Plantules size Models Fitted

Loïc Pages

2025-02-28

Introduction

```r
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.
```

```r
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
```

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

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

```r
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
```

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

spaMM.options(separation_max=70)
```

```r
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

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

Test Splines -> poly

```r
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
```

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

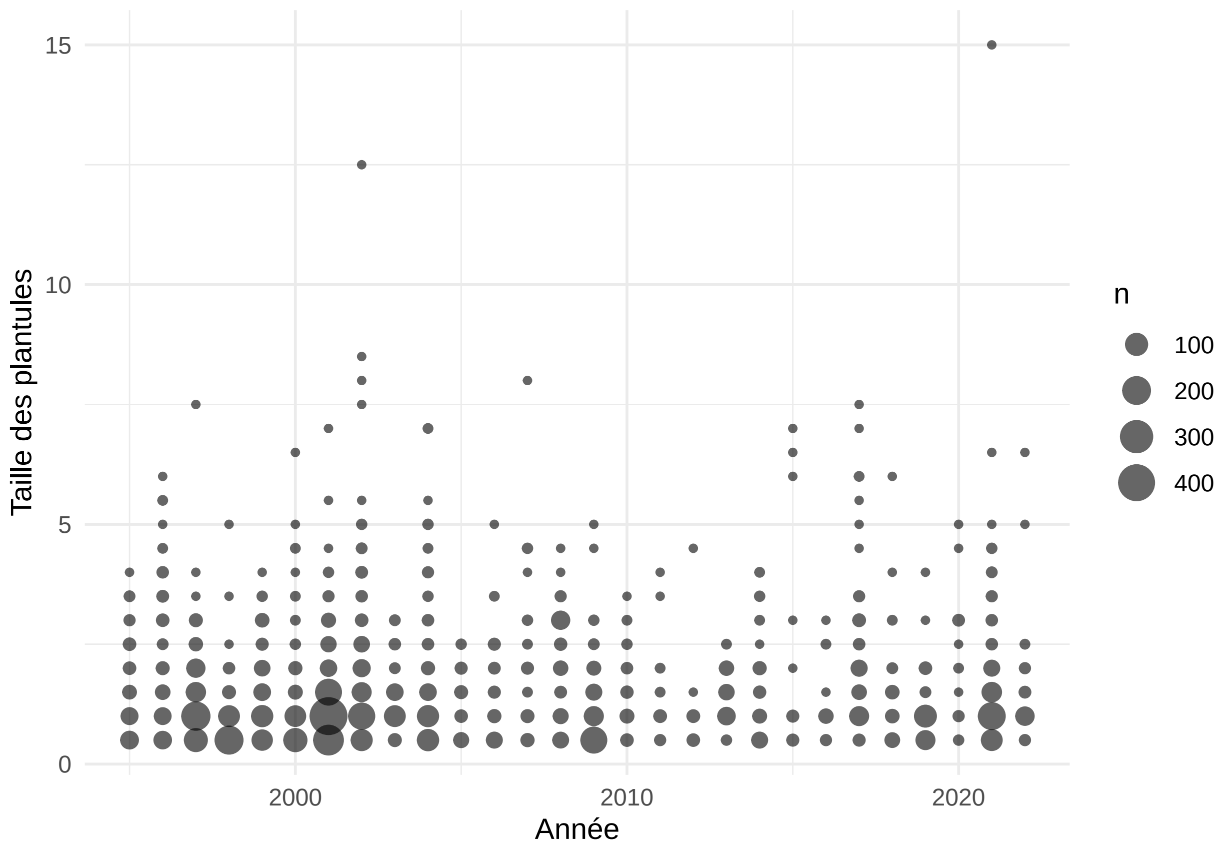
```r
spl
```

```
## 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
```

