Survival Models Fitted

Loïc Pages

2025-02-28

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,
  SizeOMars = 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(SizeOMars,df=5), data = survdata, rand
  year = pdSymm(~ SizeOMars),
  Pop = pdSymm(~ age0)))

spl <- RegSplineAsPiecePoly(spline_model, "bs(SizeOMars, 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
```

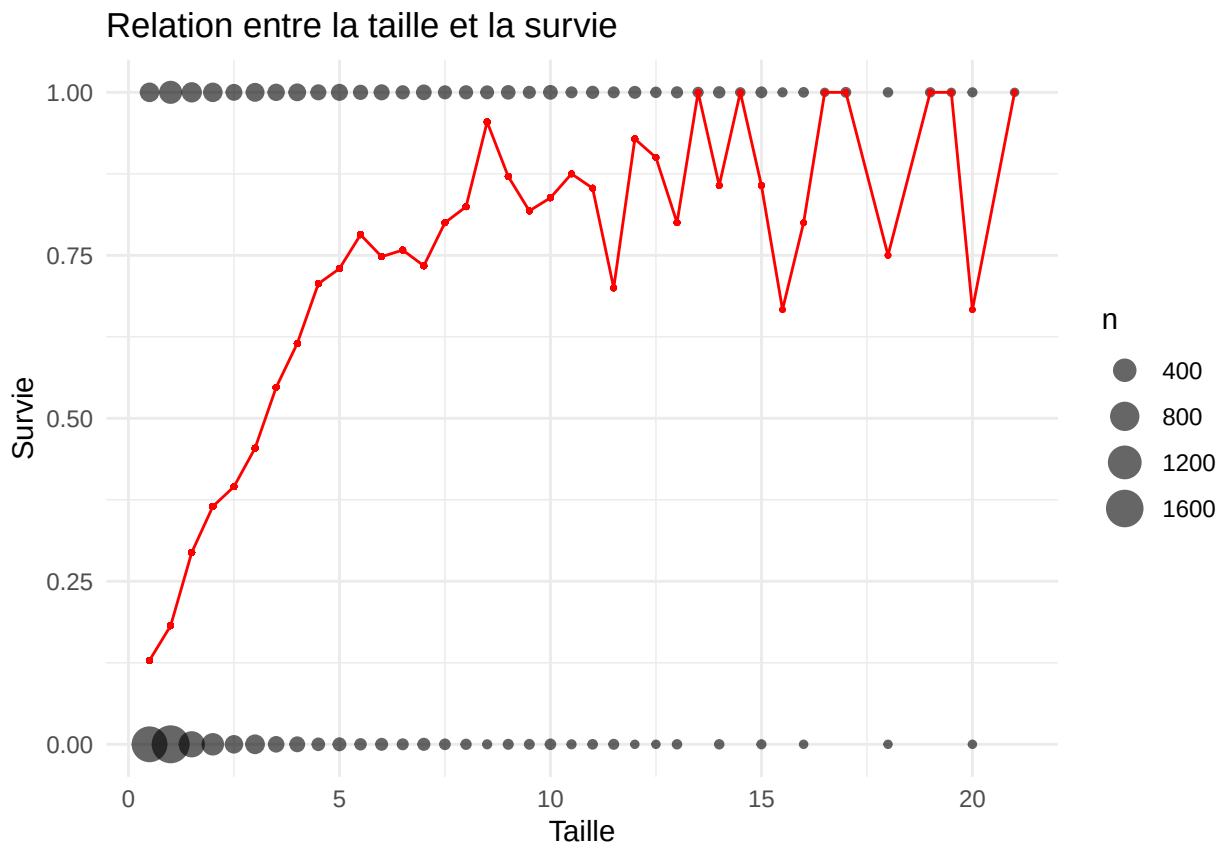
```
spl
```

```
## 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
```

Survival probability

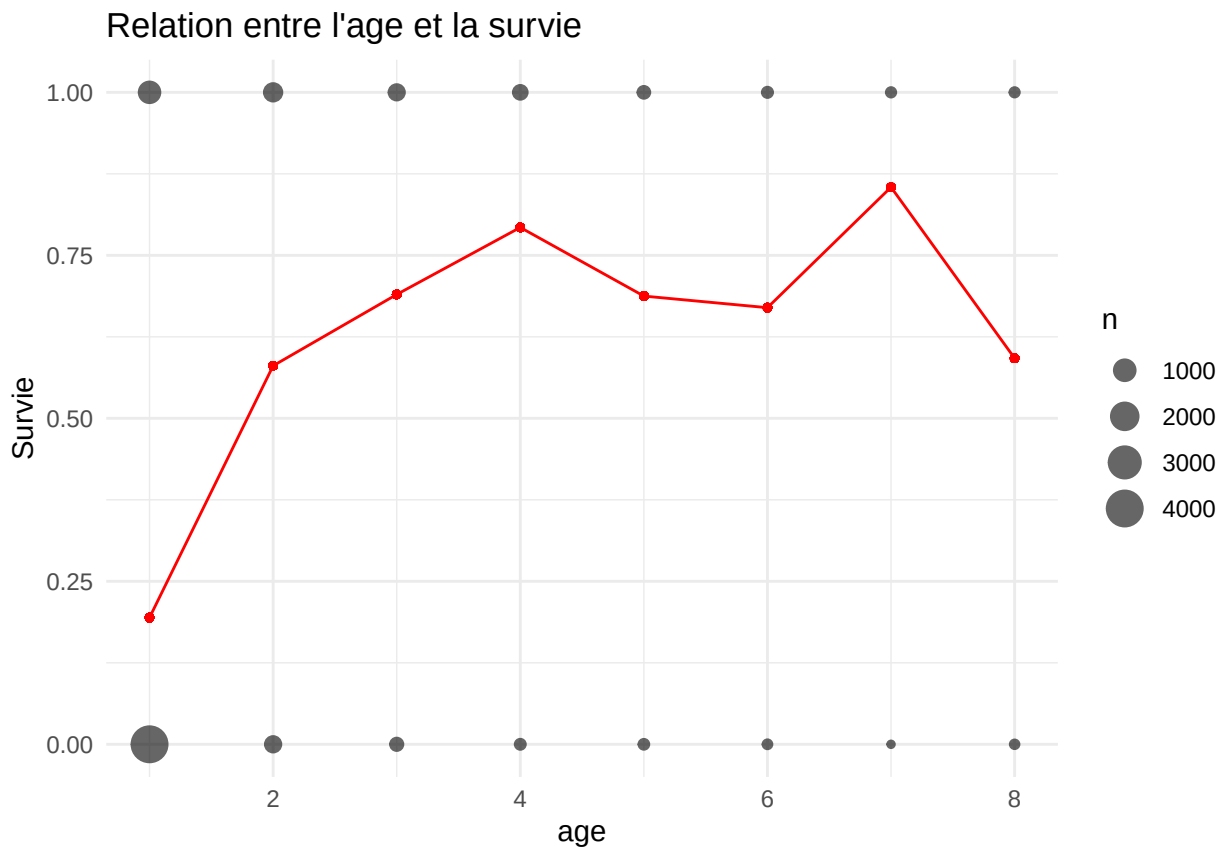
```
survdata <- centauree_data[centauree_data$Flowering0!=1,]
# survdata$SurvieMars[survdata$age0==7][1:15] <- 0

survdata %>%
  group_by(SizeOMars) %>%
  mutate(survivalProba = sum(SurvieMars, na.rm = TRUE) / n()) %>%
  ggplot(aes(x = SizeOMars, y = SurvieMars)) +
  geom_count(alpha = 0.6) +  # Points dimensionnés selon la fréquence
  geom_point(aes(y = survivalProba), color = "red", size = 0.5) +
  geom_line(aes(y = survivalProba), color = "red") +
  labs(title = "Relation entre la taille et la survie",
       x = "Taille",
       y = "Survie") +
  ylim(0, 1) +
  theme_minimal()
```

