Metrics for Recorder behaviour

Tom August

25 October 2016

Metrics

We are going to split metric into three broad groups: *Engagement profile*, *Spatial*, and *Taxanomic*

Temporal Metrics

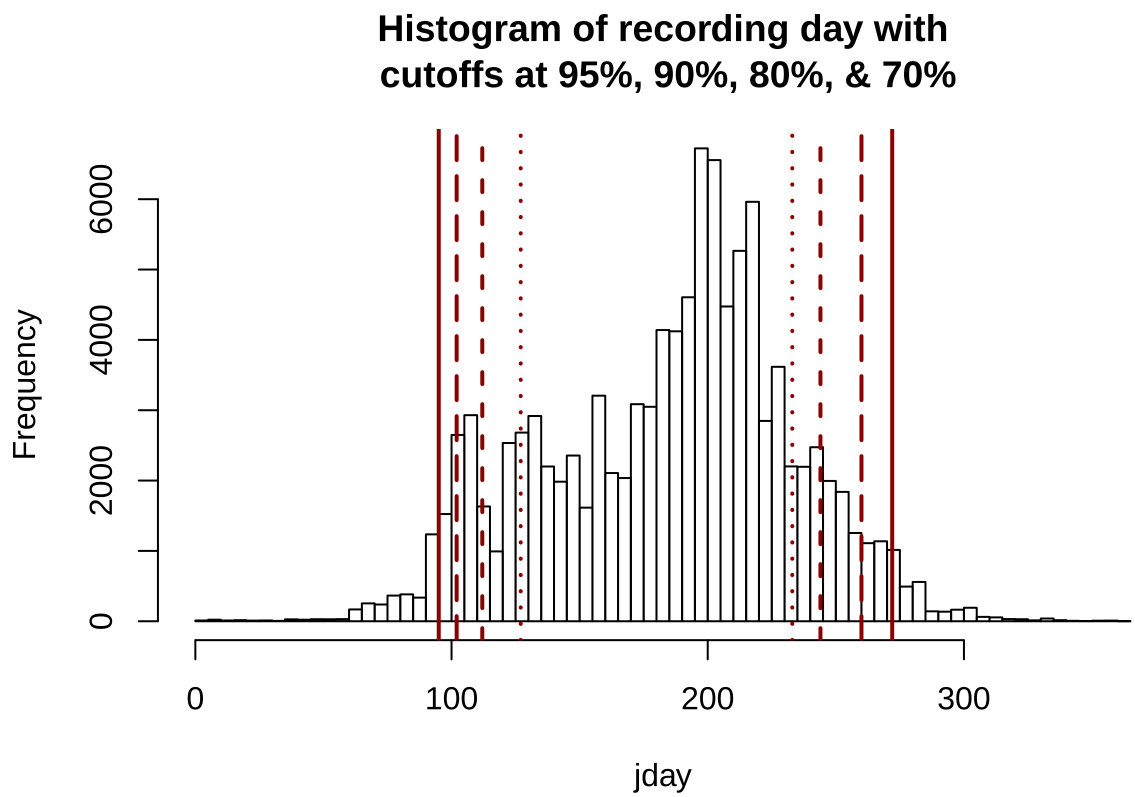
These metrics measure the recording pattern across time such as the number of days that a recorder produces records. These have been termed engagement profiles by others., The metrics here are from Ponciano and Brasileiro 2014 who used the metrics on participant of zooniverse projects. The metrics were also used by Boakes *et al* 2016.

Summer period

One issue that we have across these metrics and some others is that recording is not consistent across the year and so there can be issues with the numbers generated. To address this the data can be subset to only the summer period, when recorders are active. This period needs to be defined in such a way that the same method can be used across taxanomic groups and will be robust to changes in the start and end of teh summer period from year to year.

I suggest we use a percentage cut off, for example take the period of the year that contains 90% of the data. Lets have a look at how this might work

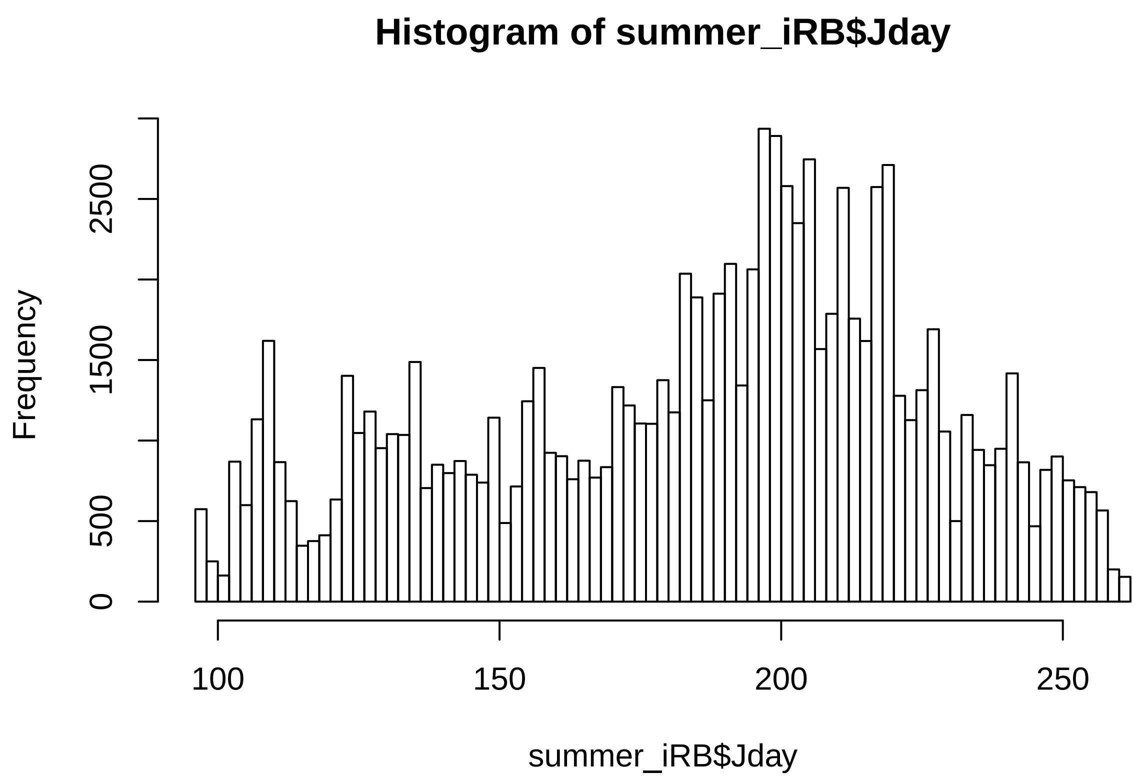
```
# First lets have a look at the distribution of
# records through the year
jday <- as.POSIXlt(iRB$date_start)$yday

# Where are the limits of different percentiles
p95 <- quantile(jday, c(0.025,0.975))
p90 <- quantile(jday, c(0.05,0.95))
p80 <- quantile(jday, c(0.1,0.9))
p70 <- quantile(jday, c(0.15,0.85))

hist(jday, 100, main = paste('Histogram of recording day with\n',
                             'cutoffs at 95%, 90%, 80%, & 70%'))
abline(v = p95, lty = 1, col = 'darkred', lwd = 2)
abline(v = p90, lty = 5, col = 'darkred', lwd = 2)
abline(v = p80, lty = 2, col = 'darkred', lwd = 2)
abline(v = p70, lty = 3, col = 'darkred', lwd = 2)
```



It looks like 90% might be a good value to go for in this case. We would then need a function that could create these values for each year of the data and throw out data that was outside the summer periods

```
summerData <- function(data, probs = c(0.05, 0.95),
                       date_col = 'date_start'){

  # check date column
  if(!inherits(data[, date_col], 'Date')){
    stop('Your date column is not a date')
  }

  # create J-day column
  data$Jday <- as.POSIXlt(data[,date_col])$yday

  # create year column
  data$year <- as.POSIXlt(data[,date_col])$year+1900

  # create summer column
  data$summer <- FALSE

  qsf <- qsl <- NULL

  # now for each year loop through and create an
  # index column
  for(i in sort(unique(data$year))){
```

```r
    year_quantiles <- quantile(data$Jday[data$year == i], probs = probs)
    qsf <- c(qsf, year_quantiles[1])
    qsl <- c(qsl, year_quantiles[2])
    data$summer[data$Jday >= year_quantiles[1]
                & data$Jday <= year_quantiles[2]
                & data$year == i] <- TRUE
  }

  summer_data <- data[data$summer, ]
  attr(summer_data, 'cutoffs') <- data.frame(year = sort(unique(data$year)),
                                             quantile_first = qsf,
                                             quantile_last = qsl)
  return(summer_data)
}

summer_iRB <- summerData(data = iRB,
                         probs = c(0.05, 0.95),
                         date_col = 'date_start')

# Look at the 'cut' data
hist(summer_iRB$Jday, 100)
```




```r
# Here are the cuts
attr(summer_iRB, 'cutoffs')
```

```
##   year quantile_first quantile_last
## 1 2014            104           260
## 2 2015             97           262
## 3 2016            107           250
```

Activity ratio

"The proportion of days on which the volunteer was active in relation to the total days he/she remained linked to the project" (Ponciano and Brasileiro 2014)

```r
# Create a function to calculate activity ratio
activityRatio <- function(recorder_name,
                          data,
                          recorder_col = 'recorders',
                          date_col = 'date_start'){

  # check date column
  if(!inherits(data[, date_col], 'Date')){
    stop('Your date column is not a date')
  }

  # Get the recorders data
  data <- data[data[,recorder_col] == recorder_name, ]

  # Some people might have no data from the summer period
  if(nrow(data) < 1){

    return(data.frame(recorder = recorder_name,
                      activity_ratio = NA,
                      total_duration = NA,
                      active_days = NA))
  } else {

  # Get unique dates
  dates <- unique(data[,date_col])

  # Get the first and last date
  first_last <- range(dates)

  # Total duration of this recorder
  duration <- as.numeric(first_last[2] - first_last[1]) + 1

  # calculate ratio
  activity_ratio <- length(dates)/duration

  # return
  return(data.frame(recorder = recorder_name,
                    activity_ratio = activity_ratio,
                    total_duration = duration,
                    active_days = length(dates)))
  }
}
```

```r
# Test on David and Tom
activityRatio(data = summer_iRB, recorder_name = 'Roy, David')
```

```
##     recorder activity_ratio total_duration active_days
## 1 Roy, David      0.1509217            868         131
```

```r
activityRatio(data = summer_iRB, recorder_name = 'August, Tom')
```

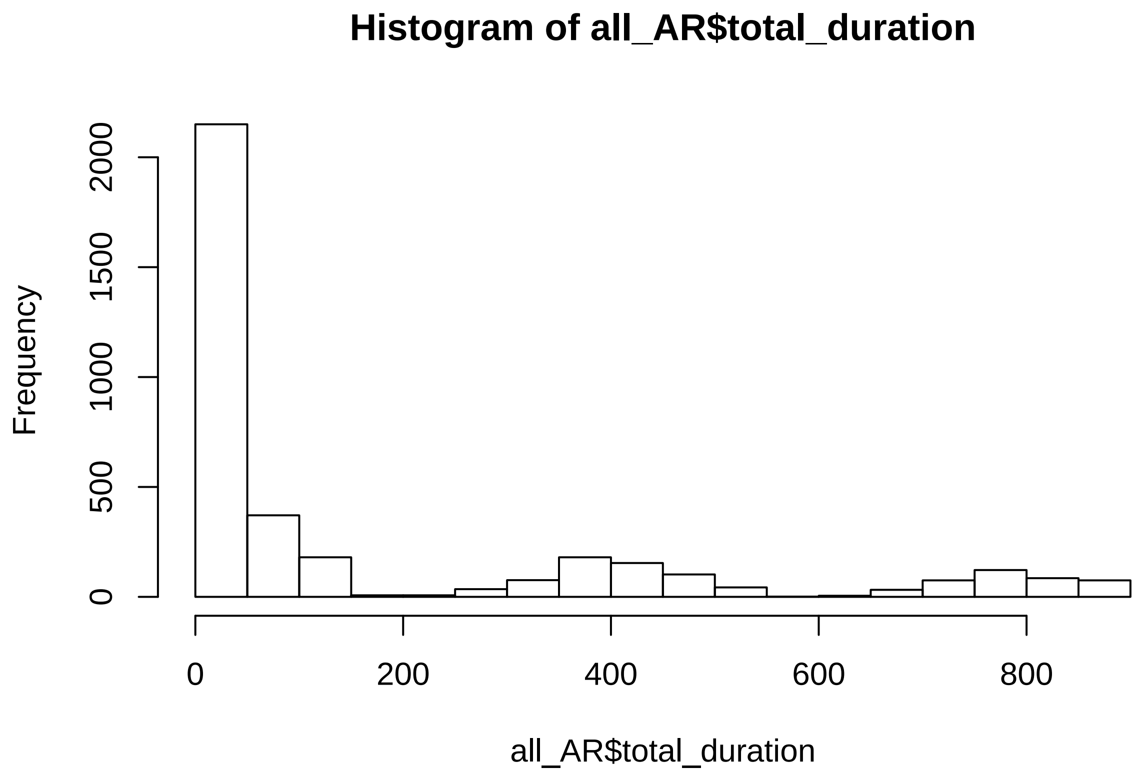
```
##      recorder activity_ratio total_duration active_days
## 1 August, Tom     0.01587302            441           7
```

```r
## David is a more active recorder than Tom ##

# Run for everyone
all_AR <- do.call(rbind, lapply(X = unique(iRB$recorders),
                                FUN = activityRatio,
                                data = summer_iRB))

# Lets have a look at some of these
head(all_AR, 20)
```