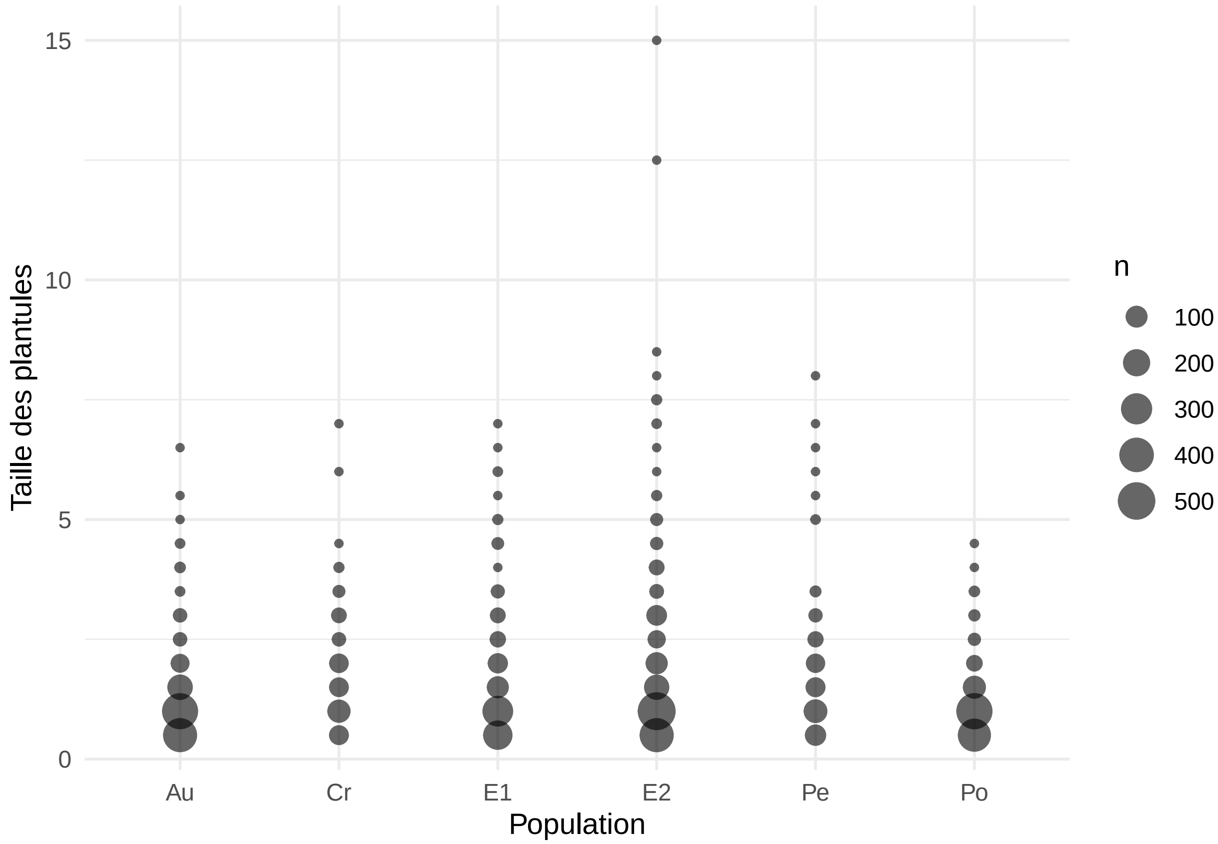
Taille des plantules

```r
plantule_data <- centauree_data[centauree_data$age0==1,]
```

```r
# Taille des plantules / année

plantule_data %>%
  ggplot(aes(x = year, y = Size0Mars)) +
  geom_count(alpha=0.6) +
  labs(x = "Année",
       y = "Taille des plantules") +
  theme_minimal()
```




```r
# Taille des plantules / population

plantule_data %>%
  ggplot(aes(x = Pop, y = Size0Mars)) +
  geom_count(alpha=0.6) +
  labs(x = "Population",
       y = "Taille des plantules") +
  theme_minimal()
```




```r
Pltglm1 <- fitme(Size0Mars ~ 1 + (1|year) + (1|Pop) + (1|Pop:year),
                 data=plantule_data,
                 family = Gamma(log))

Pltglm2 <- fitme(Size0Mars ~ 1 + (1|Pop) + (1|Pop:year),
                 data=plantule_data,
                 family = Gamma(log))

Pltglm3 <- fitme(Size0Mars ~ 1 + (1|Pop:year),
                 data=plantule_data,
                 family = Gamma(log))

Pltglm4 <- fitme(Size0Mars ~ 1 + (1|year) + (1|Pop:year),
                 data=plantule_data,
                 family = Gamma(log))

Pltglm5 <- fitme(Size0Mars ~ 1 + (1|year) + (1|Pop),
                 data=plantule_data,
                 family = Gamma(log))
```

```r
n <- length(plantule_data$Nrw)
extractAIC(Pltglm1) ; extractBIC(Pltglm1, n)
```

```
##      edf      AIC
##    1.000 8442.641
```

```
## [1] 8449.171
```

```r
extractAIC(Pltglm2) ; extractBIC(Pltglm2, n)
```

```
##      edf      AIC
##    1.000 8448.128

## [1] 8454.658
```

```r
extractAIC(Pltglm3) ; extractBIC(Pltglm3, n)
```

```
##      edf      AIC
##    1.000 8458.154

## [1] 8464.685
```

```r
extractAIC(Pltglm4) ; extractBIC(Pltglm4, n)
```

```
##      edf      AIC
##    1.000 8457.579

## [1] 8464.109
```

```r
extractAIC(Pltglm5) ; extractBIC(Pltglm5, n)
```

```
##      edf      AIC
##    1.000 8841.777

## [1] 8848.308
```

```r
summary(Pltglm1)
```

```
## formula: Size0Mars ~ 1 + (1 | year) + (1 | Pop) + (1 | Pop:year)
## Estimation of lambda and phi by ML (P_v approximation of logL).
## Estimation of fixed effects by ML (P_v approximation of logL).
## family: Gamma( link = log )
##  ------------ Fixed effects (beta) ------------
##             Estimate Cond. SE t-value
## (Intercept)   0.2132  0.07717   2.763
##  --------------- Random effects ---------------
## Family: gaussian( link = identity )
##            --- Variance parameters ('lambda'):
## lambda = var(u) for u ~ Gaussian;
##    year  :  0.02216
##    Pop  :  0.02626
##    Pop:year  :  0.07964
##              --- Coefficients for log(lambda):
##     Group        Term Estimate Cond.SE
##      year (Intercept)    -3.81  0.3809
```

```
##       Pop (Intercept)    -3.64  0.6351
##  Pop:year (Intercept)    -2.53  0.1444
## # of obs: 5065; # of groups: year, 28; Pop, 6; Pop:year, 147
##  --- Residual variation ( var = phi * mu^2 )  --
## Coefficients for log(phi)  ~ 1  :
##             Estimate Cond. SE
## (Intercept)    -1.34  0.01967
## Estimate of phi:  0.2619
##  ------------- Likelihood values  -------------
##                        logLik
## logL      (P_v(h)): -4216.321
```

```r
summary(Pltglm2)
```

```
