An automated procedure for phenology data analyses

Lea Rosa Maria Lipphardt

2024-11-19

Introduction

This automated procedure streamlines the analysis of phenology data, enabling more efficient estimation of key metrics such as hatching dates, breeding success, and chick survival. The data is sourced from the ACCESS database, specifically NEMO, and processed through a series of R scripts.

The workflow is structured as follows:

1. Import Data from the NEMO Database
2. Combine Tables into a Single Data Frame
3. Filter and Document Exceptions
4. Estimate Hatching Dates
5. Estimate Chick Survival
6. Summarize Findings

This document is written in R Markdown, a versatile format for creating HTML, PDF, and Word documents. For more details on R Markdown, visit http://rmarkdown.rstudio.com.

To execute the R code, click the green arrow in the Script Editor (Source pane). Press the Knit button to generate a final document that integrates both narrative content and the output from your R code chunks.

1. Import Data

Install the required packages and

```
install.packages("odbc") # contains drivers to connect to a database
install.packages("DBI") # contains functions for interacting with the database

library(odbc)
library(DBI)
library(tidyverse)
library(dplyr)
```

1.1 Connect to the Access database using ODBC

```
# Directs to the place the database is stored
dbname <- "N:/Midlertidig/Lea/NEMO/Spitsbergen_2024.mdb"

# Connect to the Access database using ODBC
con <- dbConnect(odbc::odbc(),
                 .connection_string = paste0("Driver={Microsoft Access Driver (*.mdb, *.accdb)};DBQ=", d
```

2. Query

2.1 Create dataframe using Structured query language (SQL)

```
query <- "
SELECT s.spcENG, n.Locality, n.Area, n.NestNumber, v.VisitDate, nc.NestContentText, v.NestClutch
FROM (((Visits AS v
INNER JOIN NestContent AS nc ON v.NestStatus = nc.NestContentID)
INNER JOIN Nest AS n ON n.NestUniqueID = v.NestUniqueID)
INNER JOIN Species AS s ON n.Species = s.EUnr)
"
results <- dbGetQuery(con, query)

head(results)
```

```
##                   spcENG    Locality Area NestNumber           VisitDate
## 1 Black-legged kittiwake Grumantbyen    -     A#0002 2011-06-18 12:00:00
## 2 Black-legged kittiwake Grumantbyen    -     A#0002 2011-06-21 12:00:00
## 3 Black-legged kittiwake Grumantbyen    -     A#0002 2011-06-23 12:00:00
## 4 Black-legged kittiwake Grumantbyen    -     A#0002 2011-06-25 12:00:00
## 5 Black-legged kittiwake Grumantbyen    -     K#0003 2011-06-13 12:00:00
## 6 Black-legged kittiwake Grumantbyen    -     K#0003 2011-06-18 12:00:00
##              NestContentText NestClutch
## 1 Nest in use, known content         1E
## 2 Nest in use, known content         2E
## 3 Nest in use, known content         2E
## 4          Nest in use empty
## 5          Nest in use empty
## 6 Nest in use, known content         2E
```

```
str(results)
```

```
## 'data.frame':    48611 obs. of  7 variables:
##  $ spcENG         : chr  "Black-legged kittiwake" "Black-legged kittiwake" "Black-legged kittiwake" "
##  $ Locality       : chr  "Grumantbyen" "Grumantbyen" "Grumantbyen" "Grumantbyen" ...
##  $ Area           : chr  "-" "-" "-" "-" ...
##  $ NestNumber     : chr  "A#0002" "A#0002" "A#0002" "A#0002" ...
##  $ VisitDate      : POSIXct, format: "2011-06-18 12:00:00" "2011-06-21 12:00:00" ...
##  $ NestContentText: chr  "Nest in use, known content" "Nest in use, known content" "Nest in use, kno
##  $ NestClutch     : chr  "1E" "2E" "2E" "" ...
```

```
## Close the connection
dbDisconnect(con) # disconnect after your work is done to free resources
```

