Thesis Script

Maria Laura Mahecha Escobar

Table of contents

This document contains the codes and results obtained by Maria Laura Mahecha Escobar in her master thesis at Lund university.

The research question we are aiming to answer is: Does the migratory behaviour of birds that breed either polar or subpolar varies with changes in daylenght? We are considering migratory behavior as three different factors: night activity, orientation and fueling.

```
setwd("E:/Users/mlmah/OneDrive/Documentos/MLME/Maestria/Animal Ecology 2022-2024/Tesis/Dat
```

For running this script you need the following packages installed in your workspace:

```
library(knitr)
library(circular)
library(ggplot2)
library(rmarkdown)
library(quarto)
library(lmerTest)
library(cowplot)
library(lme4)
library(dplyr)
library(janitor)
library(tvthemes)
library(phytools)
library(bpnreg)
```

```
library(CircStats)
library(RSQLite)
library(data.table)
library(remotes)
```

0.1 Emlen funnel analysis

In this code there are two different codes for handling emlen funnels data. The first one is manually and the second one is useng the package circular.

0.1.1 Emlen analysis manual

This code aims to conver Susanne's excel sheet into an R code step by step.

To calculate the mean direction, r value and p value for each emlen funnel experiments we need to know if the distribution is unimoda or axial (bimodal). You can notice this in the paper when you count scratches or using mathematical functions. To know if you have an axial distribution your angles need to be doubled and then compare the values from them with an unimodal distribution.

Final vectors that contain the values for the sin and cos for angle 1 and 2 in the sector and that we are going to use from this part of the code are:

@sen24sector_1
@cos24sector_1
@sen24sector_2
@cos24sector_2

In this section we are going to calculate basic trigonometric functions for posterior analysis. From this part we are interested in:

@arctan_angle1
@arctan_angle2
x an y values for both angles

```
EmlenFunnels_RawData<-read.table("E:\\Users\\mlmah\\OneDrive\\Documentos\\MLME\\Maestria\\
EmlenFunnel<- dplyr::select(EmlenFunnels_RawData, c("Ring","S1","S2","S3","S4","S5","S6","

sectors<-as.data.frame(dplyr::select(EmlenFunnels_RawData, c("S1","S2","S3","S4","S5","S6"

##angle 1##
senSectors_angle1<-mapply("*",as.data.frame(sectors),sen24sector_1) #Multiplies each colum
```

```
senData_angle1<-rowSums(senSectors_angle1) #It sums each column and creates a vector with
y_angle1<-senData_angle1/EmlenFunnel$TotalScratches #Calculates the y value for angle 1

cosSectors_angle1<-mapply("*",as.data.frame(sectors),cos24sector_1) #Multiplies each colum
cosData_angle1<-rowSums(cosSectors_angle1) #It sums each column and creates a vector with
x_angle1<-cosData_angle1/EmlenFunnel$TotalScratches #Calculates the x value for angle 1

tan_angle1<-(y_angle1/x_angle1) #Calculates the tangent for angle 1
arctan_angle1<-atan2(y_angle1,x_angle1)*(180/pi) #Calculates the arctan for angle 1
pre_dir_1<-ifelse(arctan_angle1<0, arctan_angle1+360, arctan_angle1) #Calculates the direc

##angle2##
senSectors_angle2<-mapply("*",as.data.frame(sectors),sen24sector_2) #Multiplies each colum
senData_angle2<-rowSums(senSectors_angle2) #It sums each column and creates a vector with
y_angle2<-senData_angle2/EmlenFunnel$TotalScratches #Calculates the y value for angle 2

cosSectors_angle2<-mapply("*",as.data.frame(sectors),cos24sector_2) #Multiplies each colum
cosData_angle2<-rowSums(cosSectors_angle2) #It sums each column and creates a vector with
x_angle2<-cosData_angle2/EmlenFunnel$TotalScratches #Calculates the x value for angle 2

tan_angle2<-(y_angle2/x_angle2) #Calculates the tangent for angle 2
arctan_angle2<-atan2(y_angle2,x_angle2)*(180/pi) #Calculates the arctan for angle 2
pre_dir_2<-ifelse(arctan_angle2<0, arctan_angle2+360, arctan_angle2) #Calculates the direc
pre_dir_2a<-pre_dir_2/2 #Calculates the first direction of a bimodal distribution
pre_dir_2b<-pre_dir_2a+180 #Calculates the second direction of a bimodal distribution
```

In this section we are going to handling with axial data. Axial data returns two different values for direction and we need to choose only one for it. In this case, the way of doing so is selecting the angle that has most of the scratches. You can do this manually and modifying your database or you can use the code below. This code searches for the values in the original database and compares which one is higher. Then writes the obtained direction from it.