## formula: SizeOMars ~ 1 + (1 | Pop) + (1 | Pop:year)
## Estimation of lambda and phi by ML (P_v approximation of logL).
## Estimation of fixed effects by ML (P_v approximation of logL).
## family: Gamma( link = log )
##  ------------ Fixed effects (beta) ------------
##             Estimate Cond. SE t-value
## (Intercept)   0.2024  0.06951   2.912
##  --------------- Random effects ---------------
## Family: gaussian( link = identity )
##            --- Variance parameters ('lambda'):
## lambda = var(u) for u ~ Gaussian;
##    Pop  :  0.02327
##    Pop:year  :  0.1083
##              --- Coefficients for log(lambda):
##     Group        Term Estimate Cond.SE
##       Pop (Intercept)   -3.761  0.6465
##  Pop:year (Intercept)   -2.223  0.1332
## # of obs: 5065; # of groups: Pop, 6; Pop:year, 147
##  --- Residual variation ( var = phi * mu^2 )  --
## Coefficients for log(phi)  ~ 1  :
##             Estimate Cond. SE
## (Intercept)   -1.341  0.01967
## Estimate of phi:  0.2616
##  ------------- Likelihood values  -------------
##                        logLik
## logL      (P_v(h)): -4220.064
```

```r
summary(Pltglm3)
```

```
## formula: SizeOMars ~ 1 + (1 | Pop:year)
## Estimation of lambda and phi by ML (P_v approximation of logL).
## Estimation of fixed effects by ML (P_v approximation of logL).
## family: Gamma( link = log )
##  ------------ Fixed effects (beta) ------------
##             Estimate Cond. SE t-value
## (Intercept)   0.1878  0.03325   5.648
##  --------------- Random effects ---------------
## Family: gaussian( link = identity )
##            --- Variance parameters ('lambda'):
```

```
## lambda = var(u) for u ~ Gaussian;
##    Pop:year  :  0.1333
##              --- Coefficients for log(lambda):
##     Group        Term Estimate Cond.SE
##  Pop:year (Intercept)   -2.015  0.1289
## # of obs: 5065; # of groups: Pop:year, 147
##  --- Residual variation ( var = phi * mu^2 )  --
## Coefficients for log(phi)  ~ 1  :
##             Estimate Cond. SE
## (Intercept)   -1.342  0.01968
## Estimate of phi:  0.2614
##  ------------- Likelihood values  -------------
##                         logLik
## logL      (P_v(h)): -4226.077
```

```r
summary(Pltglm4)
```

```
## formula: Size0Mars ~ 1 + (1 | year) + (1 | Pop:year)
## Estimation of lambda and phi by ML (P_v approximation of logL).
## Estimation of fixed effects by ML (P_v approximation of logL).
## family: Gamma( link = log )
##  ------------ Fixed effects (beta) ------------
##             Estimate Cond. SE t-value
## (Intercept)   0.1914  0.03913    4.89
##  --------------- Random effects ---------------
## Family: gaussian( link = identity )
##            --- Variance parameters ('lambda'):
## lambda = var(u) for u ~ Gaussian;
##    year  :  0.01472
##    Pop:year  :  0.1158
##              --- Coefficients for log(lambda):
##     Group        Term Estimate Cond.SE
##      year (Intercept)   -4.219  0.4567
##  Pop:year (Intercept)   -2.156  0.1348
## # of obs: 5065; # of groups: year, 28; Pop:year, 147
##  --- Residual variation ( var = phi * mu^2 )  --
## Coefficients for log(phi)  ~ 1  :
##             Estimate Cond. SE
## (Intercept)   -1.341  0.01968
## Estimate of phi:  0.2616
##  ------------- Likelihood values  -------------
##                         logLik
## logL      (P_v(h)): -4224.79
```

```r
summary(Pltglm5)
```

```
## formula: Size0Mars ~ 1 + (1 | year) + (1 | Pop)
## Estimation of lambda and phi by ML (P_v approximation of logL).
## Estimation of fixed effects by ML (P_v approximation of logL).
## family: Gamma( link = log )
##  ------------ Fixed effects (beta) ------------
##             Estimate Cond. SE t-value
## (Intercept)   0.2642   0.0892   2.962
```

