07 17:47:59
## 51 9048 2021-02-18 18:57:28 2021-05-07 17:49:59
## 23 6916 2021-02-01 17:47:23 2021-05-07 17:55:43
## 62 9061 2021-02-18 21:19:01 2021-05-07 18:02:34
## 29 6990 2021-02-01 18:45:59 2021-05-07 18:04:17
## 57 9055 2021-02-18 19:28:25 2021-05-07 18:10:18
## 42 9035 2021-02-18 19:09:52 2021-05-07 18:12:04
## 46 9043 2021-02-18 18:40:44 2021-05-07 18:23:28
## 45 9042 2021-02-18 18:38:02 2021-05-07 18:27:14
## 50 9047 2021-02-18 18:36:54 2021-05-07 18:29:39
## 37 9023 2021-02-01 18:20:14 2021-05-07 18:32:54
## 47 9044 2021-02-18 18:37:29 2021-05-07 18:33:27
## 40 9032 2021-02-18 19:11:12 2021-05-07 18:34:14
## 34 9006 2021-02-01 18:01:13 2021-05-07 18:35:46
## 28 6965 2021-02-01 17:57:24 2021-05-07 18:36:29
## 39 9030 2021-02-01 18:08:18 2021-05-07 18:38:28
## 58 9056 2021-02-18 18:40:19 2021-05-07 18:41:36
## 36 9015 2021-02-01 18:01:34 2021-05-07 18:44:14
## 32 9002 2021-02-01 18:06:40 2021-05-07 18:44:58
## 56 9054 2021-02-18 19:05:49 2021-05-07 18:44:58
## 21 6904 2021-02-01 17:58:13 2021-05-07 18:45:20
## 44 9041 2021-02-18 18:54:54 2021-05-07 18:46:13
## 3  6868 2021-02-06 00:34:22                <NA>
```

```
remove <- c("6888","6868","6880","6881","6893","6892", "6894","6862","6898","6882","6899","6887","6896"
bird_ids_room_11_new <- bird_ids_room_11 [! bird_ids_room_11  %in% remove ]
#generate list of the first 10 truncated XTS objects
reslist2_room_11 <- list()
for(i in 1:length(bird_ids_room_11_new)){
  res <- sep_bird_id_period(samp_name="room 11",cage_obj =room_11, samp_id = bird_ids_room_11_new[i], cu
  reslist2_room_11[[i]] <- res
}


#find wholly non-na daily timeset, to be able to feed into the transcalc
big_table_room_11 <- reslist2_room_11[[1]]$daily_obj
for(i in 2:length(reslist2_room_11)){
  current <- reslist2_room_11[[i]]$daily_obj
  big_table_room_11 <- cbind(big_table_room_11,current)
}
print("Na's in big_table_room_11")
```

```
## [1] "Na's in big_table_room_11"
```

```
print(sum(is.na(big_table_room_11)))
```

```
## [1] 0
```

```
print("top of big_table_room_11")
```

```
## [1] "top of big_table_room_11"
```

```
print(head(big_table_room_11))
```

```
##            subzone  subzone.1 subzone.2 subzone.3 subzone.4 subzone.5 subzone.6
## 2021-03-09 "top"    "top"     "bottom"  "top"     "middle"  "bottom"  "middle"
## 2021-03-10 "top"    "top"     "bottom"  "top"     "top"     "bottom"  "top"
## 2021-03-11 "bottom" "top"     "middle"  "top"     "middle"  "bottom"  "middle"
## 2021-03-12 "top"    "middle"  "bottom"  "top"     "bottom"  "bottom"  "top"
## 2021-03-13 "top"    "top"     "bottom"  "bottom"  "middle"  "bottom"  "top"
## 2021-03-14 "top"    "top"     "bottom"  "top"     "middle"  "bottom"  "top"
##            subzone.7 subzone.8 subzone.9 subzone.10 subzone.11 subzone.12
## 2021-03-09 "middle"  "top"     "top"     "bottom"   "middle"   "top"
## 2021-03-10 "middle"  "top"     "top"     "bottom"   "middle"   "top"
## 2021-03-11 "middle"  "top"     "top"     "bottom"   "middle"   "bottom"
## 2021-03-12 "middle"  "top"     "top"     "bottom"   "middle"   "top"
## 2021-03-13 "middle"  "top"     "top"     "bottom"   "bottom"   "middle"
## 2021-03-14 "middle"  "top"     "bottom"  "bottom"   "middle"   "bottom"
##            subzone.13 subzone.14 subzone.15 subzone.16 subzone.17 subzone.18
## 2021-03-09 "bottom"   "top"      "bottom"   "middle"   "middle"   "middle"
## 2021-03-10 "top"      "top"      "bottom"   "top"      "top"      "top"
## 2021-03-11 "top"      "top"      "bottom"   "top"      "middle"   "middle"
## 2021-03-12 "top"      "top"      "bottom"   "middle"   "top"      "middle"
## 2021-03-13 "bottom"   "top"      "bottom"   "bottom"   "top"      "top"
## 2021-03-14 "top"      "top"      "bottom"   "bottom"   "top"      "middle"
##            subzone.19 subzone.20 subzone.21 subzone.22 subzone.23 subzone.24
## 2021-03-09 "top"      "bottom"   "top"      "top"      "middle"   "top"
## 2021-03-10 "middle"   "bottom"   "top"      "top"      "middle"   "bottom"
## 2021-03-11 "middle"   "bottom"   "top"      "top"      "middle"   "bottom"
## 2021-03-12 "middle"   "bottom"   "top"      "top"      "middle"   "middle"
## 2021-03-13 "middle"   "top"      "bottom"   "top"      "middle"   "bottom"
## 2021-03-14 "middle"   "bottom"   "bottom"   "top"      "middle"   "bottom"
##            subzone.25 subzone.26 subzone.27 subzone.28 subzone.29 subzone.30
## 2021-03-09 "bottom"   "bottom"   "middle"   "top"      "bottom"   "bottom"
## 2021-03-10 "bottom"   "middle"   "top"      "top"      "bottom"   "bottom"
## 2021-03-11 "bottom"   "middle"   "top"      "top"      "bottom"   "bottom"
## 2021-03-12 "bottom"   "top"      "top"      "middle"   "bottom"   "bottom"
## 2021-03-13 "bottom"   "bottom"   "top"      "top"      "bottom"   "bottom"
## 2021-03-14 "bottom"   "middle"   "bottom"   "top"      "bottom"   "bottom"
##            subzone.31 subzone.32 subzone.33 subzone.34 subzone.35
## 2021-03-09 "top"      "bottom"   "middle"   "bottom"   "middle"
## 2021-03-10 "top"      "bottom"   "middle"   "bottom"   "middle"
## 2021-03-11 "top"      "bottom"   "middle"   "middle"   "top"
## 2021-03-12 "top"      "bottom"   "middle"   "bottom"   "top"
## 2021-03-13 "top"      "bottom"   "middle"   "middle"   "top"
## 2021-03-14 "top"      "bottom"   "middle"   "middle"   "middle"
```

Intra-Bird Analysis With All Rooms

```
trans_reslist_room_2 <- list()

for(i in 1:length(reslist2_room_2)){

  daily_trans_table <- calc_trans_period(reslist2_room_2[[i]]$ID,reslist2_room_2[[i]]$daily_obj,reslist2
  weekly_trans_table <- calc_trans_period(reslist2_room_2[[i]]$ID,reslist2_room_2[[i]]$weekly_obj,reslis
  monthly_trans_table <- calc_trans_period(reslist2_room_2[[i]]$ID,reslist2_room_2[[i]]$monthly_obj,resl

  result <- list("ID"=reslist2_room_2[[i]]$ID,"daily"=daily_trans_table, "weekly"=weekly_trans_table,"m
  trans_reslist_room_2[[i]] <- result
}

room_2_daily <- data.frame()
room_2_weekly <- data.frame()
room_2_monthly <- data.frame()
#TODO select sample as opposed to ID for intra bird comparison (alpha eq)
for(item in trans_reslist_room_2){
  room_2_daily <- rbind(room_2_daily, item$daily[c("sample","bottom","mid","top","total")])
  room_2_weekly <- rbind(room_2_weekly, item$weekly[c("sample","bottom","mid","top","total")])
  room_2_monthly <- rbind(room_2_monthly, item$monthly[c("sample","bottom","mid","top","total")])
}

# check to make sure only one bird is selected
print("check that the id and RFID tag are unique for each item entry")
```

```
## [1] "check that the id and RFID tag are unique for each item entry"
```

```
#this is following up with the issues of the intra-bird comparisons
for(item in trans_reslist_room_2){
  print(paste(unique(item$raw$Tagnumber)," ",unique(item$raw$tagname)))
  #print(item)
}
```

```
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-14-88   6855"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-11-64   6860"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-09-07   6872"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-11-97   6877"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-18-23   6890"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-18-73   6901"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-18-49   6903"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-09-10   6908"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-12-60   6909"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-11-12   6911"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-13-29   6919"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-03-83   6925"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-09-29   6926"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-05-21   6935"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-09-45   6937"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-04-70   6942"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-14-25   6946"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-12-80   6952"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-16-07   6955"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-12-67   6956"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-12-18   6958"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-10-61   6960"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-12-90   6962"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-17-97   6971"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-19-05   6975"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-11-50   6978"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-06-95   6981"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-17-83   6983"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-02-61   6986"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-00-28   6988"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-04-58   6989"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-02-91   9001"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-08-73   9005"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-04-14   9019"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-15-81   9021"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-15-03   9024"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-06-41   9026"
```

```
trans_reslist_room_3 <- list()
for(i in 1:length(reslist2_room_3)){

  daily_trans_table <- calc_trans_period(reslist2_room_3[[i]]$ID,reslist2_room_3[[i]]$daily_obj,reslist2
  weekly_trans_table <- calc_trans_period(reslist2_room_3[[i]]$ID,reslist2_room_3[[i]]$weekly_obj,reslis
  monthly_trans_table <- calc_trans_period(reslist2_room_3[[i]]$ID,reslist2_room_3[[i]]$monthly_obj,resl

  result <- list("ID"=reslist2_room_3[[i]]$ID,"daily"=daily_trans_table, "weekly"=weekly_trans_table,"mo
  trans_reslist_room_3[[i]] <- result
}

room_3_daily <- data.frame()
room_3_weekly <- data.frame()
room_3_monthly <- data.frame()
#TODO select sample as opposed to ID for intra bird comparison (alpha eq)
for(item in trans_reslist_room_3){
  room_3_daily <- rbind(room_3_daily, item$daily[c("sample","bottom","mid","top","total")])
  room_3_weekly <- rbind(room_3_weekly, item$weekly[c("sample","bottom","mid","top","total")])
  room_3_monthly <- rbind(room_3_monthly, item$monthly[c("sample","bottom","mid","top","total")])
}

# check to make sure only one bird is selected
print("check that the id and RFID tag are unique for each item entry")
```

```
## [1] "check that the id and RFID tag are unique for each item entry"
```

```
#this is following up with the issues of the intra-bird comparisons
for(item in trans_reslist_room_3){
  print(paste(unique(item$raw$Tagnumber)," ",unique(item$raw$tagname)))
  #print(item)
}
```

```
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-10-58   6736"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-16-46   6883"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-15-92   6902"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-19-09   6905"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-12-69   6907"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-05-59   6918"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-04-67   6921"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-02-45   6922"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-15-95   6924"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-07-52   6927"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-06-55   6933"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-03-10   6934"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-08-68   6938"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-08-92   6943"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-18-25   6944"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-19-07   6947"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-08-29   6959"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-05-76   6963"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-07-77   6967"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-05-46   6969"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-12-61   6972"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-03-79   6976"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-15-59   6977"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-03-70   6980"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-02-06   6992"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-04-06   6998"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-15-46   9004"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-06-16   9010"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-01-85   9013"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-04-10   9016"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-18-44   9018"
```

```
trans_reslist_room_8 <- list()

for(i in 1:length(reslist2_room_8)){

  daily_trans_table <- calc_trans_period(reslist2_room_8[[i]]$ID,reslist2_room_8[[i]]$daily_obj,reslist2
  weekly_trans_table <- calc_trans_period(reslist2_room_8[[i]]$ID,reslist2_room_8[[i]]$weekly_obj,reslis
  monthly_trans_table <- calc_trans_period(reslist2_room_8[[i]]$ID,reslist2_room_8[[i]]$monthly_obj,resl
  result <- list("ID"=reslist2_room_8[[i]]$ID,"daily"=daily_trans_table, "weekly"=weekly_trans_table,"m
  trans_reslist_room_8[[i]] <- result
}

room_8_daily <- data.frame()
room_8_weekly <- data.frame()
room_8_monthly <- data.frame()
#TODO select sample as opposed to ID for intra bird comparison (alpha eq)
for(item in trans_reslist_room_8){
  room_8_daily <- rbind(room_8_daily, item$daily[c("sample","bottom","mid","top","total")])
  room_8_weekly <- rbind(room_8_weekly, item$weekly[c("sample","bottom","mid","top","total")])
  room_8_monthly <- rbind(room_8_monthly, item$monthly[c("sample","bottom","mid","top","total")])
}

# check to make sure only one bird is selected
print("check that the id and RFID tag are unique for each item entry")
```

```
## [1] "check that the id and RFID tag are unique for each item entry"
```

```
#this is following up with the issues of the intra-bird comparisons
for(item in trans_reslist_room_8){
  print(paste(unique(item$raw$tagnumber)," ",unique(item$raw$tagname)))
  #print(item)
}
```

```
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-17-81   6886"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-09-71   6913"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-12-26   6917"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-09-79   6923"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-00-51   6928"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-05-45   6929"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-02-64   6931"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-09-43   6939"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-04-31   6940"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-09-76   6941"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-05-11   6945"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-02-80   6950"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-19-33   6951"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-17-70   6957"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-04-04   6961"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-12-06   6964"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-13-50   6970"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-03-55   6973"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-08-45   6979"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-01-08   6982"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-07-67   6984"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-18-97   6985"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-19-50   6993"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-15-34   6994"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-02-40   6996"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-02-89   6997"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-03-59   7000"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-09-74   9012"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-14-48   9014"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-10-60   9017"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-16-65   9020"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-10-71   9022"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-01-86   9023"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-11-80   9025"
```

```
trans_reslist_room_11 <- list()

for(i in 1:length(reslist2_room_11)){

  daily_trans_table <- calc_trans_period(reslist2_room_11[[i]]$ID,reslist2_room_11[[i]]$daily_obj,reslis
  weekly_trans_table <- calc_trans_period(reslist2_room_11[[i]]$ID,reslist2_room_11[[i]]$weekly_obj,resl
  monthly_trans_table <- calc_trans_period(reslist2_room_11[[i]]$ID,reslist2_room_11[[i]]$monthly_obj,re

  result <- list("ID"=reslist2_room_11[[i]]$ID,"daily"=daily_trans_table, "weekly"=weekly_trans_table,"r
  trans_reslist_room_11[[i]] <- result
}

room_11_daily <- data.frame()
room_11_weekly <- data.frame()
room_11_monthly <- data.frame()
#TODO select sample as opposed to ID for intra bird comparison (alpha eq)
for(item in trans_reslist_room_11){
  room_11_daily <- rbind(room_11_daily, item$daily[c("sample","bottom","mid","top","total")])
  room_11_weekly <- rbind(room_11_weekly, item$weekly[c("sample","bottom","mid","top","total")])
  room_11_monthly <- rbind(room_11_monthly, item$monthly[c("sample","bottom","mid","top","total")])
}

# check to make sure only one bird is selected
print("check that the id and RFID tag are unique for each item entry")
```

```
## [1] "check that the id and RFID tag are unique for each item entry"
```

```
#this is following up with the issues of the intra-bird comparisons
for(item in trans_reslist_room_11){
  print(paste(unique(item$raw$tagnumber)," ",unique(item$raw$tagname)))
  #print(item)
}
```

```
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-03-53   6853"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-10-20   6904"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-07-78   6916"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-19-12   6932"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-18-18   6965"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-16-24   6990"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-13-26   6999"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-09-59   9002"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-07-17   9003"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-03-56   9006"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-19-56   9007"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-07-37   9015"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-02-51   9023"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-06-94   9027"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-08-65   9030"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-09-23   9032"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-18-43   9033"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-19-81   9035"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-02-02   9038"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-02-83   9041"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-01-39   9042"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-03-50   9043"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-17-46   9044"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-11-55   9045"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-07-97   9046"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-08-00   9047"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-07-07   9048"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-15-42   9049"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-01-30   9050"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-03-35   9052"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-10-86   9053"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-19-39   9054"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-10-30   9055"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-10-89   9056"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-03-15   9058"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-06-32   9061"
```

```r
# w <- 0
# for(item in trans_reslist_room_2){
#   w <- w +1
#   print(w)
# }
# w <- 0
# for(item in trans_reslist_room_3){
#   w <- w +1
#   print(w)
# }
# w <- 0
# for(item in trans_reslist_room_8){
#   w <- w +1
#   print(w)
# }
# w <- 0
# for(item in trans_reslist_room_11){
#   w <- w +1
#   print(w)
# }
#write.csv(room_2_daily,"room_2_daily_intra.csv",row.names = F)
#write.csv(room_2_weekly,"room_2_weekly_intra.csv",row.names = F)
#write.csv(room_2_monthly,"room_2_monthly_intra.csv",row.names = F)
```

Determine which bird is the most active in the Whole Dataset

```r
daily <- data.frame()
weekly <- data.frame()
monthly <- data.frame()
#TODO select sample as opposed to ID for intra bird comparison (alpha eq)
for(item in trans_reslist_room_2){
  daily <- rbind(daily, item$daily[c("ID","bottom","mid","top","total")])
  weekly <- rbind(weekly, item$weekly[c("ID","bottom","mid","top","total")])
  monthly <- rbind(monthly, item$monthly[c("ID","bottom","mid","top","total")])
}
```

```
length(unique(daily$ID))
```

```
## [1] 37
```

```
for(item in trans_reslist_room_3){
  daily <- rbind(daily, item$daily[c("ID","bottom","mid","top","total")])
  weekly <- rbind(weekly, item$weekly[c("ID","bottom","mid","top","total")])
  monthly <- rbind(monthly, item$monthly[c("ID","bottom","mid","top","total")])
}
length(unique(daily$ID))
```

```
## [1] 68
```

```
for(item in trans_reslist_room_8){
  daily <- rbind(daily, item$daily[c("ID","bottom","mid","top","total")])
  weekly <- rbind(weekly, item$weekly[c("ID","bottom","mid","top","total")])
  monthly <- rbind(monthly, item$monthly[c("ID","bottom","mid","top","total")])
}
length(unique(daily$ID))
```

```
## [1] 102
```

```
for(item in trans_reslist_room_11){
  daily <- rbind(daily, item$daily[c("ID","bottom","mid","top","total")])
  weekly <- rbind(weekly, item$weekly[c("ID","bottom","mid","top","total")])
  monthly <- rbind(monthly, item$monthly[c("ID","bottom","mid","top","total")])
}
length(unique(daily$ID))
```

```
## [1] 137
```

```
tmp <- ""
tmp_df <- ""
tmp <- cbind(as.character(daily$ID), as.numeric(daily$total))
tmp_df <- data.frame(as.numeric(tmp[,1]), as.numeric(tmp[,2]))
colnames(tmp_df) <- c("ID","total")
transitions <- aggregate(total ~ ID, data=tmp_df, FUN=sum)
days <- table(tmp_df$ID)
norm_tots <- round(transitions$total/days,0)
norm_tots_df <- data.frame(norm_tots)
norm_tots_df <- data.frame(as.character(norm_tots_df$Var1), as.numeric(norm_tots_df$Freq))
colnames(norm_tots_df) <-c("ID","norm_tot")
most_active <- norm_tots_df$ID[norm_tots_df$norm_tot == max(norm_tots_df$norm_tot)]

print(paste("Most active bird: ",most_active))
```

```
## [1] "Most active bird:  6905"
```

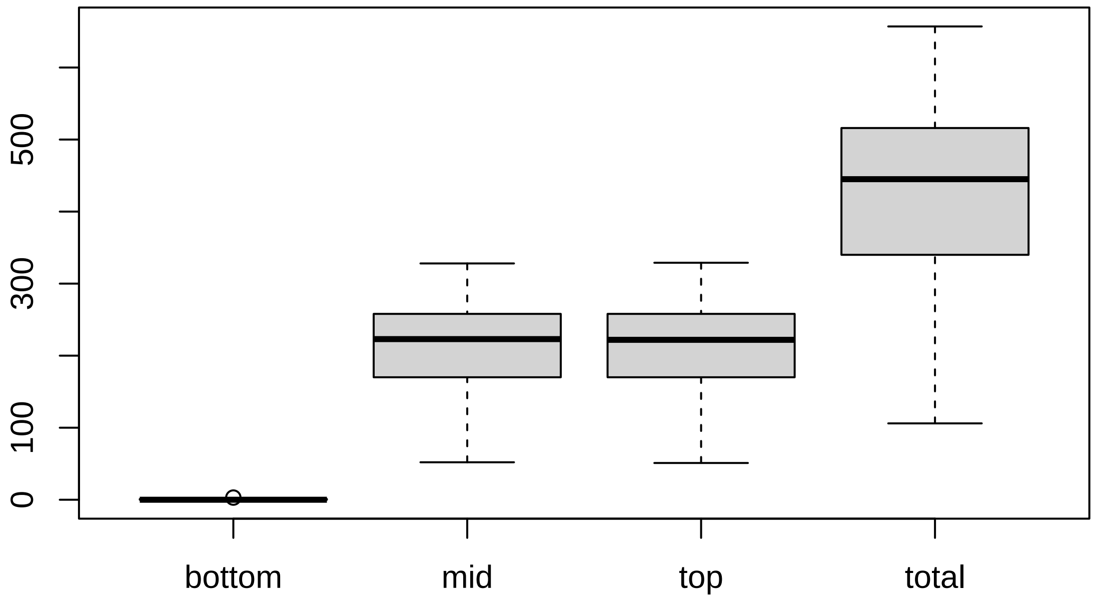
Determine which bird is the least active in the Room 2 Dataset

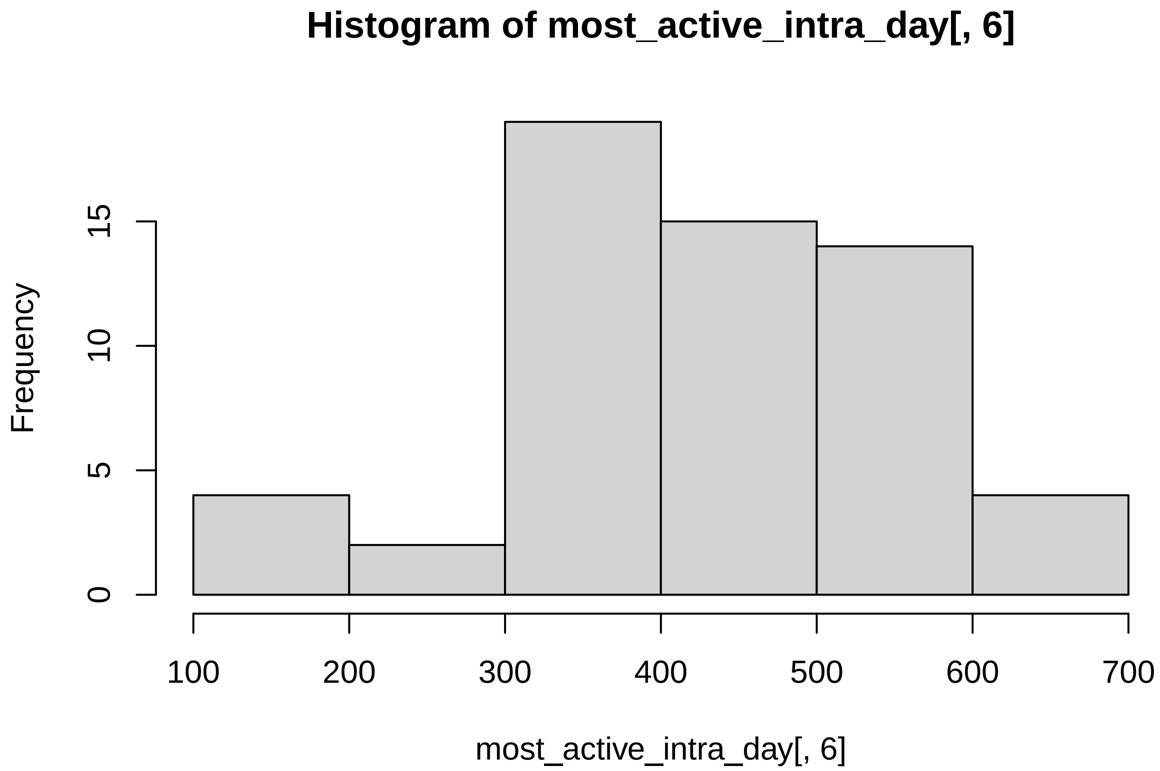
```
daily <- data.frame()
weekly <- data.frame()
monthly <- data.frame()
#TODO select sample as opposed to ID for intra bird comparison (alpha eq)
for(item in trans_reslist_room_2){
  daily <- rbind(daily, item$daily[c("ID","bottom","mid","top","total")])
  weekly <- rbind(weekly, item$weekly[c("ID","bottom","mid","top","total")])
  monthly <- rbind(monthly, item$monthly[c("ID","bottom","mid","top","total")])
}
for(item in trans_reslist_room_3){
  daily <- rbind(daily, item$daily[c("ID","bottom","mid","top","total")])
  weekly <- rbind(weekly, item$weekly[c("ID","bottom","mid","top","total")])
  monthly <- rbind(monthly, item$monthly[c("ID","bottom","mid","top","total")])
}
for(item in trans_reslist_room_8){
  daily <- rbind(daily, item$daily[c("ID","bottom","mid","top","total")])
  weekly <- rbind(weekly, item$weekly[c("ID","bottom","mid","top","total")])
  monthly <- rbind(monthly, item$monthly[c("ID","bottom","mid","top","total")])
}
for(item in trans_reslist_room_11){
  daily <- rbind(daily, item$daily[c("ID","bottom","mid","top","total")])
  weekly <- rbind(weekly, item$weekly[c("ID","bottom","mid","top","total")])
  monthly <- rbind(monthly, item$monthly[c("ID","bottom","mid","top","total")])
}