```
## Warning: Removed 121 rows containing non-finite outside the scale range
## (`stat_sum()`).
```




```
survdata %>%
  group_by(age0) %>%
  mutate(survivalProba = sum(SurvieMars, na.rm = TRUE) / n()) %>%
  ggplot(aes(x = age0, y = SurvieMars)) +
  geom_count(alpha = 0.6) +  # Points dimensionnés selon la fréquence
  geom_point(aes(x = age0, y = survivalProba), color = "red", size = 1) +
  geom_line(aes(x = age0, y = survivalProba), color = "red") +
  labs(title = "Relation entre l'age et la survie",
       x = "age",
       y = "Survie") +
  ylim(0, 1) +
  theme_minimal()
```

```
## Warning: Removed 121 rows containing non-finite outside the scale range
## (`stat_sum()`).
```



Age 1

```
survdata1 <- survdata[survdata$age0==1,]
```

```
Survglm1 <- fitme(SurvieMars ~ 1+ poly(SizeOMars,3) + (SizeOMars|year) + (1|Pop),
                  family=binomial,
                  data=survdata1,
                  method="PQL/L")

Survglm2 <- fitme(SurvieMars ~ 1+ poly(SizeOMars,4) + (SizeOMars|year) + (1|Pop),
                  family=binomial,
                  data=survdata1,
                  method="PQL/L")

Survglm3 <- fitme(SurvieMars ~ 1+ bs(SizeOMars,df=5,degree=3) + (SizeOMars|year) + (1|Pop),
                  family=binomial,
                  data=survdata1,
                  method="PQL/L")

Survglm4 <- fitme(SurvieMars ~ 1+ poly(SizeOMars,3) + (SizeOMars|year) + (SizeOMars|Pop),
                  family=binomial,
                  data=survdata1,
                  method="PQL/L")

Survglm5 <- fitme(SurvieMars ~ 1+ poly(SizeOMars,4) + (SizeOMars|year) + (SizeOMars|Pop),
                  family=binomial,
```

```
                  data=survdata1,
                  method="PQL/L")
```

```
n <- length(survdata1$Nrw)
extractAIC(Survglm1) ; extractBIC(Survglm1, n)
```

```
##      edf      AIC
##    4.000 4373.984
```

```
## [1] 4400.104
```

```
extractAIC(Survglm2) ; extractBIC(Survglm2, n)
```

```
##      edf      AIC
##    5.000 4374.731
```

```
## [1] 4407.381
```

```
extractAIC(Survglm3) ; extractBIC(Survglm3, n)
```

```
##      edf      AIC
##    5.000 4375.971
```

```
## [1] 4408.621
```

```
extractAIC(Survglm4) ; extractBIC(Survglm4, n)
```

```
##     edf     AIC
##    4.00 4376.12
```

```
## [1] 4402.24
```

```
extractAIC(Survglm5) ; extractBIC(Survglm5, n)
```

```
##      edf      AIC
##    5.000 4376.557
```

```
## [1] 4409.206
```

```
summary(Survglm1)
```

```
## formula: SurvieMars ~ 1 + poly(SizeOMars, 3) + (SizeOMars | year) + (1 |
##     Pop)
## Estimation of lambda and ranCoefs by ML (p_v approximation of logL).
## Estimation of fixed effects by h-likelihood approximation.
## family: binomial( link = logit )
##  ------------ Fixed effects (beta) ------------
##                      Estimate Cond. SE t-value
```

```
## (Intercept)           -1.506   0.2423  -6.212
## poly(SizeOMars, 3)1   36.016   5.0164   7.180
## poly(SizeOMars, 3)2   12.039   7.0270   1.713
## poly(SizeOMars, 3)3   29.460   7.1587   4.115
##  --------------- Random effects ---------------
## Family: gaussian( link = identity )
##          --- Random-coefficients Cov matrices:
##  Group        Term    Var.   Corr.
##   year (Intercept)   1.574
##   year   SizeOMars 0.06589 -0.7786
##            --- Variance parameters ('lambda'):
## lambda = var(u) for u ~ Gaussian;
##    Pop  :  0.08913
##              --- Coefficients for log(lambda):
##  Group        Term Estimate Cond.SE
##    Pop (Intercept)   -2.418  0.6505
## # of obs: 4989; # of groups: year, 27; Pop, 6
##  ------------- Likelihood values  -------------
##                          logLik
##        h-likelihood: -2176.248
## logL       (p_v(h)): -2178.992
```

```
summary(Survglm2)
```

```
## formula: SurvieMars ~ 1 + poly(SizeOMars, 4) + (SizeOMars | year) + (1 |
##     Pop)
## Estimation of lambda and ranCoefs by ML (p_v approximation of logL).
## Estimation of fixed effects by h-likelihood approximation.
## family: binomial( link = logit )
##  ------------ Fixed effects (beta) ------------
##                      Estimate Cond. SE t-value
## (Intercept)           -1.513    0.242  -6.253
## poly(SizeOMars, 4)1   30.361    4.754   6.387
## poly(SizeOMars, 4)2   -6.956    6.526  -1.066
## poly(SizeOMars, 4)3    9.480    6.769   1.401
## poly(SizeOMars, 4)4   -9.658    3.997  -2.416
##  --------------- Random effects ---------------
## Family: gaussian( link = identity )
##          --- Random-coefficients Cov matrices:
##  Group        Term   Var.   Corr.
##   year (Intercept)  1.564
##   year   SizeOMars 0.0624 -0.7827
##            --- Variance parameters ('lambda'):
## lambda = var(u) for u ~ Gaussian;
##    Pop  :  0.08803
##              --- Coefficients for log(lambda):
##  Group        Term Estimate Cond.SE
##    Pop (Intercept)    -2.43   0.651
## # of obs: 4989; # of groups: year, 27; Pop, 6
##  ------------- Likelihood values  -------------
##                          logLik
##        h-likelihood: -2175.949
## logL       (p_v(h)): -2178.366
```

```
summary(Survglm3)
```

```
## formula: SurvieMars ~ 1 + bs(SizeOMars, df = 5, degree = 3) + (SizeOMars |
##     year) + (1 | Pop)
## Estimation of lambda and ranCoefs by ML (p_v approximation of logL).
## Estimation of fixed effects by h-likelihood approximation.
## family: binomial( link = logit )
##  ------------ Fixed effects (beta) ------------
##                                    Estimate Cond. SE t-value
## (Intercept)                          -2.078   0.2757  -7.537
## bs(SizeOMars, df = 5, degree = 3)1       NA       NA      NA
## bs(SizeOMars, df = 5, degree = 3)2    0.419   0.1211   3.460
## bs(SizeOMars, df = 5, degree = 3)3    3.972   0.7476   5.313
## bs(SizeOMars, df = 5, degree = 3)4   -8.065   3.7463  -2.153
## bs(SizeOMars, df = 5, degree = 3)5   33.773  11.3136   2.985
##  --------------- Random effects ---------------
## Family: gaussian( link = identity )
##          --- Random-coefficients Cov matrices:
##  Group        Term    Var.   Corr.
##   year (Intercept)   1.573
##   year   SizeOMars 0.06546 -0.7794
##            --- Variance parameters ('lambda'):
## lambda = var(u) for u ~ Gaussian;
##    Pop  :  0.08903
##              --- Coefficients for log(lambda):
##  Group        Term Estimate Cond.SE
##    Pop (Intercept)   -2.419  0.6506
## # of obs: 4989; # of groups: year, 27; Pop, 6
##  ------------- Likelihood values  -------------
##                         logLik
##        h-likelihood: -2176.286
## logL       (p_v(h)): -2178.986
```