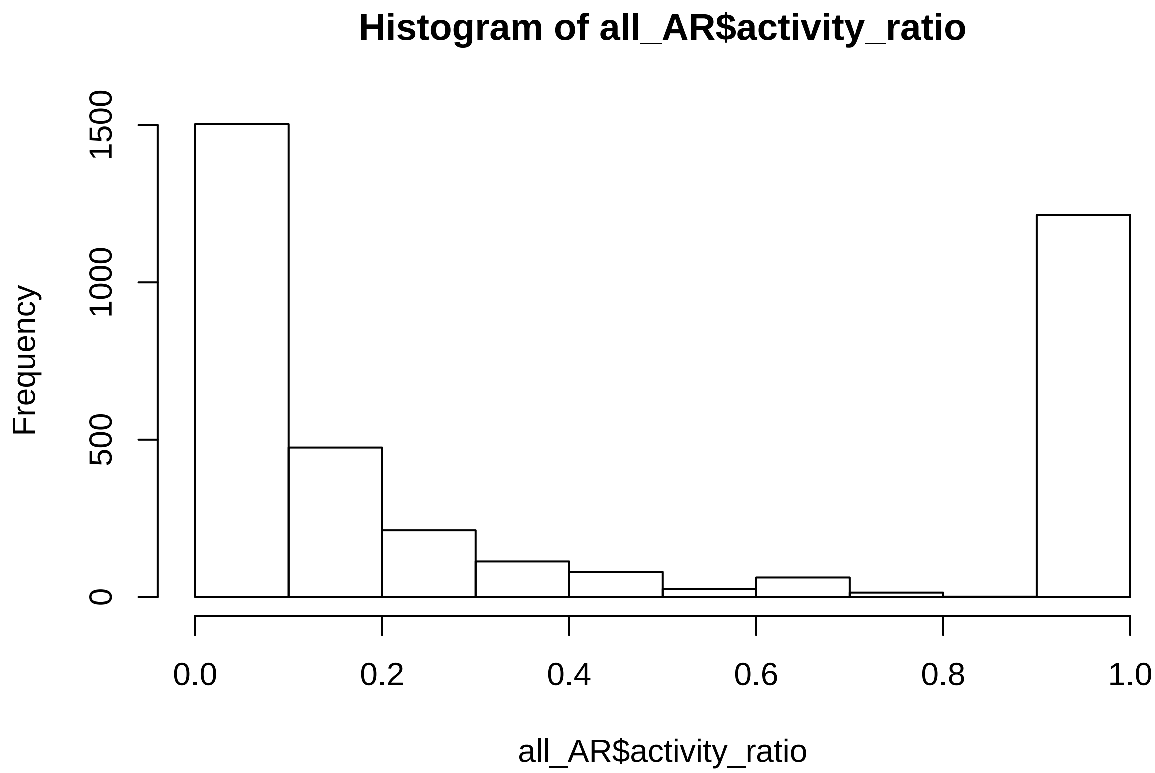
```
##                recorder activity_ratio total_duration active_days
## 1           Marsh, Nick     0.14000000            100          14
## 2             Limb, Ken     0.18464730            482          89
## 3            Ward, John     0.18181818             77          14
## 4         Hughes, Peter     0.22580645             31           7
## 5       Turner, Lindsey     0.11607143            336          39
## 6        Warren, Martin     0.19977169            876         175
## 7        Newbould, John     0.22945205            876         201
## 8            fenn, paul     0.16746411            836         140
## 9            Cox, Steve     0.18649886            874         163
## 10        Lewis, Steven     0.10432570            786          82
## 11         Anstie, John     0.17647059             34           6
## 12        Austin, David     0.13454960            877         118
## 13         Bowles, Nick     0.07531381            478          36
## 14         Binks, Rosie     0.19753086            486          96
## 15        Dean, Michael     0.07769784            695          54
## 16      Watkins, nicola     0.23913043             46          11
## 17         Kilbey, Dave     0.09239130            736          68
## 18         Dawson, John     0.25490196             51          13
## 19 rouncefield, marlene     0.05569007            413          23
## 20        Raymond, Colin     0.02816901            852          24
```

I think this metric tells a story in a combination of the ratio and the total number of days. I think the ratio means more when the recorder has been recording for a long duration

```r
# Have a look at the distribution of these 2 metrics
# There looks like there could be an effect of year
hist(all_AR$total_duration, breaks = 30)
```

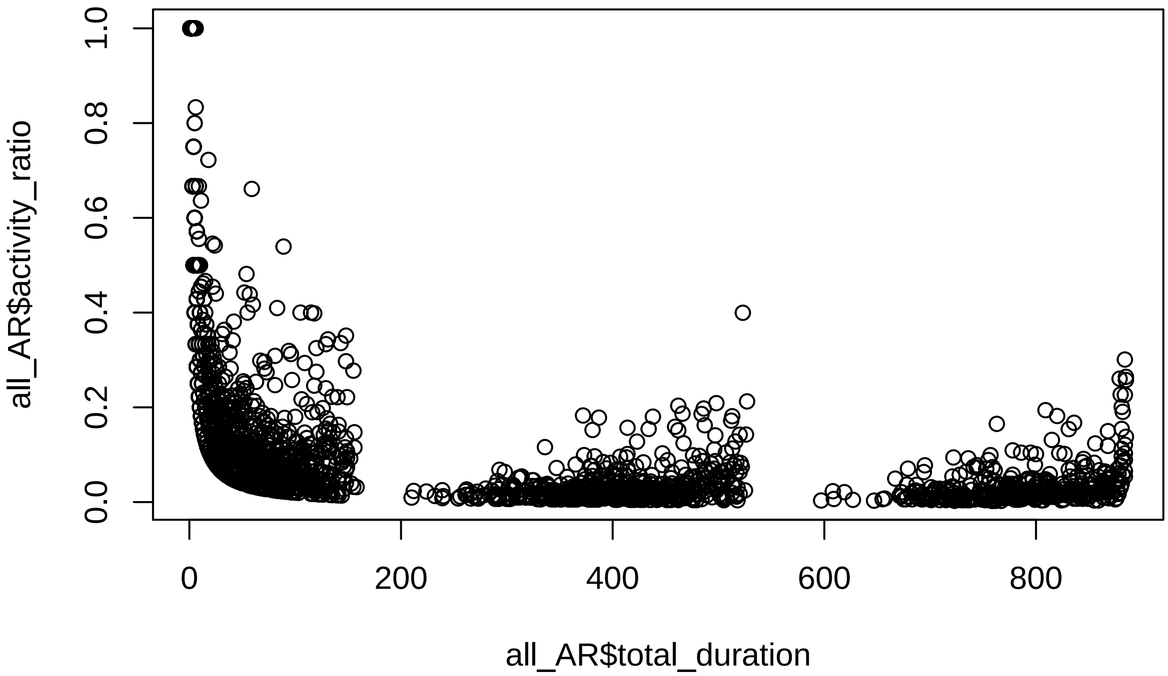



```
hist(all_AR$activity_ratio)
```

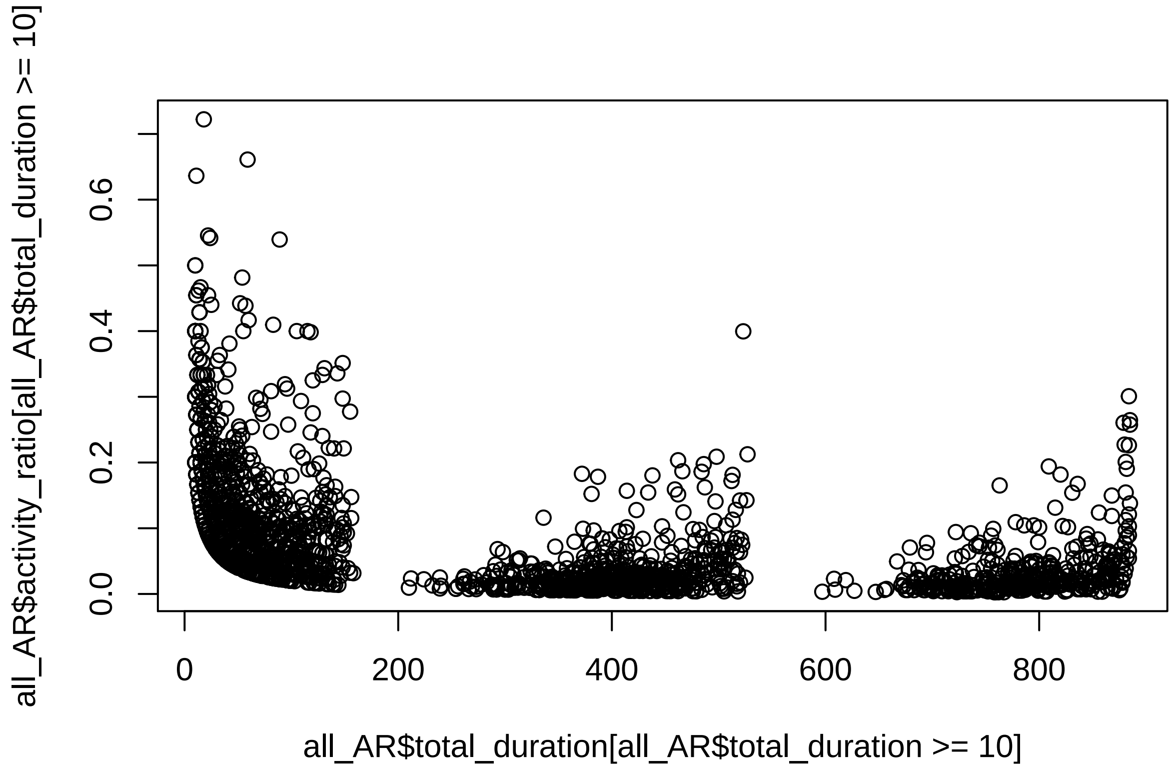


Both have nice distributions, though we can see the single record people in the ratio plot

```
# Plot activity_ratio against duration
# Here we probably want to subset to avoid bias
plot(all_AR$total_duration,
     all_AR$activity_ratio)
```




```
plot(all_AR$total_duration[all_AR$total_duration >= 10],
     all_AR$activity_ratio[all_AR$total_duration >= 10])
```