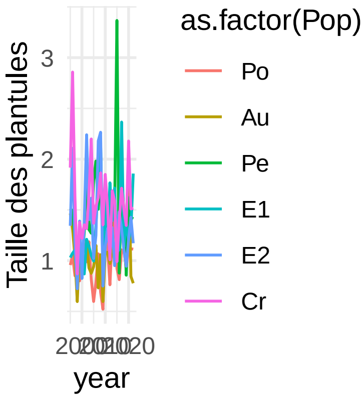
```
##  --------------- Random effects ---------------
## Family: gaussian( link = identity )
##            --- Variance parameters ('lambda'):
## lambda = var(u) for u ~ Gaussian;
##    year  :  0.05123
##    Pop  :  0.03582
##              --- Coefficients for log(lambda):
##  Group        Term Estimate Cond.SE
##   year (Intercept)   -2.971  0.2819
##    Pop (Intercept)   -3.329  0.5933
## # of obs: 5065; # of groups: year, 28; Pop, 6
##  --- Residual variation ( var = phi * mu^2 )  --
## Coefficients for log(phi)  ~ 1  :
##             Estimate Cond. SE
## (Intercept)   -1.231  0.01947
## Estimate of phi:  0.292
##  ------------- Likelihood values  -------------
##                         logLik
## logL      (P_v(h)): -4416.889
```