2.2 Control (no need to run this section)

```
df_selected <- results
nrow(df_selected)
```

```
## [1] 48611
```

```r
# Subset
#df_selected <- results[1:1000, ]

# Count the number of occurrences of each unique character in NestClutch
table(results$NestClutch)
```

```
##
##                1C    1C,1D    1C,2E       1D    1D,2C       1E    1E,1C
##    18188    12347       28       32       46        7    11001      269
## 1E,1C,1D    1E,1D    1E,1c    1E,2C    1E,2c       1c    1c,1C    1c,2E
##        4        1       29       39       13      491        1       11
##    1d,2e       1e    1e,1E    1e,1c    1e,2C    1e,2E    1e,2c       2C
##        1      177        1        2        2        1        3     1169
##       2D       2E    2E,2C       2c       2e    2e,3E       3C       3E
##        3     3199        2      137       72        1      188      905
##       3c    3c,3C       3e    3e,3C    3e,3E
##       79        1      157        2        2
```

```r
# Convert df_selected into a tibble and summarize with dplyr
results %>%
  as_tibble() %>%
  count(NestClutch)
```

```
## # A tibble: 37 x 2
##    NestClutch     n
##    <chr>      <int>
##  1 ""         18188
##  2 "1C"       12347
##  3 "1C,1D"       28
##  4 "1C,2E"       32
##  5 "1D"          46
##  6 "1D,2C"        7
##  7 "1E"       11001
##  8 "1E,1C"      269
##  9 "1E,1C,1D"     4
## 10 "1E,1D"        1
## # i 27 more rows
```

2.3 Convert dates into the right format and optional subset of data

```r
# Convert VisitDate to Date format if it's not already
df_selected$VisitDate <- as.Date(df_selected$VisitDate)

# Extract year from VisitDate (makes it easier to iterate)
df_selected$Year <- format(df_selected$VisitDate, "%Y")

# Subset the data for a specific year, specie, location
df_selected_year <- subset(df_selected, Year == "2024")
#df_selected_specie <- subset(df_selected_year, spcENG == "Black-legged kittiwake")
#df_selected_location <- subset(df_selected_specie, Locality == "Ossian Sarsfjellet")
```

3. Hatching dates

The hatching dates were estimated based on the last egg visit entry and the first chick visit entry, a mean of the two registered dates.

3.1 Filtering steps and before getting hatching dates Make a note for exceptions in a new column

```
# Apply note to only egg/chick/empty entries and makes a note in a separate column
filter_clutch_values <- function(df) {
  # Define valid clutch values for different categories
  egg_clutch_values <- c("1E", "2E", "3E", "1e", "2e", "3e", "1e,1E", "2e,2E", "3e,3E")
  chick_clutch_values <- c("1C", "2C", "3C", "1c", "2c", "3c", "1c,1C", "2c,2C", "3c,3C")
  empty_clutch_values <- c("")

  # Initialize the Note column with empty strings
  df$Note <- ""

  # Iterate over unique NestNumbers and Years
  for (nest in unique(df$NestNumber)) {
    for (year in unique(df$Year)) {
      # Subset the data for the current NestNumber and Year
      subset_df <- df[df$NestNumber == nest & df$Year == year, ]

      # Remove rows with NA values in NestClutch
      subset_df <- subset_df[!is.na(subset_df$NestClutch), ]

      # Check conditions for E, C, or empty entries and append notes
      if (all(subset_df$NestClutch %in% egg_clutch_values)) {
        df$Note[df$NestNumber == nest & df$Year == year] <- "Only E-entries"
      } else if (all(subset_df$NestClutch %in% chick_clutch_values)) {
        df$Note[df$NestNumber == nest & df$Year == year] <- "Only C-entries"
      } else if (all(subset_df$NestClutch %in% empty_clutch_values)) {
        df$Note[df$NestNumber == nest & df$Year == year] <- "Only empty-entries"
      }
    }
  }

  return(df)
}

df_selected_year <- filter_clutch_values(df_selected_year)

head(df_selected_year)
```

```
##                       spcENG           Locality Area          NestNumber
## 45988 Brünnich's guillemot Ossian Sarsfjellet    -           GLS#0004
## 45989 Brünnich's guillemot Ossian Sarsfjellet    -           GLS#0006
## 45990 Brünnich's guillemot Ossian Sarsfjellet    -           GLS#0010
## 45991 Brünnich's guillemot Ossian Sarsfjellet    -           GLS#0011
## 45992 Brünnich's guillemot Ossian Sarsfjellet    -           GLS#0018
## 45993         Glaucous gull       Kongsfjorden    - Prins Heinrich#0004
##        VisitDate            NestContentText NestClutch Year           Note
## 45988 2024-06-11 Nest in use, known content         1E 2024
## 45989 2024-06-11 Nest in use, known content         1E 2024
```

```
## 45990 2024-06-11 Nest in use, known content         1E 2024
## 45991 2024-06-11 Nest in use, known content         1E 2024
## 45992 2024-06-11 Nest in use, known content         1E 2024
## 45993 2024-06-11 Nest in use, known content         3C 2024 Only C-entries
```