Weekly devoted days

This is an adaptation of the *Daily Devoted Time* in (Ponciano and Brasileiro 2014) which is clearly not applicable to biological recording. Though Boakes *et al* 2016 don't attempt to use this measure I think the idea can be adapted by using days in a week (summer only) rather than hours in a day.

```
# Create a function
weeklyDevotedDays <- function(recorder_name,
                              data,
                              recorder_col = 'recorders',
                              date_col = 'date_start'){

  # check date column
  if(!inherits(data[, date_col], 'Date')){
    stop('Your date column is not a date')
  }

  # Get the recorders data
  data <- data[data[,recorder_col] == recorder_name, ]

  # Get unique dates as dates
  dates <- unique(data[,date_col])

  # Get all week_year combinations
  week_year <- paste(strftime(as.POSIXlt(dates), format = '%W'),
```

```
                     format(dates, '%Y'), sep = '_')

  # here are the counts
  week_counts <- table(week_year)

  # As these are counts taking the median is probably best
  weekly_devoted_days <- median(week_counts)

  return(data.frame(recorder = recorder_name,
                    median_weekly_devoted_days = weekly_devoted_days,
                    n_weeks = length(week_counts),
                    n_recs = sum(week_counts), row.names = NULL))
}

# Test on David and Tom
weeklyDevotedDays(data = summer_iRB, recorder_name = 'Roy, David')
```

```
##     recorder median_weekly_devoted_days n_weeks n_recs
## 1 Roy, David                          2      60    131
```

```
weeklyDevotedDays(data = summer_iRB, recorder_name = 'August, Tom')
```

```
##      recorder median_weekly_devoted_days n_weeks n_recs
## 1 August, Tom                          1       6      7
```

```
## David contributes more of his time than Tom ##

# Run for everyone
all_WDD <- do.call(rbind, lapply(X = unique(iRB$recorders),
                                 FUN = weeklyDevotedDays,
                                 data = summer_iRB))

# Lets have a look at some of these
head(all_WDD, 20)
```

```
##                  recorder median_weekly_devoted_days n_weeks n_recs
## 1         Brookes , Anne                        1.0      38     59
## 2         Burgoyne, Steve                       2.0      18     47
## 3            Brown, Peter                       2.0      25     57
## 4      Rutherford, Joanna                       1.0       7     11
## 5            Allan, David                       4.0      67    259
## 6        Millward, Martin                       1.0       4      4
## 7  Foulkes-Arellano, Paul                       1.0      13     18
## 8             Stewart, Tam                      3.0      65    231
## 9           Forbes, Andrew                      1.0       3      5
## 10       Richardson, Rosie                      1.5       4      7
## 11    Partridge, Francesca                      2.0      56    146
## 12          Card , Graeme                       1.5      12     22
## 13             Honey, Hawk                      1.0      39     72
## 14          Melzack, David                      1.5      12     30
## 15              Povall, Ed                      1.0       3      3
## 16           Goodwin, Paul                      2.0      26     65
```

```
## 17           Coulson, Joe                              2.0       10     22
## 18          Bailey, Peggy                              2.0       16     41
## 19             Roy, David                              2.0       60    131
## 20      Woodley, Caroline                              1.0       12     18
```

Clearly this metric is only really reliable when we have multiple weeks worth of data for an individual.

Relative activity duration

This is a metric from Ponciano and Brasileiro 2014 which is also used in Boakes *et al* 2016 but I don't think can be applied to biological records since there is no official end date for a project: *"The ratio of days during which a volunteer I remains linked to the project in relation to the total number of days elapsed since the volunteer joined the project until the project is over"*

Periodicity

There is a cluster of metrics that could be used to look at aspects of periodicity. The measure used in Ponciano and Brasileiro 2014 is 'variation in periodicity'; *"The standard deviation of the times elapsed between each pair of sequential active days"*. At the same time as calculating this I think there are another couple of metrics that might be of use. First, periodicity itself, i.e. *"The median time elapsed between each pair of sequential active days"*. Secondly, streak length, i.e. *"The average length of sequential active days"*

```
# Create a function to calculate the periodicity metrics
periodicity <- function(recorder_name,
                        data,
                        recorder_col = 'recorders',
                        date_col = 'date_start',
                        day_limit = 5){

  # check date column
  if(!inherits(data[, date_col], 'Date')){
    stop('Your date column is not a date')
  }

  # Get the recorders data
  data <- data[data[,recorder_col] == recorder_name, ]

  # Get unique dates as dates
  dates <- sort(unique(data[,date_col]))

  # we cannot calculate these metrics if people have very few
  # dates on which they record
  if(length(unique(dates)) < day_limit){

    # return
    return(data.frame(recorder = recorder_name,
                      periodicity = NA,
                      periodicity_variation = NA,
                      median_streak = NA,
                      sd_streak = NA,
                      max_streak = NA,
                      n_days = length(unique(dates))))
```

```r
} else {

  # Calculate the elapsed days between each date in sequence
  # this needs to be done within years
  elapses <- NULL

  for(year in unique(format(dates, '%Y'))){

    temp_dates <- dates[format(dates, '%Y') == year]

    # There must be at least 2 dates in a year
    if(length(temp_dates) > 1){
      temp_elapses <- sapply(1:(length(temp_dates)-1),
         FUN = function(x){
           return(as.numeric(temp_dates[x + 1] - temp_dates[x]))
         })

      elapses <- c(elapses, temp_elapses)

    }

  }

  # periodicity calculation
  periodicity <- median(elapses)

  # variation in periodicity
  periodicity_variation <- sd(elapses)

  # average streak length
  # Streaks are IDed by 1's
  non_streak <- length(elapses[elapses > 1])
  streaks <- rle(elapses)
  streaks_1 <- (streaks$lengths[streaks$value == 1]) + 1

  # Combine streaks and non-streaks
  streak_lengths <- c(rep(1, non_streak), streaks_1)

  # calculate ome metrics
  median_streak <- median(streak_lengths)
  sd_streak <- sd(streak_lengths)
  max_streak <- max(streak_lengths)

  # return
  return(data.frame(recorder = recorder_name,
                    periodicity = periodicity,
                    periodicity_variation = periodicity_variation,
                    median_streak = median_streak,
                    sd_streak = sd_streak,
                    max_streak = max_streak,
                    n_days = length(unique(dates))))

}
```

```r
}

# Test on David and Tom
periodicity(data = summer_iRB, recorder_name = 'Roy, David')
```

```
##     recorder periodicity periodicity_variation median_streak sd_streak
## 1 Roy, David           2              3.196601             1 0.9699536
##   max_streak n_days
## 1          7    131
```

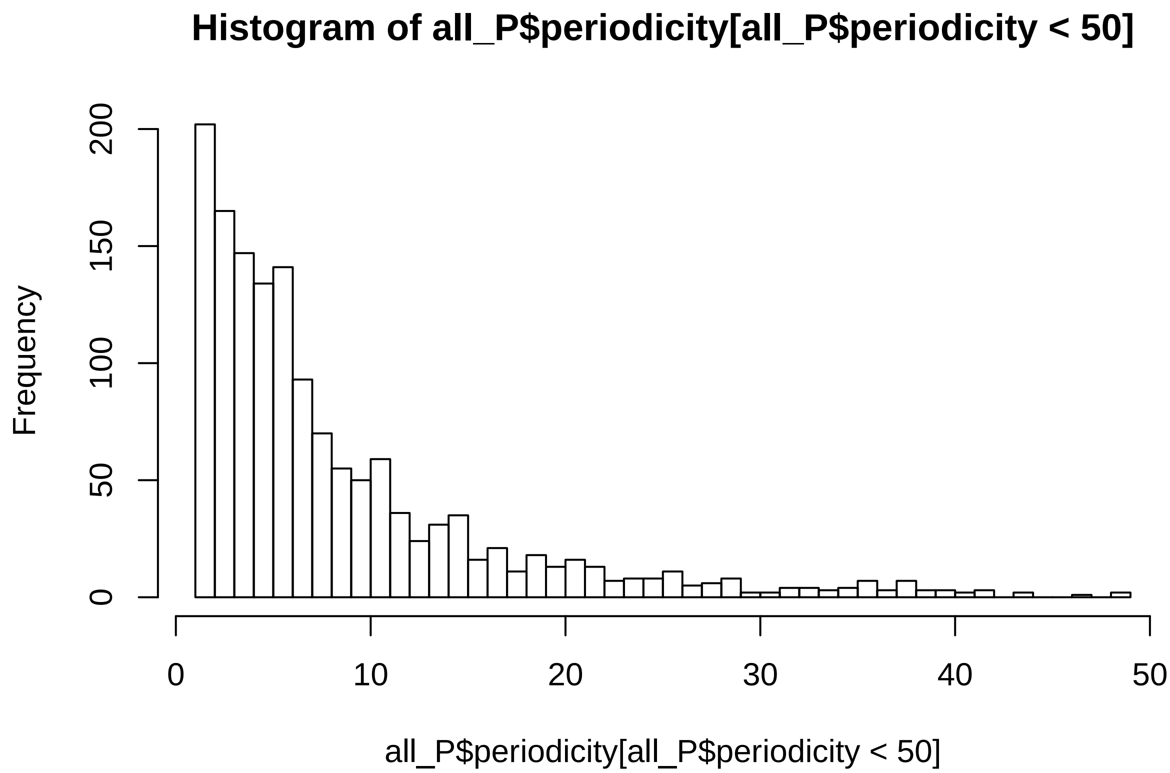
```r
periodicity(data = summer_iRB, recorder_name = 'August, Tom')
```

```
##      recorder periodicity periodicity_variation median_streak sd_streak
## 1 August, Tom          30              17.81853             1 0.4472136
##   max_streak n_days
## 1          2      7
```

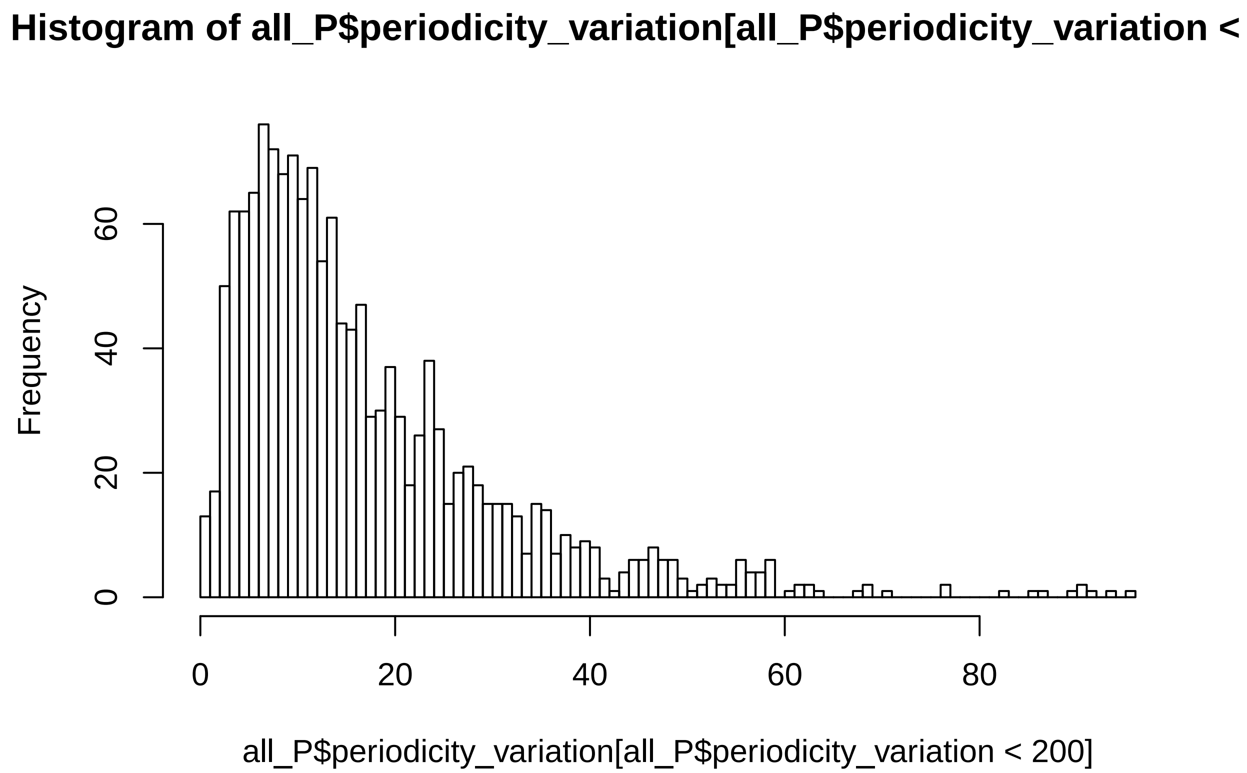
```r
# David is a much more regular recorder than Tom with less
# variation in periodicity and a longer max streak though
# Tom has less days of data to work with

# Run for everyone
all_P <- do.call(rbind, lapply(X = unique(iRB$recorders),
                               FUN = periodicity,
                               data = iRB))

# Lets have a look at some of these
head(all_P, 20)[c(5,8,1),]
```

```
##          recorder periodicity periodicity_variation median_streak
## 5    Allan, David           1              2.693493             1
## 8    Stewart, Tam           1              2.237841             1
## 1 Brookes , Anne           5              8.101543             1
##   sd_streak max_streak n_days
## 5 2.0710394         16    378
## 8 1.8107359         15    272
## 1 0.5918027          4     70
```