```
summary(Survglm4)
```

```
## formula: SurvieMars ~ 1 + poly(SizeOMars, 3) + (SizeOMars | year) + (SizeOMars |
##     Pop)
## Estimation of ranCoefs by ML (p_v approximation of logL).
## Estimation of fixed effects by h-likelihood approximation.
## family: binomial( link = logit )
##  ------------ Fixed effects (beta) ------------
##                     Estimate Cond. SE t-value
## (Intercept)           -1.516   0.2444  -6.202
## poly(SizeOMars, 3)1   35.811   5.3146   6.738
## poly(SizeOMars, 3)2   11.589   7.1402   1.623
## poly(SizeOMars, 3)3   30.423   7.2118   4.218
##  --------------- Random effects ---------------
## Family: gaussian( link = identity )
##          --- Random-coefficients Cov matrices:
##  Group        Term     Var.   Corr.
##   year (Intercept)    1.599
##   year   SizeOMars  0.06592 -0.8005
```

```
##    Pop (Intercept)   0.1493
##    Pop   SizeOMars 0.004363      -1
## # of obs: 4989; # of groups: year, 27; Pop, 6
##  ------------- Likelihood values  -------------
##                         logLik
##        h-likelihood: -2188.472
## logL       (p_v(h)): -2178.060
```

```
summary(Survglm5)
```

```
## formula: SurvieMars ~ 1 + poly(SizeOMars, 4) + (SizeOMars | year) + (SizeOMars |
##     Pop)
## Estimation of ranCoefs by ML (p_v approximation of logL).
## Estimation of fixed effects by h-likelihood approximation.
## family: binomial( link = logit )
##  ------------ Fixed effects (beta) ------------
##                     Estimate Cond. SE t-value
## (Intercept)           -1.525   0.2441  -6.245
## poly(SizeOMars, 4)1   29.932   4.9923   5.996
## poly(SizeOMars, 4)2   -8.214   5.7539  -1.428
## poly(SizeOMars, 4)3    9.745   5.9144   1.648
## poly(SizeOMars, 4)4  -10.096   3.7189  -2.715
##  --------------- Random effects ---------------
## Family: gaussian( link = identity )
##          --- Random-coefficients Cov matrices:
##  Group        Term    Var.   Corr.
##   year (Intercept)    1.59
##   year   SizeOMars 0.06218 -0.8071
##    Pop (Intercept)  0.1532
##    Pop   SizeOMars  0.0051      -1
## # of obs: 4989; # of groups: year, 27; Pop, 6
##  ------------- Likelihood values  -------------
##                         logLik
##        h-likelihood: -2188.108
## logL       (p_v(h)): -2177.278
```

```
fake_data1 <- fake_data[fake_data$age0==1,]
Survpredict1 <- predict(Survglm1, newdata = fake_data1)[,1]
Survpredict2 <- predict(Survglm2, newdata = fake_data1)[,1]
Survpredict3 <- predict(Survglm3, newdata = fake_data1)[,1]
```

```
## Warning in bs(SizeOMars, degree = 3L, knots = c(1, 1), Boundary.knots = c(0.5,
## : some 'x' values beyond boundary knots may cause ill-conditioned bases
## Warning in bs(SizeOMars, degree = 3L, knots = c(1, 1), Boundary.knots = c(0.5,
## : some 'x' values beyond boundary knots may cause ill-conditioned bases
## Warning in bs(SizeOMars, degree = 3L, knots = c(1, 1), Boundary.knots = c(0.5,
## : some 'x' values beyond boundary knots may cause ill-conditioned bases
```

```
Survpredict4 <- predict(Survglm4, newdata = fake_data1)[,1]
Survpredict5 <- predict(Survglm5, newdata = fake_data1)[,1]
```

```
plot_survie1 <- function(data = fake_data1, prediction, var, c1, valc1, fact, mindat, maxdat) {
  data %>%
    mutate(surv_predi = prediction) %>%
    filter(!!sym(c1) == valc1) %>%
    ggplot(aes(x = .data[[var]], y = surv_predi)) +
    geom_vline(xintercept=maxdat, lty="dotted")+
    geom_vline(xintercept=mindat, lty="dotted")+
    geom_line(aes(color = as.factor(.data[[fact]]))) +
    theme_minimal() +
    ylim(0, 1)+
    xlim(0,maxdat)
}
```

Survie en fonction de la taille

En fixant la population : voir l'effet année

```
var <- "SizeOMars"
c1 <- "Pop"
valc1 <- "Au"
fact <- "year"
```

En fixant l'année : voir l'effet population

```
var <- "SizeOMars"
c1 <- "year"
valc1 <- 2000
fact <- "Pop"
```

Avec splines

```
Survglm1 <- fitme(SurvieMars ~ 1+ bs(SizeOMars,df=4,degree=2) + bs(age0,degree=3,knots = 6.5)+ (age0|yea
                  family=binomial,
                  data=survdata,
                  method="PQL/L")

Survglm2 <- fitme(SurvieMars ~ 1 + bs(SizeOMars,df=5,degree=3) + bs(age0,degree=3,knots = 6.5)
                  + (age0|year) + (age0|Pop) ,
                  family=binomial,
                  data=survdata,
                  method="PQL/L")

Survglm3 <- fitme(SurvieMars ~ 1 + bs(SizeOMars,df=4,degree=2) + bs(age0,degree=3,knots = c(1.5,6.5)) +
                  family=binomial,
                  data=survdata,
                  method="PQL/L")
```

```
## Warning in (function (formula, resid.formula = NULL, data, prior.weights, :
## 'c(' detected in formula: did you mean cbind() for binomial response or for
## poly()?
```

```
Survglm4 <- fitme(SurvieMars ~ 1 + bs(Size0Mars,df=4,degree=2) + bs(age0,degree=3,knots = 6.5)
                  + (age0|year) + (Size0Mars + age0|Pop) ,
                  family=binomial,
                  data=survdata,
                  method="PQL/L")

Survglm5 <- fitme(SurvieMars ~ 1 + bs(Size0Mars,df=5,degree=3) + bs(age0,degree=3,knots = c(1.5,6.5))
                  + (age0|year) + (age0|Pop) ,
                  family=binomial,
                  data=survdata,
                  method="PQL/L")
```

```
## Warning in (function (formula, resid.formula = NULL, data, prior.weights, :
## 'c(' detected in formula: did you mean cbind() for binomial response or for
## poly()?
```

```
n <- length(centauree_data$Nrw)
extractAIC(Survglm1) ; extractBIC(Survglm1, n)
```