tmp  <- ""
tmp_df <- ""

tmp <- cbind(as.character(daily$ID), as.numeric(daily$total))
tmp_df <- data.frame(as.numeric(tmp[,1]), as.numeric(tmp[,2]))
colnames(tmp_df) <- c("ID","total")
transitions <- aggregate(total ~ ID, data=tmp_df, FUN=sum)
days <- table(tmp_df$ID)
norm_tots <- round(transitions$total/days,0)
norm_tots_df <- data.frame(norm_tots)
norm_tots_df <- data.frame(as.character(norm_tots_df$Var1), as.numeric(norm_tots_df$Freq))
colnames(norm_tots_df) <-c("ID","norm_tot")
least_active <- norm_tots_df$ID[norm_tots_df$norm_tot == min(norm_tots_df$norm_tot)]

print(paste("Least active bird: ",least_active))
```

```
## [1] "Least active bird:  6990"
```

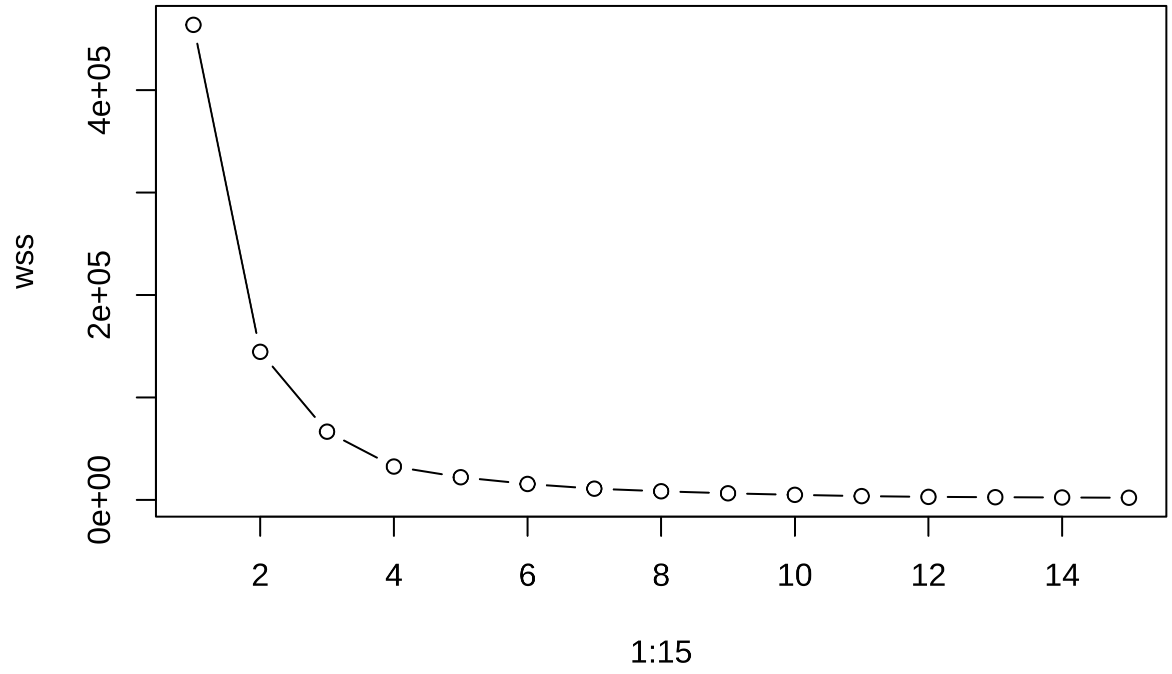
Most Active Bird

```
set.seed(34716)
library(scales)
library(stringr)
library(ggplot2)
library(lubridate)

daily <- data.frame()
weekly <- data.frame()
monthly <- data.frame()
#TODO select sample as opposed to ID for intra bird comparison (alpha eq)
for(item in trans_reslist_room_2){
  daily <- rbind(daily, item$daily[c("sample","ID","bottom","mid","top","total")])
  weekly <- rbind(weekly, item$weekly[c("sample","ID","bottom","mid","top","total")])
  monthly <- rbind(monthly, item$monthly[c("sample","ID","bottom","mid","top","total")])
}
for(item in trans_reslist_room_3){
  daily <- rbind(daily, item$daily[c("sample","ID","bottom","mid","top","total")])
  weekly <- rbind(weekly, item$weekly[c("sample","ID","bottom","mid","top","total")])
  monthly <- rbind(monthly, item$monthly[c("sample","ID","bottom","mid","top","total")])
}
for(item in trans_reslist_room_8){
  daily <- rbind(daily, item$daily[c("sample","ID","bottom","mid","top","total")])
  weekly <- rbind(weekly, item$weekly[c("sample","ID","bottom","mid","top","total")])
  monthly <- rbind(monthly, item$monthly[c("sample","ID","bottom","mid","top","total")])
}
for(item in trans_reslist_room_11){
  daily <- rbind(daily, item$daily[c("sample","ID","bottom","mid","top","total")])
  weekly <- rbind(weekly, item$weekly[c("sample","ID","bottom","mid","top","total")])
  monthly <- rbind(monthly, item$monthly[c("sample","ID","bottom","mid","top","total")])
}

for(j in 1:length(most_active)){
most_active_intra_day <- daily[grep(most_active[j], daily$ID),]
most_active_intra_day <- as.data.frame(most_active_intra_day)
most_active_intra_day[,3:6] <- sapply(most_active_intra_day[,3:6], as.numeric)
boxplot(most_active_intra_day[,3:6])
hist(most_active_intra_day[,6])
}
```





```
wss <- 0

for(i in 1:15){
  km.out <- kmeans(most_active_intra_day[,3:5], centers = i, nstart=20)
  wss[i] <- km.out$tot.withinss
}


plot(1:15, wss, type="b")
```



```
km.out <- kmeans(most_active_intra_day[,3:5],3, 20)
#km.out <- kmeans(most_active_intra_day[,2:4],3, 20)
summary(km.out)
```

```
##              Length Class  Mode
## cluster      58     -none- numeric
## centers       9     -none- numeric
## totss         1     -none- numeric
## withinss      3     -none- numeric
## tot.withinss  1     -none- numeric
## betweenss     1     -none- numeric
## size          3     -none- numeric
## iter          1     -none- numeric
## ifault        1     -none- numeric
```

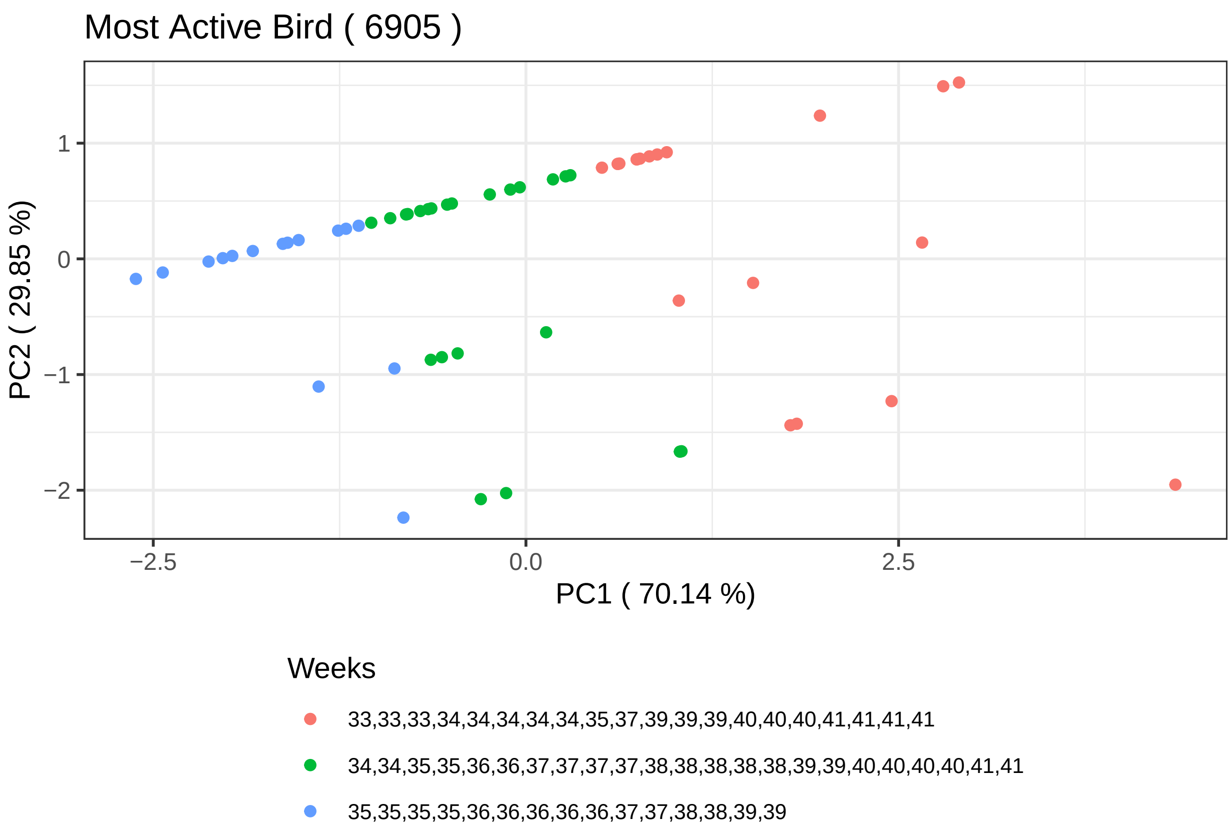
```
table(km.out$cluster)
```

```
##
##  1  2  3
## 20 23 15
```

```
pr.feb <- prcomp(x=most_active_intra_day[,3:5], scale=T, center=T)
pr.feb.prop <- summary(pr.feb)
most_active_intra_day["day"] <- as.character(str_split_fixed(most_active_intra_day$sample, ".",6)[,6])
most_active_intra_day["week"]<- (week(most_active_intra_day$day)+23)
most_active_intra_day["cluster"] <- km.out$cluster
most_active_intra_day
```

```
##                sample   ID bottom mid top total        day week cluster
## 2321 6905.2021-03-09 6905      0  68  68   136 2021-03-09   33       1
## 2322 6905.2021-03-10 6905      0  73  73   146 2021-03-10   33       1
## 2323 6905.2021-03-11 6905      0 112 112   224 2021-03-11   33       1
## 2324 6905.2021-03-12 6905      0 170 170   340 2021-03-12   34       1
## 2325 6905.2021-03-13 6905      0 175 176   351 2021-03-13   34       1
## 2326 6905.2021-03-14 6905      0 166 166   332 2021-03-14   34       1
## 2327 6905.2021-03-15 6905      0 181 181   362 2021-03-15   34       1
## 2328 6905.2021-03-16 6905      0 191 191   382 2021-03-16   34       2
## 2329 6905.2021-03-17 6905      2 190 191   383 2021-03-17   34       2
## 2330 6905.2021-03-18 6905      2 155 157   314 2021-03-18   34       1
## 2331 6905.2021-03-19 6905      0 164 163   327 2021-03-19   35       1
## 2332 6905.2021-03-20 6905      0 229 228   457 2021-03-20   35       2
## 2333 6905.2021-03-21 6905      0 280 281   561 2021-03-21   35       3
## 2334 6905.2021-03-22 6905      0 282 282   564 2021-03-22   35       3
## 2335 6905.2021-03-23 6905      0 298 298   596 2021-03-23   35       3
## 2336 6905.2021-03-24 6905      0 264 265   529 2021-03-24   35       3
## 2337 6905.2021-03-25 6905      0 239 238   477 2021-03-25   35       2
## 2338 6905.2021-03-26 6905      0 277 277   554 2021-03-26   36       3
## 2339 6905.2021-03-27 6905      1 252 253   506 2021-03-27   36       2
## 2340 6905.2021-03-28 6905      2 246 246   494 2021-03-28   36       2
## 2341 6905.2021-03-29 6905      1 289 287   577 2021-03-29   36       3
## 2342 6905.2021-03-30 6905      0 292 291   583 2021-03-30   36       3
## 2343 6905.2021-03-31 6905      0 301 301   602 2021-03-31   36       3
## 2344 6905.2021-04-01 6905      0 328 329   657 2021-04-01   36       3
## 2345 6905.2021-04-02 6905      2 279 278   559 2021-04-02   37       3
## 2346 6905.2021-04-03 6905      2 254 254   510 2021-04-03   37       2
## 2347 6905.2021-04-04 6905      0 248 248   496 2021-04-04   37       2
## 2348 6905.2021-04-05 6905      2 155 153   310 2021-04-05   37       1
## 2349 6905.2021-04-06 6905      2 192 190   384 2021-04-06   37       2
## 2350 6905.2021-04-07 6905      0 258 258   516 2021-04-07   37       3
## 2351 6905.2021-04-08 6905      0 217 216   433 2021-04-08   37       2
## 2352 6905.2021-04-09 6905      0 236 236   472 2021-04-09   38       2
## 2353 6905.2021-04-10 6905      1 249 249   499 2021-04-10   38       2
## 2354 6905.2021-04-11 6905      1 216 216   433 2021-04-11   38       2
## 2355 6905.2021-04-12 6905      0 193 192   385 2021-04-12   38       2
## 2356 6905.2021-04-13 6905      0 262 262   524 2021-04-13   38       3
## 2357 6905.2021-04-14 6905      0 320 320   640 2021-04-14   38       3
## 2358 6905.2021-04-15 6905      1 244 244   489 2021-04-15   38       2
## 2359 6905.2021-04-16 6905      1 264 264   529 2021-04-16   39       3
## 2360 6905.2021-04-17 6905      0 306 305   611 2021-04-17   39       3
## 2361 6905.2021-04-18 6905      0 230 230   460 2021-04-18   39       2
## 2362 6905.2021-04-19 6905      2 124 124   250 2021-04-19   39       1
## 2363 6905.2021-04-20 6905      3  52  51   106 2021-04-20   39       1
## 2364 6905.2021-04-21 6905      1  98  96   195 2021-04-21   39       1
## 2365 6905.2021-04-22 6905      0 207 207   414 2021-04-22   39       2
## 2366 6905.2021-04-23 6905      0 254 254   508 2021-04-23   40       2
## 2367 6905.2021-04-24 6905      0 235 235   470 2021-04-24   40       2
## 2368 6905.2021-04-25 6905      0 242 243   485 2021-04-25   40       2
## 2369 6905.2021-04-26 6905      0 243 243   486 2021-04-26   40       2
## 2370 6905.2021-04-27 6905      1 174 174   349 2021-04-27   40       1
## 2371 6905.2021-04-28 6905      1 151 150   302 2021-04-28   40       1
## 2372 6905.2021-04-29 6905      0 161 160   321 2021-04-29   40       1
## 2373 6905.2021-04-30 6905      0 170 170   340 2021-04-30   41       1
## 2374 6905.2021-05-01 6905      0 176 176    352 2021-05-01    41        1
## 2375 6905.2021-05-02 6905      0 169 169    338 2021-05-02    41        1
## 2376 6905.2021-05-03 6905      0 210 210    420 2021-05-03    41        2
## 2377 6905.2021-05-04 6905      0 197 196    393 2021-05-04    41        2
## 2378 6905.2021-05-05 6905      0 176 176    352 2021-05-05    41        1
```

```
pr.feb$x <- data.frame(pr.feb$x)

p <- ggplot(pr.feb$x[,1:2], aes(x=PC1, y=PC2, color=as.character(most_active_intra_day$cluster)))+
  geom_point() +
  labs(title=paste("Most Active Bird (",most_active,")"),x=paste("PC1 (",round(pr.feb.prop$importance[[
  scale_color_discrete(name = "Weeks", labels= c(as.character(paste(most_active_intra_day$week[most_acti
  theme_bw() +
  theme(legend.position = "bottom", legend.direction = "vertical",legend.text = element_text(size=8))
p
```



```
ggsave(p, filename = paste0("../figures/",as.character(most_active[j]),"_most_active_bird_pca_all_rooms


cluster_table <- data.frame(cbind(as.numeric(km.out$cluster),most_active_intra_day$sample))
cluster_table[order(cluster_table$X1),]
```

```
##    X1              X2
## 1   1 6905.2021-03-09
## 2   1 6905.2021-03-10
## 3   1 6905.2021-03-11
## 4   1 6905.2021-03-12
## 5    1 6905.2021-03-13
## 6    1 6905.2021-03-14
## 7    1 6905.2021-03-15
## 10   1 6905.2021-03-18
## 11   1 6905.2021-03-19
## 28   1 6905.2021-04-05
## 42   1 6905.2021-04-19
## 43   1 6905.2021-04-20
## 44   1 6905.2021-04-21
## 50   1 6905.2021-04-27
## 51   1 6905.2021-04-28
## 52   1 6905.2021-04-29
## 53   1 6905.2021-04-30
## 54   1 6905.2021-05-01
## 55   1 6905.2021-05-02
## 58   1 6905.2021-05-05
## 8    2 6905.2021-03-16
## 9    2 6905.2021-03-17
## 12   2 6905.2021-03-20
## 17   2 6905.2021-03-25
## 19   2 6905.2021-03-27
## 20   2 6905.2021-03-28
## 26   2 6905.2021-04-03
## 27   2 6905.2021-04-04
## 29   2 6905.2021-04-06
## 31   2 6905.2021-04-08
## 32   2 6905.2021-04-09
## 33   2 6905.2021-04-10
## 34   2 6905.2021-04-11
## 35   2 6905.2021-04-12
## 38   2 6905.2021-04-15
## 41   2 6905.2021-04-18
## 45   2 6905.2021-04-22
## 46   2 6905.2021-04-23
## 47   2 6905.2021-04-24
## 48   2 6905.2021-04-25
## 49   2 6905.2021-04-26
## 56   2 6905.2021-05-03
## 57   2 6905.2021-05-04
## 13   3 6905.2021-03-21
## 14   3 6905.2021-03-22
## 15   3 6905.2021-03-23
## 16   3 6905.2021-03-24
## 18   3 6905.2021-03-26
## 21   3 6905.2021-03-29
## 22   3 6905.2021-03-30
## 23   3 6905.2021-03-31
## 24   3 6905.2021-04-01
## 25   3 6905.2021-04-02
## 30   3 6905.2021-04-07
## 36   3 6905.2021-04-13
## 37   3 6905.2021-04-14
## 39   3 6905.2021-04-16
## 40   3 6905.2021-04-17
```

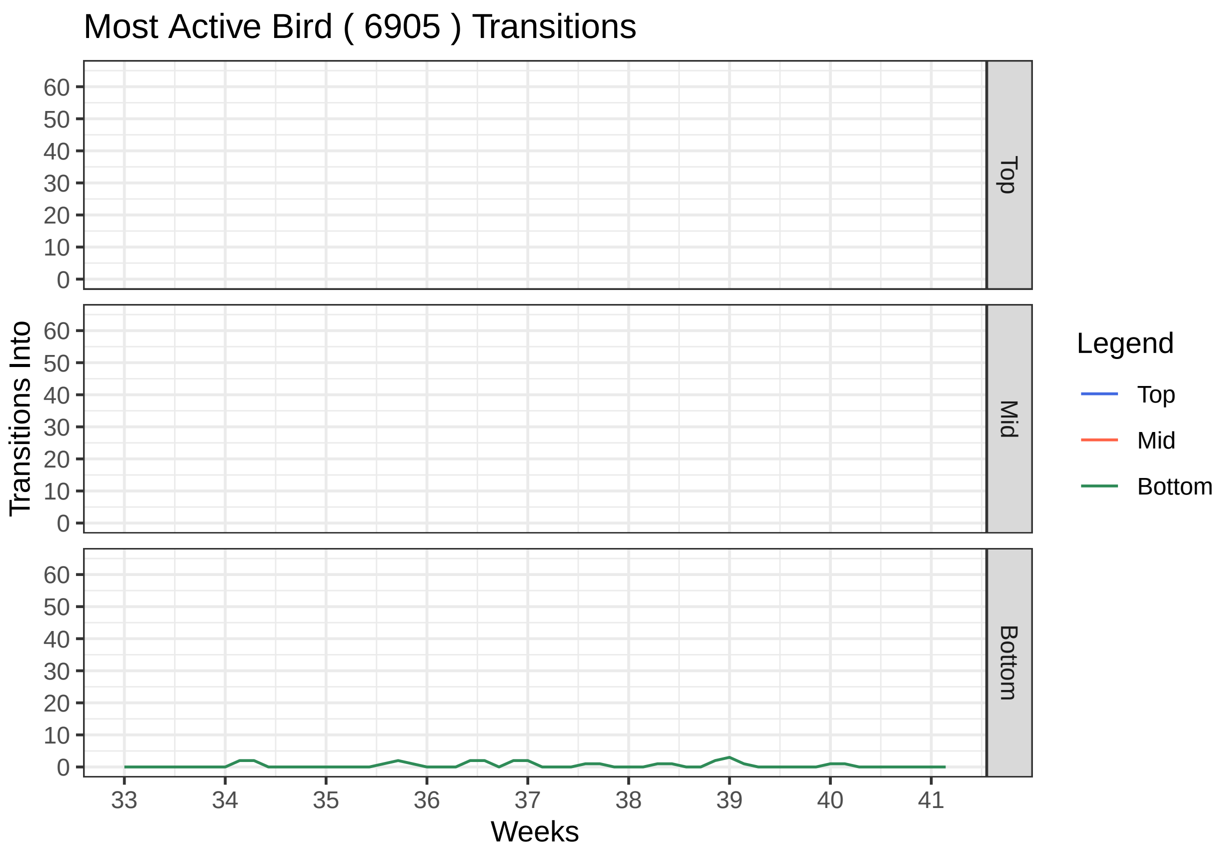
```
most_active_intra_day[order(most_active_intra_day$total),]
```