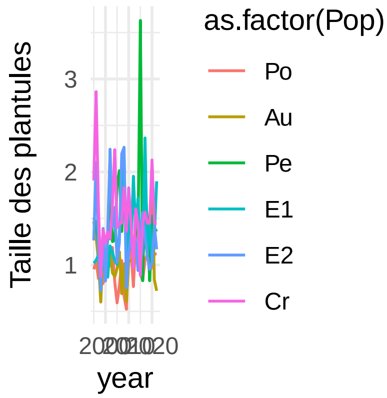
```
Pltpredict1 <- predict(Pltglm1, newdata = fake_data)[,1]
Pltpredict2 <- predict(Pltglm2, newdata = fake_data)[,1]
Pltpredict3 <- predict(Pltglm3, newdata = fake_data)[,1]
Pltpredict4 <- predict(Pltglm4, newdata = fake_data)[,1]
Pltpredict5 <- predict(Pltglm5, newdata = fake_data)[,1]
```

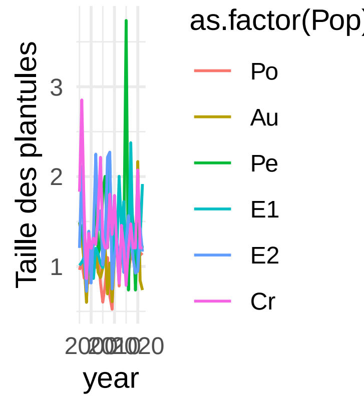
```
plot_plantule <- function(data = fake_data, prediction, var, fact) {
  data %>%
    mutate(plt_predi = prediction) %>%
    ggplot(aes(x = .data[[var]], y = plt_predi)) +
    geom_line(aes(color = as.factor(.data[[fact]]))) +
    labs(y="Taille des plantules")+
    theme_minimal()
}

plot_plantule1 <- 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="Taille des plantules")+
    theme_minimal()
}

plot_plantule2 <- function(data = fake_data, prediction) {
  data %>%
    mutate(plt_predi = prediction) %>%
    ggplot(aes(x = plt_predi)) +
    geom_histogram() +
    labs(x="Taille des plantules")+
    theme_minimal()
}
```

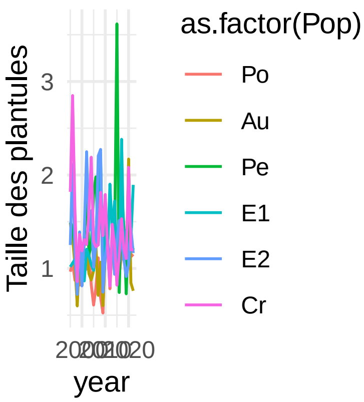
Taille des plantules en fonction de l'année

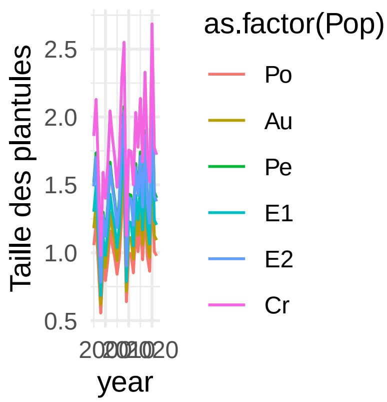
var <- "year" fact <- "Pop"