3.2 Find last E and first C

```r
# Find for last visit with egg
find_last_E <- function(df) {
  # Define the clutch values to search for
  clutch_values <- c("1E", "2E", "3E", "1e", "2e", "3e", "1e,1E", "3e,3E", "2e,3E", "1e,2E")

  # Create a new column `Last_E` initialized with NA
  df$Last_E <- as.Date(NA)

  # Iterate over unique NestNumbers and Years
  for (nest in unique(df$NestNumber)) {
    for (year in unique(df$Year)) {
      # Subset the data for current NestNumber and Year
      subset_df <- df[df$NestNumber == nest & df$Year == year, ]

      # Find the last occurrence of any specified clutch value
      matching_clutches <- subset_df[subset_df$NestClutch %in% clutch_values, ]

      if (nrow(matching_clutches) > 0) {
        # Get the last matching row based on VisitDate
        last_entry <- matching_clutches[which.max(matching_clutches$VisitDate), ]

        # Assign the Last_E value to all rows for this NestNumber and Year
        df[df$NestNumber == nest & df$Year == year, "Last_E"] <- last_entry$VisitDate
      }
    }
  }

  return(df)
}

# Find for first visit with chick
find_first_C <- function(df) {
  # Define the clutch values to search for
  clutch_values <- c("1C,2E", "1E,1C", "2C", "1C", "1E,2C", "1c", "3C", "1C,1D",
                     "2c", "3c", "1E,1c", "1e,1c", "3c,3C", "1E, 2c", "1c,2E",
                     "1e,2c", "2E,2C", "1E,1C,1D", "1c,1C", "1e,2C", "1D,2C", "3e,3C")

  # Create a new column `First_C` initialized with NA
  df$First_C <- as.Date(NA)

  # Iterate over unique NestNumbers and Years
  for (nest in unique(df$NestNumber)) {
    for (year in unique(df$Year)) {
      # Subset the data for current NestNumber and Year
      subset_df <- df[df$NestNumber == nest & df$Year == year, ]

      # Find the first occurrence of any specified clutch value
```

```
      matching_clutches <- subset_df[subset_df$NestClutch %in% clutch_values, ]

      if (nrow(matching_clutches) > 0) {
        # Get the first matching row based on VisitDate
        first_entry <- matching_clutches[which.min(matching_clutches$VisitDate), ]

        # Assign the First_C value to all rows for this NestNumber and Year
        df[df$NestNumber == nest & df$Year == year, "First_C"] <- first_entry$VisitDate
      }
    }
  }

  return(df)
}

# Find for last visit with chick
find_last_C <- function(df) {
  # Define the clutch values to search for
  clutch_values <- c("1C,2E", "1E,1C", "2C", "1C", "1E,2C", "1c", "3C", "1C,1D",
                     "2c", "3c", "1E,1c", "1e,1c", "3c,3C", "1E, 2c", "1c,2E",
                     "1e,2c", "2E,2C", "1E,1C,1D", "1c,1C", "1e,2C", "1D,2C", "3e,3C")

  # Create a new column `Last_C` initialized with NA
  df$Last_C <- as.Date(NA)

  # Iterate over unique NestNumbers and Years
  for (nest in unique(df$NestNumber)) {
    for (year in unique(df$Year)) {
      # Subset the data for current NestNumber and Year
      subset_df <- df[df$NestNumber == nest & df$Year == year, ]

      # Find the last occurrence of any specified clutch value
      matching_clutches <- subset_df[subset_df$NestClutch %in% clutch_values, ]

      if (nrow(matching_clutches) > 0) {
        # Get the last matching row based on VisitDate
        last_entry <- matching_clutches[which.max(matching_clutches$VisitDate), ]

        # Assign the Last_E value to all rows for this NestNumber and Year
        df[df$NestNumber == nest & df$Year == year, "Last_C"] <- last_entry$VisitDate
      }
    }
  }

  return(df)
}

# Get the last registered visit date
find_last_visit <- function(df) {
  # Create a new column `Last_Visit` initialized with NA
  df$Last_Visit <- as.Date(NA)

  # Iterate over unique NestNumbers and Years
```

```
  for (nest in unique(df$NestNumber)) {
    for (year in unique(df$Year)) {
      # Subset the data for the current NestNumber and Year
      subset_df <- df[df$NestNumber == nest & df$Year == year, ]

      # Find the last visit based on VisitDate
      if (nrow(subset_df) > 0) {
        last_entry <- subset_df[which.max(subset_df$VisitDate), ]

        # Assign the Last_Visit value to all rows for this NestNumber and Year
        df[df$NestNumber == nest & df$Year == year, "Last_Visit"] <- last_entry$VisitDate
      }
    }
  }

  return(df)
}

phenology <- df_selected_year %>%
  find_last_E() %>%
  find_first_C() %>%
  find_last_C() %>%
  find_last_visit()

# View the result
head(phenology)
```

```
##                       spcENG           Locality Area          NestNumber
## 45988 Brünnich's guillemot Ossian Sarsfjellet    -            GLS#0004
## 45989 Brünnich's guillemot Ossian Sarsfjellet    -            GLS#0006
## 45990 Brünnich's guillemot Ossian Sarsfjellet    -            GLS#0010
## 45991 Brünnich's guillemot Ossian Sarsfjellet    -            GLS#0011
## 45992 Brünnich's guillemot Ossian Sarsfjellet    -            GLS#0018
## 45993         Glaucous gull       Kongsfjorden    - Prins Heinrich#0004
##        VisitDate            NestContentText NestClutch Year           Note
## 45988 2024-06-11 Nest in use, known content         1E 2024
## 45989 2024-06-11 Nest in use, known content         1E 2024
## 45990 2024-06-11 Nest in use, known content         1E 2024
## 45991 2024-06-11 Nest in use, known content         1E 2024
## 45992 2024-06-11 Nest in use, known content         1E 2024
## 45993 2024-06-11 Nest in use, known content         3C 2024 Only C-entries
##           Last_E    First_C     Last_C Last_Visit
## 45988 2024-06-11 2024-07-02 2024-07-22 2024-07-30
## 45989 2024-06-22 2024-06-29 2024-07-25 2024-07-30
## 45990 2024-07-02 2024-07-09 2024-07-27 2024-07-30
## 45991 2024-06-16 2024-07-12 2024-07-30 2024-07-30
## 45992 2024-06-19 2024-07-02 2024-07-25 2024-07-30
## 45993       <NA> 2024-06-07 2024-06-11 2024-06-11
```