```
##      edf      AIC
##    9.000 6704.533
```

```
## [1] 6766.998
```

```
extractAIC(Survglm2) ; extractBIC(Survglm2, n)
```

```
##      edf      AIC
##   10.000 6705.501
```

```
## [1] 6774.906
```

```
extractAIC(Survglm3) ; extractBIC(Survglm3, n)
```

```
##      edf      AIC
##   10.000 6706.512
```

```
## [1] 6775.917
```

```
extractAIC(Survglm4) ; extractBIC(Survglm4, n)
```

```
##      edf      AIC
##    9.000 6707.213
```

```
## [1] 6769.678
```

```
extractAIC(Survglm5) ; extractBIC(Survglm5, n)
```

```
##      edf      AIC
##   11.000 6707.474
```

```
## [1] 6783.819
```

```
summary(Survglm1)
```

```
## formula: SurvieMars ~ 1 + bs(SizeOMars, df = 4, degree = 2) + bs(age0,
##     degree = 3, knots = 6.5) + (age0 | year) + (age0 | Pop)
## Estimation of ranCoefs by ML (p_v approximation of logL).
## Estimation of fixed effects by h-likelihood approximation.
## family: binomial( link = logit )
##  ------------ Fixed effects (beta) ------------
##                                     Estimate Cond. SE  t-value
## (Intercept)                         -2.24740   0.2447 -9.18514
## bs(SizeOMars, df = 4, degree = 2)1  0.29743   0.1298  2.29223
## bs(SizeOMars, df = 4, degree = 2)2  1.06568   0.1112  9.58279
## bs(SizeOMars, df = 4, degree = 2)3  5.34564   0.3837 13.93042
## bs(SizeOMars, df = 4, degree = 2)4  3.02236   0.7222  4.18469
## bs(age0, degree = 3, knots = 6.5)1  2.66779   0.3095  8.61909
## bs(age0, degree = 3, knots = 6.5)2 -1.02769   0.6027 -1.70518
## bs(age0, degree = 3, knots = 6.5)3  2.23645   0.7219  3.09782
## bs(age0, degree = 3, knots = 6.5)4  0.03102   0.5666  0.05476
##  --------------- Random effects ---------------
## Family: gaussian( link = identity )
##          --- Random-coefficients Cov matrices:
##  Group        Term    Var.   Corr.
##   year (Intercept)   1.272
##   year        age0 0.05915 -0.7272
##    Pop (Intercept)  0.1501
##    Pop        age0 0.01036 -0.7778
## # of obs: 7293; # of groups: year, 27; Pop, 6
##  ------------- Likelihood values  -------------
##                         logLik
##        h-likelihood: -3330.395
## logL       (p_v(h)): -3337.267
```

```
summary(Survglm2)
```

```
## formula: SurvieMars ~ 1 + bs(SizeOMars, df = 5, degree = 3) + bs(age0,
##     degree = 3, knots = 6.5) + (age0 | year) + (age0 | Pop)
## Estimation of ranCoefs by ML (p_v approximation of logL).
## Estimation of fixed effects by h-likelihood approximation.
## family: binomial( link = logit )
##  ------------ Fixed effects (beta) ------------
##                                     Estimate Cond. SE  t-value
## (Intercept)                          -2.2484   0.2451 -9.17448
## bs(SizeOMars, df = 5, degree = 3)1   0.1098   0.2000  0.54921
## bs(SizeOMars, df = 5, degree = 3)2   0.8527   0.1260  6.76591
```

```
## bs(SizeOMars, df = 5, degree = 3)3   4.2860   0.4539  9.44330
## bs(SizeOMars, df = 5, degree = 3)4   3.8293   0.9307  4.11457
## bs(SizeOMars, df = 5, degree = 3)5   3.7218   1.1975  3.10808
## bs(age0, degree = 3, knots = 6.5)1   2.6464   0.3106  8.52124
## bs(age0, degree = 3, knots = 6.5)2  -1.0172   0.6013 -1.69155
## bs(age0, degree = 3, knots = 6.5)3   2.2242   0.7193  3.09205
## bs(age0, degree = 3, knots = 6.5)4   0.0392   0.5643  0.06946
##  --------------- Random effects ---------------
## Family: gaussian( link = identity )
##          --- Random-coefficients Cov matrices:
##  Group        Term    Var.   Corr.
##   year (Intercept)   1.276
##   year        age0 0.05888 -0.7283
##    Pop (Intercept)   0.151
##    Pop        age0 0.01016 -0.7852
## # of obs: 7293; # of groups: year, 27; Pop, 6
##  ------------- Likelihood values  -------------
##                         logLik
##        h-likelihood: -3329.977
## logL       (p_v(h)): -3336.750
```

```
summary(Survglm3)
```

```
## formula: SurvieMars ~ 1 + bs(SizeOMars, df = 4, degree = 2) + bs(age0,
##     degree = 3, knots = c(1.5, 6.5)) + (age0 | year) + (age0 |
##     Pop)
## Estimation of ranCoefs by ML (p_v approximation of logL).
## Estimation of fixed effects by h-likelihood approximation.
## family: binomial( link = logit )
##  ------------ Fixed effects (beta) ------------
##                                            Estimate Cond. SE  t-value
## (Intercept)                                 -2.2472   0.2447 -9.18351
## bs(SizeOMars, df = 4, degree = 2)1           0.2973   0.1298  2.29121
## bs(SizeOMars, df = 4, degree = 2)2           1.0662   0.1113  9.58089
## bs(SizeOMars, df = 4, degree = 2)3           5.3476   0.3841 13.92333
## bs(SizeOMars, df = 4, degree = 2)4           3.0197   0.7226  4.17884
## bs(age0, degree = 3, knots = c(1.5, 6.5))1   0.2008   0.3476  0.57778
## bs(age0, degree = 3, knots = c(1.5, 6.5))2   2.4668   0.5927  4.16223
## bs(age0, degree = 3, knots = c(1.5, 6.5))3  -0.8651   0.8055 -1.07392
## bs(age0, degree = 3, knots = c(1.5, 6.5))4   2.2769   0.7970  2.85699
## bs(age0, degree = 3, knots = c(1.5, 6.5))5   0.0305   0.5668  0.05381
##  --------------- Random effects ---------------
## Family: gaussian( link = identity )
##          --- Random-coefficients Cov matrices:
##  Group        Term    Var.   Corr.
##   year (Intercept)   1.272
##   year        age0 0.05924 -0.7269
##    Pop (Intercept)  0.1503
##    Pop        age0 0.01037 -0.7779
## # of obs: 7293; # of groups: year, 27; Pop, 6
##  ------------- Likelihood values  -------------
##                         logLik
##        h-likelihood: -3330.367
## logL       (p_v(h)): -3337.256
```

```r
summary(Survglm4)
```

```
## formula: SurvieMars ~ 1 + bs(Size0Mars, df = 4, degree = 2) + bs(age0,
##     degree = 3, knots = 6.5) + (age0 | year) + (Size0Mars + age0 |
##     Pop)
## Estimation of ranCoefs by ML (p_v approximation of logL).
## Estimation of fixed effects by h-likelihood approximation.
## family: binomial( link = logit )
##  ------------ Fixed effects (beta) ------------
##                                    Estimate Cond. SE  t-value
## (Intercept)                         -2.2290   0.2373 -9.39369
## bs(Size0Mars, df = 4, degree = 2)1   0.2928   0.1297  2.25639
## bs(Size0Mars, df = 4, degree = 2)2   1.0543   0.1120  9.41013
## bs(Size0Mars, df = 4, degree = 2)3   5.3295   0.4181 12.74721
## bs(Size0Mars, df = 4, degree = 2)4   3.1229   0.7995  3.90601
## bs(age0, degree = 3, knots = 6.5)1   2.7114   0.3101  8.74413
## bs(age0, degree = 3, knots = 6.5)2  -1.0623   0.6075 -1.74857
## bs(age0, degree = 3, knots = 6.5)3   2.3503   0.7273  3.23129
## bs(age0, degree = 3, knots = 6.5)4   0.0546   0.5740  0.09513
##  --------------- Random effects ---------------
## Family: gaussian( link = identity )
##          --- Random-coefficients Cov matrices:
##  Group        Term     Var.   Corr. Corr..1
##   year (Intercept)    1.277
##   year        age0  0.06163 -0.7238
##    Pop (Intercept)   0.1383
##    Pop   Size0Mars 0.001352  0.4838
##    Pop        age0  0.01099 -0.9822 -0.3109
## # of obs: 7293; # of groups: year, 27; Pop, 6
##  ------------- Likelihood values  -------------
##                         logLik
##        h-likelihood: -3333.679
## logL       (p_v(h)): -3335.607
```