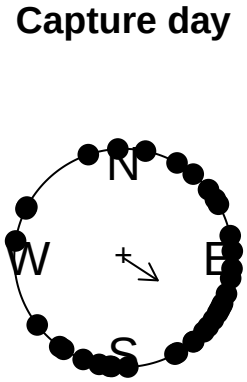
```
#To know the nearest value and find the direction in axial data

###2a###
# Create a vector to store the closest positions for direction 2a
closest_positions_2a <- numeric(length(pre_dir_2a))
for (i in 1:length(pre_dir_2a)) {# Calculate the distance to each value in sectorAngel_1_d
  distances_2a <- abs(pre_dir_2a[i] - sectorAngle_1_deg)# Find the position of the closest
  closest_positions_2a[i] <- which.min(distances_2a)
}
```

```
#print(closest_positions_2a) #prints the position of the closest value for the angle given
closest_positions_2a <-as.vector(paste("S", closest_positions_2a, sep="")) #Put an "S" bef

###2b###
# Create a vector to store the closest positions for direction 2b
closest_positions_2b <- numeric(length(pre_dir_2b))
for (i in 1:length(pre_dir_2b)) {# Calculate the distance to each value in sectorAngel_1_d
  distances_2b <- abs(pre_dir_2b[i] - sectorAngle_1_deg)# Find the position of the closest
  closest_positions_2b[i] <- which.min(distances_2b)
}
#print(closest_positions_2b) #prints the position of the closest value for the angle given
closest_positions_2b<-as.vector(paste("S", closest_positions_2b, sep="")) #Put an "S" befo


#To extract number of scratches and compare
  #For direction_2a
scratches_2a <- vector() #create an empty vector of unknown dimensions
for (i in seq_along(closest_positions_2a)) { #For a sequence of data in the vector
  scratches_2a <- c(scratches_2a, EmlenFunnel[i, closest_positions_2a[i]]) #match the name
}
#scratches_2a

  #For direction_2b
scratches_2b <- vector() #create an empty vector of unknown dimensions
for (i in seq_along(closest_positions_2b)) { #For a sequence of data in the vector
  scratches_2b <- c(scratches_2b, EmlenFunnel[i, closest_positions_2b[i]]) #match the name
}
#scratches_2b

bimodal_dir<-ifelse(scratches_2a<scratches_2b,pre_dir_2b, pre_dir_2a) #If the number of sc
```

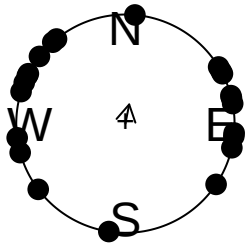
In this section we will calculate the mean lenght of the vectors (r) and the overall direction of the individual (@directionEmlen)

```
r_value_1<-sqrt(((x_angle1)^2)+((y_angle1)^2)) #Calculates the r value for angle 1
r_value_2<-sqrt(((x_angle2)^2)+((y_angle2)^2)) #calculates the r value for both angles in
rvalue<-ifelse(r_value_1>r_value_2, r_value_1, r_value_2)
#rvalue

directionEmlen<- ifelse(r_value_1>r_value_2,pre_dir_1, bimodal_dir) #compares r values and
directionEmlen<- as.circular(directionEmlen, units='degrees', template='geographics', modu
```

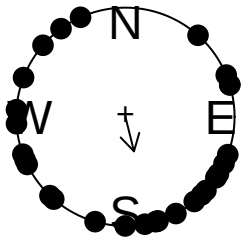
```
pvalue_1_uniformity<-(exp(-(EmlenFunnel$TotalScratches)*((r_value_1)^2)))
pvalue_2_uniformity<-(exp(-(EmlenFunnel$TotalScratches)*((r_value_2)^2)))
pvalue_uniformity<-ifelse(r_value_1>r_value_2, pvalue_1_uniformity,pvalue_2_uniformity)


EmlenFunnels_withDir<-cbind(EmlenFunnels_RawData,directionEmlen,rvalue,pvalue_uniformity)
#EmlenFunnels_withDir

EmlenDir_Capture<-subset(EmlenFunnels_withDir, Treatment=="capture")
EmlenDir_Control<-subset(EmlenFunnels_withDir, Treatment=="control")
EmlenDir_24h<-subset(EmlenFunnels_withDir, Treatment=="treatment")

rtest_capture<- rayleigh.test(EmlenDir_Capture$directionEmlen)
rtest_control<- rayleigh.test(EmlenDir_Control$directionEmlen)
rtest_treatment<- rayleigh.test(EmlenDir_24h$directionEmlen)

#circ.plot(directionEmlen, cex=1.5, bin=720, stack=FALSE, shrink=1.3)

plot(EmlenDir_Capture$directionEmlen, cex=1.5, bin=720, stack=FALSE, sep=0.035, shrink=1.3
arrows.circular(mean.circular(EmlenDir_Capture$directionEmlen), y=rtest_capture$statistic,
```



```
plot(EmlenDir_Control$directionEmlen, cex=1.5, bin=720, stack=FALSE, sep=0.035, shrink=1.3
arrows.circular(mean.circular(EmlenDir_Control$directionEmlen), y=rtest_control$statistic,
```




```
plot(EmlenDir_24h$directionEmlen, cex=1.5, bin=720, stack=FALSE, sep=0.035, shrink=1.3, ma
arrows.circular(mean.circular(EmlenDir_24h$directionEmlen), y=rtest_treatment$statistic, l
```




```
w.capture<-watson.test(as.circular(EmlenDir_Capture$directionEmlen))
print(paste("The Watson's test p-value for capture day is", w.capture$statistic))
```

```
[1] "The Watson's test p-value for capture day is 0.275968099535453"
```

```
w.control<-watson.test(EmlenDir_Control$directionEmlen)
print(paste("The Watson's test p-value for control days is", w.control$statistic))
```

```
[1] "The Watson's test p-value for control days is 0.124882816924962"
```

```
w.treatment<-watson.test(EmlenDir_24h$directionEmlen)
print(paste("The Watson's test p-value for treatment days is", w.treatment$statistic))
```

```
[1] "The Watson's test p-value for treatment days is 0.19762190591541"
```