```
##               sample   ID bottom mid top total        day week cluster
## 2363 6905.2021-04-20 6905      3  52  51   106 2021-04-20   39       1
## 2321 6905.2021-03-09 6905      0  68  68   136 2021-03-09   33       1
## 2322 6905.2021-03-10 6905      0  73  73   146 2021-03-10   33       1
## 2364 6905.2021-04-21 6905      1  98  96   195 2021-04-21   39       1
## 2323 6905.2021-03-11 6905      0 112 112   224 2021-03-11   33       1
## 2362 6905.2021-04-19 6905      2 124 124   250 2021-04-19   39       1
## 2371 6905.2021-04-28 6905      1 151 150   302 2021-04-28   40       1
## 2348 6905.2021-04-05 6905      2 155 153   310 2021-04-05   37       1
## 2330 6905.2021-03-18 6905      2 155 157   314 2021-03-18   34       1
## 2372 6905.2021-04-29 6905      0 161 160   321 2021-04-29   40       1
## 2331 6905.2021-03-19 6905      0 164 163   327 2021-03-19   35       1
## 2326 6905.2021-03-14 6905      0 166 166   332 2021-03-14   34       1
## 2375 6905.2021-05-02 6905      0 169 169   338 2021-05-02   41       1
## 2324 6905.2021-03-12 6905      0 170 170   340 2021-03-12   34       1
## 2373 6905.2021-04-30 6905      0 170 170   340 2021-04-30   41       1
## 2370 6905.2021-04-27 6905      1 174 174   349 2021-04-27   40       1
## 2325 6905.2021-03-13 6905      0 175 176   351 2021-03-13   34       1
## 2374 6905.2021-05-01 6905      0 176 176   352 2021-05-01   41       1
## 2378 6905.2021-05-05 6905      0 176 176   352 2021-05-05   41       1
## 2327 6905.2021-03-15 6905      0 181 181   362 2021-03-15   34       1
## 2328 6905.2021-03-16 6905      0 191 191   382 2021-03-16   34       2
## 2329 6905.2021-03-17 6905      2 190 191   383 2021-03-17   34       2
## 2349 6905.2021-04-06 6905      2 192 190   384 2021-04-06   37       2
## 2355 6905.2021-04-12 6905      0 193 192   385 2021-04-12   38       2
## 2377 6905.2021-05-04 6905      0 197 196   393 2021-05-04   41       2
## 2365 6905.2021-04-22 6905      0 207 207   414 2021-04-22   39       2
## 2376 6905.2021-05-03 6905      0 210 210   420 2021-05-03   41       2
## 2351 6905.2021-04-08 6905      0 217 216   433 2021-04-08   37       2
## 2354 6905.2021-04-11 6905      1 216 216   433 2021-04-11   38       2
## 2332 6905.2021-03-20 6905      0 229 228   457 2021-03-20   35       2
## 2361 6905.2021-04-18 6905      0 230 230   460 2021-04-18   39       2
## 2367 6905.2021-04-24 6905      0 235 235   470 2021-04-24   40       2
## 2352 6905.2021-04-09 6905      0 236 236   472 2021-04-09   38       2
## 2337 6905.2021-03-25 6905      0 239 238   477 2021-03-25   35       2
## 2368 6905.2021-04-25 6905      0 242 243   485 2021-04-25   40       2
## 2369 6905.2021-04-26 6905      0 243 243   486 2021-04-26   40       2
## 2358 6905.2021-04-15 6905      1 244 244   489 2021-04-15   38       2
## 2340 6905.2021-03-28 6905      2 246 246   494 2021-03-28   36       2
## 2347 6905.2021-04-04 6905      0 248 248   496 2021-04-04   37       2
## 2353 6905.2021-04-10 6905      1 249 249   499 2021-04-10   38       2
## 2339 6905.2021-03-27 6905      1 252 253   506 2021-03-27   36       2
## 2366 6905.2021-04-23 6905      0 254 254   508 2021-04-23   40       2
## 2346 6905.2021-04-03 6905      2 254 254   510 2021-04-03   37       2
## 2350 6905.2021-04-07 6905      0 258 258   516 2021-04-07   37       3
## 2356 6905.2021-04-13 6905      0 262 262   524 2021-04-13   38       3
## 2336 6905.2021-03-24 6905      0 264 265   529 2021-03-24   35       3
## 2359 6905.2021-04-16 6905      1 264 264   529 2021-04-16   39       3
## 2338 6905.2021-03-26 6905      0 277 277   554 2021-03-26   36       3
## 2345 6905.2021-04-02 6905      2 279 278   559 2021-04-02   37       3
## 2333 6905.2021-03-21 6905      0 280 281   561 2021-03-21   35       3
## 2334 6905.2021-03-22 6905      0 282 282   564 2021-03-22   35       3
## 2341 6905.2021-03-29 6905      1 289 287   577 2021-03-29   36       3
## 2342 6905.2021-03-30 6905      0 292 291   583 2021-03-30   36       3
## 2335 6905.2021-03-23 6905      0 298 298   596 2021-03-23   35       3
## 2343 6905.2021-03-31 6905      0 301 301   602 2021-03-31   36       3
## 2360 6905.2021-04-17 6905      0 306 305   611 2021-04-17   39       3
## 2357 6905.2021-04-14 6905      0 320 320   640 2021-04-14   38       3
## 2344 6905.2021-04-01 6905      0 328 329   657 2021-04-01   36       3
```

```
most_active_grp <- data.frame(x = as.Date(most_active_intra_day$day), y = c(most_active_intra_day$bottom



 q <- ggplot(most_active_grp, aes(x,y,col=group))+
  geom_line() +
  facet_grid(factor(group,levels=c("Top","Mid","Bottom"))~ .) +
  labs(title=paste("Most Active Bird (",most_active[j],") Transitions"), color="Legend")+
  scale_x_date(name="Weeks",breaks = seq(min(most_active_grp$x),max(most_active_grp$x),by="week"),label
  scale_y_continuous(name="Transitions Into",n.breaks = 8,limits = c(0,65)) +
  scale_color_manual(values=c("Top"="royalblue","Mid"="tomato","Bottom" = "seagreen")) +
  theme_bw()

q
```

```
## Warning: Removed 114 row(s) containing missing values (geom_path).
```

```
## geom_path: Each group consists of only one observation. Do you need to adjust
## the group aesthetic?
## geom_path: Each group consists of only one observation. Do you need to adjust
## the group aesthetic?
```



```
ggsave(q, filename = paste0("../figures/",as.character(most_active[j]),"_most_active_transitions_per_day
```

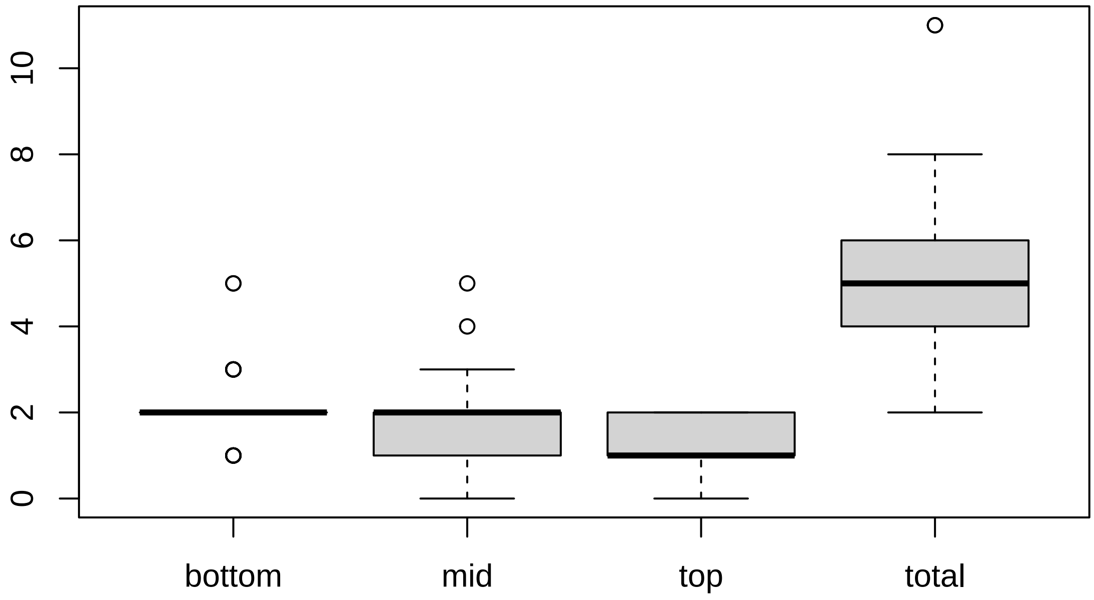
```
## Saving 6.5 x 4.5 in image
```

```
## Warning: Removed 114 row(s) containing missing values (geom_path).
```

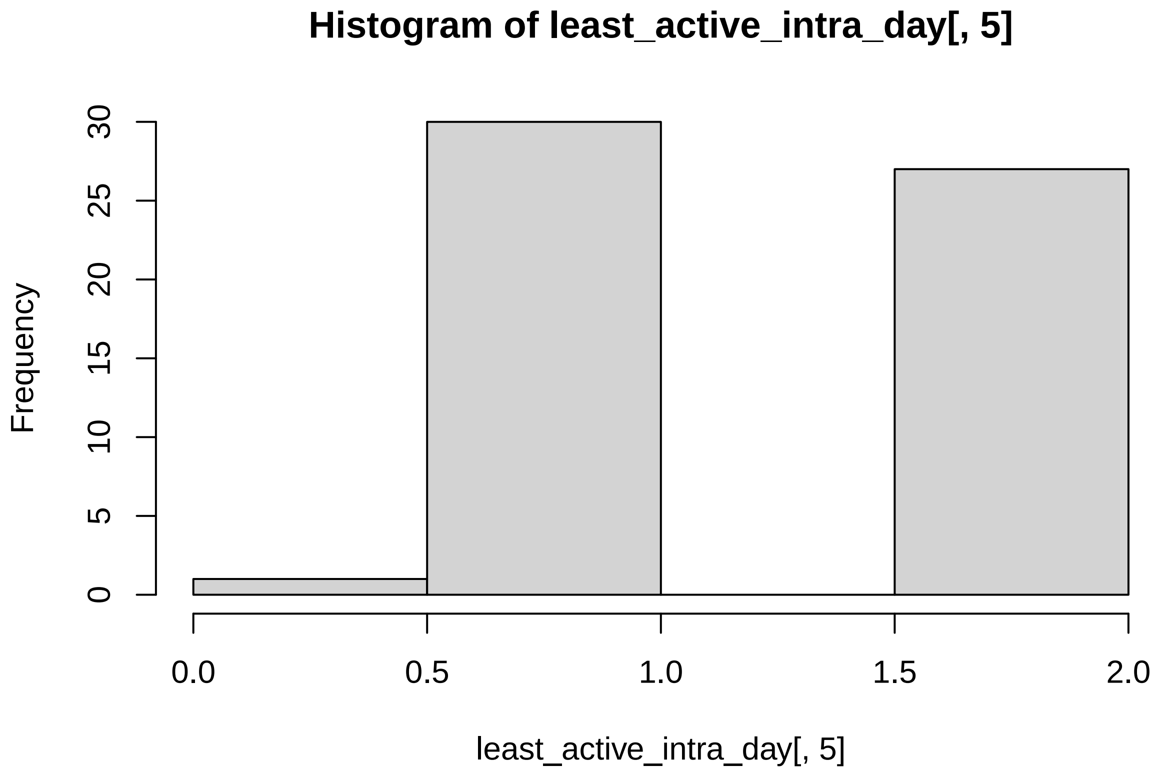
```
## geom_path: Each group consists of only one observation. Do you need to adjust
## the group aesthetic?
## geom_path: Each group consists of only one observation. Do you need to adjust
## the group aesthetic?
```

Least Active Bird

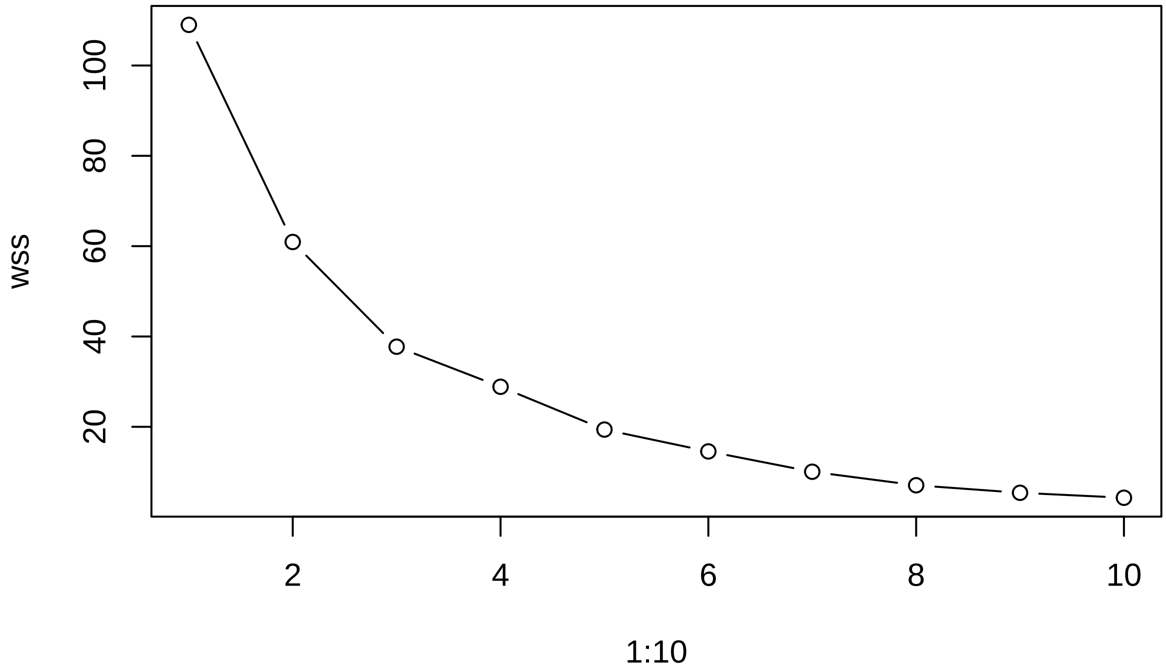
```
library(stringr)
library(ggplot2)

daily <- data.frame()
weekly <- data.frame()
monthly <- data.frame()
#TODO select sample as opposed to ID for intra bird comparison (alpha eq)
for(item in trans_reslist_room_2){
  daily <- rbind(daily, item$daily[c("sample","ID","bottom","mid","top","total")])
  weekly <- rbind(weekly, item$weekly[c("sample","ID","bottom","mid","top","total")])
  monthly <- rbind(monthly, item$monthly[c("sample","ID","bottom","mid","top","total")])
}
for(item in trans_reslist_room_3){
  daily <- rbind(daily, item$daily[c("sample","ID","bottom","mid","top","total")])
  weekly <- rbind(weekly, item$weekly[c("sample","ID","bottom","mid","top","total")])
  monthly <- rbind(monthly, item$monthly[c("sample","ID","bottom","mid","top","total")])
}
for(item in trans_reslist_room_8){
  daily <- rbind(daily, item$daily[c("sample","ID","bottom","mid","top","total")])
  weekly <- rbind(weekly, item$weekly[c("sample","ID","bottom","mid","top","total")])
  monthly <- rbind(monthly, item$monthly[c("sample","ID","bottom","mid","top","total")])
}
for(item in trans_reslist_room_11){
  daily <- rbind(daily, item$daily[c("sample","ID","bottom","mid","top","total")])
  weekly <- rbind(weekly, item$weekly[c("sample","ID","bottom","mid","top","total")])
  monthly <- rbind(monthly, item$monthly[c("sample","ID","bottom","mid","top","total")])
}

least_active_intra_day <- daily[grep(least_active,daily$ID),]
least_active_intra_day <- as.data.frame(least_active_intra_day)
least_active_intra_day[,3:6] <- sapply(least_active_intra_day[,3:6], as.numeric)
boxplot(least_active_intra_day[,3:6])
```



```
hist(least_active_intra_day[,5])
```



```
wss <- 0

for(i in 1:10){
  km.out <- kmeans(least_active_intra_day[,3:5], centers = i, nstart=20)
  wss[i] <- km.out$tot.withinss
}


plot(1:10, wss, type="b")
```



```
km.out <- kmeans(least_active_intra_day[,3:5],3, 20)
summary(km.out)
```

```
##              Length Class  Mode
## cluster      58     -none- numeric
## centers       9     -none- numeric
## totss         1     -none- numeric
## withinss      3     -none- numeric
## tot.withinss  1     -none- numeric
## betweenss     1     -none- numeric
## size          3     -none- numeric
## iter          1     -none- numeric
## ifault        1     -none- numeric
```

```
table(km.out$cluster)
```

```
##
##  1  2  3
## 32  9 17
```

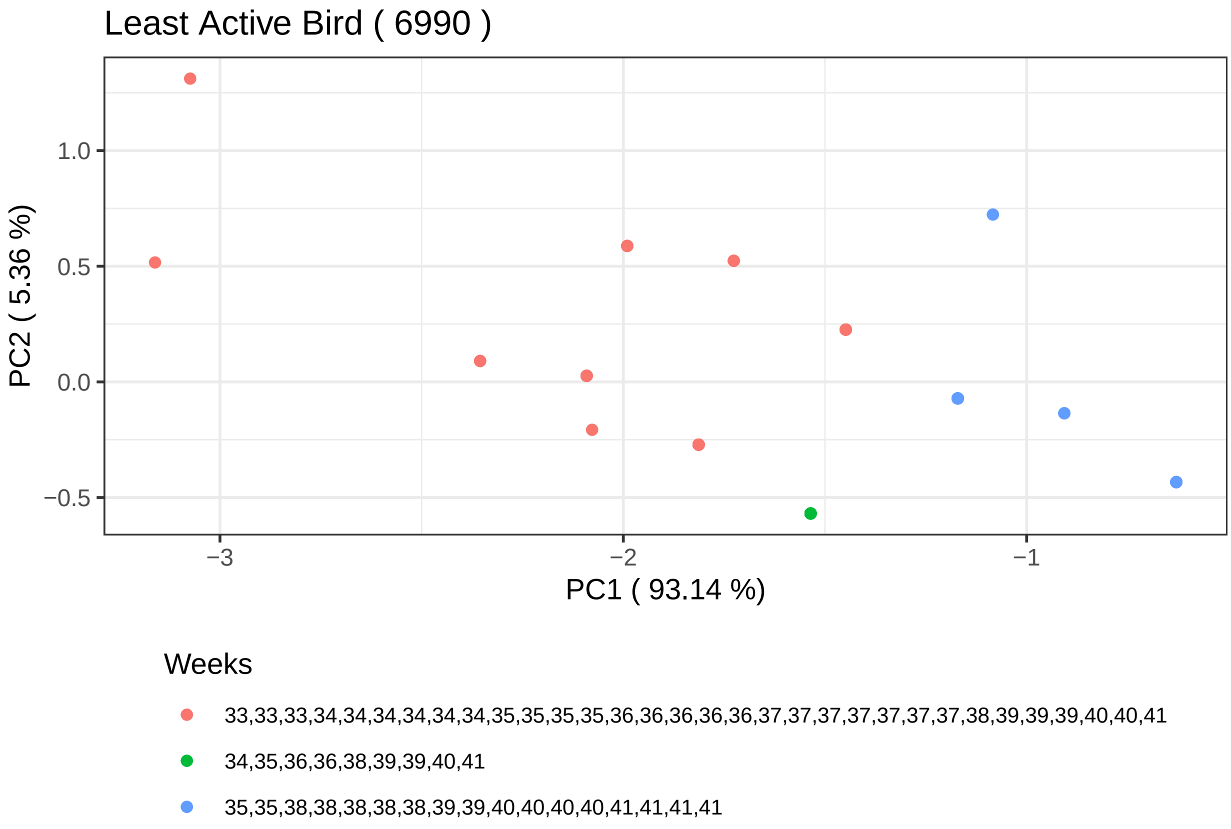
```
pr.feb <- prcomp(x=least_active_intra_day[,3:5], scale=T, center=F)
pr.feb.prop <- summary(pr.feb)

pr.feb.prop$importance
```

```
##                             PC1       PC2       PC3
## Standard deviation     1.671625 0.4009282 0.2119557
## Proportion of Variance 0.931440 0.0535800 0.0149800
## Cumulative Proportion  0.931440 0.9850200 1.0000000
```

```
pr.feb$x <- data.frame(pr.feb$x)


least_active_intra_day["day"] <- as.character(str_split_fixed(least_active_intra_day$sample, ".",6)[,6]
least_active_intra_day["week"]<- (week(least_active_intra_day$day)+23)
least_active_intra_day["cluster"] <- km.out$cluster
least_active_intra_day
```