```r
summary(Survglm5)
```

```
## formula: SurvieMars ~ 1 + bs(Size0Mars, df = 5, degree = 3) + bs(age0,
##     degree = 3, knots = c(1.5, 6.5)) + (age0 | year) + (age0 |
##     Pop)
## Estimation of ranCoefs by ML (p_v approximation of logL).
## Estimation of fixed effects by h-likelihood approximation.
## family: binomial( link = logit )
##  ------------ Fixed effects (beta) ------------
##                                              Estimate Cond. SE  t-value
## (Intercept)                                  -2.24813   0.2451 -9.17259
## bs(Size0Mars, df = 5, degree = 3)1            0.10968   0.2000  0.54838
## bs(Size0Mars, df = 5, degree = 3)2            0.85293   0.1260  6.76736
## bs(Size0Mars, df = 5, degree = 3)3            4.28917   0.4545  9.43764
## bs(Size0Mars, df = 5, degree = 3)4            3.82646   0.9312  4.10930
## bs(Size0Mars, df = 5, degree = 3)5            3.72175   1.1976  3.10775
## bs(age0, degree = 3, knots = c(1.5, 6.5))1    0.19193   0.3476  0.55220
## bs(age0, degree = 3, knots = c(1.5, 6.5))2    2.45805   0.5921  4.15141
```

```
## bs(age0, degree = 3, knots = c(1.5, 6.5))3 -0.86775   0.8037 -1.07965
## bs(age0, degree = 3, knots = c(1.5, 6.5))4  2.27125   0.7942  2.85965
## bs(age0, degree = 3, knots = c(1.5, 6.5))5  0.03866   0.5646  0.06847
##  --------------- Random effects ---------------
## Family: gaussian( link = identity )
##          --- Random-coefficients Cov matrices:
##  Group        Term    Var.   Corr.
##   year (Intercept)   1.275
##   year        age0 0.05897  -0.728
##    Pop (Intercept)  0.1512
##    Pop        age0 0.01017 -0.7853
## # of obs: 7293; # of groups: year, 27; Pop, 6
##  ------------- Likelihood values  -------------
##                         logLik
##        h-likelihood: -3329.945
## logL       (p_v(h)): -3336.737
```

```
Survpredict1 <- predict(Survglm1, newdata = fake_data)[,1]
```

```
## Warning in bs(Size0Mars, degree = 2L, knots = c(1, 2), Boundary.knots = c(0.5,
## : some 'x' values beyond boundary knots may cause ill-conditioned bases
## Warning in bs(Size0Mars, degree = 2L, knots = c(1, 2), Boundary.knots = c(0.5,
## : some 'x' values beyond boundary knots may cause ill-conditioned bases
## Warning in bs(Size0Mars, degree = 2L, knots = c(1, 2), Boundary.knots = c(0.5,
## : some 'x' values beyond boundary knots may cause ill-conditioned bases
## Warning in bs(Size0Mars, degree = 2L, knots = c(1, 2), Boundary.knots = c(0.5,
## : some 'x' values beyond boundary knots may cause ill-conditioned bases
## Warning in bs(Size0Mars, degree = 2L, knots = c(1, 2), Boundary.knots = c(0.5,
## : some 'x' values beyond boundary knots may cause ill-conditioned bases
## Warning in bs(Size0Mars, degree = 2L, knots = c(1, 2), Boundary.knots = c(0.5,
## : some 'x' values beyond boundary knots may cause ill-conditioned bases
## Warning in bs(Size0Mars, degree = 2L, knots = c(1, 2), Boundary.knots = c(0.5,
## : some 'x' values beyond boundary knots may cause ill-conditioned bases
## Warning in bs(Size0Mars, degree = 2L, knots = c(1, 2), Boundary.knots = c(0.5,
## : some 'x' values beyond boundary knots may cause ill-conditioned bases
## Warning in bs(Size0Mars, degree = 2L, knots = c(1, 2), Boundary.knots = c(0.5,
## : some 'x' values beyond boundary knots may cause ill-conditioned bases
## Warning in bs(Size0Mars, degree = 2L, knots = c(1, 2), Boundary.knots = c(0.5,
## : some 'x' values beyond boundary knots may cause ill-conditioned bases
## Warning in bs(Size0Mars, degree = 2L, knots = c(1, 2), Boundary.knots = c(0.5,
## : some 'x' values beyond boundary knots may cause ill-conditioned bases
## Warning in bs(Size0Mars, degree = 2L, knots = c(1, 2), Boundary.knots = c(0.5,
## : some 'x' values beyond boundary knots may cause ill-conditioned bases
## Warning in bs(Size0Mars, degree = 2L, knots = c(1, 2), Boundary.knots = c(0.5,
## : some 'x' values beyond boundary knots may cause ill-conditioned bases
## Warning in bs(Size0Mars, degree = 2L, knots = c(1, 2), Boundary.knots = c(0.5,
## : some 'x' values beyond boundary knots may cause ill-conditioned bases
```

```
Survpredict2 <- predict(Survglm2, newdata = fake_data)[,1]
```

```
## Warning in bs(Size0Mars, degree = 3L, knots = c(1, 2), Boundary.knots = c(0.5,
## : some 'x' values beyond boundary knots may cause ill-conditioned bases
```

```
## Warning in bs(Size0Mars, degree = 3L, knots = c(1, 2), Boundary.knots = c(0.5,
## : some 'x' values beyond boundary knots may cause ill-conditioned bases
## Warning in bs(Size0Mars, degree = 3L, knots = c(1, 2), Boundary.knots = c(0.5,
## : some 'x' values beyond boundary knots may cause ill-conditioned bases
## Warning in bs(Size0Mars, degree = 3L, knots = c(1, 2), Boundary.knots = c(0.5,
## : some 'x' values beyond boundary knots may cause ill-conditioned bases
## Warning in bs(Size0Mars, degree = 3L, knots = c(1, 2), Boundary.knots = c(0.5,
## : some 'x' values beyond boundary knots may cause ill-conditioned bases
## Warning in bs(Size0Mars, degree = 3L, knots = c(1, 2), Boundary.knots = c(0.5,
## : some 'x' values beyond boundary knots may cause ill-conditioned bases
## Warning in bs(Size0Mars, degree = 3L, knots = c(1, 2), Boundary.knots = c(0.5,
## : some 'x' values beyond boundary knots may cause ill-conditioned bases
## Warning in bs(Size0Mars, degree = 3L, knots = c(1, 2), Boundary.knots = c(0.5,
## : some 'x' values beyond boundary knots may cause ill-conditioned bases
## Warning in bs(Size0Mars, degree = 3L, knots = c(1, 2), Boundary.knots = c(0.5,
## : some 'x' values beyond boundary knots may cause ill-conditioned bases
## Warning in bs(Size0Mars, degree = 3L, knots = c(1, 2), Boundary.knots = c(0.5,
## : some 'x' values beyond boundary knots may cause ill-conditioned bases
## Warning in bs(Size0Mars, degree = 3L, knots = c(1, 2), Boundary.knots = c(0.5,
## : some 'x' values beyond boundary knots may cause ill-conditioned bases
## Warning in bs(Size0Mars, degree = 3L, knots = c(1, 2), Boundary.knots = c(0.5,
## : some 'x' values beyond boundary knots may cause ill-conditioned bases
## Warning in bs(Size0Mars, degree = 3L, knots = c(1, 2), Boundary.knots = c(0.5,
## : some 'x' values beyond boundary knots may cause ill-conditioned bases
## Warning in bs(Size0Mars, degree = 3L, knots = c(1, 2), Boundary.knots = c(0.5,
## : some 'x' values beyond boundary knots may cause ill-conditioned bases
```