In this section we will perform the models we want to test. At the moment bpnme models are not running. I have errors on them. Needs to be solved in the inmediate future.

```
EmlenFunnels_withDir$NumericID<-as.factor(EmlenFunnels_withDir$NumericID)
EmlenFunnels_withDir$Treatment<-as.factor(EmlenFunnels_withDir$Treatment)
EmlenFunnels_withDir$DayLength<-as.numeric(EmlenFunnels_withDir$DayLength)
EmlenFunnels_withDir$BreedingDistribution<-as.factor(EmlenFunnels_withDir$BreedingDistribu
EmlenFunnels_withDir$Rad_Dir<-as.circular(deg2rad(EmlenFunnels_withDir$directionEmlen), ty
EmlenFunnels_withDir$DummyTreatment<-ifelse(EmlenFunnels_withDir$Treatment== "control", 0,
EmlenFunnels_withDir$DummyBreed<-ifelse(EmlenFunnels_withDir$BreedingDistribution== "subpo
EmlenFunnels_withDir<-as.data.frame(EmlenFunnels_withDir)

#to interpret the output https://cran.r-project.org/web/packages/bpnreg/vignettes/FAQ.html

#pred_orientation_1<-bpnme(Rad_Dir ~ DummyTreatment * DummyBreed + (1|NumericID), EmlenFun

# pred_orientation_1<-bpnme(Rad_Dir ~ Treatment * BreedingDistribution + (1|NumericID), Em
# pred_orientation_2<-bpnme(Rad_Dir ~ DayLength * BreedingDistribution + (1|NumericID), Em
```

0.1.2 Using 'circular' package

In this section (I need to tidy up the text and the comment son the code), I will perform the same analysis and some other test on the emlen funnel data using 'circular' package.