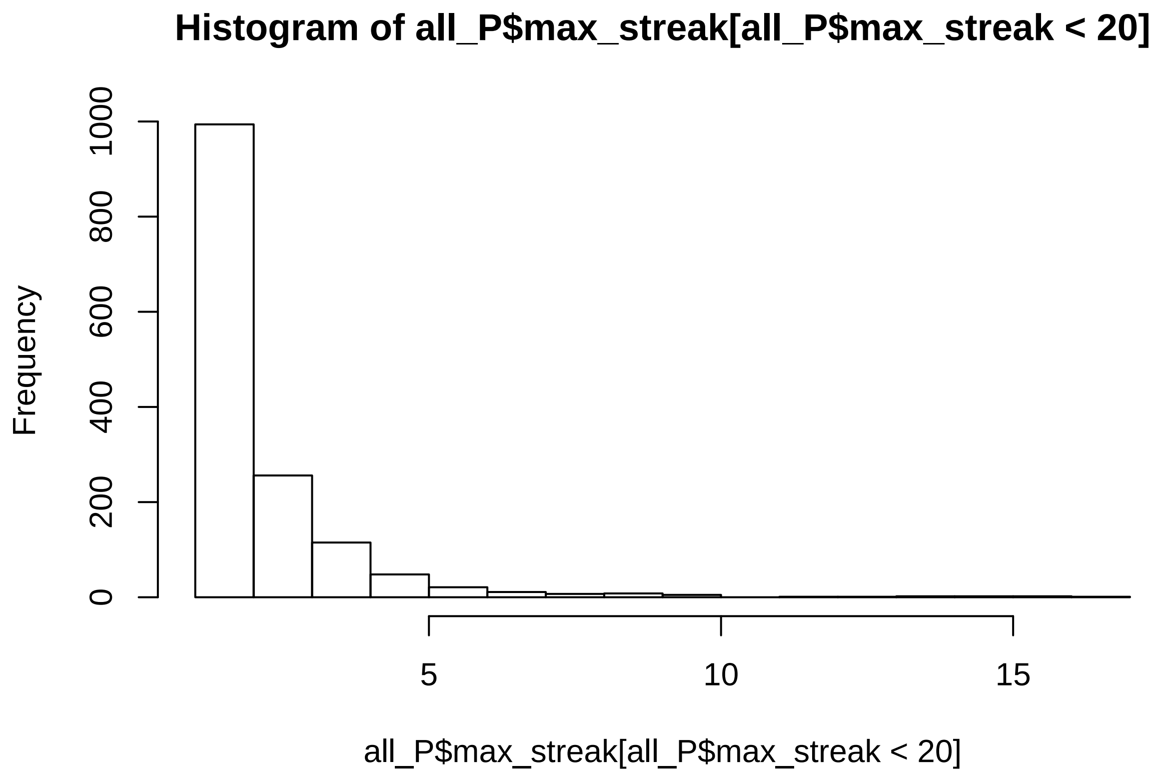
```r
# David a Tam are both very studious recorders with
# long max streaks and very low periodicity.
# Anne is less studious but still has a low periodicity

# Nice poission dist. for periodicity
hist(all_P$periodicity[all_P$periodicity < 50],
     breaks = 50)
```




```
# Nice poission dist. for periodicity_variation (long tail)
hist(all_P$periodicity_variation[all_P$periodicity_variation < 200],
     breaks = 100)
```




```r
# Dist. of max_streak
hist(all_P$max_streak[all_P$max_streak < 20],
     breaks = 20)
```



By using the summer data only this analysis seems to be better than an earlier one that included all data. These metrics cannot be calculate for people who have only made one record. I have included a parameter `day_limit` to allow us to set a limit at which we calculate these metrics.

Spatial Meterics

These metrics deal with the spatial distribution of records

Area and heterogenity of recording

I think the first step for all of these metrics is to turn the points into a SpatialPoints object which will allow us to manipulate then more easily. Once we have done that we can calculate MCP (minimum convex polygons) around the points. We might want to change this method to a method that is less susceptible to outliers such as alpha hull (we can talk to Colin about this). Here I use 95% MCP as the total recording area (hopefully removing outliers), and use the ratio of 95%:50% as a measure of heterogeneity.

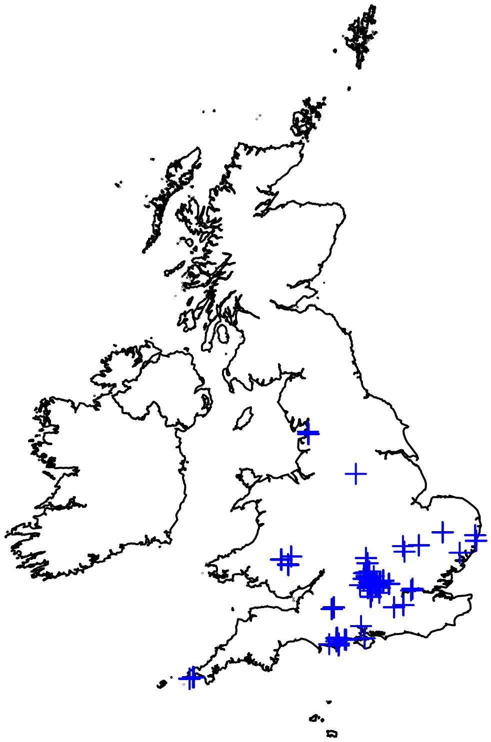
```
# Function takes data and username and returns spatial metrics
spatial_behaviour <- function(data, recorder_name,
                              latitude_col, longitude_col,
                              recorder_col = 'recorders',
                              upper_percentile = 95,
                              lower_percentile = 60,
                              h = 5000,
                              res = 1000){
```

```
if(is.factor(recorder_name)){
  recorder_name <- as.character(recorder_name)
}

n_row <- nrow(iRB[iRB[,recorder_col] == recorder_name, ])

if(n_row >= 5){

  # Convert to SpatialPoints
  spPoints_LL <- SpatialPoints(iRB[iRB[,recorder_col] == recorder_name,
                                   c(longitude_col, latitude_col)])
  # Data is lat long
  proj4string(spPoints_LL) <- CRS("+init=epsg:4326")

  # Convert to Eastings Northings to get meters on X and Y
  spPoint_UK <- spTransform(spPoints_LL, "+init=epsg:27700")

  # set up grid
  # This allows us to ensure there is space for the
  # isoclines to be drawn and that the pixel res is the
  # same - here 1km
  minlong <- floor(bbox(spPoint_UK)['long','min'] - (10*h))
  minlat <- floor(bbox(spPoint_UK)['lat','min'] - (10*h))
  maxlong <- ceiling(bbox(spPoint_UK)['long','max'] + (10*h))
  maxlat <- ceiling(bbox(spPoint_UK)['lat','max'] + (10*h))

  grid_ras <- raster(ext = extent(minlong, maxlong, minlat, maxlat),
                     res = res,
                     crs = projection(spPoint_UK))

                     # matrix(NA,
                     # ncol = ceiling(nlat),
                     # nrow = ceiling(nlong)))

  grid_SP <- as(grid_ras, "SpatialPixels")

  # Try kernel density
  KD <- kernelUD(xy = spPoint_UK, h = h, grid = grid_SP)
  # image(KD)
  KA <- kernel.area(KD,
              percent = c(lower_percentile, upper_percentile),
              unin = "m",
              unout = "km2")
  area_upper <- KA[2]
  area_lower <- KA[1]
  poly_lower <- getverticeshr(KD, percent = lower_percentile)
  poly_upper <- getverticeshr(KD, percent = upper_percentile)
  rm(list = 'KD')
  npolys_upper <- length(poly_upper@polygons[[1]]@Polygons)
  npolys_lower <- length(poly_lower@polygons[[1]]@Polygons)

  # # Calculate the Local convex hull
  # LCH_poly <- LoCoH.k(xy = spPoint_UK,
```

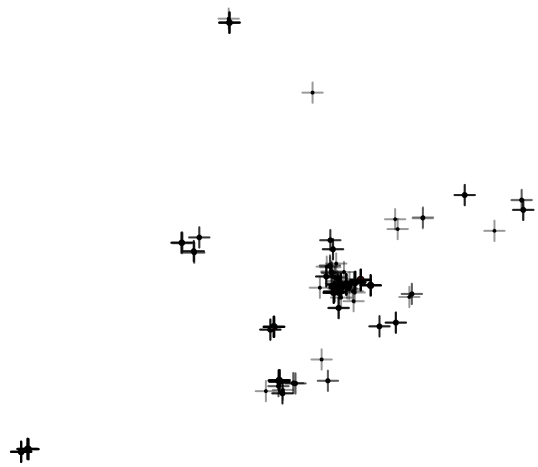
```
    #                      k =  10,
    #                      unin = 'm',
    #                      unout = 'km',
    #                      duplicates = 'remove')
    #
    # # Extract percentiles
    # LCH_MCHu <- spoldf2MCHu(LCH_poly)
    # poly_upper <- getverticeshr.MCHu(LCH_MCHu,
    #                                  percent = upper_percentile)
    # poly_lower <- getverticeshr.MCHu(LCH_MCHu,
    #                                  percent = lower_percentile)
    #
    # npolys_upper <- length(poly_upper@polygons[[1]]@Polygons)
    # npolys_lower <- length(poly_lower@polygons[[1]]@Polygons)
    #
    # area_upper <- MCHu2hrsize(x = LCH_MCHu, percent = upper_percentile,
    #                           plotit = FALSE)
    # area_lower <- MCHu2hrsize(x = LCH_MCHu, percent = lower_percentile,
    #                           plotit = FALSE)

    return(list(recorder = recorder_name,
                spPoint_UK = spPoint_UK,
                # poly = LCH_poly,
                poly_upper = poly_upper,
                poly_lower = poly_lower,
                upper_n_poly = npolys_upper,
                lower_n_poly = npolys_lower,
                upper_area = area_upper,
                lower_area = area_lower,
                ratio = area_lower/area_upper,
                n = n_row))
  } else {
    return(list(recorder = recorder_name,
                spPoint_UK = NA,
                # poly = NA,
                poly_upper = NA,
                poly_lower = NA,
                upper_n_poly = NA,
                lower_n_poly = NA,
                upper_area = NA,
                lower_area = NA,
                ratio = NA,
                n = n_row))
  }

}

# Function for plotting records
plot_ratio <- function(data){
  om <- par("mar")
  omf <- par('mfrow')
  par(mfrow = c(1,2), mar = c(1,1,1,1))
  data(UK)
```

```
  plot_GIS(UK, new.window = FALSE,
           main = 'Distribution of records',
           show.axis = FALSE, show.grid = FALSE)
  points(data$spPoint_UK, pch = 3, col = 'blue')

  # Plot heat map
  plot(data$poly_upper,
       main = paste('\n\n', data$recorder, '-', 'Ratio:',
                    round(data$ratio, 4), '\n',
                    'Upper/lower polygons:', data$upper_n_poly,
                    '/', data$lower_n_poly, '\n',
                    'Total Area:', data$upper_area),
       col = 'grey')
  plot(data$poly_lower, add = TRUE,
       col = 'red', border = 'red')
  points(data$spPoint_UK, col = rgb(0,0,0,0.4),
         pch = 3)
  par(mfrow = omf,
      mar = om)
}

for(h in c(1000, 5000, 10000)){
  DD <- spatial_behaviour(data = iRB, recorder_name = 'Roy, David',
                          latitude_col = 'lat', longitude_col = 'long',
                          h = h)
  plot_ratio(data = DD)

}
```