```r
Survpredict3 <- predict(Survglm3, newdata = fake_data)[,1]
```

```
## Warning in bs(Size0Mars, degree = 2L, knots = c(1, 2), Boundary.knots = c(0.5,
## : some 'x' values beyond boundary knots may cause ill-conditioned bases
```

```
## Warning in bs(Size0Mars, degree = 2L, knots = c(1, 2), Boundary.knots = c(0.5,
## : some 'x' values beyond boundary knots may cause ill-conditioned bases
## Warning in bs(Size0Mars, degree = 2L, knots = c(1, 2), Boundary.knots = c(0.5,
## : some 'x' values beyond boundary knots may cause ill-conditioned bases
## Warning in bs(Size0Mars, degree = 2L, knots = c(1, 2), Boundary.knots = c(0.5,
## : some 'x' values beyond boundary knots may cause ill-conditioned bases
## Warning in bs(Size0Mars, degree = 2L, knots = c(1, 2), Boundary.knots = c(0.5,
## : some 'x' values beyond boundary knots may cause ill-conditioned bases
## Warning in bs(Size0Mars, degree = 2L, knots = c(1, 2), Boundary.knots = c(0.5,
## : some 'x' values beyond boundary knots may cause ill-conditioned bases
## Warning in bs(Size0Mars, degree = 2L, knots = c(1, 2), Boundary.knots = c(0.5,
## : some 'x' values beyond boundary knots may cause ill-conditioned bases
## Warning in bs(Size0Mars, degree = 2L, knots = c(1, 2), Boundary.knots = c(0.5,
## : some 'x' values beyond boundary knots may cause ill-conditioned bases
## Warning in bs(Size0Mars, degree = 2L, knots = c(1, 2), Boundary.knots = c(0.5,
## : some 'x' values beyond boundary knots may cause ill-conditioned bases
## Warning in bs(Size0Mars, degree = 2L, knots = c(1, 2), Boundary.knots = c(0.5,
## : some 'x' values beyond boundary knots may cause ill-conditioned bases
## Warning in bs(Size0Mars, degree = 2L, knots = c(1, 2), Boundary.knots = c(0.5,
## : some 'x' values beyond boundary knots may cause ill-conditioned bases
## Warning in bs(Size0Mars, degree = 2L, knots = c(1, 2), Boundary.knots = c(0.5,
```

```
## : some 'x' values beyond boundary knots may cause ill-conditioned bases
## Warning in bs(Size0Mars, degree = 2L, knots = c(1, 2), Boundary.knots = c(0.5,
## : some 'x' values beyond boundary knots may cause ill-conditioned bases
## Warning in bs(Size0Mars, degree = 2L, knots = c(1, 2), Boundary.knots = c(0.5,
## : some 'x' values beyond boundary knots may cause ill-conditioned bases
```

```
Survpredict4 <- predict(Survglm4, newdata = fake_data)[,1]
```

```
## Warning in bs(Size0Mars, degree = 2L, knots = c(1, 2), Boundary.knots = c(0.5,
## : some 'x' values beyond boundary knots may cause ill-conditioned bases
## Warning in bs(Size0Mars, degree = 2L, knots = c(1, 2), Boundary.knots = c(0.5,
## : some 'x' values beyond boundary knots may cause ill-conditioned bases
## Warning in bs(Size0Mars, degree = 2L, knots = c(1, 2), Boundary.knots = c(0.5,
## : some 'x' values beyond boundary knots may cause ill-conditioned bases
## Warning in bs(Size0Mars, degree = 2L, knots = c(1, 2), Boundary.knots = c(0.5,
## : some 'x' values beyond boundary knots may cause ill-conditioned bases
## Warning in bs(Size0Mars, degree = 2L, knots = c(1, 2), Boundary.knots = c(0.5,
## : some 'x' values beyond boundary knots may cause ill-conditioned bases
## Warning in bs(Size0Mars, degree = 2L, knots = c(1, 2), Boundary.knots = c(0.5,
## : some 'x' values beyond boundary knots may cause ill-conditioned bases
## Warning in bs(Size0Mars, degree = 2L, knots = c(1, 2), Boundary.knots = c(0.5,
## : some 'x' values beyond boundary knots may cause ill-conditioned bases
## Warning in bs(Size0Mars, degree = 2L, knots = c(1, 2), Boundary.knots = c(0.5,
## : some 'x' values beyond boundary knots may cause ill-conditioned bases
## Warning in bs(Size0Mars, degree = 2L, knots = c(1, 2), Boundary.knots = c(0.5,
## : some 'x' values beyond boundary knots may cause ill-conditioned bases
## Warning in bs(Size0Mars, degree = 2L, knots = c(1, 2), Boundary.knots = c(0.5,
## : some 'x' values beyond boundary knots may cause ill-conditioned bases
## Warning in bs(Size0Mars, degree = 2L, knots = c(1, 2), Boundary.knots = c(0.5,
## : some 'x' values beyond boundary knots may cause ill-conditioned bases
## Warning in bs(Size0Mars, degree = 2L, knots = c(1, 2), Boundary.knots = c(0.5,
## : some 'x' values beyond boundary knots may cause ill-conditioned bases
## Warning in bs(Size0Mars, degree = 2L, knots = c(1, 2), Boundary.knots = c(0.5,
## : some 'x' values beyond boundary knots may cause ill-conditioned bases
## Warning in bs(Size0Mars, degree = 2L, knots = c(1, 2), Boundary.knots = c(0.5,
## : some 'x' values beyond boundary knots may cause ill-conditioned bases
```

```
Survpredict5 <- predict(Survglm5, newdata = fake_data)[,1]
```

```
## Warning in bs(Size0Mars, degree = 3L, knots = c(1, 2), Boundary.knots = c(0.5,
## : some 'x' values beyond boundary knots may cause ill-conditioned bases
```

```
## Warning in bs(Size0Mars, degree = 3L, knots = c(1, 2), Boundary.knots = c(0.5,
## : some 'x' values beyond boundary knots may cause ill-conditioned bases
## Warning in bs(Size0Mars, degree = 3L, knots = c(1, 2), Boundary.knots = c(0.5,
## : some 'x' values beyond boundary knots may cause ill-conditioned bases
## Warning in bs(Size0Mars, degree = 3L, knots = c(1, 2), Boundary.knots = c(0.5,
## : some 'x' values beyond boundary knots may cause ill-conditioned bases
## Warning in bs(Size0Mars, degree = 3L, knots = c(1, 2), Boundary.knots = c(0.5,
## : some 'x' values beyond boundary knots may cause ill-conditioned bases
## Warning in bs(Size0Mars, degree = 3L, knots = c(1, 2), Boundary.knots = c(0.5,
## : some 'x' values beyond boundary knots may cause ill-conditioned bases
```

```
## Warning in bs(Size0Mars, degree = 3L, knots = c(1, 2), Boundary.knots = c(0.5,
## : some 'x' values beyond boundary knots may cause ill-conditioned bases
## Warning in bs(Size0Mars, degree = 3L, knots = c(1, 2), Boundary.knots = c(0.5,
## : some 'x' values beyond boundary knots may cause ill-conditioned bases
## Warning in bs(Size0Mars, degree = 3L, knots = c(1, 2), Boundary.knots = c(0.5,
## : some 'x' values beyond boundary knots may cause ill-conditioned bases
## Warning in bs(Size0Mars, degree = 3L, knots = c(1, 2), Boundary.knots = c(0.5,
## : some 'x' values beyond boundary knots may cause ill-conditioned bases
## Warning in bs(Size0Mars, degree = 3L, knots = c(1, 2), Boundary.knots = c(0.5,
## : some 'x' values beyond boundary knots may cause ill-conditioned bases
## Warning in bs(Size0Mars, degree = 3L, knots = c(1, 2), Boundary.knots = c(0.5,
## : some 'x' values beyond boundary knots may cause ill-conditioned bases
## Warning in bs(Size0Mars, degree = 3L, knots = c(1, 2), Boundary.knots = c(0.5,
## : some 'x' values beyond boundary knots may cause ill-conditioned bases
## Warning in bs(Size0Mars, degree = 3L, knots = c(1, 2), Boundary.knots = c(0.5,
## : some 'x' values beyond boundary knots may cause ill-conditioned bases
```