```
##                 sample   ID bottom mid top total        day week cluster
## 6207 6990.2021-03-09 6990      3   2   2     7 2021-03-09   33       1
## 6208 6990.2021-03-10 6990      3   3   2     8 2021-03-10   33       1
## 6209 6990.2021-03-11 6990      5   5   1    11 2021-03-11   33       1
## 6210 6990.2021-03-12 6990      5   4   2    11 2021-03-12   34       1
## 6211 6990.2021-03-13 6990      2   2   1     5 2021-03-13   34       1
## 6212 6990.2021-03-14 6990      2   1   2     5 2021-03-14   34       2
## 6213 6990.2021-03-15 6990      2   2   2     6 2021-03-15   34       1
## 6214 6990.2021-03-16 6990      2   3   2     7 2021-03-16   34       1
## 6215 6990.2021-03-17 6990      2   2   2     6 2021-03-17   34       1
## 6216 6990.2021-03-18 6990      3   3   1     7 2021-03-18   34       1
## 6217 6990.2021-03-19 6990      3   3   1     7 2021-03-19   35       1
## 6218 6990.2021-03-20 6990      2   1   2     5 2021-03-20   35       2
## 6219 6990.2021-03-21 6990      2   1   1     4 2021-03-21   35       3
## 6220 6990.2021-03-22 6990      2   2   0     4 2021-03-22   35       3
## 6221 6990.2021-03-23 6990      2   2   1     5 2021-03-23   35       1
## 6222 6990.2021-03-24 6990      2   2   2     6 2021-03-24   35       1
## 6223 6990.2021-03-25 6990      2   2   2     6 2021-03-25   35       1
## 6224 6990.2021-03-26 6990      2   3   2     7 2021-03-26   36       1
## 6225 6990.2021-03-27 6990      2   3   1     6 2021-03-27   36       1
## 6226 6990.2021-03-28 6990      2   2   1     5 2021-03-28   36       1
## 6227 6990.2021-03-29 6990      2   2   2     6 2021-03-29   36       1
## 6228 6990.2021-03-30 6990      2   1   2     5 2021-03-30   36       2
## 6229 6990.2021-03-31 6990      2   1   2     5 2021-03-31   36       2
## 6230 6990.2021-04-01 6990      2   2   1     5 2021-04-01   36       1
## 6231 6990.2021-04-02 6990      3   3   1     7 2021-04-02   37       1
## 6232 6990.2021-04-03 6990      3   3   2     8 2021-04-03   37       1
## 6233 6990.2021-04-04 6990      2   2   2     6 2021-04-04   37       1
## 6234 6990.2021-04-05 6990      2   2   2     6 2021-04-05   37       1
## 6235 6990.2021-04-06 6990      2   3   2     7 2021-04-06   37       1
## 6236 6990.2021-04-07 6990      2   3   2     7 2021-04-07   37       1
## 6237 6990.2021-04-08 6990      2   2   1     5 2021-04-08   37       1
## 6238 6990.2021-04-09 6990      2   1   1     4 2021-04-09   38       3
## 6239 6990.2021-04-10 6990      2   1   1     4 2021-04-10   38       3
## 6240 6990.2021-04-11 6990      2   2   1     5 2021-04-11   38       1
## 6241 6990.2021-04-12 6990      2   1   1     4 2021-04-12   38       3
## 6242 6990.2021-04-13 6990      1   0   1     2 2021-04-13   38       3
## 6243 6990.2021-04-14 6990      1   0   1     2 2021-04-14   38       3
## 6244 6990.2021-04-15 6990      2   1   2     5 2021-04-15   38       2
## 6245 6990.2021-04-16 6990      2   2   2     6 2021-04-16   39       1
## 6246 6990.2021-04-17 6990      2   1   2     5 2021-04-17   39       2
## 6247 6990.2021-04-18 6990      2   1   1     4 2021-04-18   39       3
## 6248 6990.2021-04-19 6990      2   3   1     6 2021-04-19   39       1
## 6249 6990.2021-04-20 6990      2   3   2     7 2021-04-20   39       1
## 6250 6990.2021-04-21 6990      2   1   2     5 2021-04-21   39       2
## 6251 6990.2021-04-22 6990      2   1   1     4 2021-04-22   39       3
## 6252 6990.2021-04-23 6990      2   1   1     4 2021-04-23   40       3
## 6253 6990.2021-04-24 6990      2   1   2     5 2021-04-24   40       2
## 6254 6990.2021-04-25 6990      2   2   2     6 2021-04-25   40       1
## 6255 6990.2021-04-26 6990      1   1   1     3 2021-04-26   40       3
## 6256 6990.2021-04-27 6990      1   1   1     3 2021-04-27   40       3
## 6257 6990.2021-04-28 6990      2   2   1     5 2021-04-28   40       1
## 6258 6990.2021-04-29 6990      2   1   1     4 2021-04-29   40       3
## 6259 6990.2021-04-30 6990      2   1   2     5 2021-04-30   41       2
## 6260 6990.2021-05-01 6990      2   2   1     5 2021-05-01   41       1
## 6261 6990.2021-05-02 6990      2   1   1     4 2021-05-02   41       3
## 6262 6990.2021-05-03 6990      1   0   1     2 2021-05-03   41       3
## 6263 6990.2021-05-04 6990      1   0   1     2 2021-05-04   41       3
## 6264 6990.2021-05-05 6990      1   0   1     2 2021-05-05   41       3
```

```
p <- ggplot(pr.feb$x[,1:2], aes(x=PC1, y=PC2, color=as.character(least_active_intra_day$cluster)))+
  geom_point() +
  labs(title=paste("Least Active Bird (",least_active,")"),x=paste("PC1 (",round(pr.feb.prop$importance
  scale_color_discrete(name = "Weeks", labels= c(as.character(paste(least_active_intra_day$week[least_ac
  theme_bw() +
  theme(legend.position = "bottom", legend.direction = "vertical",legend.text = element_text(size=8))
p
```



```
ggsave(p, filename = "../figures/least_active_bird_pca_all_rooms.png",device = "png",width = unit(7.5,"

cluster_table <- data.frame(cbind(as.numeric(km.out$cluster),least_active_intra_day$sample))
cluster_table[order(cluster_table$X1),]
```

```
##    X1              X2
## 1   1 6990.2021-03-09
## 2   1 6990.2021-03-10
## 3   1 6990.2021-03-11
## 4   1 6990.2021-03-12
## 5   1 6990.2021-03-13
## 7   1 6990.2021-03-15
## 8   1 6990.2021-03-16
## 9   1 6990.2021-03-17
## 10  1 6990.2021-03-18
## 11  1 6990.2021-03-19
## 15  1 6990.2021-03-23
## 16  1 6990.2021-03-24
## 17  1 6990.2021-03-25
## 18  1 6990.2021-03-26
## 19  1 6990.2021-03-27
## 20  1 6990.2021-03-28
## 21  1 6990.2021-03-29
## 24  1 6990.2021-04-01
## 25  1 6990.2021-04-02
## 26  1 6990.2021-04-03
## 27  1 6990.2021-04-04
## 28  1 6990.2021-04-05
## 29  1 6990.2021-04-06
## 30  1 6990.2021-04-07
## 31  1 6990.2021-04-08
## 34  1 6990.2021-04-11
## 39  1 6990.2021-04-16
## 42  1 6990.2021-04-19
## 43  1 6990.2021-04-20
## 48  1 6990.2021-04-25
## 51  1 6990.2021-04-28
## 54  1 6990.2021-05-01
## 6   2 6990.2021-03-14
## 12  2 6990.2021-03-20
## 22  2 6990.2021-03-30
## 23  2 6990.2021-03-31
## 38  2 6990.2021-04-15
## 40  2 6990.2021-04-17
## 44  2 6990.2021-04-21
## 47  2 6990.2021-04-24
## 53  2 6990.2021-04-30
## 13  3 6990.2021-03-21
## 14  3 6990.2021-03-22
## 32  3 6990.2021-04-09
## 33  3 6990.2021-04-10
## 35  3 6990.2021-04-12
## 36  3 6990.2021-04-13
## 37  3 6990.2021-04-14
## 41  3 6990.2021-04-18
## 45  3 6990.2021-04-22
## 46  3 6990.2021-04-23
## 49  3 6990.2021-04-26
## 50  3 6990.2021-04-27
## 52  3 6990.2021-04-29
## 55  3 6990.2021-05-02
## 56  3 6990.2021-05-03
## 57  3 6990.2021-05-04
## 58  3 6990.2021-05-05
```

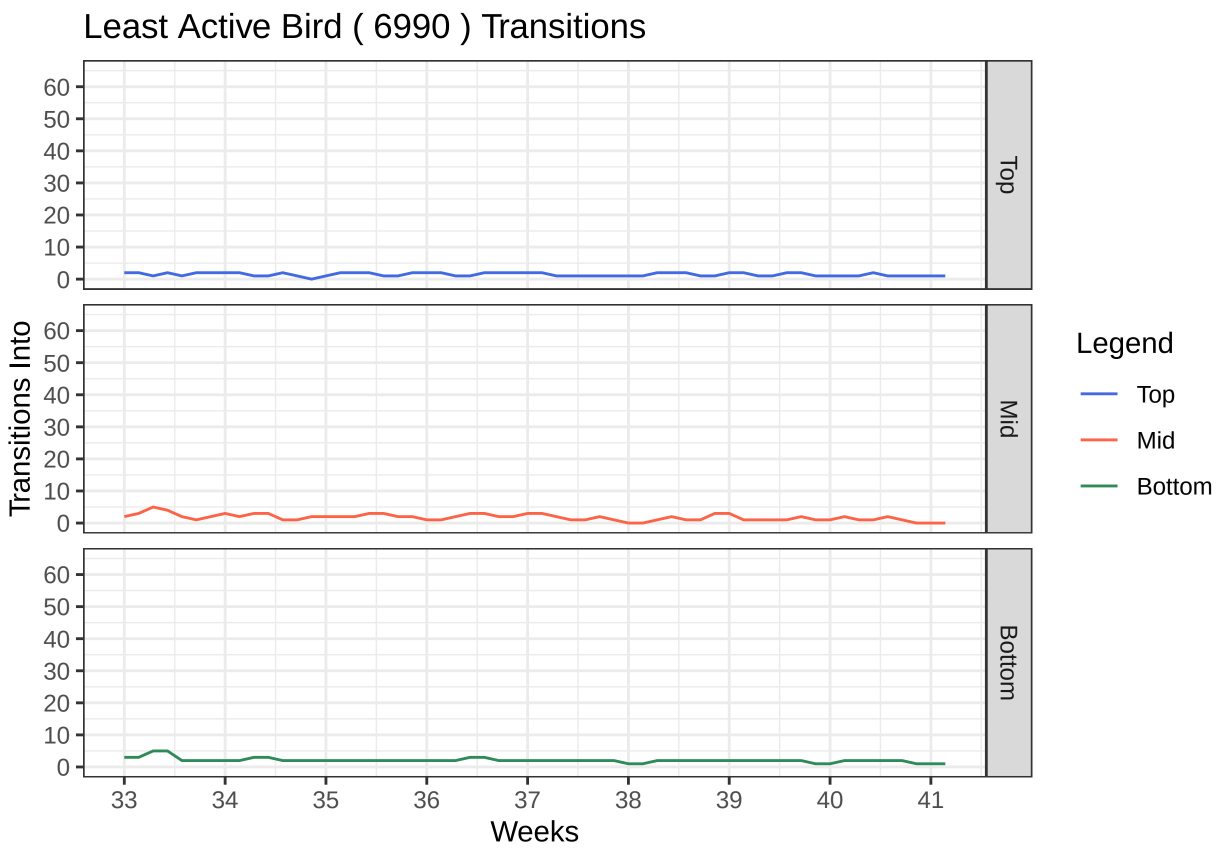
```
least_active_intra_day[order(least_active_intra_day$total),]
```

```
##                sample   ID bottom mid top total        day week cluster
## 6242 6990.2021-04-13 6990      1   0   1     2 2021-04-13   38       3
## 6243 6990.2021-04-14 6990      1   0   1     2 2021-04-14   38       3
## 6262 6990.2021-05-03 6990      1   0   1     2 2021-05-03   41       3
## 6263 6990.2021-05-04 6990      1   0   1     2 2021-05-04   41       3
## 6264 6990.2021-05-05 6990      1   0   1     2 2021-05-05   41       3
## 6255 6990.2021-04-26 6990      1   1   1     3 2021-04-26   40       3
## 6256 6990.2021-04-27 6990      1   1   1     3 2021-04-27   40       3
## 6219 6990.2021-03-21 6990      2   1   1     4 2021-03-21   35       3
## 6220 6990.2021-03-22 6990      2   2   0     4 2021-03-22   35       3
## 6238 6990.2021-04-09 6990      2   1   1     4 2021-04-09   38       3
## 6239 6990.2021-04-10 6990      2   1   1     4 2021-04-10   38       3
## 6241 6990.2021-04-12 6990      2   1   1     4 2021-04-12   38       3
## 6247 6990.2021-04-18 6990      2   1   1     4 2021-04-18   39       3
## 6251 6990.2021-04-22 6990      2   1   1     4 2021-04-22   39       3
## 6252 6990.2021-04-23 6990      2   1   1     4 2021-04-23   40       3
## 6258 6990.2021-04-29 6990      2   1   1     4 2021-04-29   40       3
## 6261 6990.2021-05-02 6990      2   1   1     4 2021-05-02   41       3
## 6211 6990.2021-03-13 6990      2   2   1     5 2021-03-13   34       1
## 6212 6990.2021-03-14 6990      2   1   2     5 2021-03-14   34       2
## 6218 6990.2021-03-20 6990      2   1   2     5 2021-03-20   35       2
## 6221 6990.2021-03-23 6990      2   2   1     5 2021-03-23   35       1
## 6226 6990.2021-03-28 6990      2   2   1     5 2021-03-28   36       1
## 6228 6990.2021-03-30 6990      2   1   2     5 2021-03-30   36       2
## 6229 6990.2021-03-31 6990      2   1   2     5 2021-03-31   36       2
## 6230 6990.2021-04-01 6990      2   2   1     5 2021-04-01   36       1
## 6237 6990.2021-04-08 6990      2   2   1     5 2021-04-08   37       1
## 6240 6990.2021-04-11 6990      2   2   1     5 2021-04-11   38       1
## 6244 6990.2021-04-15 6990      2   1   2     5 2021-04-15   38       2
## 6246 6990.2021-04-17 6990      2   1   2     5 2021-04-17   39       2
## 6250 6990.2021-04-21 6990      2   1   2     5 2021-04-21   39       2
## 6253 6990.2021-04-24 6990      2   1   2     5 2021-04-24   40       2
## 6257 6990.2021-04-28 6990      2   2   1     5 2021-04-28   40       1
## 6259 6990.2021-04-30 6990      2   1   2     5 2021-04-30   41       2
## 6260 6990.2021-05-01 6990      2   2   1     5 2021-05-01   41       1
## 6213 6990.2021-03-15 6990      2   2   2     6 2021-03-15   34       1
## 6215 6990.2021-03-17 6990      2   2   2     6 2021-03-17   34       1
## 6222 6990.2021-03-24 6990      2   2   2     6 2021-03-24   35       1
## 6223 6990.2021-03-25 6990      2   2   2     6 2021-03-25   35       1
## 6225 6990.2021-03-27 6990      2   3   1     6 2021-03-27   36       1
## 6227 6990.2021-03-29 6990      2   2   2     6 2021-03-29   36       1
## 6233 6990.2021-04-04 6990      2   2   2     6 2021-04-04   37       1
## 6234 6990.2021-04-05 6990      2   2   2     6 2021-04-05   37       1
## 6245 6990.2021-04-16 6990      2   2   2     6 2021-04-16   39       1
## 6248 6990.2021-04-19 6990      2   3   1     6 2021-04-19   39       1
## 6254 6990.2021-04-25 6990      2   2   2     6 2021-04-25   40       1
## 6207 6990.2021-03-09 6990      3   2   2     7 2021-03-09   33       1
## 6214 6990.2021-03-16 6990      2   3   2     7 2021-03-16   34       1
## 6216 6990.2021-03-18 6990      3   3   1     7 2021-03-18   34       1
## 6217 6990.2021-03-19 6990      3   3   1     7 2021-03-19   35       1
## 6224 6990.2021-03-26 6990      2   3   2     7 2021-03-26   36       1
## 6231 6990.2021-04-02 6990      3   3   1     7 2021-04-02   37       1
## 6235 6990.2021-04-06 6990      2   3   2     7 2021-04-06   37       1
## 6236 6990.2021-04-07 6990      2   3   2     7 2021-04-07   37       1
## 6249 6990.2021-04-20 6990      2   3   2     7 2021-04-20   39       1
## 6208 6990.2021-03-10 6990      3   3   2     8 2021-03-10   33       1
## 6232 6990.2021-04-03 6990      3   3   2     8 2021-04-03   37       1
## 6209 6990.2021-03-11 6990      5   5   1    11 2021-03-11   33       1
## 6210 6990.2021-03-12 6990      5   4   2    11 2021-03-12   34       1
```

```
least_active_intra_day
```

```
##               sample   ID bottom mid top total        day week cluster
## 6207 6990.2021-03-09 6990      3   2   2     7 2021-03-09   33       1
## 6208 6990.2021-03-10 6990      3   3   2     8 2021-03-10   33       1
## 6209 6990.2021-03-11 6990      5   5   1    11 2021-03-11   33       1
## 6210 6990.2021-03-12 6990      5   4   2    11 2021-03-12   34       1
## 6211 6990.2021-03-13 6990      2   2   1     5 2021-03-13   34       1
## 6212 6990.2021-03-14 6990      2   1   2     5 2021-03-14   34       2
## 6213 6990.2021-03-15 6990      2   2   2     6 2021-03-15   34       1
## 6214 6990.2021-03-16 6990      2   3   2     7 2021-03-16   34       1
## 6215 6990.2021-03-17 6990      2   2   2     6 2021-03-17   34       1
## 6216 6990.2021-03-18 6990      3   3   1     7 2021-03-18   34       1
## 6217 6990.2021-03-19 6990      3   3   1     7 2021-03-19   35       1
## 6218 6990.2021-03-20 6990      2   1   2     5 2021-03-20   35       2
## 6219 6990.2021-03-21 6990      2   1   1     4 2021-03-21   35       3
## 6220 6990.2021-03-22 6990      2   2   0     4 2021-03-22   35       3
## 6221 6990.2021-03-23 6990      2   2   1     5 2021-03-23   35       1
## 6222 6990.2021-03-24 6990      2   2   2     6 2021-03-24   35       1
## 6223 6990.2021-03-25 6990      2   2   2     6 2021-03-25   35       1
## 6224 6990.2021-03-26 6990      2   3   2     7 2021-03-26   36       1
## 6225 6990.2021-03-27 6990      2   3   1     6 2021-03-27   36       1
## 6226 6990.2021-03-28 6990      2   2   1     5 2021-03-28   36       1
## 6227 6990.2021-03-29 6990      2   2   2     6 2021-03-29   36       1
## 6228 6990.2021-03-30 6990      2   1   2     5 2021-03-30   36       2
## 6229 6990.2021-03-31 6990      2   1   2     5 2021-03-31   36       2
## 6230 6990.2021-04-01 6990      2   2   1     5 2021-04-01   36       1
## 6231 6990.2021-04-02 6990      3   3   1     7 2021-04-02   37       1
## 6232 6990.2021-04-03 6990      3   3   2     8 2021-04-03   37       1
## 6233 6990.2021-04-04 6990      2   2   2     6 2021-04-04   37       1
## 6234 6990.2021-04-05 6990      2   2   2     6 2021-04-05   37       1
## 6235 6990.2021-04-06 6990      2   3   2     7 2021-04-06   37       1
## 6236 6990.2021-04-07 6990      2   3   2     7 2021-04-07   37       1
## 6237 6990.2021-04-08 6990      2   2   1     5 2021-04-08   37       1
## 6238 6990.2021-04-09 6990      2   1   1     4 2021-04-09   38       3
## 6239 6990.2021-04-10 6990      2   1   1     4 2021-04-10   38       3
## 6240 6990.2021-04-11 6990      2   2   1     5 2021-04-11   38       1
## 6241 6990.2021-04-12 6990      2   1   1     4 2021-04-12   38       3
## 6242 6990.2021-04-13 6990      1   0   1     2 2021-04-13   38       3
## 6243 6990.2021-04-14 6990      1   0   1     2 2021-04-14   38       3
## 6244 6990.2021-04-15 6990      2   1   2     5 2021-04-15   38       2
## 6245 6990.2021-04-16 6990      2   2   2     6 2021-04-16   39       1
## 6246 6990.2021-04-17 6990      2   1   2     5 2021-04-17   39        2
## 6247 6990.2021-04-18 6990      2   1   1     4 2021-04-18   39        3
## 6248 6990.2021-04-19 6990      2   3   1     6 2021-04-19   39        1
## 6249 6990.2021-04-20 6990      2   3   2     7 2021-04-20   39        1
## 6250 6990.2021-04-21 6990      2   1   2     5 2021-04-21   39        2
## 6251 6990.2021-04-22 6990      2   1   1     4 2021-04-22   39        3
## 6252 6990.2021-04-23 6990      2   1   1     4 2021-04-23   40        3
## 6253 6990.2021-04-24 6990      2   1   2     5 2021-04-24   40        2
## 6254 6990.2021-04-25 6990      2   2   2     6 2021-04-25   40        1
## 6255 6990.2021-04-26 6990      1   1   1     3 2021-04-26   40        3
## 6256 6990.2021-04-27 6990      1   1   1     3 2021-04-27   40        3
## 6257 6990.2021-04-28 6990      2   2   1     5 2021-04-28   40        1
## 6258 6990.2021-04-29 6990      2   1   1     4 2021-04-29   40        3
## 6259 6990.2021-04-30 6990      2   1   2     5 2021-04-30   41        2
## 6260 6990.2021-05-01 6990      2   2   1     5 2021-05-01   41        1
## 6261 6990.2021-05-02 6990      2   1   1     4 2021-05-02   41        3
## 6262 6990.2021-05-03 6990      1   0   1     2 2021-05-03   41        3
## 6263 6990.2021-05-04 6990      1   0   1     2 2021-05-04   41        3
## 6264 6990.2021-05-05 6990      1   0   1     2 2021-05-05   41        3
```

```
least_active_grp <- data.frame(x = as.Date(least_active_intra_day$day), y = c(least_active_intra_day$bot



q <-ggplot(least_active_grp, aes(x,y,col=group))+
  geom_line() +
  facet_grid(factor(group,levels=c("Top","Mid","Bottom"))~ .) +
  labs(title=paste("Least Active Bird (",least_active,") Transitions"), color="Legend")+
  scale_x_date(name="Weeks",breaks = seq(min(least_active_grp$x),max(least_active_grp$x),by="week"),labe
  scale_y_continuous(name="Transitions Into",n.breaks = 8,limits = c(0,65)) +
  scale_color_manual(values=c("Top"="royalblue","Mid"="tomato","Bottom" = "seagreen")) +
  theme_bw()
q
```



```
ggsave(q, filename="../figures/least_active_transitions_per_day_all_rooms.png",device="png")
```

```
## Saving 6.5 x 4.5 in image
```

Organize Birds By activity

```
library(qpcR)

trans_reslists <- c(trans_reslist_room_2,trans_reslist_room_3,trans_reslist_room_8,trans_reslist_room_1
#trans_reslists <- c(trans_reslist_room_2,trans_reslist_room_8,trans_reslist_room_11)
daily_colsum <- data.frame()