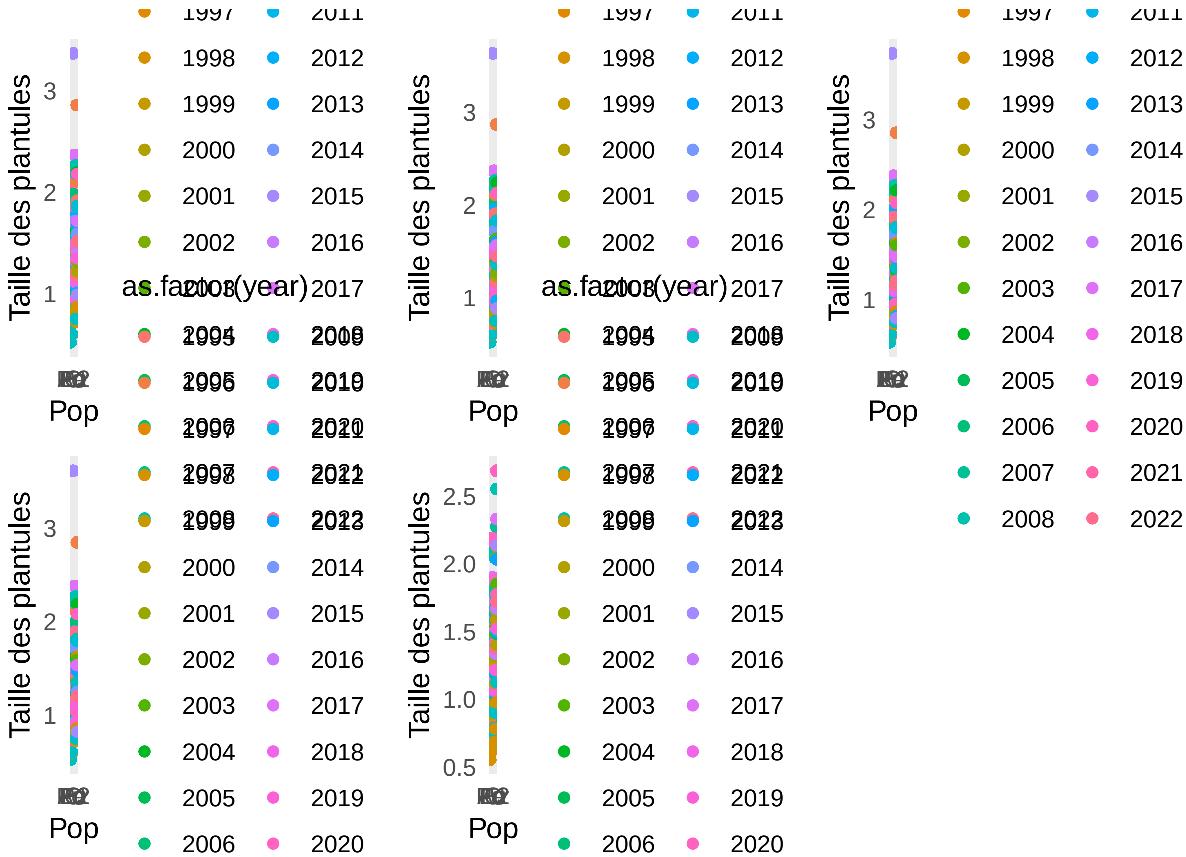






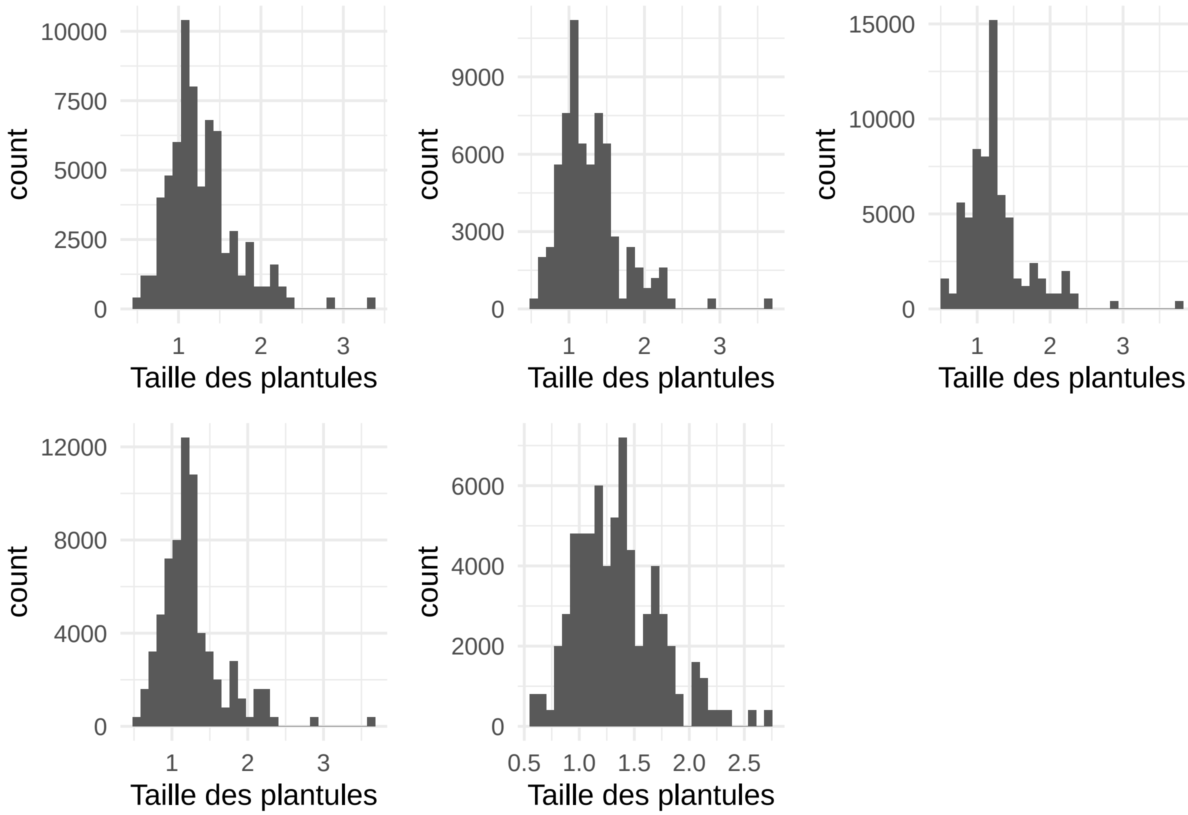
Taille des plantules en fonction de la population

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



\## Densité de taille de plantule

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```


count
Taille des plantules
count
Taille des plantules
count
Taille des plantules
count
Taille des plantules
count
Taille des plantules