The code needs some tidy as well

```
setwd("E:/Users/mlmah/OneDrive/Documentos/MLME/Maestria/Animal Ecology 2022-2024/Tesis/Dat
#read .csv data file and assign it a name
Emlen_IndData_2<-read.table("E:/Users/mlmah/OneDrive/Documentos/MLME/Maestria/Animal Ecolo

#check data
names(Emlen_IndData_2) #returns names of the columns
```

```
 [1] "Date"                "Species"             "BreedingDistribution"
 [4] "NumericID"           "Treatment"           "Group"
 [7] "DayLength"           "StartTime"           "WindDirection"
[10] "WindSpeed"           "X.m.s."              "CloudCover"
[13] "Ring"                "Line"                "S1"
[16] "S2"                  "S3"                  "S4"
[19] "S5"                  "S6"                  "S7"
[22] "S8"                  "S9"                  "S10"
[25] "S11"                 "S12"                 "S13"
[28] "S14"                 "S15"                 "S16"
[31] "S17"                 "S18"                 "S19"
```

```
[34] "S20"                "S21"                "S22"
[37] "S23"                "S24"                "TotalScratches"
```

```
head(Emlen_IndData_2) #returns first 6 rows
```

```
       Date Species BreedingDistribution NumericID Treatment Group DayLength
1 20/04/2023   Robin                Polar         1   capture local     14:34
2 20/04/2023   Robin                Polar         2   capture local     14:34
3 20/04/2023   Robin                Polar         3   capture local     14:34
4 20/04/2023   Robin                Polar         4   capture local     14:34
5 20/04/2023   Robin                Polar         5   capture local     14:34
6 26/04/2023   Robin                Polar         5 treatment total     24:00
  StartTime WindDirection WindSpeed X.m.s. CloudCover    Ring Line S1 S2 S3 S4
1     19:11            NO        46   12,2          1 DK51988    C 16 25 34 70
2     19:11            NO        46   12,2          1 DK52084    C 54 83 46 30
3     19:11            NO        46   12,2          1 DK51990    D 23 30 23 33
4     19:11            NO        46   12,2          1 DK51980    C 70 71 71 86
5     19:11            NO        46   12,2          1 DK52068    C 44 39 63 82
6     19:22             V       267    6,1          2 DK52068    C 76 95 86 87
   S5  S6  S7  S8  S9 S10 S11 S12 S13 S14 S15 S16 S17 S18 S19 S20 S21 S22 S23
1  84  82  95 106 111 103 107 115  89  73  41  26  15  20  20  25  29  28  21
2  75  92  82  87  99  80 110  99 113 118  85  33  42  40  21  17  30  29  55
3  26  27  26  42  31  46  40  50  53  42  52  36  44  28  27  25  34  47  77
4  91 102 104 118 107 123 120 125  80  97  94  98  95  96  87  93  81  56  82
5 103 120 119 105 162 171 150 153 137 111  81  70  64  73  63  68  77  75  46
6  93  89  84  96 102 105  98  89  83  87  79  94  99  78  66  78  82  61  60
  S24 TotalScratches
1  15           1350
2  44           1564
3  70            932
4  82           2229
5  51           2227
6  58           2025
```

```
dim(Emlen_IndData_2) #returns dimensions - rows and columns
```

```
[1] 87 39
```

```
str(Emlen_IndData_2) #compilation of the above
```

```
'data.frame':   87 obs. of  39 variables:
 $ Date                : chr  "20/04/2023" "20/04/2023" "20/04/2023" "20/04/2023" ...
 $ Species             : chr  "Robin" "Robin" "Robin" "Robin" ...
 $ BreedingDistribution: chr  "Polar" "Polar" "Polar" "Polar" ...
 $ NumericID           : int  1 2 3 4 5 5 1 3 4 2 ...
 $ Treatment           : chr  "capture" "capture" "capture" "capture" ...
 $ Group               : chr  "local" "local" "local" "local" ...
 $ DayLength           : chr  "14:34" "14:34" "14:34" "14:34" ...
 $ StartTime           : chr  "19:11" "19:11" "19:11" "19:11" ...
 $ WindDirection       : chr  "NO" "NO" "NO" "NO" ...
 $ WindSpeed           : int  46 46 46 46 46 267 267 267 267 267 ...
 $ X.m.s.              : chr  "12,2" "12,2" "12,2" "12,2" ...
 $ CloudCover          : int  1 1 1 1 1 2 2 2 2 2 ...
 $ Ring                : chr  "DK51988" "DK52084" "DK51990" "DK51980" ...
 $ Line                : chr  "C" "C" "D" "C" ...
 $ S1                  : int  16 54 23 70 44 76 43 18 38 26 ...
 $ S2                  : int  25 83 30 71 39 95 36 45 55 35 ...
 $ S3                  : int  34 46 23 71 63 86 43 52 45 30 ...
 $ S4                  : int  70 30 33 86 82 87 50 55 71 50 ...
 $ S5                  : int  84 75 26 91 103 93 88 48 90 49 ...
 $ S6                  : int  82 92 27 102 120 89 96 55 99 58 ...
 $ S7                  : int  95 82 26 104 119 84 94 33 123 69 ...
 $ S8                  : int  106 87 42 118 105 96 121 43 127 64 ...
 $ S9                  : int  111 99 31 107 162 102 129 43 118 73 ...
 $ S10                 : int  103 80 46 123 171 105 115 47 115 82 ...
 $ S11                 : int  107 110 40 120 150 98 98 24 103 81 ...
 $ S12                 : int  115 99 50 125 153 89 93 26 91 86 ...
 $ S13                 : int  89 113 53 80 137 83 89 30 82 72 ...
 $ S14                 : int  73 118 42 97 111 87 70 28 83 72 ...
 $ S15                 : int  41 85 52 94 81 79 65 10 78 43 ...
 $ S16                 : int  26 33 36 98 70 94 11 17 53 53 ...
 $ S17                 : int  15 42 44 95 64 99 8 17 48 51 ...
 $ S18                 : int  20 40 28 96 73 78 10 25 42 53 ...
 $ S19                 : int  20 21 27 87 63 66 37 28 52 50 ...
 $ S20                 : int  25 17 25 93 68 78 35 25 34 42 ...
 $ S21                 : int  29 30 34 81 77 82 41 32 33 38 ...
