Establishment Models Fitted

Loïc Pages

28/02/2025

Introduction

```
rm(list=ls())
library(knitr)
library(spaMM)
```

```
## Registered S3 methods overwritten by 'registry':
##   method               from
##   print.registry_field proxy
##   print.registry_entry proxy
```

```
## spaMM (Rousset & Ferdy, 2014, version 4.5.30) is loaded.
## Type 'help(spaMM)' for a short introduction,
## 'news(package='spaMM')' for news,
## and 'citation('spaMM')' for proper citation.
## Further infos, slides, etc. at https://gitlab.mbb.univ-montp2.fr/francois/spamm-ref.
```

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
## v lubridate 1.9.4     v tidyr     1.3.1
## v purrr     1.0.2
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(splines)
library(patchwork)
library(SplinesUtils)

setwd("/media/loic/Commun/0Travail/Stage 2025 ISEM/Models")
```

```
centauree_data <- read.csv("donnesIPM_short.csv")
centauree_data_complet <- read.csv("donnesIPM.csv")


#Supprimer plantes dont l'age est inconnu
centauree_data <- centauree_data[!is.na(centauree_data$age0), ]
centauree_data$age1 <- ifelse(centauree_data$Stage1=="V",centauree_data$age0+1,NA)

#Forcer l'age maximal à 8
length(centauree_data$age0[centauree_data$age0 >= 8])
```

```
## [1] 93
```

```
centauree_data$age0[centauree_data$age0 > 8] <- 8

spaMM.options(separation_max=70)
```

```
annees <- 1995:2022
populations <- c("Po","Au","Pe","E1","E2","Cr")
taille_range <- seq(0.5, 25, by = 0.5)
age_range <- 1:8

fake_data <- expand.grid(
  year = annees,
  Pop = populations,
  Size0Mars = taille_range,
  age0 = age_range
)

fake_data <- fake_data %>%
  mutate(Nrw = row_number())
```

BIC

```
extractBIC <- function(fit, n){
  extractAIC(fit)[[2]]+(log(n)-2)*DoF(fit)[[3]]
}
```

Test Splines -> poly

```
survdata <- centauree_data[centauree_data$Flowering0!=1,]
survdata <- survdata[!is.na(survdata$SurvieMars),]

library(nlme)
```

```
##
## Attaching package: 'nlme'

## The following object is masked from 'package:dplyr':
##
##     collapse
```

```
library(SplinesUtils)
spline_model <- lme(SurvieMars ~ bs(age0, degree=3, knots=6.5)+bs(Size0Mars,df=5), data = survdata, ran
  year = pdSymm(~ Size0Mars),
  Pop = pdSymm(~ age0)))

spl <- RegSplineAsPiecePoly(spline_model, "bs(Size0Mars, df = 5)")
spl2 <- RegSplineAsPiecePoly(spline_model, "bs(age0, degree = 3, knots = 6.5)")

spl$PiecePoly$coef
```

```
##               [,1]        [,2]          [,3]
## [1,] -1.040834e-17  0.06889436  0.1911940010
## [2,]  4.934356e-02  0.16645927  0.0961884958
## [3,]  2.964395e-01 -0.06220810 -0.0080626749
## [4,] -2.390984e-01  0.01804847  0.0002194813
```

```
spl
```

```
## 3 piecewise polynomials of degree 3 are constructed!
## Use 'summary' to export all of them.
## The first 3 are printed below.
## -1.04e-17 + 0.0493 * (x - 0.5) + 0.296 * (x - 0.5) ^ 2 - 0.239 * (x - 0.5) ^ 3
## 0.0689 + 0.166 * (x - 1) - 0.0622 * (x - 1) ^ 2 + 0.018 * (x - 1) ^ 3
## 0.191 + 0.0962 * (x - 2) - 0.00806 * (x - 2) ^ 2 + 0.000219 * (x - 2) ^ 3
```

Establishment rate

Remplir les données manquantes de nombres de capitules avec des prédictions.