```
plot_survie <- function(data = fake_data, prediction, var, c1, valc1 = 1, c2, valc2 = "Au", fact, minda
  data %>%
    mutate(surv_predi = prediction) %>%
    filter(!!sym(c1) == valc1, !!sym(c2) == valc2) %>%
    ggplot(aes(x = .data[[var]], y = surv_predi)) +
    geom_vline(xintercept=maxdat, lty="dotted")+
    geom_vline(xintercept=mindat, lty="dotted")+
    geom_line(aes(color = as.factor(.data[[fact]]))) +
    theme_minimal() +
    ylim(0, 1)
}
```

Survie en fonction de la taille

En fixant la population : voir l'effet année

```
var <- "Size0Mars"
c1 <- "age0"
c2 <- "Pop"
valc2 <- "Au"
fact <- "year"
```

En fixant l'année : voir l'effet population

```
var <- "Size0Mars"
c1 <- "age0"
c2 <- "year"
valc2 <- 2000
fact <- "Pop"
```

Survie en fonction de l'age

En fixant la population : voir l'effet année

```
var <- "age0"
c1 <- "Size0Mars"
c2 <- "Pop"
valc2 <- "Au"
fact <- "year"
```

En fixant l'année : voir l'effet population

```
var <- "age0"
c1 <- "Size0Mars"
c2 <- "year"
valc2 <- 2000
fact <- "Pop"
```

Avec poly (et Nrw)

Avec polynomes de degré 4 et effet aléatoire individus

```
Survglm1 <- fitme(SurvieMars ~ 1+ poly(Size0Mars,4) + poly(age0,4)+ (age0|year) + (age0|Pop) ,
                  family=binomial,
                  data=survdata,
                  method="PQL/L")

Survglm2 <- fitme(SurvieMars ~ 1+ poly(Size0Mars,3) + poly(age0,4)+ (age0|year) + (age0|Pop) ,
                  family=binomial,
                  data=survdata,
                  method="PQL/L")

Survglm3 <- fitme(SurvieMars ~ 1+ poly(Size0Mars,4) + poly(age0,4)+ (age0|year) + (age0|Pop) + (1|Nrw),
                  family=binomial,
                  data=survdata,
                  method="PQL/L")

Survglm4 <- fitme(SurvieMars ~ 1+ poly(Size0Mars,3) + poly(age0,4)+ (age0|year) + (age0|Pop) +(1|Nrw),
                  family=binomial,
                  data=survdata,
                  method="PQL/L")

Survglm5 <- fitme(SurvieMars ~ 1+ poly(Size0Mars,4) + poly(age0,4)+ (age0|year) + (Size0Mars + age0|Pop)
                  family=binomial,
                  data=survdata,
                  method="PQL/L")
```

```r
n <- length(centauree_data$Nrw)
extractAIC(Survglm1) ; extractBIC(Survglm1, n)
```

```
##      edf      AIC
##    9.000 6715.552
```

```
## [1] 6778.017
```

```r
extractAIC(Survglm2) ; extractBIC(Survglm2, n)
```

```
##      edf      AIC
##    8.000 6716.118
```

```
## [1] 6771.642
```

```r
extractAIC(Survglm3) ; extractBIC(Survglm3, n)
```

```
##      edf      AIC
##    9.000 6716.191
```

```
## [1] 6778.656
```

```r
extractAIC(Survglm4) ; extractBIC(Survglm4, n)
```

```
##      edf      AIC
##    8.000 6716.653
```

```
## [1] 6772.177
```

```r
extractAIC(Survglm5) ; extractBIC(Survglm5, n)
```

```
##      edf      AIC
##    9.000 6718.384
```

```
## [1] 6780.848
```

```r
summary(Survglm1)
```

```
## formula: SurvieMars ~ 1 + poly(SizeOMars, 4) + poly(age0, 4) + (age0 |
##     year) + (age0 | Pop)
## Estimation of ranCoefs by ML (p_v approximation of logL).
## Estimation of fixed effects by h-likelihood approximation.
## family: binomial( link = logit )
##  ------------ Fixed effects (beta) ------------
##                     Estimate Cond. SE t-value
## (Intercept)          -0.8316   0.2089  -3.981
## poly(SizeOMars, 4)1  81.1109   4.2340  19.157
## poly(SizeOMars, 4)2 -28.7092   3.2348  -8.875
```

```
## poly(SizeOMars, 4)3  10.0305   2.9389   3.413
## poly(SizeOMars, 4)4  -4.8679   2.9373  -1.657
## poly(age0, 4)1       25.8434   8.5546   3.021
## poly(age0, 4)2      -25.2941   3.1175  -8.114
## poly(age0, 4)3       10.3349   2.7834   3.713
## poly(age0, 4)4       -7.7561   2.6666  -2.909
##  --------------- Random effects ---------------
## Family: gaussian( link = identity )
##          --- Random-coefficients Cov matrices:
##  Group        Term     Var.   Corr.
##   year (Intercept)    1.275
##   year        age0  0.05704  -0.723
##    Pop (Intercept)   0.1496
##    Pop        age0 0.009923 -0.7796
## # of obs: 7293; # of groups: year, 27; Pop, 6
##  -------------- Likelihood values  -------------
##                         logLik
##        h-likelihood: -3335.976
## logL       (p_v(h)): -3342.776
```

```
summary(Survglm2)
```

```
## formula: SurvieMars ~ 1 + poly(SizeOMars, 3) + poly(age0, 4) + (age0 |
##     year) + (age0 | Pop)
## Estimation of ranCoefs by ML (p_v approximation of logL).
## Estimation of fixed effects by h-likelihood approximation.
## family: binomial( link = logit )
##  ------------ Fixed effects (beta) ------------
##                     Estimate Cond. SE t-value
## (Intercept)          -0.8266   0.2082  -3.971
## poly(SizeOMars, 3)1  80.7460   4.2179  19.144
## poly(SizeOMars, 3)2 -28.7984   3.4245  -8.410
## poly(SizeOMars, 3)3  10.6991   3.3016   3.241
## poly(age0, 4)1       25.8296   8.4834   3.045
## poly(age0, 4)2      -25.2809   3.1139  -8.119
## poly(age0, 4)3       10.5462   2.7810   3.792
## poly(age0, 4)4       -7.7687   2.6691  -2.911
##  --------------- Random effects ---------------
## Family: gaussian( link = identity )
##          --- Random-coefficients Cov matrices:
##  Group        Term     Var.   Corr.
##   year (Intercept)    1.268
##   year        age0  0.05551 -0.7228
##    Pop (Intercept)   0.1494
##    Pop        age0 0.009805 -0.7931
## # of obs: 7293; # of groups: year, 27; Pop, 6
##  -------------- Likelihood values  -------------
##                         logLik
##        h-likelihood: -3337.661
## logL       (p_v(h)): -3344.059
```