#each item is one transition list for an intra bird sample
#we want to sum up the columns for each day and compare bird to bird


for(i in 1:length(trans_reslists)){
  res <- colSums(sapply(trans_reslists[[i]]$daily[,2:6], as.numeric))
  res[1] <- res[1]/length(trans_reslists[[i]]$daily$ID)
  res[6] <- round(res[2]/length(trans_reslists[[i]]$daily$bottom),0)
  res[7] <- round(res[3]/length(trans_reslists[[i]]$daily$mid),0)
  res[8] <- round(res[4]/length(trans_reslists[[i]]$daily$top),0)
  res[9] <- round(res[5]/length(trans_reslists[[i]]$daily$total),0)
  daily_colsum <- rbind(daily_colsum,res)
}
```

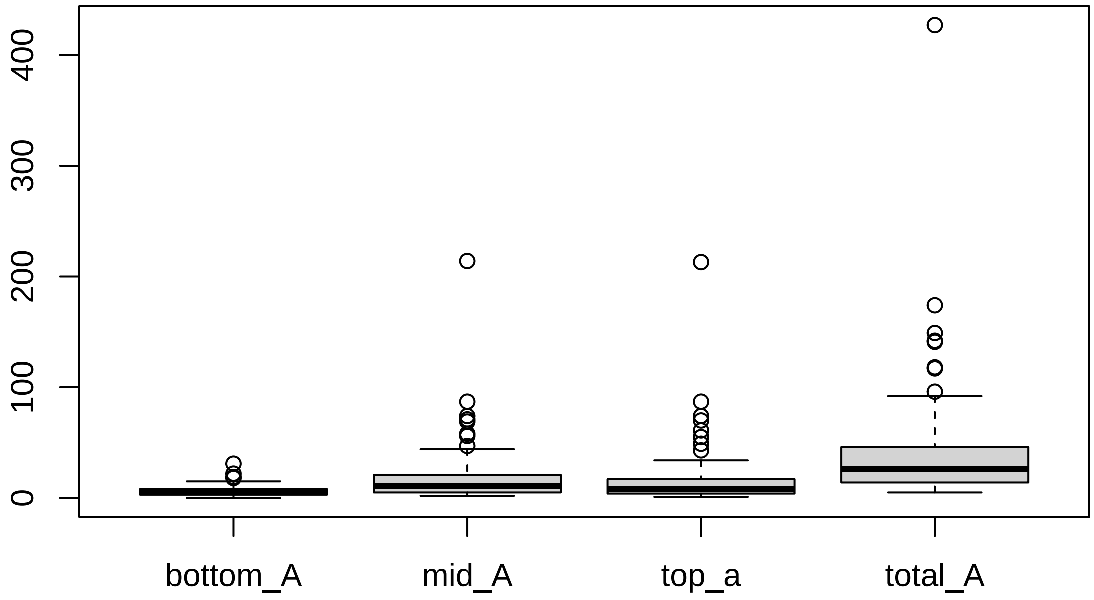
```
colnames(daily_colsum) <- c("ID","bottom","mid","top","total","bottom_A","mid_A","top_a","total_A")
daily_colsum
```

```
##      ID bottom   mid   top total bottom_A mid_A top_a total_A
## 1  6855     95   719   748  1562        2    12    13      27
## 2  6860    238   196   182   616        4     3     3      11
## 3  6872    283   713   633  1629        5    12    11      28
## 4  6877    334   533   459  1326        6     9     8      23
## 5  6890      9   685   674  1368        0    12    12      24
## 6  6901     20  1492  1491  3003        0    26    26      52
## 7  6903    472   678   563  1713        8    12    10      30
## 8  6908    229   244   196   669        4     4     3      12
## 9  6909    181   146    87   414        3     3     2       7
## 10 6911    392   364   192   948        7     6     3      16
## 11 6919     43   990   977  2010        1    17    17      35
## 12 6925    278   548   464  1290        5     9     8      22
## 13 6926   1109  1971  1768  4848       19    34    30      84
## 14 6935    428   500   358  1286        7     9     6      22
## 15 6937    485   490   393  1368        8     8     7      24
## 16 6942    166   141   140   447        3     2     2       8
## 17 6946    146   508   473  1127        3     9     8      19
## 18 6952    211   358   341   910        4     6     6      16
## 19 6955    412   352   353  1117        7     6     6      19
## 20 6956     34  1052  1044  2130        1    18    18      37
## 21 6958    190   193   182   565        3     3     3      10
## 22 6960    377   880   799  2056        6    15    14      35
## 23 6962    539   577   421  1537        9    10     7      26
## 24 6971    238   261   119   618        4     4     2      11
## 25 6975    154   692   680  1526        3    12    12      26
## 26 6978    446   464   347  1257        8     8     6      22
## 27 6981    538   623   492  1653        9    11     8      28
## 28 6983    166   156   151   473        3     3     3       8
## 29 6986    223   200   130   553        4     3     2      10
## 30 6988    126   150   115   391        2     3     2       7
## 31 6989    325   391   312  1028        6     7     5      18
## 32 9001    413   390   267  1070        7     7     5      18
## 33 9005    487   763   729  1979        8    13    13      34
## 34 9019    245   481   437  1163        4     8     8      20
## 35 9021    821  1619  1220  3660       14    28    21      63
## 36 9024    118   221   168   507        2     4     3       9
## 37 9026    501   435   474  1410        9     8     8      24
## 38 6736     41  1921  1900  3862        1    33    33      67
## 39 6883    551   771   426  1748       10    13     7      30
## 40 6902    656  1625  1213  3494       11    28    21      60
## 41 6905     28 12384 12373 24785        0   214   213     427
## 42 6907    380  1315  1142  2837        7    23    20      49
## 43 6918    569  1329   985  2883       10    23    17      50
## 44 6921      8  4310  4314  8632        0    74    74     149
## 45 6922    549   775   417  1741        9    13     7      30
## 46 6924    517  1885  1512  3914        9    32    26      67
## 47 6927    824  2557  1955  5336       14    44    34      92
## 48 6933    848  1016   295  2159       15    18     5      37
## 49 6934   1024  1915  1213  4152       18    33    21      72
## 50  6938     73  1720  1674  3467        1    30    29      60
## 51  6943    573   801   413  1787       10    14     7      31
## 52  6944    306   749   550  1605        5    13     9      28
## 53  6947    856  1191   474  2521       15    21     8      43
## 54  6959    692  1577  1122  3391       12    27    19      58
## 55  6963    317  1422  1167  2906        5    25    20      50
## 56  6967     97  4110  4041  8248        2    71    70     142
## 57  6969    307   950   796  2053        5    16    14      35
## 58  6972    593  1232   843  2668       10    21    15      46
## 59  6976   1272  2424  1460  5156       22    42    25      89
## 60  6977    367  2710  2508  5585        6    47    43      96
## 61  6980    628  4011  3537  8176       11    69    61     141
## 62  6992    743  1180   573  2496       13    20    10      43
## 63  6998     24  5024  5022 10070        0    87    87     174
## 64  9004    627  2424  2001  5052       11    42    34      87
## 65  9010    697  3274  2842  6813       12    56    49     117
## 66  9013    297  3389  3183  6869        5    58    55     118
## 67  9016    776  2335  1975  5086       13    40    34      88
## 68  9018    222   662   488  1372        4    11     8      24
## 69  6886    302   823   796  1921        5    14    14      33
## 70  6913    455   979   641  2075        8    17    11      36
## 71  6917    187   299   330   816        3     5     6      14
## 72  6923    454   593   518  1565        8    10     9      27
## 73  6928    169   177   155   501        3     3     3       9
## 74  6929    118   129   123   370        2     2     2       6
## 75  6931    159   200   164   523        3     3     3       9
## 76  6939    154   141    89   384        3     2     2       7
## 77  6940    384   438   270  1092        7     8     5      19
## 78  6941    271   786   715  1772        5    14    12      31
## 79  6945    181   158   152   491        3     3     3       8
## 80  6950    170   355   312   837        3     6     5      14
## 81  6951   1822  2188   937  4947       31    38    16      85
## 82  6957    229   314   197   740        4     5     3      13
## 83  6961    205   137   147   489        4     2     3       8
## 84  6964    194   264   204   662        3     5     4      11
## 85  6970    878  1456   800  3134       15    25    14      54
## 86  6973    744   572   727  2043       13    10    13      35
## 87  6979     75   543   493  1111        1     9     8      19
## 88  6982     28   799   800  1627        0    14    14      28
## 89  6984     14  1088  1072  2174        0    19    18      37
## 90  6985    327   544   535  1406        6     9     9      24
## 91  6993     70   256   254   580        1     4     4      10
## 92  6994    156   157   163   476        3     3     3       8
## 93  6996    163   136   119   418        3     2     2       7
## 94  6997     81   194   171   446        1     3     3       8
## 95  7000    510   570   198  1278        9    10     3      22
## 96  9012     37   815   803  1655        1    14    14      29
## 97  9014    247   266   135   648        4     5     2      11
## 98  9017    785  1346  1011  3142       14    23    17      54
## 99  9020    263  1298  1222  2783        5    22    21      48
## 100 9022     20  1292  1271  2583        0    22    22      45
## 101 9023    232   344   400   976        4     6     7      17
## 102 9025     73   412   427   912        1     7     7      16
## 103 6853    702  1077  1135  2914       12    19    20      50
## 104 6904     21  1336  1322  2679        0    23    23      46
## 105 6916    283   306   150   739        5     5     3      13
## 106 6932    216   290   273   779        4     5     5      13
## 107 6965    354   975   834  2163        6    17    14      37
## 108 6990    121   103    84   308        2     2     1       5
## 109 6999    536   684   506  1726        9    12     9      30
## 110 9002     30  1870  1856  3756        1    32    32      65
## 111 9003    443   621   573  1637        8    11    10      28
## 112 9006    327   438   439  1204        6     8     8      21
## 113 9007    419   642   412  1473        7    11     7      25
## 114 9015    236   381   272   889        4     7     5      15
## 115 9023    293   352   192   837        5     6     3      14
## 116 9027    279   652   586  1517        5    11    10      26
## 117 9030    131  1802  1730  3663        2    31    30      63
## 118 9032    163   142    85   390        3     2     1       7
## 119 9033    388   523   338  1249        7     9     6      22
## 120 9035    160   762   730  1652        3    13    13      28
## 121 9038    254   672   578  1504        4    12    10      26
## 122 9041     29  1542  1526  3097        0    27    26      53
## 123 9042    194   178    95   467        3     3     2       8
## 124 9043    128   173   157   458        2     3     3       8
## 125 9044    261   653   627  1541        4    11    11      27
## 126 9045    435  1097   870  2402        8    19    15      41
## 127 9046    187   148   156   491        3     3     3       8
## 128 9047    179   227   181   587        3     4     3      10
## 129 9048    517   545   138  1200        9     9     2      21
## 130 9049    229   248   253   730        4     4     4      13
## 131 9050    400   420   238  1058        7     7     4      18
## 132 9052    282   299   148   729        5     5     3      13
## 133 9053    256   234    72   562        4     4     1      10
## 134 9054    449   910   757  2116        8    16    13      36
## 135 9055    446   500   278  1224        8     9     5      21
## 136 9056     15  1373  1361  2749        0    24    23      47
## 137 9058    437   467   256  1160        8     8     4      20
## 138 9061     15  1078  1078  2171        0    19    19      37
```

```
hold <- daily_colsum[order(daily_colsum$total_A,decreasing = T),]

summary(hold[,6:9])
```

```
##     bottom_A          mid_A            top_a          total_A
##  Min.   : 0.000   Min.   :  2.00   Min.   :  1.0   Min.   :  5.00
##  1st Qu.: 3.000   1st Qu.:  5.25   1st Qu.:  4.0   1st Qu.: 14.00
##  Median : 5.000   Median : 11.00   Median :  8.0   Median : 26.00
##  Mean   : 5.862   Mean   : 17.22   Mean   : 14.7   Mean   : 37.77
##  3rd Qu.: 8.000   3rd Qu.: 21.00   3rd Qu.: 17.0   3rd Qu.: 45.75
##  Max.   :31.000   Max.   :214.00   Max.   :213.0   Max.   :427.00
```

```
boxplot(hold[,6:9])
```



```
low_act <- c()
med_act <- c()
high_act <- c()

#TODO Check these values
for(i in 1:length(hold[,1])){
  cur_tol <- hold[i,9]
  if(cur_tol<=14){
    low_act <- c(low_act,hold[i,1])
  }
  if((cur_tol>14) & (cur_tol<=45)){
    med_act <- c(med_act,hold[i,1])
  }
  if(cur_tol>45){
    high_act <- c(high_act, hold[i,1])
  }
}


cluster_table <- data.frame(qpcR:::cbind.na(low_act,med_act,high_act))
cluster_table
```

```
##   low_act med_act high_act
## 1    6917    9022     6905
## 2    6950    6947     6998
## 3    9023    6992     6921
## 4     6957    9045     6967
## 5     6916    6956     6980
## 6     6932    6933     9013
## 7     9049    6984     9010
## 8     9052    6965     6977
## 9     6908    9061     6927
## 10    6860    6913     6976
## 11    6971    9054     9016
## 12    6964    6919     9004
## 13    9014    6960     6951
## 14    6958    6969     6926
## 15    6986    6973     6934
## 16    6993    9005     6736
## 17    9047    6886     6924
## 18    9053    6943     9002
## 19    9024    6941     9021
## 20    6928    6903     9030
## 21    6931    6883     6902
## 22    6942    6922     6938
## 23    6983    6999     6959
## 24    6945    9012     6970
## 25    6961    6872     9017
## 26    6994    6981     9041
## 27    6997    6944     6901
## 28    9042    6982     6918
## 29    9043    9003     6963
## 30    9046    9035     6853
## 31    6909    6855     6907
## 32    6988    6923     9020
## 33    6939    9044     9056
## 34    6996    6962     6972
## 35    9032    6975     6904
## 36    6929    9027       NA
## 37    6990    9038       NA
## 38      NA    9007       NA
## 39      NA    6890       NA
## 40      NA    6937       NA
## 41      NA    9026       NA
## 42      NA    9018       NA
## 43      NA    6985       NA
## 44      NA    6877       NA
## 45      NA    6925       NA
## 46      NA    6935       NA
## 47      NA    6978       NA
## 48      NA    7000       NA
## 49      NA    9033       NA
## 50      NA    9006       NA
## 51      NA    9048       NA
## 52      NA    9055       NA
## 53      NA    9019       NA
## 54      NA    9058       NA
## 55      NA    6946       NA
## 56      NA    6955       NA
## 57      NA    6940       NA
## 58      NA    6979       NA
## 59      NA    6989       NA
## 60      NA    9001       NA
## 61      NA    9050       NA
## 62      NA    9023       NA
## 63      NA    6911       NA
## 64      NA    6952       NA
## 65      NA    9025       NA
## 66      NA    9015       NA
```

```
write.csv(cluster_table, "../output/organized_birds_by_activity_daily_all_rooms.csv",row.names = F)
```

```
# find which room each ID is in:
updated_table <- data.frame()

all_selected_ids <- c(low_act,med_act,high_act)
new_row <- ""
activity <- ""
room <- ""

for(id in all_selected_ids){
  #find out which activity level id is in
  if(id %in% low_act){
      activity <- "low"
  } else if( id %in% med_act){
      activity <- "medium"
  } else {
      activity <- "high"
  }
  #find out which room id is in
  if(id %in% bird_ids_room_2){
      room <- "2"
  } else if(id %in% bird_ids_room_3 ){
      room <- "3"
  } else if( id %in% bird_ids_room_8 ){
      room <- "8"
  } else if( id %in% bird_ids_room_11 ) {
      room <- "11"
  }
  #add row to updated table
  new_row <- c(room, id, activity)
  updated_table <- rbind(updated_table, new_row)
}

colnames(updated_table) <- c("Room", "Bird ID", "Activity Level")
write.csv(updated_table, "../output/organized_birds_by_activity_daily_all_rooms_nice_table.csv",row.name
```

Jumpy table for the separated samples

```
# Aggregate the low activity birds
low_act_bird_ids <- na.trim(cluster_table$low_act)

#make a table of the samples
low_act_bird_frame <- data.frame()
```

```r
for(id in low_act_bird_ids){
  curr_bird <- daily[grep(id,daily$sample),]
  low_act_bird_frame <- rbind(low_act_bird_frame,curr_bird)
}

low_act_bird_frame[,3:6] <- sapply(low_act_bird_frame[,3:6], as.numeric)
low_act_bird_frame["day"] <- as.character(str_split_fixed(low_act_bird_frame$sample, ".",6)[,6])
low_act_agg_tab <- aggregate(low_act_bird_frame[,3:6],by = list(low_act_bird_frame$day), FUN = mean)
colnames(low_act_agg_tab) <- c("day","bottom","mid","top","total")

# Aggregate the medium activity birds
med_act_bird_ids <- na.trim(cluster_table$med_act)

#make a table of the samples
med_act_bird_frame <- data.frame()

for(id in med_act_bird_ids){
  curr_bird <- daily[grep(id,daily$sample),]
  med_act_bird_frame <- rbind(med_act_bird_frame,curr_bird)
}

med_act_bird_frame[,3:6] <- sapply(med_act_bird_frame[,3:6], as.numeric)
med_act_bird_frame["day"] <- as.character(str_split_fixed(med_act_bird_frame$sample, ".",6)[,6])
med_act_agg_tab <- aggregate(med_act_bird_frame[,3:6],by = list(med_act_bird_frame$day), FUN = mean)
colnames(med_act_agg_tab) <- c("day","bottom","mid","top","total")

# Aggregate the high activity birds
high_act_bird_ids <- na.trim(cluster_table$high_act)

#make a table of the samples
high_act_bird_frame <- data.frame()

for(id in high_act_bird_ids){
  curr_bird <- daily[grep(id,daily$sample),]
  high_act_bird_frame <- rbind(high_act_bird_frame,curr_bird)
}

high_act_bird_frame[,3:6] <- sapply(high_act_bird_frame[,3:6], as.numeric)
high_act_bird_frame["day"] <- as.character(str_split_fixed(high_act_bird_frame$sample, ".",6)[,6])
high_act_agg_tab <- aggregate(high_act_bird_frame[,3:6],by = list(high_act_bird_frame$day), FUN = mean)
colnames(high_act_agg_tab) <- c("day","bottom","mid","top","total")

# plot the low activity birds
low_act_grp <- data.frame(x = as.Date(low_act_agg_tab$day), y = c(low_act_agg_tab$bottom, low_act_agg_ta

med_act_grp <- data.frame(x = as.Date(med_act_agg_tab$day), y = c(med_act_agg_tab$bottom, med_act_agg_ta

# plot the low activity birds
high_act_grp <- data.frame(x = as.Date(high_act_agg_tab$day), y = c(high_act_agg_tab$bottom, high_act_ag

combo_table <- rbind(low_act_grp, med_act_grp, high_act_grp)

q <-ggplot(combo_table, aes(x,y,col=activity))+
  geom_line(aes(col=activity)) +
  facet_grid(factor(group,levels=c("Top","Mid","Bottom"))~ ., scales = "free_y") +
  labs(title=paste0("Activity Birds Transitions (n=",length(unique(daily$ID)),")"), color="Activity Leve
  scale_x_date(name="Weeks",breaks = seq(min(combo_table$x),max(combo_table$x),by="week"),labels = as.nu
  scale_y_continuous(name="Average Transitions Into",n.breaks = 8) +
  scale_color_manual(values=c("Low"="royalblue","Medium"="tomato","High" = "seagreen")) +
  theme_bw()
  ggsave(q, filename="../figures/active_transitions_all_rooms_set_2.png",device="png")
```

```
## Saving 6.5 x 4.5 in image
```

Zone Occupying Table for all rooms

```
library(dplyr)

all_room_zone_table <- data.frame()
result <- ""

for(item in trans_reslist_room_2){
  result <- calc_zone_duration("room 2", item$ID, item$raw)
  all_room_zone_table <- rbind(all_room_zone_table, result)
}
for(item in trans_reslist_room_3){
  result <- calc_zone_duration("room 3", item$ID, item$raw)
  all_room_zone_table <- rbind(all_room_zone_table, result)
}
for(item in trans_reslist_room_8){
  result <- calc_zone_duration("room 8", item$ID, item$raw)
  all_room_zone_table <- rbind(all_room_zone_table, result)
}
for(item in trans_reslist_room_11){
  result <- calc_zone_duration("room 11", item$ID, item$raw)
  all_room_zone_table <- rbind(all_room_zone_table, result)
}

# change the resultant times to numerics as opposed to chars
all_room_zone_table[,3:5] <- sapply(all_room_zone_table[,3:5],function(x) as.numeric(as.character(x)))
all_room_zone_table[,3:5] <- (all_room_zone_table[,3:5])/60
# calculate the number of days observed
all_room_zone_table["days_observed"]<- round((rowSums(all_room_zone_table[,3:5])/60)/24)
colnames(all_room_zone_table) <-  c("Room","ID", "Time in Bottom","Time in Middle", "Time in Top","Days


write.csv(all_room_zone_table, "../output/zone_time_per_id_all_rooms.csv", row.names=F)

ave_time_per_room <- aggregate(cbind(all_room_zone_table$"Time in Bottom",all_room_zone_table$"Time in 

colnames(ave_time_per_room) <- c("Room", "Average Time in Bottom", "Average Time in Middle", "Average Ti

write.csv(ave_time_per_room, "../output/zone_time_per_room_all_rooms.csv", row.names=F)
```

Intra-Bird Analysis For Longer Series

```
remove <- c("6910","6966","6914")
bird_ids_room_2_new <- bird_ids_room_2 [! bird_ids_room_2  %in% remove ]
#generate list of the first 10 truncated XTS objects
reslist2_room_2 <- list()
for(i in 1:length(bird_ids_room_2_new)){
  res <- sep_bird_id_period(samp_name="room 2",cage_obj =room_2, samp_id = bird_ids_room_2_new[i],cutof
  reslist2_room_2[[i]] <- res
}

remove <-c("6949","6936","6974","9009","9011","6995")
bird_ids_room_8_new <- bird_ids_room_8 [! bird_ids_room_8  %in% remove ]
#generate list of the first 10 truncated XTS objects
reslist2_room_8 <- list()
for(i in 1:length(bird_ids_room_8_new)){
  res <- sep_bird_id_period(samp_name="room 8",cage_obj =room_8, samp_id = bird_ids_room_8_new[i],cutof
  reslist2_room_8[[i]] <- res
}

remove <- c("6888","6868","6880","6881","6893","6892", "6894","6862","6898","6882","6899","6887","6896"
bird_ids_room_11_new <- bird_ids_room_11 [! bird_ids_room_11  %in% remove ]
#generate list of the first 10 truncated XTS objects
reslist2_room_11 <- list()
for(i in 1:length(bird_ids_room_11_new)){
  res <- sep_bird_id_period(samp_name="room 11",cage_obj =room_11, samp_id = bird_ids_room_11_new[i], c
  reslist2_room_11[[i]] <- res
}
```

```
trans_reslist_room_2 <- list()

for(i in 1:length(reslist2_room_2)){

  daily_trans_table <- calc_trans_period(reslist2_room_2[[i]]$ID,reslist2_room_2[[i]]$daily_obj,reslist
  weekly_trans_table <- calc_trans_period(reslist2_room_2[[i]]$ID,reslist2_room_2[[i]]$weekly_obj,reslis
  monthly_trans_table <- calc_trans_period(reslist2_room_2[[i]]$ID,reslist2_room_2[[i]]$monthly_obj,resl

  result <- list("ID"=reslist2_room_2[[i]]$ID,"daily"=daily_trans_table, "weekly"=weekly_trans_table,"m
  trans_reslist_room_2[[i]] <- result
}

room_2_daily <- data.frame()
room_2_weekly <- data.frame()
room_2_monthly <- data.frame()
#TODO select sample as opposed to ID for intra bird comparison (alpha eq)
for(item in trans_reslist_room_2){
  room_2_daily <- rbind(room_2_daily, item$daily[c("sample","bottom","mid","top","total")])
  room_2_weekly <- rbind(room_2_weekly, item$weekly[c("sample","bottom","mid","top","total")])
  room_2_monthly <- rbind(room_2_monthly, item$monthly[c("sample","bottom","mid","top","total")])
}