```
cptl_data <- centauree_data_complet[!centauree_data_complet$Flowering0==0,]
Cptlglm1 <- fitme(Cptl0 ~ 1 + Size0Mars,
                  data=cptl_data)
# NbrCptl = -3.769 + 2.683*Size0Mars

cptl_data_predi <- cptl_data %>%
  mutate(Cptl0 = ifelse(is.na(Cptl0), round(-3.769+2.683*Size0Mars), Cptl0))
```

```
plt <- centauree_data_complet %>%
  filter(age0==1) %>%
  group_by(Quadrat,year,Pop) %>%
  summarize(NombrePlantules = sum(age0))
```

```
## `summarise()` has grouped output by 'Quadrat', 'year'. You can override using
## the `.groups` argument.
```

```
cptl <- cptl_data_predi %>%
  group_by(Quadrat,year,Pop) %>%
  summarize(NombresCapitules = sum(Cptl0))
```

```
## ‘summarise()‘ has grouped output by ’Quadrat’, ’year’. You can override using
## the ‘.groups‘ argument.
```

```
Estb <- inner_join(plt,cptl, by=join_by(Quadrat,year,Pop))
summary(Estb)
```

```
##     Quadrat          year          Pop            NombrePlantules
##  Min.   : 1.0   Min.   :1995   Length:162         Min.   :  1.0
##  1st Qu.: 6.0   1st Qu.:1997   Class :character   1st Qu.:  2.0
##  Median :26.5   Median :2000   Mode  :character   Median :  8.0
##  Mean   :22.5   Mean   :2002                      Mean   : 17.7
##  3rd Qu.:34.0   3rd Qu.:2004                      3rd Qu.: 19.0
##  Max.   :80.0   Max.   :2021                      Max.   :203.0
##  NombresCapitules
##  Min.   :  1.00
##  1st Qu.: 20.25
##  Median : 34.50
##  Mean   : 50.10
##  3rd Qu.: 60.75
##  Max.   :214.00
```

```
Estb <- Estb %>% mutate(EstbRate=rep(NA)) %>%
  arrange(Quadrat)

for (i in 2:length(Estb$Quadrat)){
  if (Estb$Quadrat[i]!=Estb$Quadrat[i-1]){next}
  if (Estb$year[i]!=Estb$year[i-1]+1){next}
  Estb$EstbRate[i] <- Estb$NombrePlantules[i]/Estb$NombresCapitules[i-1]
}
```

```
Estbglm1 <- fitme(EstbRate ~ 1 + (1|Pop:year), data=Estb)
Estbglm2 <- fitme(EstbRate ~ 1 +(1|year), data=Estb)
Estbglm3 <- fitme(EstbRate ~ 1 + (1|year) + (1|Pop:year), data=Estb)
Estbglm4 <- fitme(EstbRate ~ 1, data=Estb)
Estbglm5 <- fitme(EstbRate ~ 1 + (1|Pop) + (1|Pop:year), data=Estb)

Estbglm1
```

```
## formula: EstbRate ~ 1 + (1 | Pop:year)
## Estimation of fixed effects by ML.
## Estimation of lambda and phi by ’outer’ ML, maximizing logL.
## family: gaussian( link = identity )
##  ------------ Fixed effects (beta) ------------
##             Estimate Cond. SE t-value
## (Intercept)   0.4852  0.05292   9.169
##  --------------- Random effects ---------------
## Family: gaussian( link = identity )
##            --- Variance parameters (’lambda’):
## lambda = var(u) for u ~ Gaussian;
##    Pop:year  :  0.04924
## # of obs: 95; # of groups: Pop:year, 58
##  -------------- Residual variance  ------------
## phi estimate was 0.173107
##  ------------- Likelihood values  -------------
##                         logLik
## logL       (p_v(h)): -62.26091
```

```
Estbglm2
```

```
## formula: EstbRate ~ 1 + (1 | year)
## Estimation of fixed effects by ML.
## Estimation of lambda and phi by ’outer’ ML, maximizing logL.
## family: gaussian( link = identity )
##  ------------ Fixed effects (beta) ------------
##             Estimate Cond. SE t-value
## (Intercept)    0.486  0.06047   8.036
##  --------------- Random effects ---------------
## Family: gaussian( link = identity )
##            --- Variance parameters (’lambda’):
## lambda = var(u) for u ~ Gaussian;
##    year  :  0.01659
## # of obs: 95; # of groups: year, 16
##  -------------- Residual variance  ------------
## phi estimate was 0.206833
##  ------------- Likelihood values  -------------
##                        logLik
## logL       (p_v(h)): -62.8351
```