```
summary(Survglm3)
```

```
## formula: SurvieMars ~ 1 + poly(SizeOMars, 4) + poly(age0, 4) + (age0 |
##     year) + (age0 | Pop) + (1 | Nrw)
## Estimation of lambda and ranCoefs by ML (p_v approximation of logL).
## Estimation of fixed effects by h-likelihood approximation.
## family: binomial( link = logit )
##  ------------ Fixed effects (beta) ------------

## [one-time computation of covariance matrix, which may be slow]

##                     Estimate Cond. SE t-value
## (Intercept)          -0.8677   0.2132  -4.069
## poly(SizeOMars, 4)1  82.1551   4.3039  19.088
## poly(SizeOMars, 4)2 -28.5717   3.2652  -8.750
## poly(SizeOMars, 4)3   9.7555   2.9692   3.286
## poly(SizeOMars, 4)4  -4.7928   2.9664  -1.616
## poly(age0, 4)1       20.2503   8.6429   2.343
## poly(age0, 4)2      -24.0855   3.1696  -7.599
## poly(age0, 4)3        9.7873   2.8056   3.488
## poly(age0, 4)4       -7.5383   2.6797  -2.813
##  --------------- Random effects ---------------
## Family: gaussian( link = identity )
##          --- Random-coefficients Cov matrices:
##  Group        Term     Var.   Corr.
##   year (Intercept)    1.321
##   year        age0  0.05831 -0.7256
##    Pop (Intercept)    0.154
##    Pop        age0 0.009912 -0.7696
##            --- Variance parameters ('lambda'):
## lambda = var(u) for u ~ Gaussian;
##    Nrw  :  0.1093
##              --- Coefficients for log(lambda):
##  Group        Term Estimate Cond.SE
##    Nrw (Intercept)   -2.213  0.1352
## # of obs: 7293; # of groups: year, 27; Pop, 6; Nrw, 5017
##  ------------- Likelihood values  -------------
##                          logLik
##        h-likelihood: -2336.431
## logL       (p_v(h)): -3342.096
## Estimates did not converge; increase control.HLfit's 'max.iter' above 200,
##  or try control.HLfit=list(LevenbergM=TRUE) (see help('control.HLfit') for details).
```

```
summary(Survglm4)
```

```
## formula: SurvieMars ~ 1 + poly(SizeOMars, 3) + poly(age0, 4) + (age0 |
##     year) + (age0 | Pop) + (1 | Nrw)
## Estimation of lambda and ranCoefs by ML (p_v approximation of logL).
## Estimation of fixed effects by h-likelihood approximation.
## family: binomial( link = logit )
##  ------------ Fixed effects (beta) ------------

## [one-time computation of covariance matrix, which may be slow]
```

```
##                     Estimate Cond. SE t-value
## (Intercept)          -0.8642   0.2127  -4.063
## poly(SizeOMars, 3)1  81.8385   4.2889  19.082
## poly(SizeOMars, 3)2 -28.6829   3.4416  -8.334
## poly(SizeOMars, 3)3  10.3754   3.3122   3.132
## poly(age0, 4)1       20.0306   8.5679   2.338
## poly(age0, 4)2      -24.0265   3.1671  -7.586
## poly(age0, 4)3        9.9703   2.8038   3.556
## poly(age0, 4)4       -7.5461   2.6822  -2.813
##  --------------- Random effects ---------------
## Family: gaussian( link = identity )
##          --- Random-coefficients Cov matrices:
##  Group        Term     Var.   Corr.
##   year (Intercept)    1.316
##   year        age0  0.05674 -0.7257
##    Pop (Intercept)    0.154
##    Pop        age0 0.009761  -0.783
##            --- Variance parameters ('lambda'):
## lambda = var(u) for u ~ Gaussian;
##    Nrw  :  0.1132
##             --- Coefficients for log(lambda):
##  Group        Term Estimate Cond.SE
##    Nrw (Intercept)   -2.179  0.1329
## # of obs: 7293; # of groups: year, 27; Pop, 6; Nrw, 5017
##  ------------- Likelihood values  -------------
##                          logLik
##        h-likelihood: -2423.785
## logL       (p_v(h)): -3343.327
## Estimates did not converge; increase control.HLfit's 'max.iter' above 200,
##  or try control.HLfit=list(LevenbergM=TRUE) (see help('control.HLfit') for details).
```

```
summary(Survglm5)
```

```
## formula: SurvieMars ~ 1 + poly(SizeOMars, 4) + poly(age0, 4) + (age0 |
##     year) + (SizeOMars + age0 | Pop)
## Estimation of ranCoefs by ML (p_v approximation of logL).
## Estimation of fixed effects by h-likelihood approximation.
## family: binomial( link = logit )
##  ------------ Fixed effects (beta) ------------
##                     Estimate Cond. SE t-value
## (Intercept)          -0.8135   0.2064  -3.942
## poly(SizeOMars, 4)1  81.4302   5.4094  15.053
## poly(SizeOMars, 4)2 -28.1166   3.2756  -8.584
## poly(SizeOMars, 4)3   9.8745   2.9351   3.364
## poly(SizeOMars, 4)4  -4.8495   2.8973  -1.674
## poly(age0, 4)1       26.6245   8.7001   3.060
## poly(age0, 4)2      -25.4679   3.1280  -8.142
## poly(age0, 4)3       10.4278   2.7882   3.740
## poly(age0, 4)4       -8.1169   2.6718  -3.038
##  --------------- Random effects ---------------
## Family: gaussian( link = identity )
##          --- Random-coefficients Cov matrices:
##  Group        Term     Var.   Corr. Corr..1
##   year (Intercept)    1.278
```

```
##   year        age0  0.05918 -0.7187
##    Pop (Intercept)   0.1376
##    Pop   Size0Mars 0.001237  0.5189
##    Pop        age0  0.01077 -0.9789 -0.3331
## # of obs: 7293; # of groups: year, 27; Pop, 6
##  ------------- Likelihood values  -------------
##                          logLik
##        h-likelihood: -3339.371
## logL       (p_v(h)): -3341.192
```

```
Survpredict1 <- predict(Survglm1, newdata = fake_data)[,1]
Survpredict2 <- predict(Survglm2, newdata = fake_data)[,1]
Survpredict3 <- predict(Survglm3, newdata = fake_data)[,1]
Survpredict4 <- predict(Survglm4, newdata = fake_data)[,1]
Survpredict5 <- predict(Survglm5, newdata = fake_data)[,1]
```

```
plot_survie <- function(data = fake_data, prediction, var, c1, valc1 = 1, c2, valc2 = "Au", fact, minda
  data %>%
    mutate(surv_predi = prediction) %>%
    filter(!!sym(c1) == valc1, !!sym(c2) == valc2) %>%
    ggplot(aes(x = .data[[var]], y = surv_predi)) +
    geom_vline(xintercept=maxdat, lty="dotted")+
    geom_vline(xintercept=mindat, lty="dotted")+
    geom_line(aes(color = as.factor(.data[[fact]]))) +
    theme_minimal() +
    ylim(0, 1)
}
```

Survie en fonction de la taille

En fixant la population : voir l'effet année

```
var <- "Size0Mars"
c1 <- "age0"
c2 <- "Pop"
valc2 <- "Au"
fact <- "year"
```

En fixant l'année : voir l'effet population

```
var <- "Size0Mars"
c1 <- "age0"
c2 <- "year"
valc2 <- 2000
fact <- "Pop"
```

Survie en fonction de l'age

En fixant la population : voir l'effet année

```
var <- "age0"
c1 <- "Size0Mars"
c2 <- "Pop"
valc2 <- "Au"
fact <- "year"
```

En fixant l'année : voir l'effet population

```
var <- "age0"
c1 <- "Size0Mars"
c2 <- "year"
valc2 <- 2000
fact <- "Pop"
```