# check to make sure only one bird is selected
print("check that the id and RFID tag are unique for each item entry")
```

```
## [1] "check that the id and RFID tag are unique for each item entry"
```

```r
#this is following up with the issues of the intra-bird comparisons
for(item in trans_reslist_room_2){
  print(paste(unique(item$raw$tagnumber)," ",unique(item$raw$tagname)))
  #print(item)
}
```

```
## [1] "   6855"
## [1] "   6860"
## [1] "   6872"
## [1] "   6877"
## [1] "   6890"
## [1] "   6901"
## [1] "   6903"
## [1] "   6908"
## [1] "   6909"
## [1] "   6911"
## [1] "   6919"
## [1] "   6925"
## [1] "   6926"
## [1] "   6935"
## [1] "   6937"
## [1] "   6942"
## [1] "   6946"
## [1] "   6952"
## [1] "   6955"
## [1] "   6956"
## [1] "   6958"
## [1] "   6960"
## [1] "   6962"
## [1] "   6971"
## [1] "   6975"
## [1] "   6978"
## [1] "   6981"
## [1] "   6983"
## [1] "   6986"
## [1] "   6988"
## [1] "   6989"
## [1] "   9001"
## [1] "   9005"
## [1] "   9019"
## [1] "   9021"
## [1] "   9024"
## [1] "   9026"
```

```r
trans_reslist_room_8 <- list()

for(i in 1:length(reslist2_room_8)){

  daily_trans_table <- calc_trans_period(reslist2_room_8[[i]]$ID,reslist2_room_8[[i]]$daily_obj,reslist2
  weekly_trans_table <- calc_trans_period(reslist2_room_8[[i]]$ID,reslist2_room_8[[i]]$weekly_obj,reslis
  monthly_trans_table <- calc_trans_period(reslist2_room_8[[i]]$ID,reslist2_room_8[[i]]$monthly_obj,resl

  result <- list("ID"=reslist2_room_8[[i]]$ID,"daily"=daily_trans_table, "weekly"=weekly_trans_table,"m
  trans_reslist_room_8[[i]] <- result
}

room_8_daily <- data.frame()
room_8_weekly <- data.frame()
room_8_monthly <- data.frame()
#TODO select sample as opposed to ID for intra bird comparison (alpha eq)
for(item in trans_reslist_room_8){
  room_8_daily <- rbind(room_8_daily, item$daily[c("sample","bottom","mid","top","total")])
  room_8_weekly <- rbind(room_8_weekly, item$weekly[c("sample","bottom","mid","top","total")])
  room_8_monthly <- rbind(room_8_monthly, item$monthly[c("sample","bottom","mid","top","total")])
}

# check to make sure only one bird is selected
print("check that the id and RFID tag are unique for each item entry")
```

```
## [1] "check that the id and RFID tag are unique for each item entry"
```

```
#this is following up with the issues of the intra-bird comparisons
for(item in trans_reslist_room_8){
  print(paste(unique(item$raw$tagnumber)," ",unique(item$raw$tagname)))
  #print(item)
}
```

```
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-17-81   6886"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-09-71   6913"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-12-26   6917"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-09-79   6923"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-00-51   6928"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-05-45   6929"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-02-64   6931"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-09-43   6939"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-04-31   6940"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-09-76   6941"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-05-11   6945"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-02-80   6950"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-19-33   6951"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-17-70   6957"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-04-04   6961"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-12-06   6964"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-13-50   6970"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-03-55   6973"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-08-45   6979"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-01-08   6982"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-07-67   6984"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-18-97   6985"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-19-50   6993"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-15-34   6994"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-02-40   6996"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-02-89   6997"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-03-59   7000"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-09-74   9012"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-14-48   9014"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-10-60   9017"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-16-65   9020"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-10-71   9022"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-01-86   9023"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-11-80   9025"
```

```
trans_reslist_room_11 <- list()

for(i in 1:length(reslist2_room_11)){

  daily_trans_table <- calc_trans_period(reslist2_room_11[[i]]$ID,reslist2_room_11[[i]]$daily_obj,reslis
  weekly_trans_table <- calc_trans_period(reslist2_room_11[[i]]$ID,reslist2_room_11[[i]]$weekly_obj,resl
  monthly_trans_table <- calc_trans_period(reslist2_room_11[[i]]$ID,reslist2_room_11[[i]]$monthly_obj,re

  result <- list("ID"=reslist2_room_11[[i]]$ID,"daily"=daily_trans_table, "weekly"=weekly_trans_table,"
  trans_reslist_room_11[[i]] <- result
}

room_11_daily <- data.frame()
room_11_weekly <- data.frame()
room_11_monthly <- data.frame()
#TODO select sample as opposed to ID for intra bird comparison (alpha eq)
for(item in trans_reslist_room_11){
  room_11_daily <- rbind(room_11_daily, item$daily[c("sample","bottom","mid","top","total")])
  room_11_weekly <- rbind(room_11_weekly, item$weekly[c("sample","bottom","mid","top","total")])
  room_11_monthly <- rbind(room_11_monthly, item$monthly[c("sample","bottom","mid","top","total")])
}

# check to make sure only one bird is selected
print("check that the id and RFID tag are unique for each item entry")
```

```
## [1] "check that the id and RFID tag are unique for each item entry"
```

```
#this is following up with the issues of the intra-bird comparisons
for(item in trans_reslist_room_11){
  print(paste(unique(item$raw$tagnumber)," ",unique(item$raw$tagname)))
  #print(item)
}
```

```
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-03-53   6853"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-10-20   6904"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-07-78   6916"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-19-12   6932"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-18-18   6965"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-16-24   6990"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-13-26   6999"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-09-59   9002"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-07-17   9003"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-03-56   9006"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-19-56   9007"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-07-37   9015"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-02-51   9023"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-06-94   9027"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-08-65   9030"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-09-23   9032"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-18-43   9033"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-19-81   9035"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-02-02   9038"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-02-83   9041"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-01-39   9042"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-03-50   9043"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-17-46   9044"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-11-55   9045"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-07-97   9046"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-08-00   9047"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-07-07   9048"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-15-42   9049"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-01-30   9050"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-03-35   9052"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-10-86   9053"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-19-39   9054"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-10-30   9055"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-10-89   9056"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-03-15   9058"
## [1] "E2-00-9A-01-20-03-9A-F0-00-00-06-32   9061"
```

```
#write.csv(room_2_daily,"room_2_daily_intra.csv",row.names = F)
#write.csv(room_2_weekly,"room_2_weekly_intra.csv",row.names = F)
#write.csv(room_2_monthly,"room_2_monthly_intra.csv",row.names = F)
```

Determine which bird is the most active in the Whole Dataset without room 3

```
daily <- data.frame()
weekly <- data.frame()
monthly <- data.frame()
#TODO select sample as opposed to ID for intra bird comparison (alpha eq)
for(item in trans_reslist_room_2){
  daily <- rbind(daily, item$daily[c("ID","bottom","mid","top","total")])
  weekly <- rbind(weekly, item$weekly[c("ID","bottom","mid","top","total")])
  monthly <- rbind(monthly, item$monthly[c("ID","bottom","mid","top","total")])
}

for(item in trans_reslist_room_8){
  daily <- rbind(daily, item$daily[c("ID","bottom","mid","top","total")])
  weekly <- rbind(weekly, item$weekly[c("ID","bottom","mid","top","total")])
  monthly <- rbind(monthly, item$monthly[c("ID","bottom","mid","top","total")])
}
for(item in trans_reslist_room_11){
  daily <- rbind(daily, item$daily[c("ID","bottom","mid","top","total")])
  weekly <- rbind(weekly, item$weekly[c("ID","bottom","mid","top","total")])
  monthly <- rbind(monthly, item$monthly[c("ID","bottom","mid","top","total")])
}
length(unique(daily$ID))
```

```
## [1] 106
```

```
unique(daily$ID)
```

```
##   [1] "6855" "6860" "6872" "6877" "6890" "6901" "6903" "6908" "6909" "6911"
##  [11] "6919" "6925" "6926" "6935" "6937" "6942" "6946" "6952" "6955" "6956"
##  [21] "6958" "6960" "6962" "6971" "6975" "6978" "6981" "6983" "6986" "6988"
##  [31] "6989" "9001" "9005" "9019" "9021" "9024" "9026" "6886" "6913" "6917"
##  [41] "6923" "6928" "6929" "6931" "6939" "6940" "6941" "6945" "6950" "6951"
##  [51] "6957" "6961" "6964" "6970" "6973" "6979" "6982" "6984" "6985" "6993"
##  [61] "6994" "6996" "6997" "7000" "9012" "9014" "9017" "9020" "9022" "9023"
##  [71] "9025" "6853" "6904" "6916" "6932" "6965" "6990" "6999" "9002" "9003"
##  [81] "9006" "9007" "9015" "9027" "9030" "9032" "9033" "9035" "9038" "9041"
##  [91] "9042" "9043" "9044" "9045" "9046" "9047" "9048" "9049" "9050" "9052"
## [101] "9053" "9054" "9055" "9056" "9058" "9061"
```

```
tmp <- cbind(as.character(daily$ID), as.numeric(daily$total))
tmp_df <- data.frame(as.numeric(tmp[,1]), as.numeric(tmp[,2]))
colnames(tmp_df) <- c("ID","total")
transitions <- aggregate(total ~ ID, data=tmp_df, FUN=sum)
days <- table(tmp_df$ID)
norm_tots <- round(transitions$total/days,0)
norm_tots_df <- data.frame(norm_tots)
norm_tots_df <- data.frame(as.character(norm_tots_df$Var1), as.numeric(norm_tots_df$Freq))
colnames(norm_tots_df) <-c("ID","norm_tot")
most_active <- norm_tots_df$ID[norm_tots_df$norm_tot == max(norm_tots_df$norm_tot)]

print(paste("Most active bird: ",most_active))
```

```
## [1] "Most active bird:  6951"
```

Determine which bird is the least active in the Room 2 Dataset

```
daily <- data.frame()
weekly <- data.frame()
monthly <- data.frame()
#TODO select sample as opposed to ID for intra bird comparison (alpha eq)
for(item in trans_reslist_room_2){
  daily <- rbind(daily, item$daily[c("ID","bottom","mid","top","total")])
  weekly <- rbind(weekly, item$weekly[c("ID","bottom","mid","top","total")])
  monthly <- rbind(monthly, item$monthly[c("ID","bottom","mid","top","total")])
}

for(item in trans_reslist_room_8){
  daily <- rbind(daily, item$daily[c("ID","bottom","mid","top","total")])
  weekly <- rbind(weekly, item$weekly[c("ID","bottom","mid","top","total")])
  monthly <- rbind(monthly, item$monthly[c("ID","bottom","mid","top","total")])
}
for(item in trans_reslist_room_11){
  daily <- rbind(daily, item$daily[c("ID","bottom","mid","top","total")])
  weekly <- rbind(weekly, item$weekly[c("ID","bottom","mid","top","total")])
  monthly <- rbind(monthly, item$monthly[c("ID","bottom","mid","top","total")])
}
tmp <- cbind(as.character(daily$ID), as.numeric(daily$total))
tmp_df <- data.frame(as.numeric(tmp[,1]), as.numeric(tmp[,2]))
colnames(tmp_df) <- c("ID","total")
transitions <- aggregate(total ~ ID, data=tmp_df, FUN=sum)
days <- table(tmp_df$ID)
norm_tots <- round(transitions$total/days,0)
norm_tots_df <- data.frame(norm_tots)
norm_tots_df <- data.frame(as.character(norm_tots_df$Var1), as.numeric(norm_tots_df$Freq))
colnames(norm_tots_df) <-c("ID","norm_tot")
least_active <- norm_tots_df$ID[norm_tots_df$norm_tot == min(norm_tots_df$norm_tot)]
```

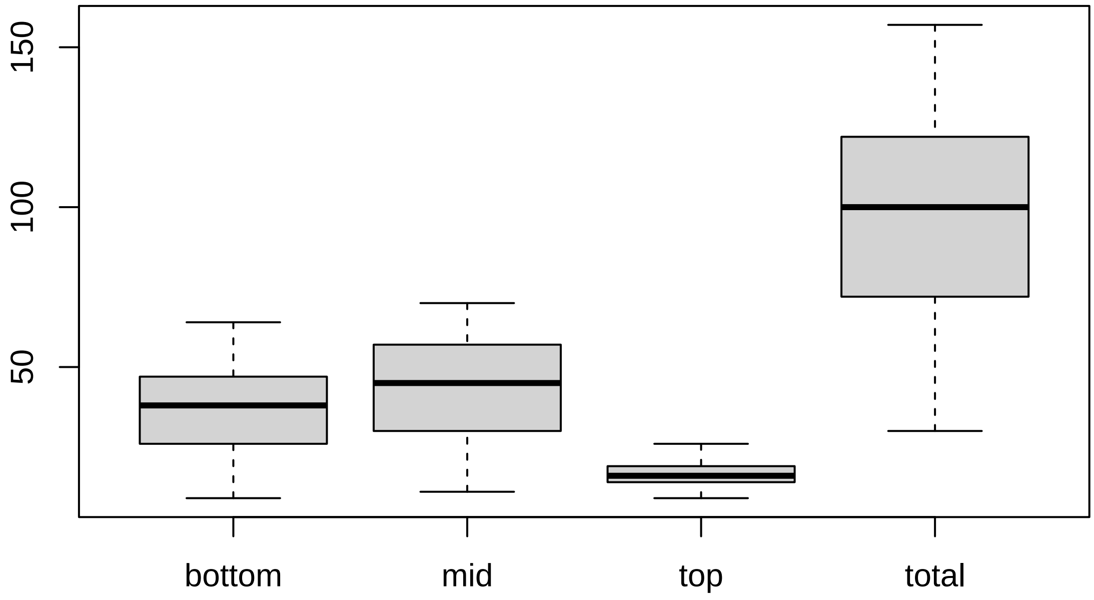
```r
print(paste("Least active bird: ",least_active))
```

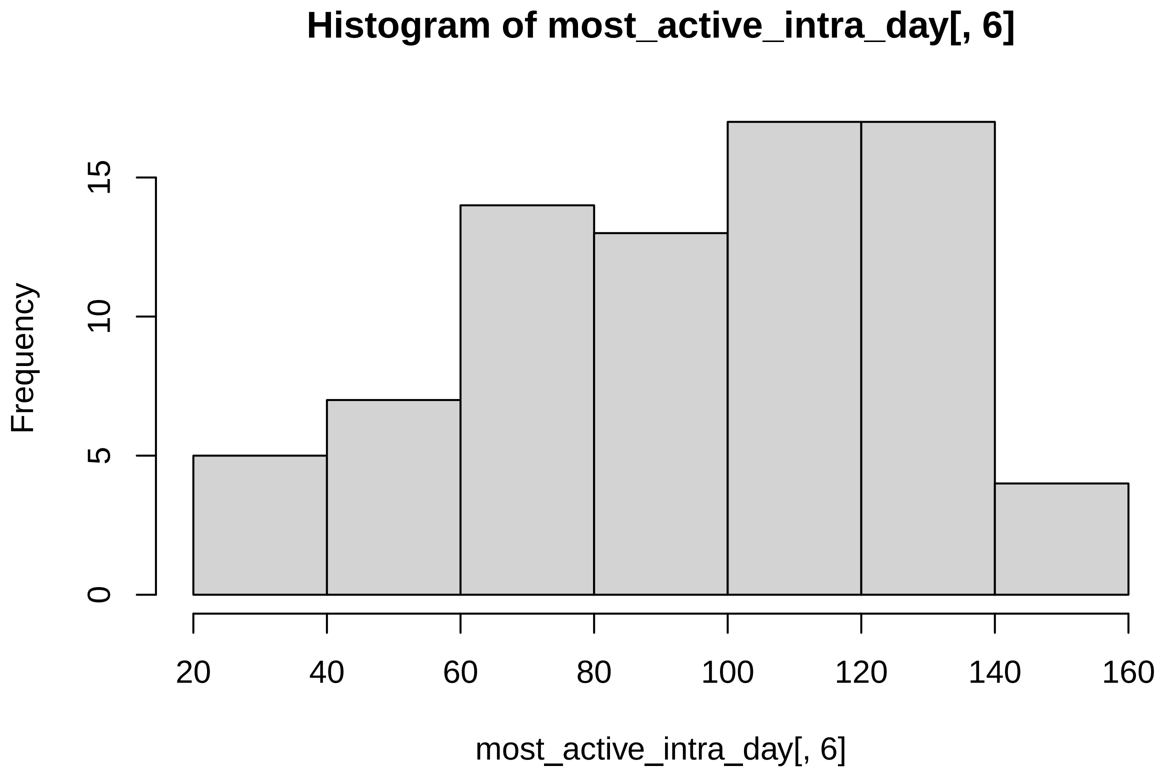
```
## [1] "Least active bird:  6929" "Least active bird:  6939"
## [3] "Least active bird:  6990"
```

Most Active Bird

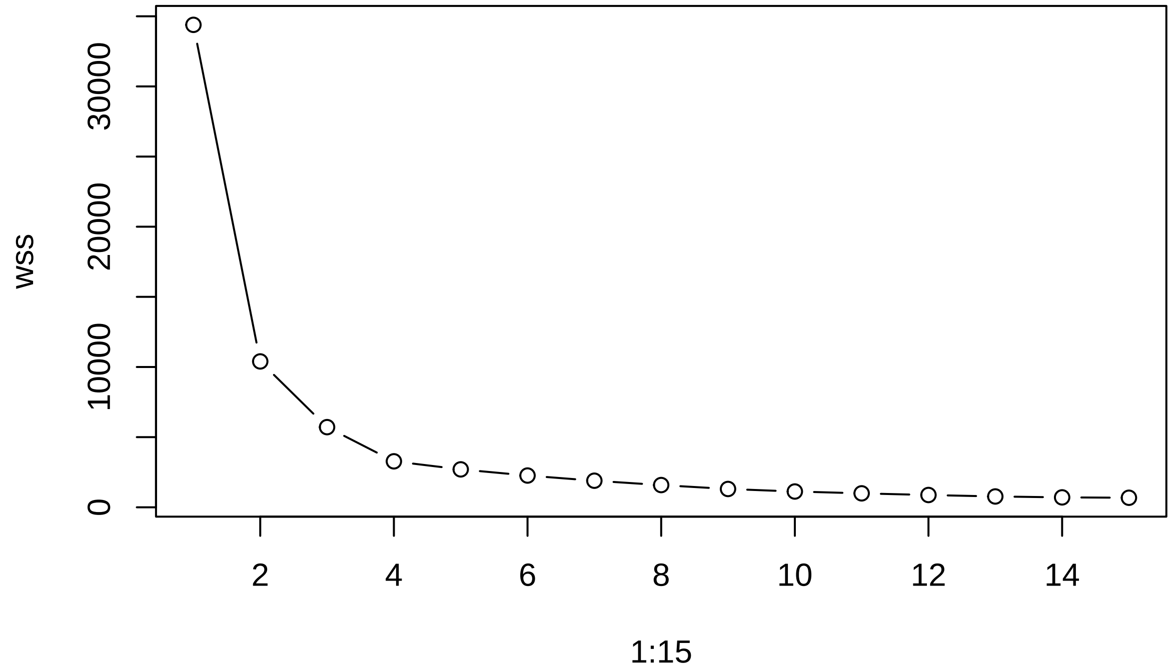
```r
set.seed(34716)
library(scales)
library(stringr)
library(ggplot2)
library(lubridate)

daily <- data.frame()
weekly <- data.frame()
monthly <- data.frame()
#TODO select sample as opposed to ID for intra bird comparison (alpha eq)
for(item in trans_reslist_room_2){
  daily <- rbind(daily, item$daily[c("sample","ID","bottom","mid","top","total")])
  weekly <- rbind(weekly, item$weekly[c("sample","ID","bottom","mid","top","total")])
  monthly <- rbind(monthly, item$monthly[c("sample","ID","bottom","mid","top","total")])
}

for(item in trans_reslist_room_8){
  daily <- rbind(daily, item$daily[c("sample","ID","bottom","mid","top","total")])
  weekly <- rbind(weekly, item$weekly[c("sample","ID","bottom","mid","top","total")])
  monthly <- rbind(monthly, item$monthly[c("sample","ID","bottom","mid","top","total")])
}
for(item in trans_reslist_room_11){
  daily <- rbind(daily, item$daily[c("sample","ID","bottom","mid","top","total")])
  weekly <- rbind(weekly, item$weekly[c("sample","ID","bottom","mid","top","total")])
  monthly <- rbind(monthly, item$monthly[c("sample","ID","bottom","mid","top","total")])
}

for(j in 1:length(most_active)){
most_active_intra_day <- daily[grep(most_active[j], daily$ID),]
most_active_intra_day <- as.data.frame(most_active_intra_day)
most_active_intra_day[,3:6] <- sapply(most_active_intra_day[,3:6], as.numeric)
boxplot(most_active_intra_day[,3:6])
hist(most_active_intra_day[,6])

wss <- 0

for(i in 1:15){
  km.out <- kmeans(most_active_intra_day[,3:5], centers = i, nstart=20)
  wss[i] <- km.out$tot.withinss
}


plot(1:15, wss, type="b")

km.out <- kmeans(most_active_intra_day[,3:5],3, 20)
```

```
#km.out <- kmeans(most_active_intra_day[,2:4],3, 20)
summary(km.out)
table(km.out$cluster)

pr.feb <- prcomp(x=most_active_intra_day[,3:5], scale=T, center=T)
pr.feb.prop <- summary(pr.feb)
most_active_intra_day["day"] <- as.character(str_split_fixed(most_active_intra_day$sample, ".",6)[,6])
most_active_intra_day["week"]<- (week(most_active_intra_day$day)+23)
most_active_intra_day["cluster"] <- km.out$cluster
most_active_intra_day
pr.feb$x <- data.frame(pr.feb$x)

p <- ggplot(pr.feb$x[,1:2], aes(x=PC1, y=PC2, color=as.character(most_active_intra_day$cluster)))+
  geom_point() +
  labs(title=paste("Most Active Bird (",most_active,")"),x=paste("PC1 (",round(pr.feb.prop$importance[[
  scale_color_discrete(name = "Weeks", labels= c(as.character(paste(most_active_intra_day$week[most_acti
  theme_bw() +
  theme(legend.position = "bottom", legend.direction = "vertical",legend.text = element_text(size=8))
p
ggsave(p, filename = paste0("../figures/",as.character(most_active[j]),"_most_active_bird_pca_no_rm_3.p


cluster_table <- data.frame(cbind(as.numeric(km.out$cluster),most_active_intra_day$sample))
cluster_table[order(cluster_table$X1),]

most_active_intra_day[order(most_active_intra_day$total),]




most_active_grp <- data.frame(x = as.Date(most_active_intra_day$day), y = c(most_active_intra_day$botto




 q <- ggplot(most_active_grp, aes(x,y,col=group))+
  geom_line() +
  facet_grid(factor(group,levels=c("Top","Mid","Bottom"))~ .) +
  labs(title=paste("Most Active Bird (",most_active[j],") Transitions"), color="Legend")+
  scale_x_date(name="Weeks",breaks = seq(min(most_active_grp$x),max(most_active_grp$x),by="week"),label
  scale_y_continuous(name="Transitions Into",n.breaks = 8,limits = c(0,70)) +
  scale_color_manual(values=c("Top"="royalblue","Mid"="tomato","Bottom" = "seagreen")) +
  theme_bw()