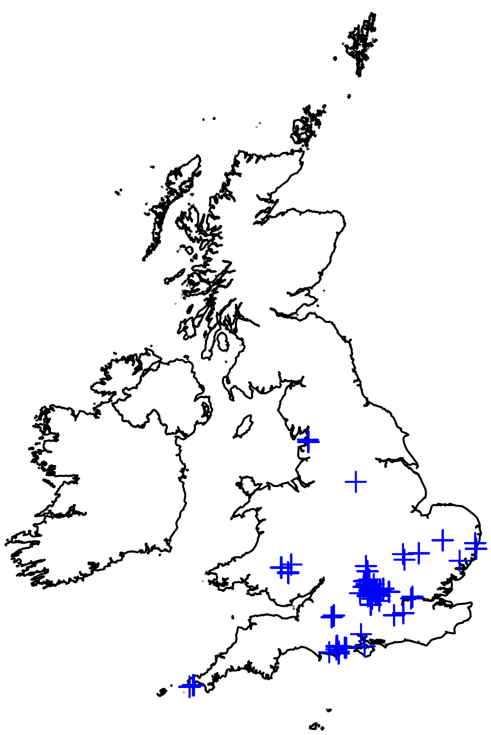
Distribution of records



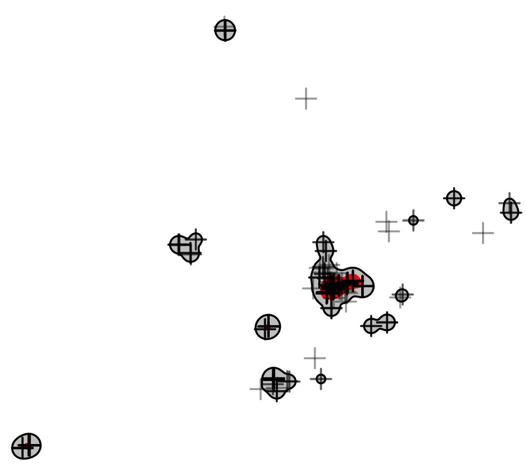
Roy, David − Ratio: 0.1118
Upper/lower polygons: 42 / 10
Total Area: 662

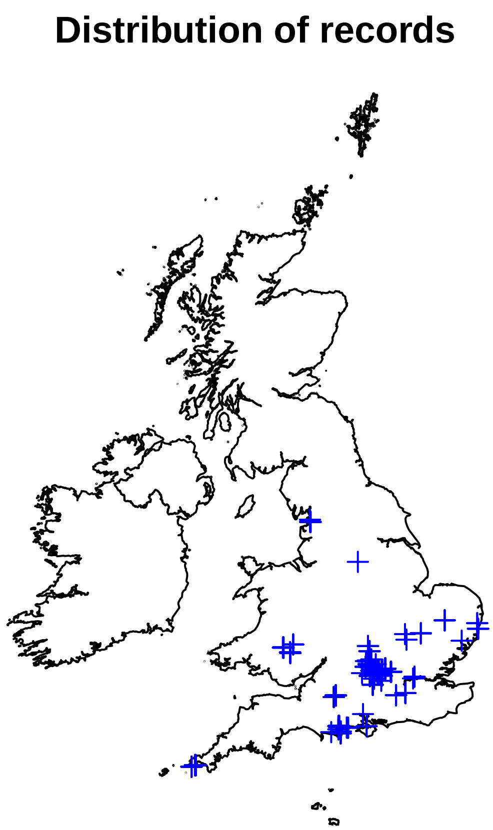


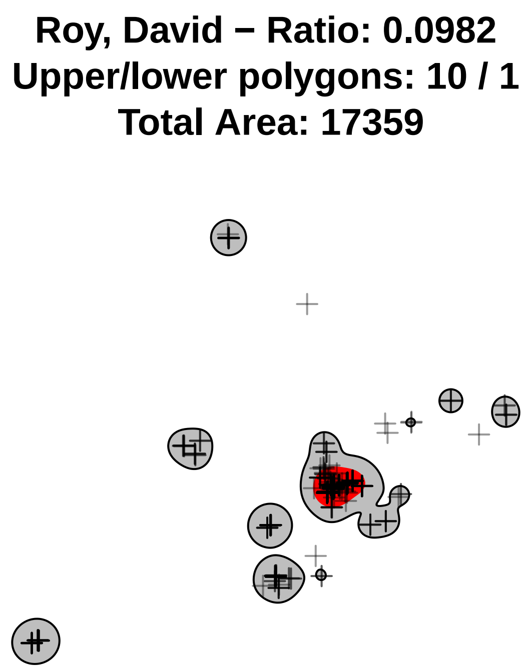
Distribution of records



Roy, David − Ratio: 0.1089
Upper/lower polygons: 12 / 3
Total Area: 6683








```r
# for(recorder in c('Partridge, Francesca', 'Harley, Ross')){
#
#   RD <- spatial_behaviour(data = iRB, recorder_name = recorder,
#                           latitude_col = 'lat', longitude_col = 'long',
#                           h = 5000)
#   plot_ratio(data = RD)
#
# }

# Apply to all recorders
pdf(file = 'recorderAreas.pdf')
all_spatial <- lapply(unique(iRB$recorders), FUN = function(x){
  # cat(paste(x, '\n'))
  recorder_info <- spatial_behaviour(data = iRB, recorder_name = x,
                                     latitude_col = 'lat', longitude_col = 'long')
  if(!is.na(recorder_info$ratio)) plot_ratio(recorder_info)
  return(data.frame(recorder = recorder_info$recorder,
                    upper_area = recorder_info$upper_area,
                    lower_area = recorder_info$lower_area,
                    upper_n_poly = recorder_info$upper_n_poly,
                    lower_n_poly = recorder_info$lower_n_poly,
                    ratio = recorder_info$ratio,
                    n = recorder_info$n))
})
dev.off()
```

```
## pdf
##   2
```

```r
# combine results
temp <- do.call(rbind, all_spatial)
temp <- temp[temp$n > 400, ]

# Lets have a look at some people who have recorded a lot
temp[order(temp$ratio, decreasing = TRUE),]
```

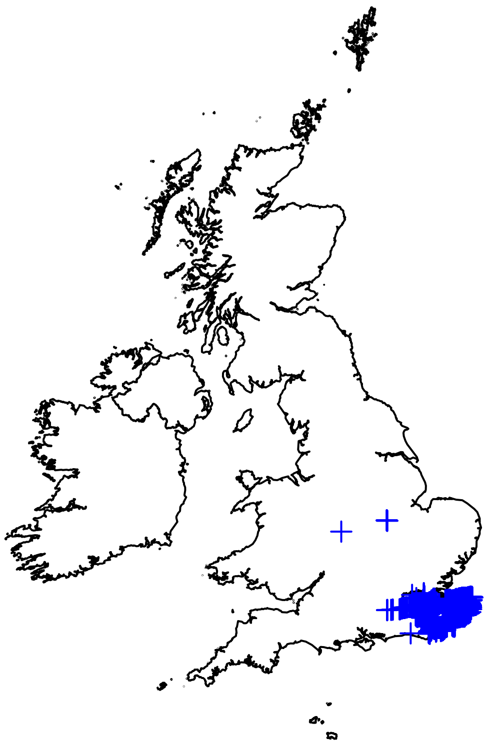
```
##                      recorder upper_area lower_area upper_n_poly
## 9541      Partridge, Francesca       5905       1905            2
## 9542          Cornish, Stephen        493        146            1
## 951                  Limb, Ken       6531       1881            7
## 95123           Hunter, Amands       1653        464            1
## 9575                Jones, Dave        564        156            1
## 9555                Leaver, Kim       2169        561            3
## 95129           Saville, Simon       4290       1058            6
## 95187              Atkin, Paul       2480        592            3
## 9557             Lunnon, Marie        822        193            1
## 9583              Gillie, Tony       2635        613            3
## 958                 Cox, Steve       7884       1633           13
## 9533             Shanks, Scott       5888       1219           10
## 9562              Sell, Claire       1646        338            1
## 9516              Kilbey, Dave       6253       1276            9
## 95205             Ford, Rachel        843        163            2
## 9556            Steele, Andrew      16070       3002           28
## 9582          Checkley, Graham       1746        323            2
## 9554               Hill, Brian       4648        856           10
## 9540              Cowton, Keith       4424        790            7
## 9512              Bowles, Nick       1941        333            4
## 957                 fenn, paul       3192        539            6
## 9578               Sims, Clive       3392        514            6
## 956             Newbould, John       5372        811            8
## 955             Warren, Martin      11701       1747           20
## 9568              Stewart, Tam       9098       1286           16
## 95264            Dawson, Steve       1596        213            1
## 9564              Fox, Richard       7776       1012           14
## 95130 Lonsdale, Liz and Steve      11053       1429           25
## 9511             Austin, David       3607        434            5
## 95190               Roy, David       6683        728           12
## 9523            Shersby, Megan       4605        496            7
## 95169             Harley, Ross       2544        256            7
## 95234        Pennington, Robert       3798        335            8
## 9538              Allan, David       4360        242           13
## 95145           shilland, ewan       9871        242           26
##       lower_n_poly      ratio    n
## 9541             5 0.32260796 1438
## 9542             1 0.29614604  487
## 951              7 0.28801102  622
## 95123            1 0.28070175 1109
## 9575             1 0.27659574 2240
## 9555             1 0.25864454  542
## 95129            4 0.24662005  441
```

```
## 95187             2 0.23870968  615
## 9557              1 0.23479319  450
## 9583              2 0.23263757 1123
## 958               5 0.20712836 1004
## 9533              5 0.20703125  513
## 9562              1 0.20534629  555
## 9516              2 0.20406205  764
## 95205             1 0.19335706  433
## 9556             14 0.18680772  571
## 9582              1 0.18499427 1810
## 9554              2 0.18416523  858
## 9540              3 0.17857143  407
## 9512              1 0.17156105  609
## 957               2 0.16885965 2525
## 9578              2 0.15153302  873
## 956               2 0.15096798 1012
## 955               1 0.14930348 2483
## 9568              3 0.14134975 1799
## 95264             1 0.13345865  838
## 9564              4 0.13014403 1158
## 95130             6 0.12928617  565
## 9511              2 0.12032160  444
## 95190             3 0.10893311  590
## 9523              2 0.10770901  478
## 95169             1 0.10062893  656
## 95234             1 0.08820432  985
## 9538              1 0.05550459 3307
## 95145             1 0.02451626 1641
```

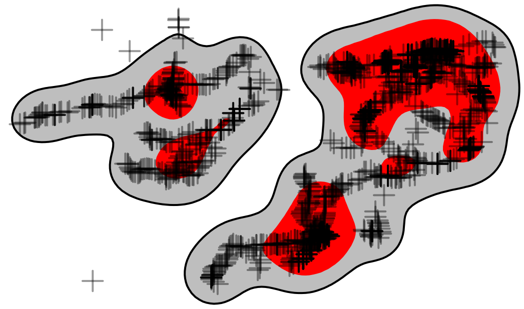
Lets have a look at two people with very different ratios

```r
# Get the names of top and bottom
temp <- temp[order(temp$ratio, decreasing = TRUE),]
top <- as.character(head(temp$recorder, 1))
bottom <- as.character(tail(temp$recorder, 1))

# Plot the top and bottom ratio recorder
for(i in c(top, bottom)){
  top_d <- spatial_behaviour(data = iRB,
                             recorder_name = i,
                             latitude_col = 'lat',
                             longitude_col = 'long')
  plot_ratio(data = top_d)
}
```