```
Estbglm3
```

```
## formula: EstbRate ~ 1 + (1 | year) + (1 | Pop:year)
## Estimation of fixed effects by ML.
## Estimation of lambda and phi by ’outer’ ML, maximizing logL.
## family: gaussian( link = identity )
##  ------------ Fixed effects (beta) ------------
##             Estimate Cond. SE t-value
## (Intercept)   0.4839  0.05963   8.116
##  --------------- Random effects ---------------
## Family: gaussian( link = identity )
##            --- Variance parameters (’lambda’):
## lambda = var(u) for u ~ Gaussian;
##    year  :  0.01134
##    Pop:year  :  0.03626
## # of obs: 95; # of groups: year, 16; Pop:year, 58
##  -------------- Residual variance  ------------
## phi estimate was 0.174002
##  ------------- Likelihood values  -------------
##                         logLik
## logL       (p_v(h)): -61.91014
```

```
Estbglm4
```

```
## formula: EstbRate ~ 1
## ML: Estimation of phi by ML.
##     Estimation of fixed effects by ML.
## family: gaussian( link = identity )
##  ------------ Fixed effects (beta) ------------
##             Estimate Cond. SE t-value
## (Intercept)   0.4912  0.04868   10.09
##  -------------- Residual variance  ------------
## Coefficients for log(phi)  ~ 1  :
##             Estimate Cond. SE
## (Intercept)   -1.491   0.1451
## Estimate of phi=residual var:  0.2251
##  ------------- Likelihood values  -------------
##                           logLik
## logL               : -63.96959
```

```
Estbglm5
```

```
## formula: EstbRate ~ 1 + (1 | Pop) + (1 | Pop:year)
## Estimation of fixed effects by ML.
## Estimation of lambda and phi by ’outer’ ML, maximizing logL.
## family: gaussian( link = identity )
##  ------------ Fixed effects (beta) ------------
##             Estimate Cond. SE t-value
## (Intercept)   0.4852  0.05292   9.169
##  --------------- Random effects ---------------
## Family: gaussian( link = identity )
##            --- Variance parameters (’lambda’):
## lambda = var(u) for u ~ Gaussian;
##    Pop  :  1.98e-07
##    Pop:year  :  0.04924
## # of obs: 95; # of groups: Pop, 6; Pop:year, 58
##  -------------- Residual variance  ------------
## phi estimate was 0.173107
##  ------------- Likelihood values  -------------
##                           logLik
## logL       (p_v(h)): -62.26092
```

```
n <- length(Estb$Quadrat)
extractAIC(Estbglm1) ; extractBIC(Estbglm1, n)
```

```
##      edf      AIC
##   1.0000 130.5218
```

```
## [1] 133.6094
```

```
extractAIC(Estbglm2) ; extractBIC(Estbglm2, n)
```

```
##      edf      AIC
##   1.0000 131.6702
```

```
## [1] 134.7578
```

```r
extractAIC(Estbglm3) ; extractBIC(Estbglm3, n)
```

```
##      edf      AIC 
##   1.0000 131.8203

## [1] 134.9079
```

```r
extractAIC(Estbglm4) ; extractBIC(Estbglm4, n)
```

```
##      edf      AIC 
##   1.0000 131.9392

## [1] 135.0268
```

```r
extractAIC(Estbglm5) ; extractBIC(Estbglm5, n)
```

```
##      edf      AIC 
##   1.0000 132.5218

## [1] 135.6094
```

```r
Estbpredict1 <- predict(Estbglm1, newdata = fake_data)[,1]
Estbpredict2 <- predict(Estbglm2, newdata = fake_data)[,1]
Estbpredict3 <- predict(Estbglm3, newdata = fake_data)[,1]
Estbpredict4 <- predict(Estbglm4, newdata = fake_data)[,1]
Estbpredict5 <- predict(Estbglm5, newdata = fake_data)[,1]
```

```r
plot_estb <- function(data = fake_data, prediction, var, fact) {
  data %>%
    mutate(plt_predi = prediction) %>%
    ggplot(aes(x = .data[[var]], y = plt_predi)) +
    geom_line(aes(color = as.factor(.data[[fact]]))) +
    labs(y="Establishment rate")+
    theme_minimal()
}

plot_estb2 <- function(data = fake_data, prediction, var, fact) {
  data %>%
    mutate(plt_predi = prediction) %>%
    ggplot(aes(x = .data[[var]], y = plt_predi)) +
    geom_point(aes(color = as.factor(.data[[fact]]))) +
    labs(y="Establishment rate")+
    theme_minimal()
}
```

Establishment rate en fonction de l'année

var <- "year"; fact <- "Pop"

Establishment rate en fonction de la population

var <- "Pop"; fact <- "year"