q

ggsave(q, filename = paste0("../figures/",as.character(most_active[j]),"_most_active_transitions_per_da
}
```


150
100
50
bottom
mid
top
total



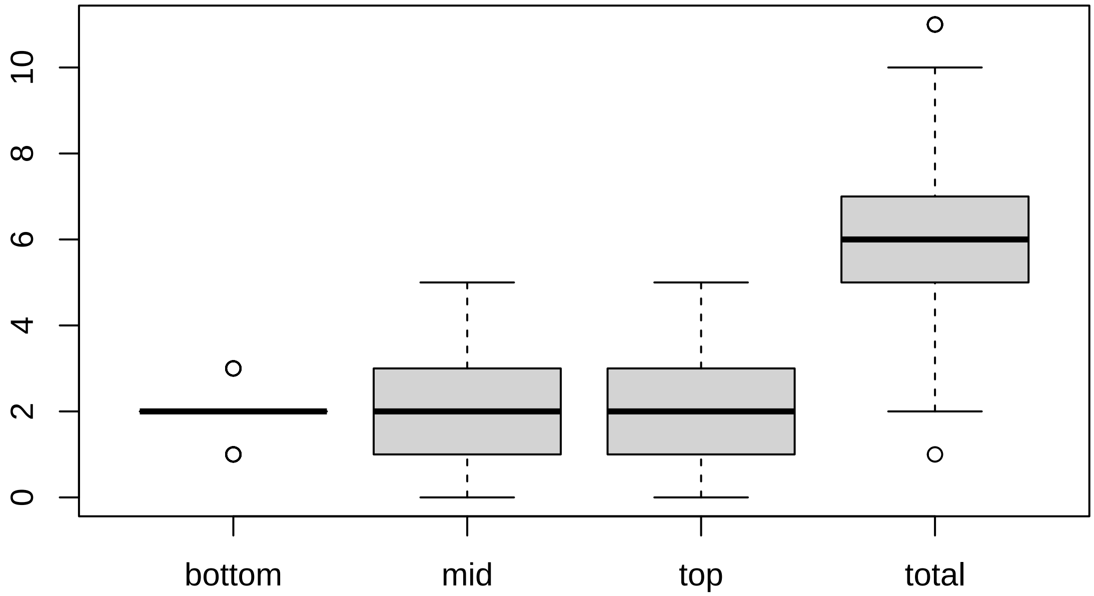
```
## Saving 6.5 x 4.5 in image
```

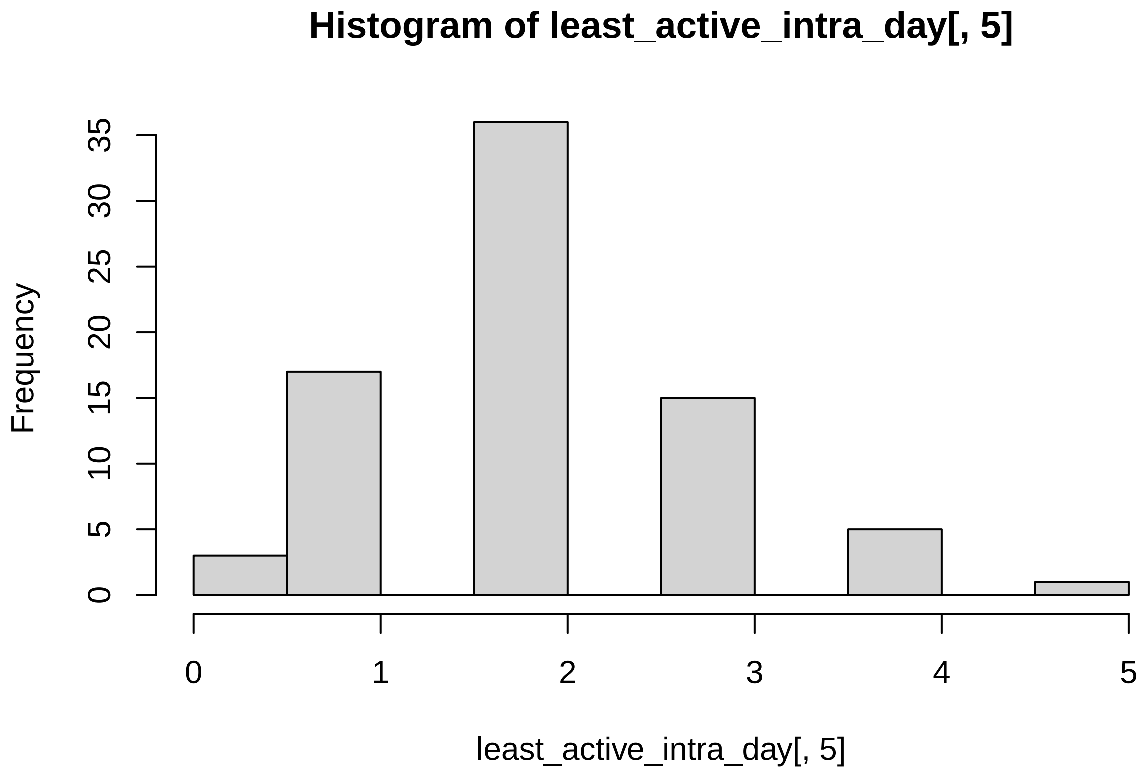


Least Active Bird

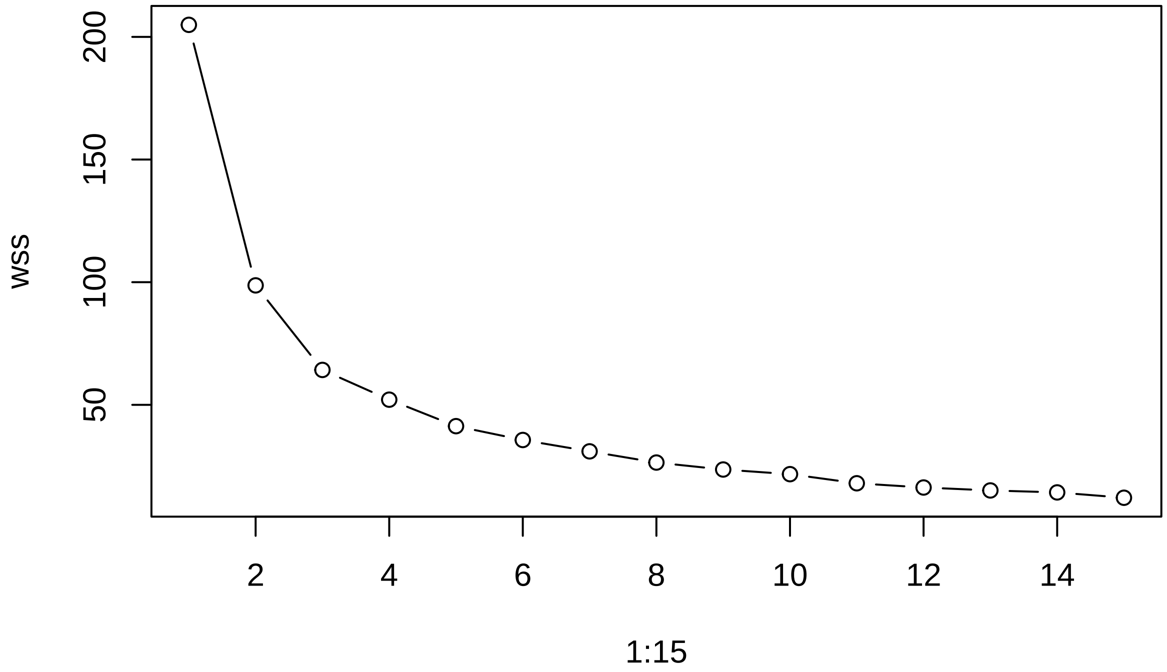
```
library(stringr)
library(ggplot2)

daily <- data.frame()
weekly <- data.frame()
monthly <- data.frame()
#TODO select sample as opposed to ID for intra bird comparison (alpha eq)
for(item in trans_reslist_room_2){
  daily <- rbind(daily, item$daily[c("sample","ID","bottom","mid","top","total")])
  weekly <- rbind(weekly, item$weekly[c("sample","ID","bottom","mid","top","total")])
  monthly <- rbind(monthly, item$monthly[c("sample","ID","bottom","mid","top","total")])
}

for(item in trans_reslist_room_8){
  daily <- rbind(daily, item$daily[c("sample","ID","bottom","mid","top","total")])
  weekly <- rbind(weekly, item$weekly[c("sample","ID","bottom","mid","top","total")])
  monthly <- rbind(monthly, item$monthly[c("sample","ID","bottom","mid","top","total")])
}
for(item in trans_reslist_room_11){
  daily <- rbind(daily, item$daily[c("sample","ID","bottom","mid","top","total")])
  weekly <- rbind(weekly, item$weekly[c("sample","ID","bottom","mid","top","total")])
  monthly <- rbind(monthly, item$monthly[c("sample","ID","bottom","mid","top","total")])
```

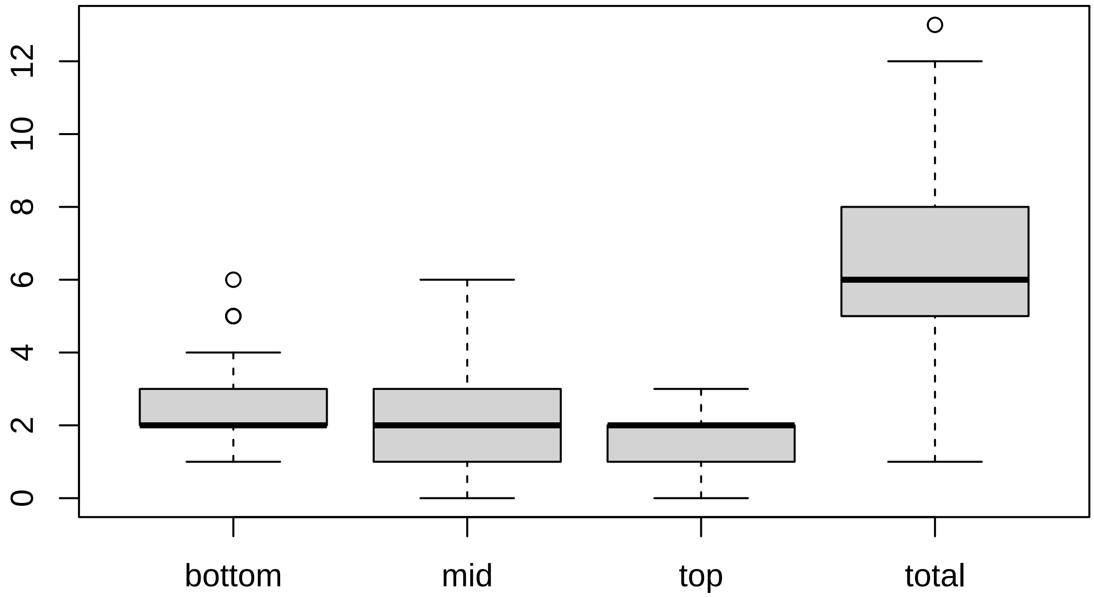
```
}

for (j in 1:length(least_active))
{
  least_active_intra_day <- daily[grep(least_active[j],daily$ID),]
  least_active_intra_day <- as.data.frame(least_active_intra_day)
  least_active_intra_day[,3:6] <- sapply(least_active_intra_day[,3:6], as.numeric)
  boxplot(least_active_intra_day[,3:6])
  hist(least_active_intra_day[,5])

  wss <- 0

  for(i in 1:15){
    km.out <- kmeans(least_active_intra_day[,3:5], centers = i, nstart=20)
    wss[i] <- km.out$tot.withinss
  }


  plot(1:15, wss, type="b")

  km.out <- kmeans(least_active_intra_day[,3:5],3, 20)
  summary(km.out)
  table(km.out$cluster)

  pr.feb <- prcomp(x=least_active_intra_day[,3:5], scale=T, center=F)
  pr.feb.prop <- summary(pr.feb)

  pr.feb.prop$importance
  pr.feb$x <- data.frame(pr.feb$x)


  least_active_intra_day["day"] <- as.character(str_split_fixed(least_active_intra_day$sample, ".",6)[,
  least_active_intra_day["week"]<- (week(least_active_intra_day$day)+23)
  least_active_intra_day["cluster"] <- km.out$cluster
  least_active_intra_day

  p <- ggplot(pr.feb$x[,1:2], aes(x=PC1, y=PC2, color=as.character(least_active_intra_day$cluster)))+
    geom_point() +
    labs(title=paste("Least Active Bird (",least_active[j],")"),x=paste("PC1 (",round(pr.feb.prop$impor
    scale_color_discrete(name = "Weeks", labels= c(as.character(paste(least_active_intra_day$week[least_
    theme_bw() +
    theme(legend.position = "bottom", legend.direction = "vertical",legend.text = element_text(size=8))
  p

  ggsave(p, filename = paste0("../figures/",as.character(least_active[j]),"_least_active_bird_pca.png")

  cluster_table <- data.frame(cbind(as.numeric(km.out$cluster),least_active_intra_day$sample))
  cluster_table[order(cluster_table$X1),]

  least_active_intra_day[order(least_active_intra_day$total),]
  least_active_intra_day
  least_active_grp <- data.frame(x = as.Date(least_active_intra_day$day), y = c(least_active_intra_day$l



  q <-ggplot(least_active_grp, aes(x,y,col=group))+
    geom_line() +
    facet_grid(factor(group,levels=c("Top","Mid","Bottom"))~ .) +
    labs(title=paste("Least Active Bird (",least_active[j],") Transitions"), color="Legend")+
    scale_x_date(name="Weeks",breaks = seq(min(least_active_grp$x),max(least_active_grp$x),by="week"),la
    scale_y_continuous(name="Transitions Into",n.breaks = 8,limits = c(0,70)) +
    scale_color_manual(values=c("Top"="royalblue","Mid"="tomato","Bottom" = "seagreen")) +
    theme_bw()
  q
  ggsave(q, filename=paste0("../figures/",as.character(least_active[j]),"_least_active_transitions_per_c
}
```

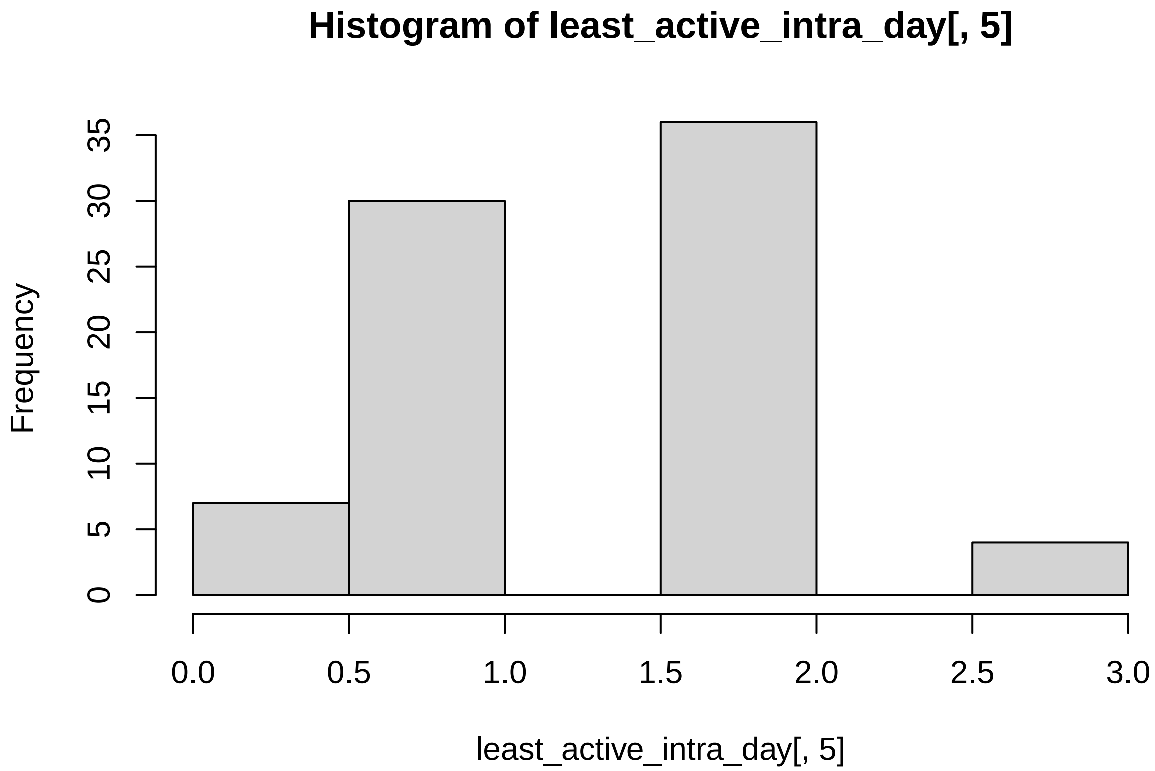




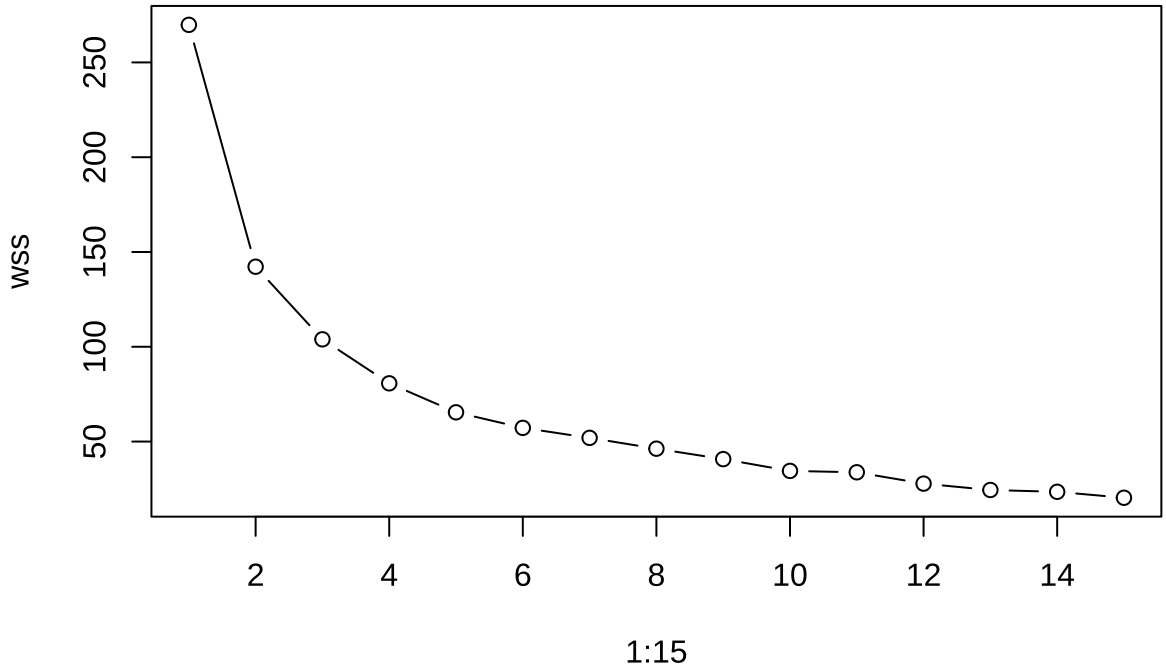
```
## Saving 6.5 x 4.5 in image
```

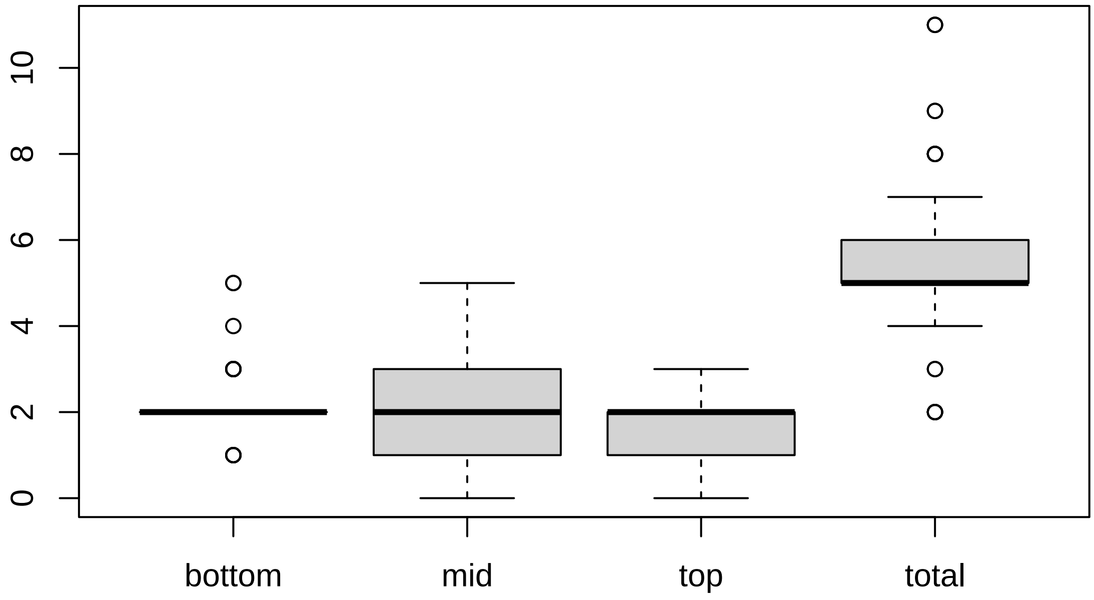

200
150
100
50
wss
2
4
6
8
10
12
14
1:15


0
2
4
6
8
10
12
bottom
mid
top
total

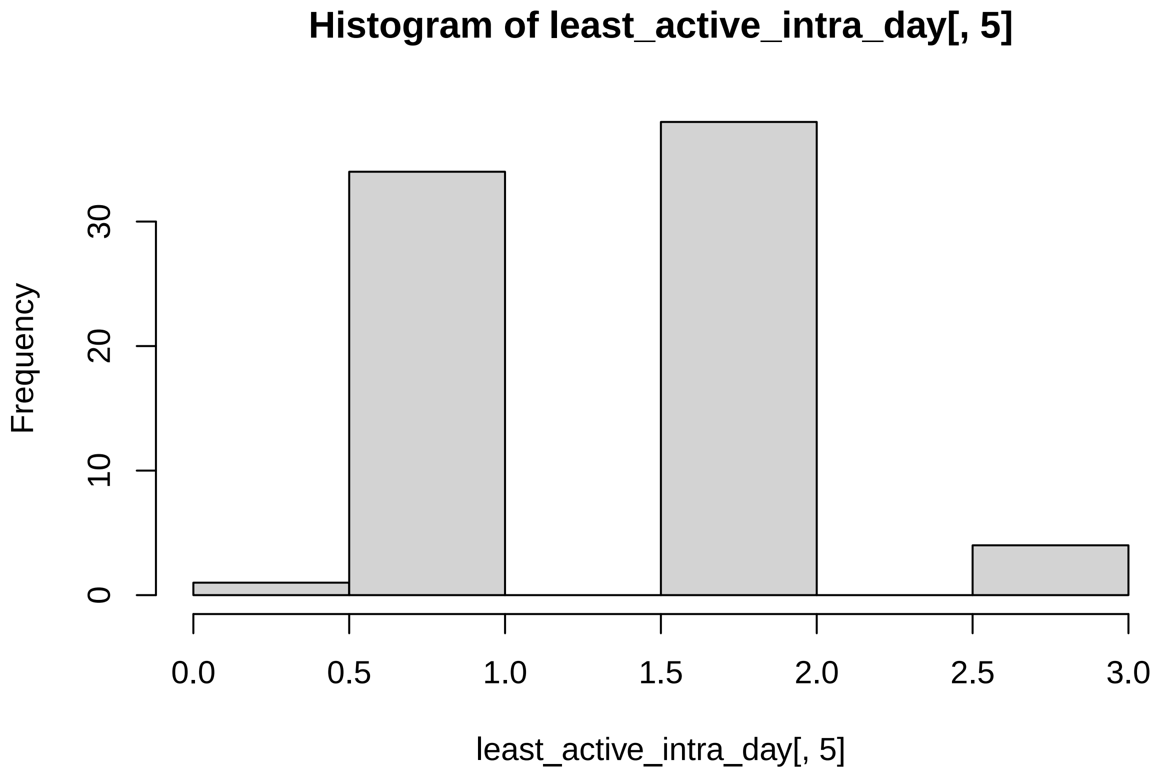