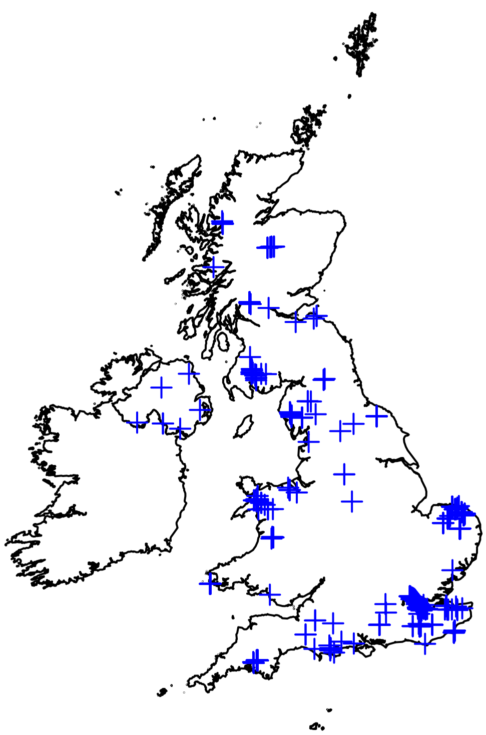
Distribution of records



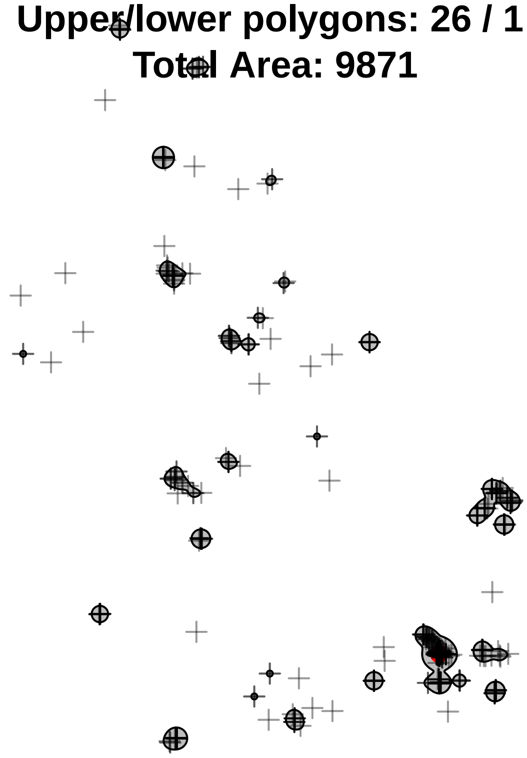
Partridge, Francesca − Ratio: 0.3226
Upper/lower polygons: 2 / 5
Total Area: 5905



Distribution of records



shilland, ewan − Ratio: 0.0245
Upper/lower polygons: 26 / 1
Total Area: 9871



Taxanomic Metrics

These metric relate the the species that people record

Taxanomic Breadth

This is simply a measure of the proportion of taxa a person has recorded. Note this is going to be correlated to the number of records.

```
taxa_breadth <- function(data, recorder_name,
                         sp_col = 'preferred_taxon',
                         recorder_col = 'recorders'){

  data_rec <- data[data[,recorder_col] == recorder_name, c(sp_col, recorder_col)]

  return(data.frame(recorder = recorder_name,
                    taxa_breadth = length(unique(data_rec[ ,sp_col])),
                    taxa_prop = length(unique(data_rec[ ,sp_col]))/length(unique(data[ ,sp_col])),
                    n = nrow(data_rec)))
}

taxa_breadth <- do.call(rbind, lapply(unique(iRB$recorders), FUN = taxa_breadth, data = iRB))

temp <- taxa_breadth[taxa_breadth$n > 400, ]

# Lets have a look at some people who have recorded a lot
temp[order(temp$taxa_prop, decreasing = TRUE),]
```

```
##                    recorder taxa_breadth taxa_prop    n
## 39           Warren, Martin           52 0.6265060 2434
## 5              Allan, David           51 0.6144578 3180
## 103       Pennington, Robert          49 0.5903614  969
## 113             Hill, Brian           48 0.5783133  851
## 1356         Saville, Simon           48 0.5783133  441
## 123              Cox, Steve           47 0.5662651  991
## 175             Sims, Clive           47 0.5662651  864
## 143            Fox, Richard           46 0.5542169 1147
## 158            Harley, Ross           45 0.5421687  682
## 383          Steele, Andrew           45 0.5421687  563
## 256           Cowton, Keith           42 0.5060241  445
## 395             Atkin, Paul           42 0.5060241  615
## 26               fenn, paul           41 0.4939759 2503
## 180               Limb, Ken           41 0.4939759  622
## 488            Kilbey, Dave           41 0.4939759  780
## 87            Dawson, Steve           40 0.4819277  789
## 65             Gillie, Tony           39 0.4698795 1112
## 523          Shersby, Megan           38 0.4578313  478
## 19               Roy, David           37 0.4457831  615
## 41   Lonsdale, Liz and Steve          37 0.4457831  542
## 78           shilland, ewan           36 0.4337349 1636
## 339             Bowles, Nick          36 0.4337349  590
## 43            Newbould, John          33 0.3975904 1001
## 11      Partridge, Francesca          32 0.3855422 1418
```

```
## 45              Sell, Claire          32 0.3855422  555
## 139           Hunter, Amands          31 0.3734940 1090
## 8               Stewart, Tam          29 0.3493976 1811
## 197            Lunnon, Marie          28 0.3373494  444
## 109           Shanks, Scott           26 0.3132530  513
## 104              Leaver, Kim          24 0.2891566  537
## 72               Jones, Dave          23 0.2771084 2207
## 96          Checkley, Graham          22 0.2650602 1813
## 140            Austin, David          22 0.2650602  441
## 52          Cornish, Stephen          19 0.2289157  487
## 100             Ford, Rachel          15 0.1807229  431
```

Species Rarity

We want to capture the rarity of the species that people record. For example are they just recording the common species or are they only recording the rare ones, or perhaps they are recording everything. Since we don't know the real frequency distribution we can only compare people to the global average in the dataset. We can look to see what the distribution of species rank for each recorder is and how this compares to all records. A recorder only interested in rare species will have a median rank higher than the average. A recorder only recording common species will have a value lower than the average.

```
# Lets look at a recorder
species_rank <- function(data, recorder_name,
                         sp_col = 'preferred_taxon',
                         recorder_col = 'recorders'){

  data <- data[,c(sp_col, recorder_col)]
  rank_species <- rank(abs(table(data[,sp_col])-max(table(data[,sp_col]))))
  sp_counts <- table(data[,sp_col])

  rank_reps <- rep(rank_species, sp_counts)
  grand_median <- median(rank_reps)
  grand_sd <- sd(rank_reps)

  recorder_data <- data[data[,recorder_col] == recorder_name,]
  recorder_data$rank <- rank_species[recorder_data[ ,sp_col]]

  return(data.frame(recorder = as.character(recorder_name),
                    median = median(recorder_data$rank),
                    median_diff = median(recorder_data$rank) - grand_median,
                    stdev = sd(recorder_data$rank),
                    n = nrow(recorder_data)))
}

rarity_preference <- do.call(rbind,
                             lapply(unique(iRB$recorders),
                                    FUN = species_rank,
                                    data = iRB))

temp <- rarity_preference[rarity_preference$n > 400, ]

# Lets have a look at some people who have recorded a lot
temp[order(temp$median_diff, decreasing = TRUE),]
```

```
##                     recorder median median_diff     stdev    n
## 1356          Saville, Simon     13           5 12.191833  441
## 256            Cowton, Keith     12           4 10.283900  445
## 39            Warren, Martin     11           3 10.754206 2434
## 175              Sims, Clive     11           3 10.132960  864
## 339             Bowles, Nick     10           2  8.557264  590
## 395              Atkin, Paul     10           2  9.738285  615
## 523           Shersby, Megan     10           2  8.613459  478
## 8               Stewart, Tam      9           1 10.764394 1811
## 19                Roy, David      9           1  9.647095  615
## 26                fenn, paul      9           1  8.779256 2503
## 43            Newbould, John      9           1  8.245020 1001
## 45              Sell, Claire      9           1  8.912894  555
## 65              Gillie, Tony      9           1  8.645367 1112
## 103       Pennington, Robert      9           1  9.100094  969
## 109           Shanks, Scott       9           1  9.482688  513
## 113              Hill, Brian      9           1 10.226885  851
## 139           Hunter, Amands      9           1  7.199181 1090
## 158             Harley, Ross      9           1  9.410956  682
## 180                Limb, Ken      9           1  9.165788  622
## 197            Lunnon, Marie      9           1  7.004225  444
## 41   Lonsdale, Liz and Steve      8           0  8.646054  542
## 78            shilland, ewan      8           0  8.303214 1636
## 96          Checkley, Graham      8           0  6.931797 1813
## 104              Leaver, Kim      8           0  6.082150  537
## 143             Fox, Richard      8           0  9.681677 1147
## 383           Steele, Andrew      8           0  9.108308  563
## 488             Kilbey, Dave      8           0  9.170174  780
## 87             Dawson, Steve      7          -1  7.926813  789
## 100             Ford, Rachel      7          -1  5.281118  431
## 123               Cox, Steve      7          -1  9.048282  991
## 5               Allan, David      6          -2  8.643921 3180
## 11      Partridge, Francesca      6          -2  6.888191 1418
## 72               Jones, Dave      6          -2  4.862982 2207
## 52          Cornish, Stephen      5          -3  5.081520  487
## 140            Austin, David      5          -3  5.474312  441
```

Here `median_diff` gives the difference between the grand median for all records and the recorders median. This suggests `Saville, Simon` prefers to record rare species and `Cornish, Stephen` prefers to record common species.

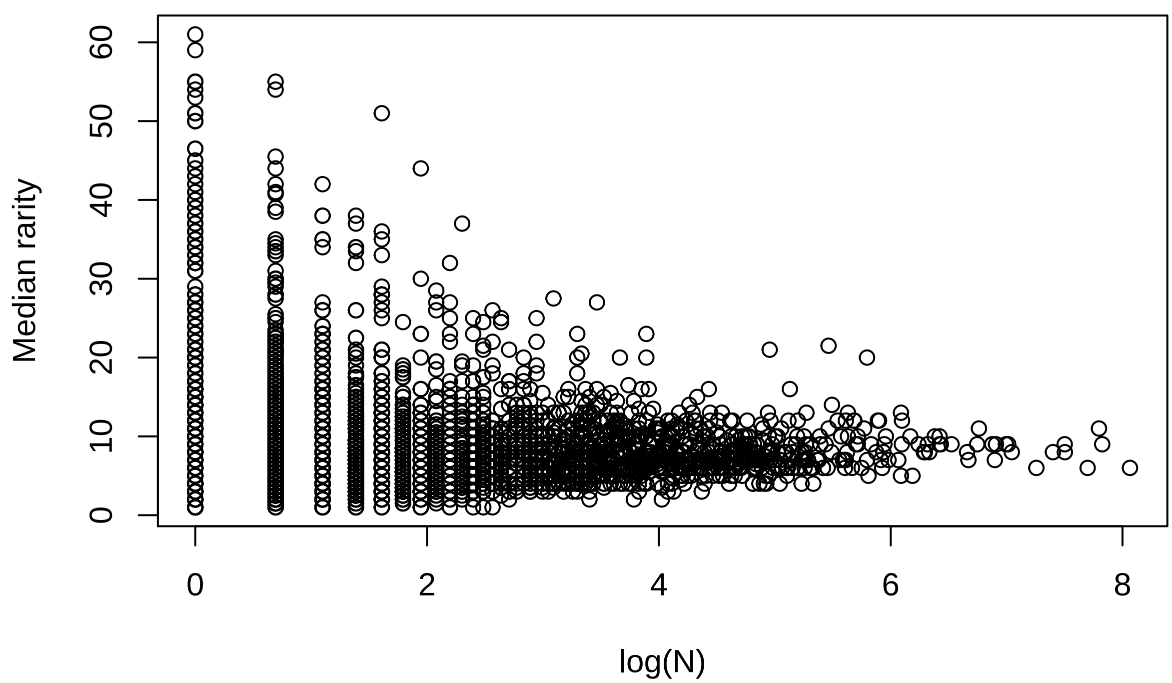
This could be correlated to the number of records.

```r
mod <- glm(median ~ log(n), data = rarity_preference, family = 'quasipoisson')
summary(mod)
```

```
## 
## Call:
## glm(formula = median ~ log(n), family = "quasipoisson", data = rarity_preference)
## 
## Deviance Residuals: 
##     Min       1Q   Median       3Q      Max  
## -3.7397  -1.6376  -0.4224   0.7604  10.7394  
## 
```

```
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.334807   0.018526 126.030   <2e-16 ***
## log(n)      -0.070761   0.008457  -8.367   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for quasipoisson family taken to be 5.26387)
##
##     Null deviance: 17611  on 3944  degrees of freedom
## Residual deviance: 17232  on 3943  degrees of freedom
## AIC: NA
##
## Number of Fisher Scoring iterations: 5
```

```r
plot(log(rarity_preference$n),
     rarity_preference$median,
     xlab = 'log(N)',
     ylab = 'Median rarity')
```