 $ S22                 : int  28 29 47 56 75 61 31 17 30 34 ...
 $ S23                 : int  21 55 77 82 46 60 43 18 41 30 ...
 $ S24                 : int  15 44 70 82 51 58 47 22 38 35 ...
 $ TotalScratches      : int  1350 1564 932 2229 2227 2025 1493 758 1689 1276 ...
```

```r
#load libraries
library(circular)
library(dplyr)
library(data.table)
library(tidyr)
library(dplyr)
library(readr)
library(remotes)
library(bpnreg)

# ANALYSIS FOR INDIVIDUAL BIRDS - UNIMODAL -------------------------------------
#import my data, update column name and assigning a new row as heading
setnames(Emlen_IndData_2, old=c("S1","S2","S3","S4","S5","S6","S7","S8","S9","S10","S11","
         new = c("0","15","30","45","60","75","90","105","120","135","150","165","180","19

Emlen_IndData_2<-tibble::rowid_to_column(Emlen_IndData_2, "Identificator")
deg2rad <- function(deg) {(deg * pi) / (180)} #converts degrees to radians.
rad2deg <- function(rad) {(rad * 180) / (pi)} #converts radians to degrees.

Emlen_IndData_2$Date<-lubridate::dmy(Emlen_IndData_2$Date)
str(Emlen_IndData_2)
```

```
'data.frame':   87 obs. of  40 variables:
 $ Identificator       : int  1 2 3 4 5 6 7 8 9 10 ...
 $ Date                : Date, format: "2023-04-20" "2023-04-20" ...
 $ Species             : chr  "Robin" "Robin" "Robin" "Robin" ...
 $ BreedingDistribution: chr  "Polar" "Polar" "Polar" "Polar" ...
 $ NumericID           : int  1 2 3 4 5 5 1 3 4 2 ...
 $ Treatment           : chr  "capture" "capture" "capture" "capture" ...
 $ Group               : chr  "local" "local" "local" "local" ...
 $ DayLength           : chr  "14:34" "14:34" "14:34" "14:34" ...
 $ StartTime           : chr  "19:11" "19:11" "19:11" "19:11" ...
 $ WindDirection       : chr  "NO" "NO" "NO" "NO" ...
 $ WindSpeed           : int  46 46 46 46 46 267 267 267 267 267 ...
 $ X.m.s.              : chr  "12,2" "12,2" "12,2" "12,2" ...
 $ CloudCover          : int  1 1 1 1 1 2 2 2 2 2 ...
 $ Ring                : chr  "DK51988" "DK52084" "DK51990" "DK51980" ...
 $ Line                : chr  "C" "C" "D" "C" ...
 $ 0                   : int  16 54 23 70 44 76 43 18 38 26 ...
 $ 15                  : int  25 83 30 71 39 95 36 45 55 35 ...
 $ 30                  : int  34 46 23 71 63 86 43 52 45 30 ...
 $ 45                  : int  70 30 33 86 82 87 50 55 71 50 ...
```

```
$ 60                : int  84 75 26 91 103 93 88 48 90 49 ...
$ 75                : int  82 92 27 102 120 89 96 55 99 58 ...
$ 90                : int  95 82 26 104 119 84 94 33 123 69 ...
$ 105               : int  106 87 42 118 105 96 121 43 127 64 ...
$ 120               : int  111 99 31 107 162 102 129 43 118 73 ...
$ 135               : int  103 80 46 123 171 105 115 47 115 82 ...
$ 150               : int  107 110 40 120 150 98 98 24 103 81 ...
$ 165               : int  115 99 50 125 153 89 93 26 91 86 ...
$ 180               : int  89 113 53 80 137 83 89 30 82 72 ...
$ 195               : int  73 118 42 97 111 87 70 28 83 72 ...
$ 210               : int  41 85 52 94 81 79 65 10 78 43 ...
$ 225               : int  26 33 36 98 70 94 11 17 53 53 ...
$ 240               : int  15 42 44 95 64 99 8 17 48 51 ...
$ 255               : int  20 40 28 96 73 78 10 25 42 53 ...
$ 270               : int  20 21 27 87 63 66 37 28 52 50 ...
$ 285               : int  25 17 25 93 68 78 35 25 34 42 ...
$ 300               : int  29 30 34 81 77 82 41 32 33 38 ...
$ 315               : int  28 29 47 56 75 61 31 17 30 34 ...
$ 330               : int  21 55 77 82 46 60 43 18 41 30 ...
$ 345               : int  15 44 70 82 51 58 47 22 38 35 ...
$ TotalScratches    : int  1350 1564 932 2229 2227 2025 1493 758 1689 1276 ...
```

```r
Emlen_IndData_2 <- Emlen_IndData_2 %>%
  mutate(Week = case_when(
    between(Date, as.Date("2023-04-20"), as.Date("2023-04-26")) ~ 1,
    between(Date, as.Date("2023-04-28"), as.Date("2023-05-04")) ~ 2,
    between(Date, as.Date("2023-05-08"), as.Date("2023-05-14")) ~ 3,
    between(Date, as.Date("2023-05-16"), as.Date("2023-05-22")) ~ 4,
    between(Date, as.Date("2023-05-25"), as.Date("2023-05-31")) ~ 5,
    TRUE ~ 6 # Defining values for each week with date intervals
  ))

View(Emlen_IndData_2)


datalong_EmlenInd_2<-pivot_longer( #makes it a frequency to rep after
  Emlen_IndData_2,
  cols=c("0","15","30","45","60","75","90","105","120","135","150","165","180","195","210"
  names_to="sector",
  values_to="count")
```

```r
uniqueValues_ID<-unique(datalong_EmlenInd_2$Identificator)

results_df <- data.frame(Identificator = uniqueValues_ID,   #create an empty dataframe to
                         test_statistic = numeric(length(uniqueValues_ID)),
                         p_value = numeric(length(uniqueValues_ID)),
                         Mean = numeric(length(uniqueValues_ID)),
                         rho= numeric(length(uniqueValues_ID)))

# Loop through unique identifiers
for (i in uniqueValues_ID) {
  subset_data <- subset(datalong_EmlenInd_2, Identificator == i) #subsets data for each id
  IndData <- as.data.frame(as.numeric(unlist(subset_data)))
  IndData <- na.omit(IndData)
  longdata <- as.numeric(rep(subset_data$sector, times = subset_data$count)) #repeats the
  longdata <- deg2rad(longdata)  # Convert to radians
  rayleigh_test <- rayleigh.test(longdata) #performs rayleigh test for each entry in the d
  r1 <- rho.circular(longdata, na.rm = FALSE)
  test_statistic <- rayleigh_test$statistic
  p_value <- rayleigh_test$p.value