```
## Saving 6.5 x 4.5 in image
```

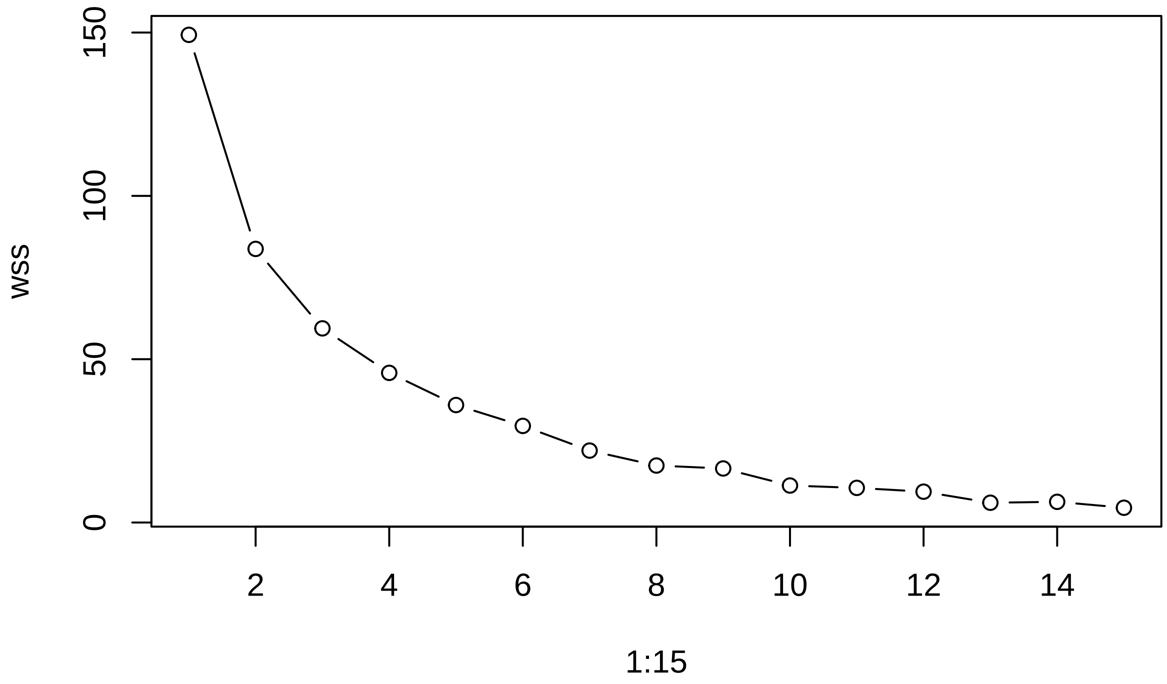

wss
250
200
150
100
50
2
4
6
8
10
12
14
1:15



```
## Warning: did not converge in 10 iterations
```



```
## Saving 6.5 x 4.5 in image
```



Organize Birds By activity

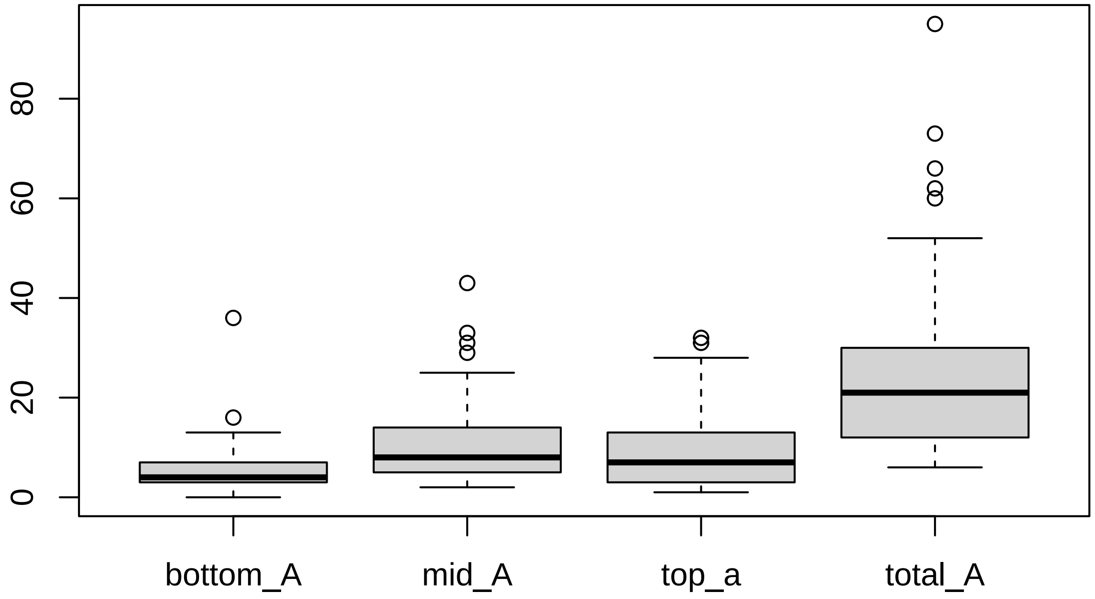
```
library(qpcR)

#trans_reslists <- c(trans_reslist_room_2,trans_reslist_room_3,trans_reslist_room_8,trans_reslist_room_
trans_reslists <- c(trans_reslist_room_2,trans_reslist_room_8,trans_reslist_room_11)
daily_colsum <- data.frame()

#each item is one transition list for an intra bird sample
#we want to sum up the columns for each day and compare bird to bird


for(i in 1:length(trans_reslists)){
  res <- colSums(sapply(trans_reslists[[i]]$daily[,2:6], as.numeric))
  res[1] <- res[1]/length(trans_reslists[[i]]$daily$ID)
  res[6] <- round(res[2]/length(trans_reslists[[i]]$daily$bottom),0)
  res[7] <- round(res[3]/length(trans_reslists[[i]]$daily$mid),0)
  res[8] <- round(res[4]/length(trans_reslists[[i]]$daily$top),0)
  res[9] <- round(res[5]/length(trans_reslists[[i]]$daily$total),0)
  daily_colsum <- rbind(daily_colsum,res)
}

colnames(daily_colsum) <- c("ID","bottom","mid","top","total","bottom_A","mid_A","top_a","total_A")
daily_colsum
```

```
##        ID bottom  mid  top total bottom_A mid_A top_a total_A
## 1   6855    108  901  934  1943        1    12    12      25
## 2   6860    329  261  294   884        4     3     4      11
## 3   6872    335  876  841  2052        4    11    11      27
## 4   6877    455  643  674  1772        6     8     9      23
## 5   6890     14  886  880  1780        0    12    11      23
## 6   6901     37 1952 1952  3941        0    25    25      51
## 7   6903    618  803  792  2213        8    10    10      29
## 8   6908    397  323  430  1150        5     4     6      15
## 9   6909    230  165  139   534        3     2     2       7
## 10  6911    519  415  346  1280        7     5     4      17
## 11  6919     49 1319 1311  2679        1    17    17      35
## 12  6925    287 1068  990  2345        4    14    13      30
## 13  6926   1201 2271 2139  5611       16    29    28      73
## 14  6935    537  551  492  1580        7     7     6      21
## 15  6937    617  624  601  1842        8     8     8      24
## 16  6942    229  218  239   686        3     3     3       9
## 17  6946    203  872  871  1946        3    11    11      25
## 18  6952    271  525  522  1318        4     7     7      17
## 19  6955    622  427  605  1654        8     6     8      21
## 20  6956     46 1384 1382  2812        1    18    18      37
## 21  6958    281  310  309   900        4     4     4      12
## 22  6960    476 1286 1256  3018        6    17    16      39
## 23  6962    670  720  652  2042        9     9     8      27
## 24  6971    316  291  217   824        4     4     3      11
## 25  6975    219  857  865  1941        3    11    11      25
## 26  6978    600  538  566  1704        8     7     7      22
## 27  6981    729  774  792  2295        9    10    10      30
## 28  6983    207  180  203   590        3     2     3       8
## 29  6986    313  257  263   833        4     3     3      11
## 30  6988    188  197  213   598        2     3     3       8
## 31  6989    453  464  476  1393        6     6     6      18
## 32  9001    545  485  453  1483        7     6     6      19
## 33  9005    645 1094 1149  2888        8    14    15      38
## 34  9019    315  645  623  1583        4     8     8      21
## 35  9021   1003 1941 1682  4626       13    25    22      60
## 36  9024    158  267  230   655        2     3     3       9
## 37  9026    577  501  557  1635        7     7     7      21
## 38  6886    344 1051 1013  2408        4    14    13      31
## 39  6913    729 1339  754  2822        9    17    10      37
## 40  6917    226  523  550  1299        3     7     7      17
## 41  6923    568  805  712  2085        7    10     9      27
## 42  6928    227  240  203   670        3     3     3       9
## 43  6929    153  171  159   483        2     2     2       6
## 44  6931    222  304  238   764        3     4     3      10
## 45  6939    200  183  114   497        3     2     1       6
## 46  6940    450  538  364  1352        6     7     5      18
## 47  6941    331 1036  953  2320        4    13    12      30
## 48  6945    233  202  197   632        3     3     3       8
## 49  6950    176  537  501  1214        2     7     7      16
## 50  6951   2784 3286 1261  7331       36    43    16      95
## 51  6957    289  406  268   963        4     5     3      13
## 52  6961    258  180  187   625        3     2     2       8
## 53  6964    311  388  250   949        4     5     3      12
## 54  6970    971 1687  971  3629       13    22    13      47
## 55  6973    851  714  909  2474       11     9    12      32
## 56  6979     83  675  624  1382        1     9     8      18
## 57  6982     36 1131 1129  2296        0    15    15      30
## 58  6984     22 1350 1329  2701        0    18    17      35
## 59  6985    391  755  734  1880        5    10    10      24
## 60  6993     80  387  386   853        1     5     5      11
## 61  6994    208  208  210   626        3     3     3       8
## 62  6996    228  232  190   650        3     3     2       8
## 63  6997    101  240  216   557        1     3     3       7
## 64  7000    618  708  277  1603        8     9     4      21
## 65  9012     40 1105 1091  2236        1    14    14      29
## 66  9014    339  380  176   895        4     5     2      12
## 67  9017    965 1696 1342  4003       13    22    17      52
## 68  9020    355 1559 1470  3384        5    20    19      44
## 69  9022     24 1563 1541  3128        0    20    20      41
## 70  9023    280  466  530  1276        4     6     7      17
## 71  9025     79  595  604  1278        1     8     8      17
## 72  6853    937 1449 1515  3901       12    19    20      51
## 73  6904     30 1752 1729  3511        0    23    22      46
## 74  6916    429  473  201  1103        6     6     3      14
## 75  6932    270  421  387  1078        4     5     5      14
## 76  6965    481 1318 1103  2902        6    17    14      38
## 77  6990    165  146  122   433        2     2     2       6
## 78  6999    713  878  689  2280        9    11     9      30
## 79  9002     39 2393 2379  4811        1    31    31      62
## 80  9003    562  893  874  2329        7    12    11      30
## 81  9006    413  587  580  1580        5     8     8      21
## 82  9007    547  862  555  1964        7    11     7      26
## 83  9015    349  611  455  1415        5     8     6      18
## 84  9023    380  494  273  1147        5     6     4      15
## 85  9027    320  717  635  1672        4     9     8      22
## 86  9030    142 2511 2432  5085        2    33    32      66
## 87  9032    215  191  129   535        3     2     2       7
## 88  9033    468  657  444  1569        6     9     6      20
## 89  9035    195 1017  976  2188        3    13    13      28
## 90  9038    360  957  811  2128        5    12    11      28
## 91  9041     33 1859 1841  3733        0    24    24      48
## 92  9042    246  225  137   608        3     3     2       8
## 93  9043    166  276  253   695        2     4     3       9
## 94  9044    325  885  850  2060        4    11    11      27
## 95  9045    640 1529 1175  3344        8    20    15      43
## 96  9046    240  211  203   654        3     3     3       8
## 97  9047    274  444  352  1070        4     6     5      14
## 98  9048    686  716  182  1584        9     9     2      21
## 99  9049    284  397  401  1082        4     5     5      14
## 100 9050    457  546  351  1354        6     7     5      18
## 101 9052    332  343  180   855        4     4     2      11
## 102 9053    349  324   95   768        5     4     1      10
## 103 9054    585 1226 1055  2866        8    16    14      37
## 104 9055    509  562  330  1401        7     7     4      18
## 105 9056     17 1792 1773  3582        0    23    23      47
## 106 9058    601  651  343  1595        8     8     4      21
## 107 9061     16 1508 1501  3025        0    20    19      39
```

```
hold <- daily_colsum[order(daily_colsum$total_A,decreasing = T),]

summary(hold[,6:9])
```

```
##     bottom_A          mid_A           top_a           total_A
##  Min.   : 0.000   Min.   : 2.00   Min.   : 1.000   Min.   : 6.00
##  1st Qu.: 3.000   1st Qu.: 5.00   1st Qu.: 3.000   1st Qu.:12.00
##  Median : 4.000   Median : 8.00   Median : 7.000   Median :21.00
##  Mean   : 4.944   Mean   :10.35   Mean   : 9.178   Mean   :24.58
##  3rd Qu.: 7.000   3rd Qu.:14.00   3rd Qu.:13.000   3rd Qu.:30.00
##  Max.   :36.000   Max.   :43.00   Max.   :32.000   Max.   :95.00
```

```
boxplot(hold[,6:9])
```



```
low_act <- c()
med_act <- c()
high_act <- c()

for(i in 1:length(hold[,1])){
  cur_tol <- hold[i,9]
  if(cur_tol<=12){
    low_act <- c(low_act,hold[i,1])
  }
  if((cur_tol>12) & (cur_tol<=30)){
    med_act <- c(med_act,hold[i,1])
  }
  if(cur_tol>30){
    high_act <- c(high_act, hold[i,1])
  }
}


cluster_table <- data.frame(qpcR:::cbind.na(low_act,med_act,high_act))
cluster_table
```

```
##    low_act med_act high_act
## 1     6958    6925     6951
## 2     6964    6981     6926
## 3     9014    6941     9030
## 4     6860    6982     9002
## 5     6971    6999     9021
## 6     6986    9003     9017
## 7     6993    6903     6901
## 8     9052    9012     6853
## 9     6931    9035     9041
## 10    9053    9038     6970
## 11    6942    6872     9056
## 12    9024    6962     6904
## 13    6928    6923     9020
## 14    9043    9044     9045
## 15    6983    9007     9022
## 16    6988    6855     6960
## 17    6945    6946     9061
## 18    6961    6975     9005
## 19    6994    6937     6965
## 20    6996    6985     6956
## 21    9042    6877     6913
## 22    9046    6890     9054
## 23    6909    6978     6919
## 24    6997    9027     6984
## 25    9032    6935     6973
## 26    6929    6955     6886
## 27    6939    9019       NA
## 28    6990    9026       NA
## 29      NA    7000       NA
## 30      NA    9006       NA
## 31      NA    9048       NA
## 32      NA    9058       NA
## 33      NA    9033       NA
## 34      NA    9001       NA
## 35      NA    6989       NA
## 36      NA    6940       NA
## 37      NA    6979       NA
## 38      NA    9015       NA
## 39      NA    9050       NA
## 40      NA    9055       NA
## 41      NA    6911       NA
## 42      NA    6952       NA
## 43      NA    6917       NA
## 44      NA    9023       NA
## 45      NA    9025       NA
## 46      NA    6950       NA
## 47      NA    6908       NA
## 48      NA    9023       NA
## 49      NA    6916       NA
## 50      NA    6932       NA
## 51      NA    9047       NA
## 52      NA    9049       NA
## 53      NA    6957       NA
```

```
write.csv(cluster_table, "../output/organized_birds_by_activity_daily_no_rm_3.csv",row.names = F)
```

```
# find which room each ID is in:
updated_table <- data.frame()

all_selected_ids <- c(low_act,med_act,high_act)
new_row <- ""
activity <- ""
room <- ""

for(id in all_selected_ids){
  #find out which activity level id is in
  if(id %in% low_act){
      activity <- "low"
  } else if( id %in% med_act){
      activity <- "medium"
  } else {
      activity <- "high"
  }
  #find out which room id is in
  if(id %in% bird_ids_room_2){
      room <- "2"
  } else if(id %in% bird_ids_room_3 ){
      room <- "3"
  } else if( id %in% bird_ids_room_8 ){
      room <- "8"
  } else if( id %in% bird_ids_room_11 ) {
      room <- "11"
  }
  #add row to updated table
  new_row <- c(room, id, activity)
  updated_table <- rbind(updated_table, new_row)
}

colnames(updated_table) <- c("Room", "Bird ID", "Activity Level")
write.csv(updated_table, "../output/organized_birds_by_activity_daily_no_rm_3_nice_table.csv",row.names
```

Jumpy table for the separated samples

```
# Aggregate the low activity birds
low_act_bird_ids <- na.trim(cluster_table$low_act)

#make a table of the samples
low_act_bird_frame <- data.frame()
```

```r
for(id in low_act_bird_ids){
  curr_bird <- daily[grep(id,daily$sample),]
  low_act_bird_frame <- rbind(low_act_bird_frame,curr_bird)
}

low_act_bird_frame[,3:6] <- sapply(low_act_bird_frame[,3:6], as.numeric)
low_act_bird_frame["day"] <- as.character(str_split_fixed(low_act_bird_frame$sample, ".",6)[,6])
low_act_agg_tab <- aggregate(low_act_bird_frame[,3:6],by = list(low_act_bird_frame$day), FUN = mean)
colnames(low_act_agg_tab) <- c("day","bottom","mid","top","total")

# Aggregate the medium activity birds
med_act_bird_ids <- na.trim(cluster_table$med_act)

#make a table of the samples
med_act_bird_frame <- data.frame()

for(id in med_act_bird_ids){
  curr_bird <- daily[grep(id,daily$sample),]
  med_act_bird_frame <- rbind(med_act_bird_frame,curr_bird)
}

med_act_bird_frame[,3:6] <- sapply(med_act_bird_frame[,3:6], as.numeric)
med_act_bird_frame["day"] <- as.character(str_split_fixed(med_act_bird_frame$sample, ".",6)[,6])
med_act_agg_tab <- aggregate(med_act_bird_frame[,3:6],by = list(med_act_bird_frame$day), FUN = mean)
colnames(med_act_agg_tab) <- c("day","bottom","mid","top","total")

# Aggregate the high activity birds
high_act_bird_ids <- na.trim(cluster_table$high_act)

#make a table of the samples
high_act_bird_frame <- data.frame()

for(id in high_act_bird_ids){
  curr_bird <- daily[grep(id,daily$sample),]
  high_act_bird_frame <- rbind(high_act_bird_frame,curr_bird)
}

high_act_bird_frame[,3:6] <- sapply(high_act_bird_frame[,3:6], as.numeric)
high_act_bird_frame["day"] <- as.character(str_split_fixed(high_act_bird_frame$sample, ".",6)[,6])
high_act_agg_tab <- aggregate(high_act_bird_frame[,3:6],by = list(high_act_bird_frame$day), FUN = mean)
colnames(high_act_agg_tab) <- c("day","bottom","mid","top","total")

# plot the low activity birds
low_act_grp <- data.frame(x = as.Date(low_act_agg_tab$day), y = c(low_act_agg_tab$bottom, low_act_agg_ta

med_act_grp <- data.frame(x = as.Date(med_act_agg_tab$day), y = c(med_act_agg_tab$bottom, med_act_agg_ta

# plot the low activity birds
high_act_grp <- data.frame(x = as.Date(high_act_agg_tab$day), y = c(high_act_agg_tab$bottom, high_act_ag

combo_table <- rbind(low_act_grp, med_act_grp, high_act_grp)

q <-ggplot(combo_table, aes(x,y,col=activity))+
  geom_line(aes(col=activity)) +
  facet_grid(factor(group,levels=c("Top","Mid","Bottom"))~ ., scales='free_y') +
  labs(title=paste0("Activity Birds Transitions no Room 3 (n=",length(unique(daily$ID)),")"), color="Ac
  scale_x_date(name="Weeks",breaks = seq(min(combo_table$x),max(combo_table$x),by="week"),labels = as.nu
  scale_y_continuous(name="Average Transitions Into",n.breaks = 8) +
  scale_color_manual(values=c("Low"="royalblue","Medium"="tomato","High" = "seagreen")) +
  theme_bw()
  ggsave(q, filename="../figures/active_transitions_no_rm3_set_2.png",device="png")
```

```
## Saving 6.5 x 4.5 in image
```

Zone Occupying Table No Room 3

```
library(dplyr)

no_room_3_zone_table <- data.frame()
result <- ""

for(item in trans_reslist_room_2){
  result <- calc_zone_duration("room 2", item$ID, item$raw)
  no_room_3_zone_table <- rbind(no_room_3_zone_table, result)
}
for(item in trans_reslist_room_8){
  result <- calc_zone_duration("room 8", item$ID, item$raw)
  no_room_3_zone_table <- rbind(no_room_3_zone_table, result)
}
for(item in trans_reslist_room_11){
  result <- calc_zone_duration("room 11", item$ID, item$raw)
  no_room_3_zone_table <- rbind(no_room_3_zone_table, result)
}

# change the resultant times to numerics as opposed to chars
no_room_3_zone_table[,3:5] <- sapply(no_room_3_zone_table[,3:5],function(x) as.numeric(as.character(x))
no_room_3_zone_table[,3:5] <- (no_room_3_zone_table[,3:5])/60
# calculate the number of days observed
no_room_3_zone_table["days_observed"]<- round((rowSums(no_room_3_zone_table[,3:5])/60)/24)
colnames(no_room_3_zone_table) <-  c("Room","ID", "Time in Bottom","Time in Middle", "Time in Top","Day


write.csv(no_room_3_zone_table, "../output/zone_time_per_id_no_3.csv", row.names=F)

ave_time_per_room_no_3 <- aggregate(cbind(no_room_3_zone_table$"Time in Bottom",no_room_3_zone_table$"Ti

colnames(ave_time_per_room_no_3) <- c("Room", "Average Time in Bottom", "Average Time in Middle", "Avera

write.csv(ave_time_per_room_no_3, "../output/zone_time_per_room_no_3.csv", row.names=F)
```