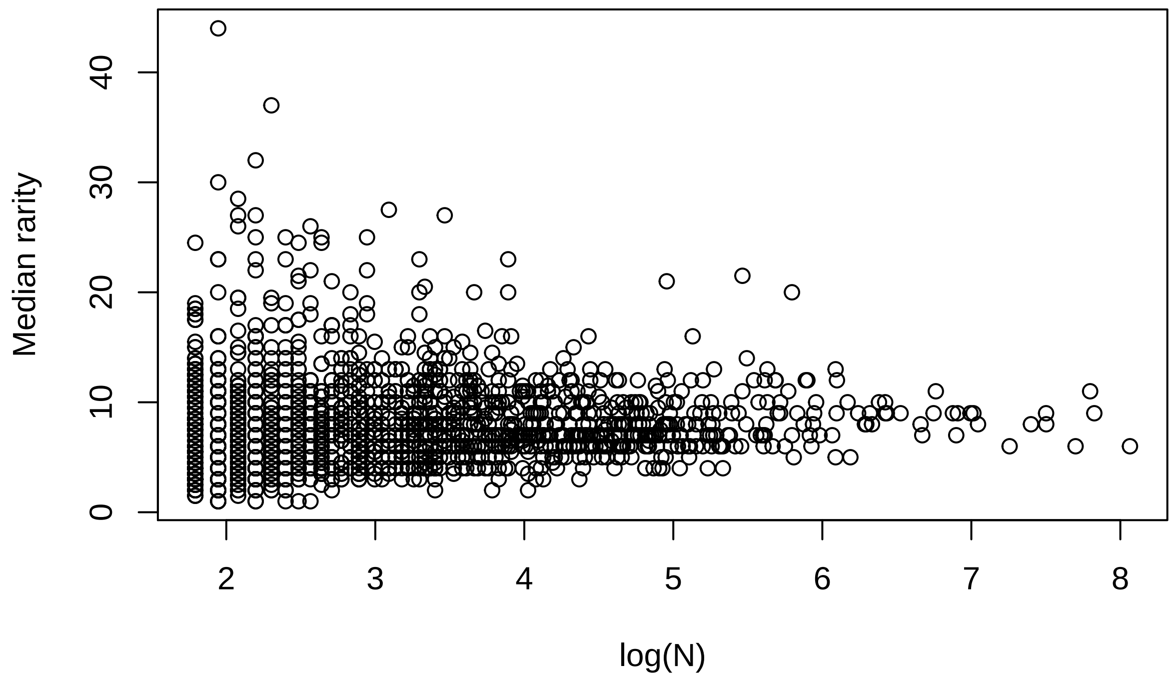


There is a significant negative relationship. The more records you make the lower your median value. This could be a result of the fact that people who make only a few records record rare stuff?

```r
rarity_preference_above <- rarity_preference[rarity_preference$n > 5, ]
mod <- glm(median ~ log(n), data = rarity_preference_above, family = 'quasipoisson')
summary(mod)
```

```
##
## Call:
## glm(formula = median ~ log(n), family = "quasipoisson", data = rarity_preference_above)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.1729  -0.9646  -0.2134   0.6260   8.7863
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 2.076269   0.034427  60.310   <2e-16 ***
## log(n)      0.007479   0.010513   0.711    0.477
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for quasipoisson family taken to be 2.087498)
##
##     Null deviance: 3474.5  on 1877  degrees of freedom
## Residual deviance: 3473.5  on 1876  degrees of freedom
## AIC: NA
##
## Number of Fisher Scoring iterations: 4
```

```r
plot(log(rarity_preference_above$n),
     rarity_preference_above$median,
     xlab = 'log(N)',
     ylab = 'Median rarity')
```



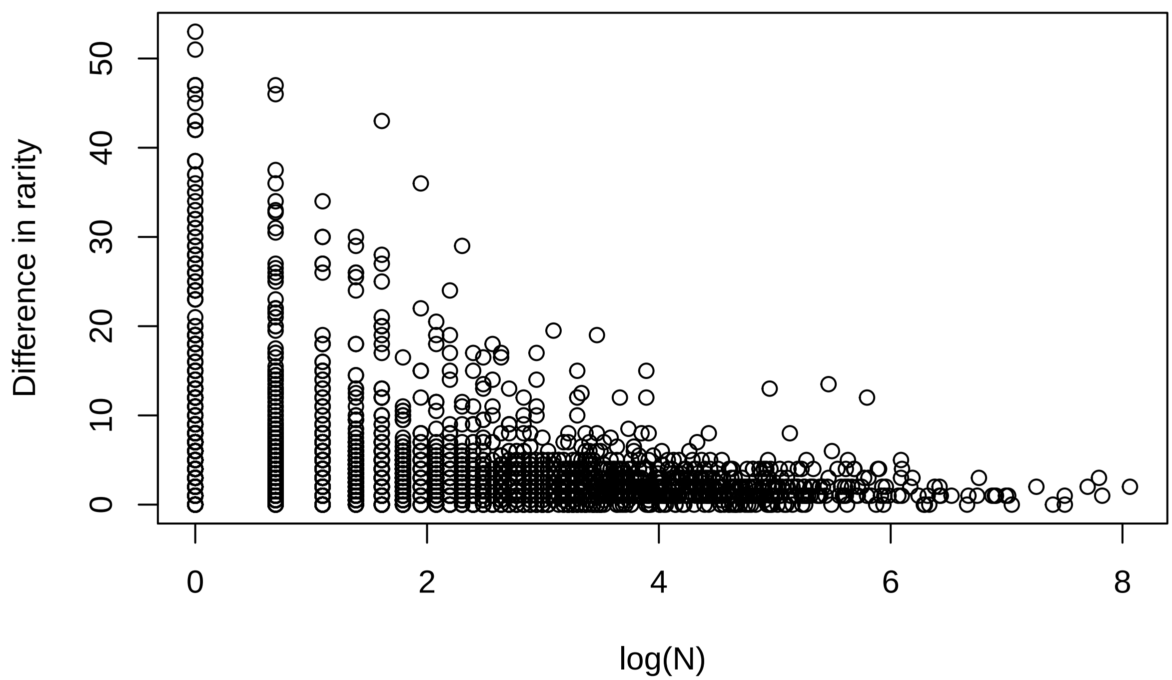
Okay, the relationship falls down once we get rid of the people who only record a few species. I suggest this metric not be estimates for people who contribute only a few records. The relationship might actually be between deviation from the median and `n`.

```
rarity_preference$median_diff_abs <- abs(rarity_preference$median_diff)
mod <- glm(median_diff_abs ~ log(n), data = rarity_preference, family = 'quasipoisson')
summary(mod)
```

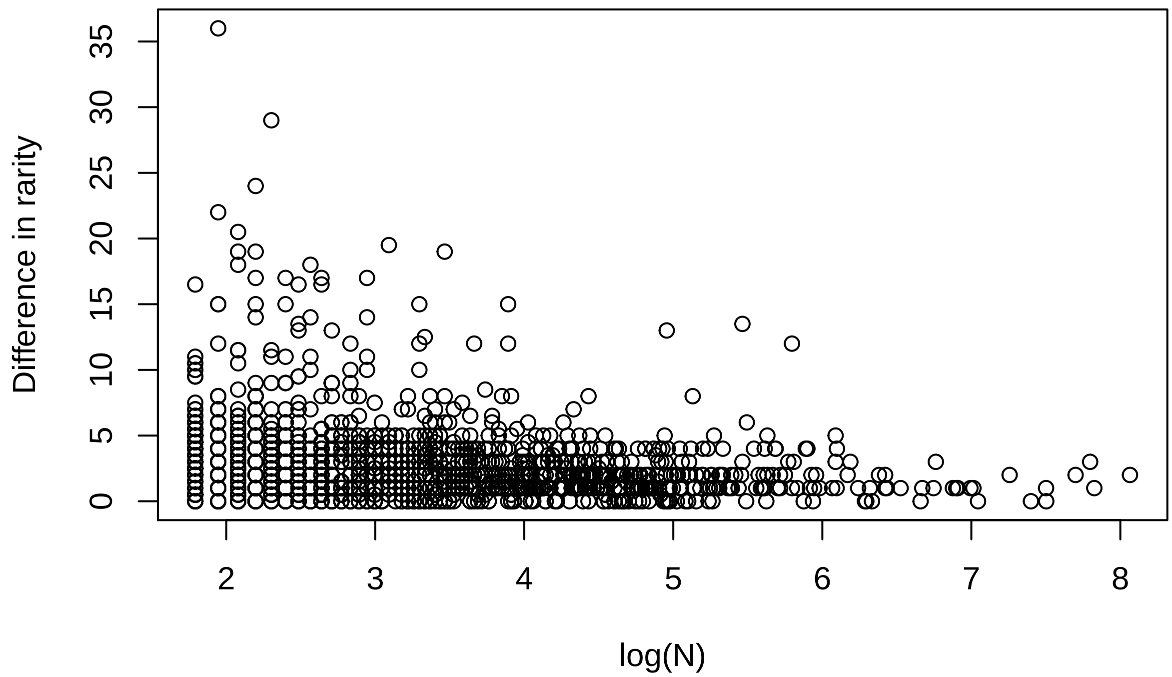
```
##
## Call:
## glm(formula = median_diff_abs ~ log(n), family = "quasipoisson",
##     data = rarity_preference)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.8262  -1.3717  -0.5038   0.3857  10.8928
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.99062    0.02414   82.47   <2e-16 ***
## log(n)      -0.31472    0.01388  -22.68   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for quasipoisson family taken to be 5.259472)
##
```

```
##     Null deviance: 18546  on 3944  degrees of freedom
## Residual deviance: 15454  on 3943  degrees of freedom
## AIC: NA
##
## Number of Fisher Scoring iterations: 5
```

```r
plot(log(rarity_preference$n),
     rarity_preference$median_diff_abs,
     xlab = 'log(N)',
     ylab = 'Difference in rarity')
```




```r
plot(log(rarity_preference$n[rarity_preference$n >5]),
     rarity_preference$median_diff_abs[rarity_preference$n >5],
     xlab = 'log(N)',
     ylab = 'Difference in rarity')
```



The more records you record the less you deviate from the median. This is probably because you only get extreme values where the sample size is small.

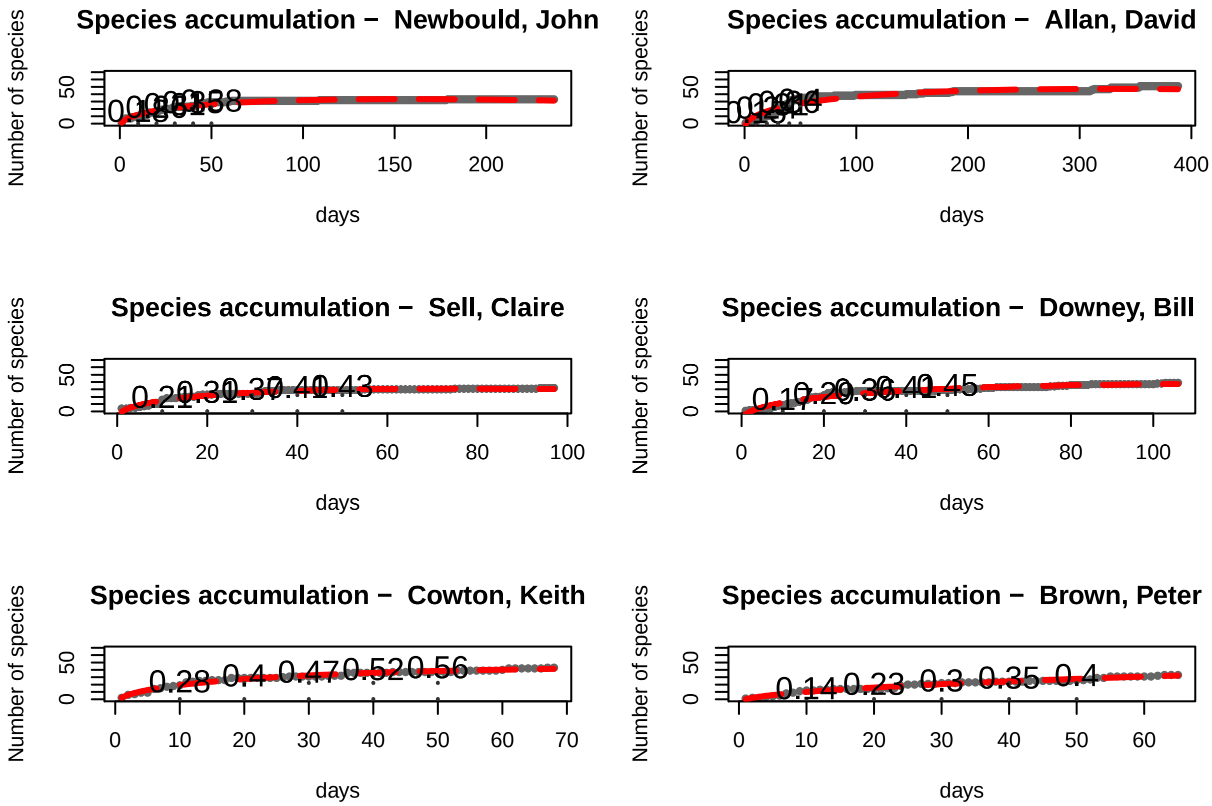
Species accumulation curve

The rate at which a recorder accumulates species over time recorded could be seen as a measure of their effort and expertise. The time scale will be measured in days recorded (not days elapsed). This measure will need to be normalised within a taxonomic group as clearly one can accumulate species more rapidly for birds than amphibians and reptiles. Note that while we look only at active days we have no way of accounting for actual time (hrs) recorded as a 10 minute search on one days appears the same as a 3 hour search on another. This metric is therefore an approximation only.