  MeanDir<- mean.circular(longdata)

  # Convert the mean direction from radians to degrees and correct negative values
  MeanDir_deg <- rad2deg(MeanDir)  # Convert to degrees
  if (MeanDir_deg < 0) {
    MeanDir_deg <- 360 + MeanDir_deg  # Wrap around negative values
  }

  # Assign results to the correct rows
  row_index <- which(results_df$Identificator == i)
  results_df$rho[row_index] <- r1
  results_df$test_statistic[row_index] <- test_statistic
  results_df$p_value[row_index] <- p_value
  results_df$Mean[row_index]<- MeanDir_deg
}

View(results_df)

mergedEmlen<-merge(results_df, Emlen_IndData_2, by = "Identificator") #merge both datafram
View(mergedEmlen)

mergedEmlen$Mean<-as.circular(deg2rad(mergedEmlen$Mean))
```

```r
mergedEmlen<-filter(mergedEmlen, p_value<0.05)
str(mergedEmlen)
```

```
'data.frame':   72 obs. of  45 variables:
 $ Identificator       : int  1 2 3 4 5 6 7 8 9 10 ...
 $ test_statistic      : num  0.432 0.273 0.06 0.106 0.261 ...
 $ p_value             : num  3.74e-110 3.23e-51 3.50e-02 1.36e-11 1.51e-66 ...
 $ Mean                : 'circular' num  2.17 2.31 4.17 2.62 2.45 ...
  ..- attr(*, "circularp")=List of 6
  .. ..$ type    : chr "angles"
  .. ..$ units   : chr "radians"
  .. ..$ template: chr "none"
  .. ..$ modulo  : chr "asis"
  .. ..$ zero    : num 0
  .. ..$ rotation: chr "counter"
 $ rho                 : num  0.432 0.273 0.06 0.106 0.261 ...
 $ Date                : Date, format: "2023-04-20" "2023-04-20" ...
 $ Species             : chr  "Robin" "Robin" "Robin" "Robin" ...
 $ BreedingDistribution: chr  "Polar" "Polar" "Polar" "Polar" ...
 $ NumericID           : int  1 2 3 4 5 5 1 3 4 2 ...
 $ Treatment           : chr  "capture" "capture" "capture" "capture" ...
 $ Group               : chr  "local" "local" "local" "local" ...
 $ DayLength           : chr  "14:34" "14:34" "14:34" "14:34" ...
 $ StartTime           : chr  "19:11" "19:11" "19:11" "19:11" ...
 $ WindDirection       : chr  "NO" "NO" "NO" "NO" ...
 $ WindSpeed           : int  46 46 46 46 46 267 267 267 267 267 ...
 $ X.m.s.              : chr  "12,2" "12,2" "12,2" "12,2" ...
 $ CloudCover          : int  1 1 1 1 1 2 2 2 2 2 ...
 $ Ring                : chr  "DK51988" "DK52084" "DK51990" "DK51980" ...
 $ Line                : chr  "C" "C" "D" "C" ...
 $ 0                   : int  16 54 23 70 44 76 43 18 38 26 ...
 $ 15                  : int  25 83 30 71 39 95 36 45 55 35 ...
 $ 30                  : int  34 46 23 71 63 86 43 52 45 30 ...
 $ 45                  : int  70 30 33 86 82 87 50 55 71 50 ...
 $ 60                  : int  84 75 26 91 103 93 88 48 90 49 ...
 $ 75                  : int  82 92 27 102 120 89 96 55 99 58 ...
 $ 90                  : int  95 82 26 104 119 84 94 33 123 69 ...
 $ 105                 : int  106 87 42 118 105 96 121 43 127 64 ...
 $ 120                 : int  111 99 31 107 162 102 129 43 118 73 ...
 $ 135                 : int  103 80 46 123 171 105 115 47 115 82 ...
 $ 150                 : int  107 110 40 120 150 98 98 24 103 81 ...
 $ 165                 : int  115 99 50 125 153 89 93 26 91 86 ...
```

```
 $ 180                : int  89 113 53 80 137 83 89 30 82 72 ...
 $ 195                : int  73 118 42 97 111 87 70 28 83 72 ...
 $ 210                : int  41 85 52 94 81 79 65 10 78 43 ...
 $ 225                : int  26 33 36 98 70 94 11 17 53 53 ...
 $ 240                : int  15 42 44 95 64 99 8 17 48 51 ...
 $ 255                : int  20 40 28 96 73 78 10 25 42 53 ...
 $ 270                : int  20 21 27 87 63 66 37 28 52 50 ...
 $ 285                : int  25 17 25 93 68 78 35 25 34 42 ...
 $ 300                : int  29 30 34 81 77 82 41 32 33 38 ...
 $ 315                : int  28 29 47 56 75 61 31 17 30 34 ...
 $ 330                : int  21 55 77 82 46 60 43 18 41 30 ...
 $ 345                : int  15 44 70 82 51 58 47 22 38 35 ...
 $ TotalScratches     : int  1350 1564 932 2229 2227 2025 1493 758 1689 1276 ...
 $ Week               : num  1 1 1 1 1 1 1 1 1 1 ...
```

```
View(mergedEmlen)

#To check unique values and how many data points a ring has to perform a paired test
uniqueValues_Ring<-unique(mergedEmlen$Ring)
length(uniqueValues_Ring)
```

```
[1] 37
```

```
occurrences <- mergedEmlen %>% count(Ring) #to see how many times a ring occours in the da
print(occurrences)
```

```
      Ring n
1  DK51980 2
2  DK51988 2
3  DK51990 2
4  DK51996 2
5  DK52030 1
6  DK52068 2
7  DK52084 2
8  DK52429 1
9  DK52626 1
10 DK52630 2
11 DK52631 2
12 DK52636 2
13 DK52638 2
```

```
14 DK52639 2
15 DK52640 2
16 DK52642 1
17 DK52700 1
18 DK52702 2
19 DK52704 1
20 DK52705 3
21 DK52712 1
22 DK52713 3
23 DK52976 1
24 DK53160 1
25 DK53182 1
26 DK53193 3
27 DK53195 2
28 DK53341 3
29 DK53342 2
30 DK53343 2
31 DK53345 3
32 DK53346 3
33 DK53347 3
34 DK53348 1
35 DK53349 2
36 DK53350 3
37 DK53359 3
```

```r
#tO TESTS
EmlenData_ForTest<- dplyr::select(mergedEmlen, c("Identificator", "test_statistic","p_valu

Emlen_local<-filter(EmlenData_ForTest, Group=="local") #just local
Emlen_local<-Emlen_local %>%
  group_by(Ring) %>%
  summarise(circular_mean = mean.circular(Mean))

Emlen_total<-filter(EmlenData_ForTest, Group=="total") #just 24h treatment

nrow(Emlen_local)
```

```
[1] 34
```

```r
nrow(Emlen_total)
```

```
[1] 25
```

```
#watson.williams.test FOR ALL THE DATA

Watson_data <- list(
  Emlen_total$Mean <- as.circular(Emlen_total$Mean, units="degrees", template="geographics