3.3 Estimate hatcing dates

```r
# Initialize new columns for hatching date and its accuracy
phenology$Hatching_date <- NA
phenology$Hatching_date_accuracy <- NA

# Get the hatching date and hatching date accuracy
phenology <- phenology %>%
  mutate(
    # Calculate mean date for Hatching_date
    Hatching_date = if_else(
      !is.na(Last_E) & !is.na(First_C),
      as.Date((as.numeric(Last_E) + as.numeric(First_C)) / 2, origin = "1970-01-01"),
      NA_Date_
    ),

    # Calculate accuracy as the number of days divided by 2
    Hatching_date_accuracy = if_else(
      !is.na(Last_E) & !is.na(First_C),
      round(as.numeric(First_C - Last_E) / 2),
      NA_real_
    )
  )


# Select the columns of interest
phenology <- phenology %>%
  select(spcENG, Locality, Area, NestNumber, Year, Note, Last_E, First_C, Last_C, Last_Visit, Hatching_
head(phenology)
```

```
##                     spcENG           Locality Area          NestNumber Year
## 45988 Brünnich's guillemot Ossian Sarsfjellet    -            GLS#0004 2024
## 45989 Brünnich's guillemot Ossian Sarsfjellet    -            GLS#0006 2024
## 45990 Brünnich's guillemot Ossian Sarsfjellet    -            GLS#0010 2024
## 45991 Brünnich's guillemot Ossian Sarsfjellet    -            GLS#0011 2024
## 45992 Brünnich's guillemot Ossian Sarsfjellet    -            GLS#0018 2024
## 45993         Glaucous gull       Kongsfjorden    - Prins Heinrich#0004 2024
##                 Note     Last_E    First_C     Last_C Last_Visit Hatching_date
## 45988               2024-06-11 2024-07-02 2024-07-22 2024-07-30    2024-06-21
## 45989               2024-06-22 2024-06-29 2024-07-25 2024-07-30    2024-06-25
## 45990               2024-07-02 2024-07-09 2024-07-27 2024-07-30    2024-07-05
## 45991               2024-06-16 2024-07-12 2024-07-30 2024-07-30    2024-06-29
## 45992               2024-06-19 2024-07-02 2024-07-25 2024-07-30    2024-06-25
## 45993 Only C-entries       <NA> 2024-06-07 2024-06-11 2024-06-11          <NA>
##       Hatching_date_accuracy
## 45988                     10
## 45989                      4
## 45990                      4
## 45991                     13
## 45992                      6
## 45993                     NA
```