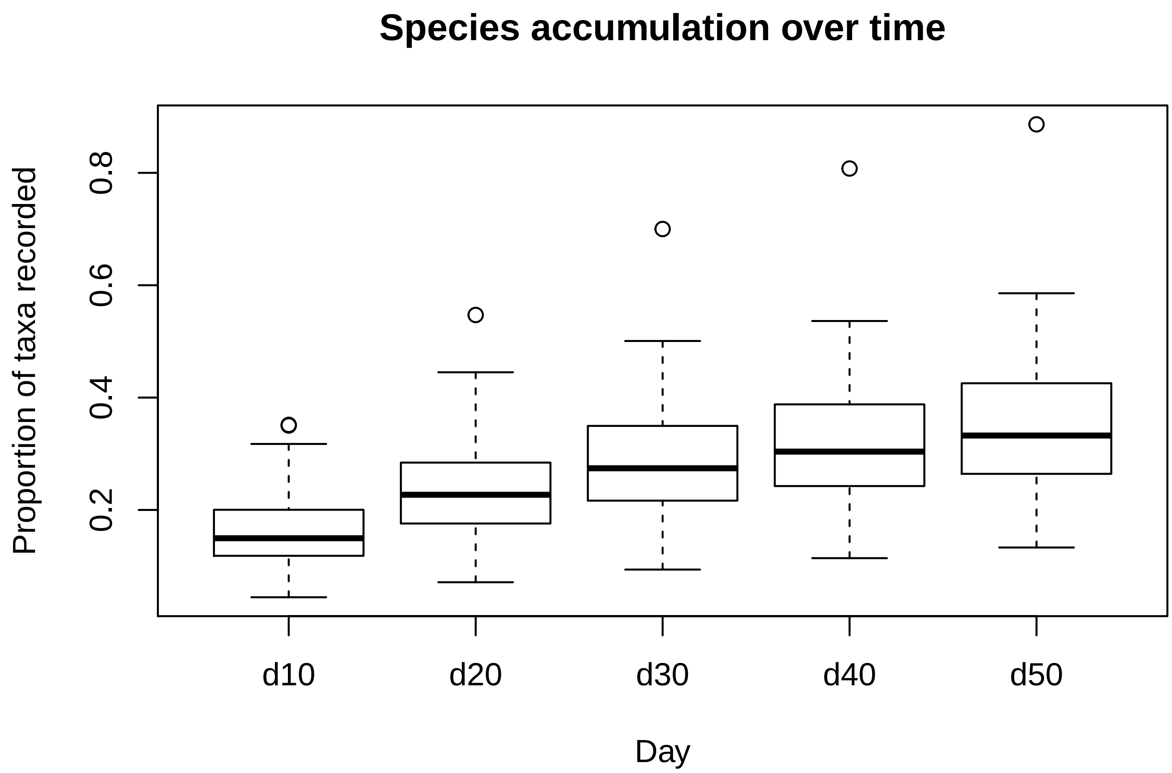
```
# This can be run with all data
species_accumulation <- function(data, recorder_name,
                                 n_taxa, prediction_days = c(10, 50),
                                 sp_col = 'preferred_taxon',
                                 date_col = 'date_start',
                                 recorder_col = 'recorders',
                                 plot = FALSE, ...){

  # Get the data for this recorder
  rec_data <- data[data[,recorder_col] == recorder_name,
                   c(date_col, sp_col)]

  # sort dates
  dates <- sort(unique(rec_data$date_start))
```

```
# Only carry out this test on people with more than the
# required number of days in their data
if(length(dates) >= max(prediction_days)){

  # accumulation function
  acc <- function(x){
    length(unique(rec_data[,sp_col][rec_data[,date_col] <= x]))
  }

  species_accumulation_data <- sapply(dates, FUN = acc)
  day <- seq_along(species_accumulation_data)

  # Fit a model
  m <- glm(formula = species_accumulation_data ~ day + sqrt(day))
  m_sum <- summary(m)
  # predict atleast up to day 100 (could be dangerous)
  # but just for visualiation purposes
  days_to_predict <- ifelse(test = length(dates) > max(prediction_days),
                            yes = length(dates),
                            no = prediction_days)
  predicted <- predict(m, newdata = data.frame(day = 1:days_to_predict))

  if(plot){
    plot(species_accumulation_data,
         main = paste('Species accumulation - ', recorder_name),
         xlab = 'days',
         ylab = 'Number of species',
         col = 'grey40',
         pch = 20,
         ... = ...)
    lines(predicted, col = 'red', lwd = 3, lty = 5)
    for(pred_day in prediction_days){
      lines(x = rep(pred_day, 2), y = c(0, predicted[pred_day]),
      lty = 3, col = 'grey20', lwd = 2)
      text(x = pred_day, y = predicted[pred_day] + 5, cex = 1.5,
         labels = round(predicted[pred_day]/n_taxa, 2))
    }
  }

  # create named vector for predictions
  x <- predicted[prediction_days]/n_taxa
  names(x) <- paste0('d', prediction_days)
  x <- data.frame(t(x))
  x$recorder <- as.character(recorder_name)

  return(list(species_accumulation_data = species_accumulation_data,
              predicted_data = predicted,
              day_pred = x,
              n_day = length(dates)))

} else {

  # empty df
```

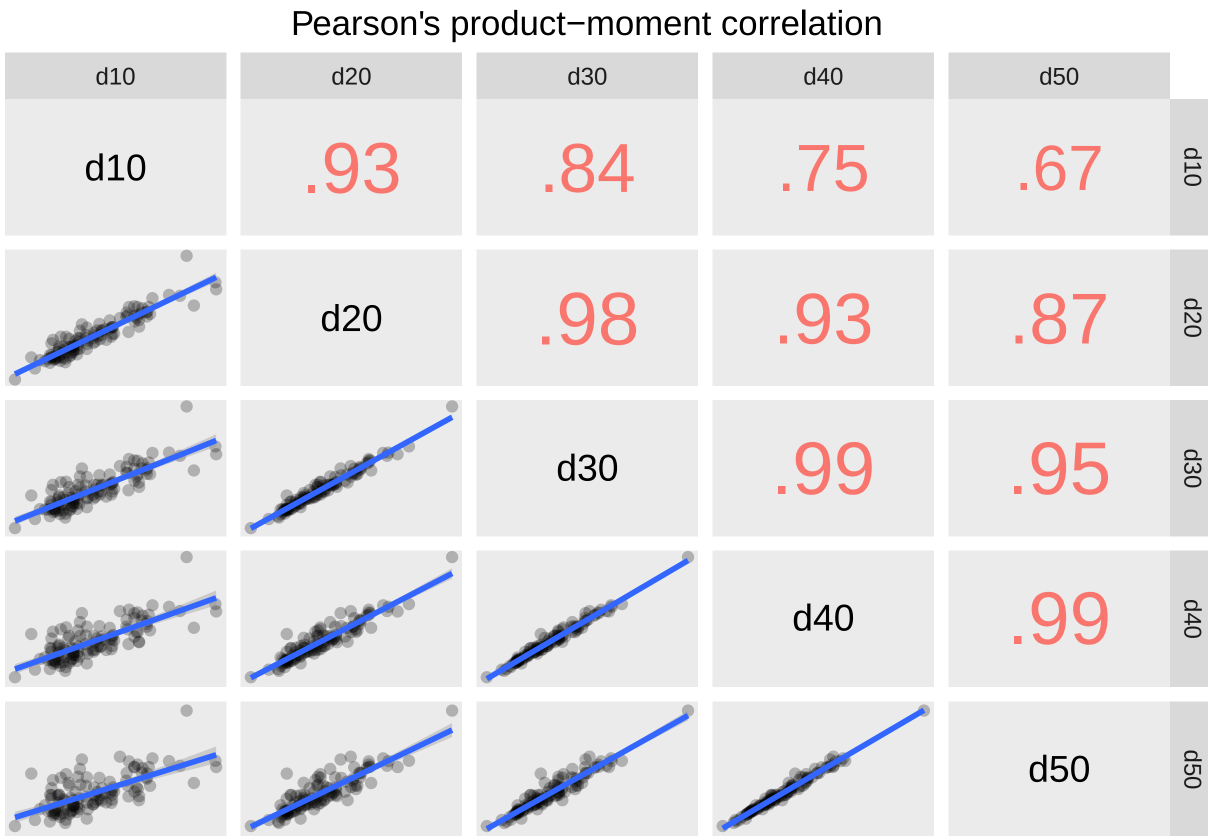
```
    eDF <- as.data.frame(cbind(matrix(data = rep(NA,
                                                 length(prediction_days)),
                                      nrow = 1,
                                      dimnames = list(recorder_name, paste0('d', prediction_days)))))

    eDF$recorder <- as.character(recorder_name)

    return(list(species_acculation_data = NA,
                predicted_data = NA,
                day_pred = eDF,
                n_day = length(dates)))

  }
}

# Run for a few people
par(mfrow = c(3, 2))

# top recorders
user_dates <- tapply(iRB$date_start, iRB$recorders, FUN = function(x) length(unique(x)))
recorders <- sample(names(sort(user_dates[user_dates > 50], decreasing = TRUE)), size = 6)

for(recorder in recorders){
user_acc <- species_accumulation(data = iRB, n_taxa = length(unique(iRB$preferred_taxon)),
                                 recorder_name = recorder,
                                 plot = TRUE,
                                 prediction_days = seq(10, 50, 10),
                                 # xlim = c(0, 100),
                                 ylim = c(0, length(unique(iRB$preferred_taxon))))
}
```




```
par(mfrow = c(1, 1))

# Calculate for everyone
all_acc_list <- lapply(X = unique(iRB$recorders), FUN = function(x){
  day_pred <- species_accumulation(recorder_name = x,
                                   data = iRB,
                                   n_taxa = length(unique(iRB$preferred_taxon)),
                                   plot = FALSE,
                                   prediction_days = seq(10, 50, 10))$day_pred
})

all_acc <- do.call(rbind, all_acc_list)

# These values clearly rise over time
boxplot(na.omit(all_acc[,!colnames(all_acc) %in% 'recorder']),
        xlab = 'Day',
        ylab = 'Proportion of taxa recorded',
        main = 'Species accumulation over time')
```




```
# How do these correlate?
cor_data <- cor.matrix(na.omit(all_acc[,!colnames(all_acc) %in% 'recorder']))
ggcorplot(cor.mat = cor_data,
          data = na.omit(all_acc[,!colnames(all_acc) %in% 'recorder']),
          var_text_size = 5,
          cor_text_limits = c(5,10))
```



Pearson's product−moment correlation
d10
d20
d30
d40
d50
.93
.84
.75
.67
.98
.93
.87
.99
.95
.99