  Emlen_local$circular_mean <- as.circular(Emlen_local$circular_mean, units="degrees", tem
)
watson.williams.test(Watson_data) #Watson test not paired
```

```
    Watson-Williams test for homogeneity of means

data:  1 and 2
F = 1.7475, df1 = 1, df2 = 57, p-value = 0.1915
sample estimates:
Circular Data:
Type = angles
Units = radians
Template = none
Modulo = asis
Zero = 0
Rotation = counter
mean of 1 mean of 2
 3.063224  2.482513
```

```
#To perform a paired test (Hotelling test)

common_ids <- intersect(Emlen_local$Ring, Emlen_total$Ring)

# Filter dataframes to include only common IDs
Emlen_local_ID <- Emlen_local[Emlen_local$Ring %in% common_ids, ]
Emlen_total_ID <- Emlen_total[Emlen_total$Ring %in% common_ids, ]

nrow(Emlen_local_ID)
```

```
[1] 22
```

```
nrow(Emlen_total_ID)
```

```
[1] 22
```

```r
Emlen_hotelling<-merge(Emlen_local_ID, Emlen_total_ID, by="Ring")
Emlen_hotelling<-dplyr::select(Emlen_hotelling, c("Ring","circular_mean","Mean", "Week"))
Emlen_hotelling$circular_mean<-as.vector(rad2deg(Emlen_hotelling$circular_mean))
Emlen_hotelling$circular_mean <- (Emlen_hotelling$circular_mean + 360) %% 360
Emlen_hotelling$Mean<-as.vector(rad2deg(Emlen_hotelling$Mean))

source("https://raw.githubusercontent.com/olitroski/circular/master/paired.hotelling.r") #
paired.hotelling(Emlen_hotelling$circular_mean, Emlen_hotelling$Mean)
```

```
  Test de Hottelling para muestras circulares pareadas
      k pares       =  22
      Estadistico F =  0.54868
      Critico 5%    =  3.4928
      p-valor       =  0.58616
```

```r
#And for a circular ANOVA
Emlen_Anova<-Emlen_hotelling
colnames(Emlen_Anova)<-c("Ring","local","total","Week")
Emlen_Anova<-pivot_longer(data= Emlen_Anova,
                          cols = "local":"total",
                          names_to = "Group",
                          values_to = "Mean")
Emlen_Anova$Mean<-as.circular(deg2rad(Emlen_Anova$Mean))
```

```r
emlenanova <- bpnr(pred.I = Mean ~ Group + Week, data = Emlen_Anova,
                   its = 1000, burn = 100, n.lag = 3, seed = 101)
```

```r
summary(emlenanova)
```

```
               Length Class  Mode
beta1           3000  -none- numeric
beta2           3000  -none- numeric
Likelihood     44000  -none- numeric
its                1  -none- numeric
n.lag              1  -none- numeric
burn-in            1  -none- numeric
p1                 1  -none- numeric
p2                 1  -none- numeric
theta             44  -none- numeric
```

```
a.x              1000  -none- numeric
a.c              1000  -none- numeric
b.c              1000  -none- numeric
SAM              1000  -none- numeric
AS               1000  -none- numeric
SSDO             1000  -none- numeric
circ.diff        1000  -none- numeric
Call                7  -none- call
lin.coef.I         15  -none- numeric
lin.coef.II        15  -none- numeric
circ.coef          30  -none- numeric
circ.coef.cat       5  -none- numeric
circ.coef.means    10  -none- numeric
model.fit          13  -none- list
mm                  8  -none- list
```

```
emlenanova
```

```
Projected Normal Regression

Model

Call:
bpnr(pred.I = Mean ~ Group + Week, data = Emlen_Anova, its = 1000,
    burn = 100, n.lag = 3, seed = 101)

MCMC:
iterations = 1000
burn-in = 100
lag = 3

Model Fit:
        Statistic Parameters
lppd      -72.964     6.0000
DIC       157.851     5.9647
DIC.alt   157.580     5.8292
WAIC1     157.845     5.9582
WAIC2     158.548     6.3100


Linear Coefficients
```

```
Component I:
                mean     mode      sd   LB HPD  UB HPD
(Intercept) -0.71242 -0.48253 0.55443 -1.76561 0.36218
Grouptotal  -0.23298 -0.13957 0.33761 -0.91715 0.39341
Week         0.10793  0.11613 0.21696 -0.30855 0.49709

Component II:
                mean     mode      sd    LB HPD    UB HPD
(Intercept)  1.10825  1.08267 0.56824  0.099835  2.305498
Grouptotal  -0.14152 -0.17201 0.34855 -0.754507  0.573949
Week        -0.44013 -0.38520 0.22669 -0.921133 -0.042738


Circular Coefficients

Continuous variables:
mean ax mode ax   sd ax   LB ax   UB ax
2.61770 2.66831 1.34098 0.26688 4.67955

 mean ac  mode ac    sd ac    LB ac    UB ac
-2.97964 -2.85531  0.73359  0.64358  4.32347

 mean bc  mode bc    sd bc    LB bc    UB bc
 4.31192  0.90676 48.93835 -6.32760  9.75045

 mean AS  mode AS    sd AS    LB AS    UB AS
 0.92403  0.68983  3.86373 -0.94339  3.90253

 mean SAM  mode SAM    sd SAM    LB SAM    UB SAM
1.0261573 0.6671395 3.6441613 0.0060955 5.2933585

mean SSDO mode SSDO   sd SSDO   LB SSSO   UB SSDO
 -0.83139  -1.07788   0.63357  -1.96911   0.76106

Categorical variables:

Means:
              mean   mode      sd     LB     UB
(Intercept) 2.1327 2.1653 0.48719 1.2274 3.1837
Grouptotal  2.3466 2.4769 0.44449 1.4646 3.2576

Differences:
               mean     mode      sd      LB     UB
```

```
Grouptotal -0.20163 -0.14363 0.37222 -1.0115 0.4306
```

```
coef_circ(emlenanova, type = "continuous", units = "degrees")
```

```
               mean      mode         sd     LB HPD    UB HPD
Week ax     2.61770    2.6683    1.34098    0.26688   4.67955
Week ac  -170.72066 -163.5973   42.03164   36.87463 247.71682
Week bc   247.05500   51.9536 2803.96079 -362.54496 558.65956
Week AS    52.94328   39.5245  221.37551  -54.05252 223.59867
Week SAM   58.79448   38.2243  208.79506    0.34924 303.28710
Week SSDO  -0.83139   -1.0779    0.63357   -1.96911   0.76106
```

```
coef_circ(emlenanova, type = "categorical", units = "degrees")
```

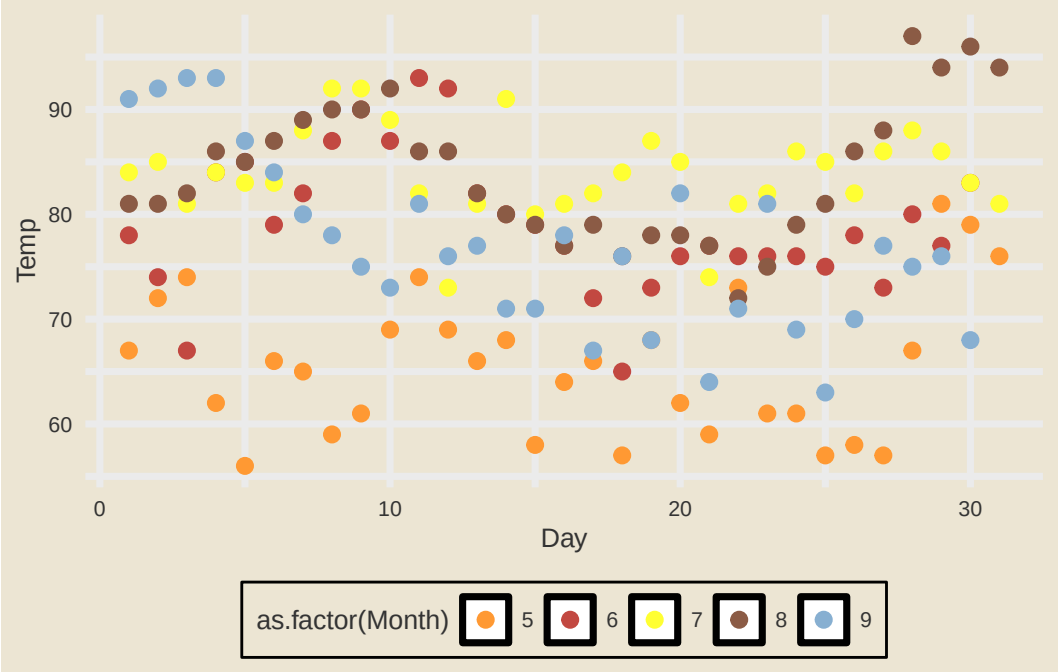
```
$Means
              mean   mode     sd     LB     UB
(Intercept) 122.20 124.06 27.914 70.325 182.41
Grouptotal  134.45 141.92 25.467 83.918 186.65

$Differences
              mean    mode     sd      LB     UB
Grouptotal -11.553 -8.2294 21.327 -57.953 24.672
```

```
fit(emlenanova)
```

```
        Statistic Parameters
lppd      -72.964     6.0000
DIC       157.851     5.9647
DIC.alt   157.580     5.8292
WAIC1     157.845     5.9582
WAIC2     158.548     6.3100
```

0.2 Color palette

This is the code for the color palette, the last step when I finish everything and I want to make it pretty. Im using a default dataset



0.3 Cuartito de San Alejo

This is the part of the code where I put everything I tried and does not worked but I don't want to delete