

Mise en forme des données issus du fichier `piege0.xlsx`

Objectif

Ce document présente la mise en forme des données brutes du fichier `piege0.xlsx`. L'objectif est d'obtenir pour une parcelle donnée (bloc, sous-bloc) et pour chaque date :

- le nombre de larves s'éjectant de l'arbre ;
- le nombre d'inflorescences vivantes ;
- le nombre d'inflorescences mortes ;
- le rapport larves / inflorescences vivantes.

Les variables des données brutes utilisées sont :

- `bloc` : le bloc dans lequel l'observation a été effectuée ;
- `trait` : la modalité de couverture du sol ;
- `arbre` : l'identifiant de l'arbre sur lequel l'observation a été effectuée ;
- `piege` : l'identifiant du piège relatif aux observations ;
- `date` : la date de l'observation ;
- `nb.larves` : le nombre de larves présentes dans le piège ;
- `nb.inflos.piege` : le nombre d'inflorescences présentes au-dessus du piège ;
- `nb.inflos` : le nombre d'inflorescences sur l'arbre sur lequel l'observation a été effectuée ;
- `nb.inflos.mortes` : le nombre d'inflorescences mortes sur l'arbre sur lequel l'observation a été effectuée.

Si l'on suppose que pour une parcelle considérée et une date fixée il y a n arbres et n_p arbres observés (*i.e* qui ont un piège en dessous), alors le nombre de larves s'éjectant du manguier est

$$\frac{n}{n_p} \left(\sum_{j=1}^{n_p} \text{nb.larves}_j \times \frac{\text{nb.inflos}_j}{\text{nb.inflos.piege}_j} \right),$$

où `nb.larvesj`, `nb.inflosj` et `nb.inflos.piegej` correspondent respectivement au nombres de larves contenus dans les pièges, au nombre d'inflorescences de l'arbre et au nombre d'inflorescences au-dessus des pièges pour le $j^{\text{ème}}$ arbre de l'échantillon.

De manière similaire on définit le nombre d'inflorescences vivantes, pour une parcelle et une date donnée, par

$$\frac{n}{n_p} \left(\sum_{j=1}^{n_p} \text{nb.inflos}_j \right),$$

et le nombre d'inflorescences mortes par

$$\frac{n}{n_p} \left(\sum_{j=1}^{n_p} \text{nb.inflos.mortes}_j \right).$$

Réalisation

Chargement des librairies

```
library(tidyverse) ## Pour nettoyer les données
library(readxl)    ## Pour lire le fichier excel
library(lubridate)  ## Pour gérer les dates
library(magrittr)
```

Importation du fichier

On importe le fichier dans R. Les arbres 19 et 36 du bloc 2 ont été enlevés car, contrairement aux autres, leurs relevés n'ont commencé qu'en septembre. Si cela entraîne une perte d'information, cela a l'avantage de permettre d'avoir le même nombre d'arbre par bloc/sous-bloc (à savoir 30 par bloc, 10 pour chaque modalité de couverture du sol).

```
piege <- read_excel("/home/bastien/Stage/Moi/piege0.xlsx", sheet = 3)

## suppression arbre 19
piege <- piege[-which(piege$bloc==2 & piege$trait=='enh.ras' & piege$arbre==19),]
## suppression arbre 36
piege <- piege[-which(piege$bloc==2 & piege$trait=='enh.ras' & piege$arbre==36),]
```

Arrangement des données

On attribue le type de données à chacune des variables, on supprime celles qui ne sont pas nécessaires et enfin on ordonne les individus par date puis par modalité de couverture du sol et enfin par l'identifiant de l'arbre.

```
piege %<>% mutate_at(vars(starts_with("nb.")), list(as.numeric))
piege %<>% mutate_at(c("piege", "trait"), list(as.factor))
piege %<>% mutate_at("date", list(as.Date))
piege %<>% select(-c(observations, nb.larves.estima, larve))
piege %<>% arrange(arbre) %>% arrange(trait) %>% arrange(date)
```

Premier bloc

Par souci de simplicité, on traite d'abord les données du premier bloc

```
piege2 <- piege %>% filter(bloc==2)
piege %<>% filter(bloc==1)
```

Correction d'erreurs et gestion des valeurs manquantes

Il y a deux individus pour lesquels le piège n'est pas le bon (B au lieu de A). On corrige.

```
bad_ind <- which(piege$piege=="B")[which(which(piege$piege=="B") %% 2 == 1)]
piege[bad_ind,]$piege <- "A"
```

Certaines valeurs manquantes sont présentes. Lorsque le nombre d'inflorescences au-dessus du piège est nul et que le nombre de larves dans le piège est manquant, on le fixe à 0. Dans les autres cas, la valeur manquante a été remplacée par la moyenne des autres valeurs dans le même bloc à la même date et avec la même modalité de couverture du sol.

```
piege[which(is.na(piege$nb.larves) & piege$nb.inflo.piege == 0), ]$nb.larves <- 0
piege[which(is.na(piege$nb.inflo.piege)), ]$nb.inflo.piege <- 2.25
piege[which(is.na(piege$nb.inflo)), ]$nb.inflo <- 26.55
piege[which(is.na(piege$nb.larves) & piege$trait == "bache"), ]$nb.larves <- 2.57
piege[which(is.na(piege$nb.larves) & piege$trait == "enh.haut"), ]$nb.larves <- 1.5
piege[which(is.na(piege$nb.larves) & piege$trait == "enh.ras"), ]$nb.larves <- 2.63
```

Gestion des pièges

Pour gérer les pièges, on sépare en deux le jeu de données. La première partie correspond aux observations réalisées avant le 17 août 2017 (soit avant la pose du deuxième piège) et la seconde correspond aux observations effectuées après ladite date.

```
piegeA <- piege %>% filter(date < "2017-08-17")
piegeAB <- piege %>% filter(date > "2017-08-17")
```

Fonction

La fonction suivante renvoie un fichier csv contenant les informations voulues pour le bloc 1.

```
estimation_piege_bloc1 <- function(modalite = NULL){
  ## Renvoie un fichier .csv contenant les estimations du nombre de larves dans
  ## la parcelle, le nombre d'inflorescences vivante dans la parcelle, le nombre
  ## d'inflorescences mortes et le rapport de larves/inflo_vivantes. Les estimations
  ## sont données pour chaque date.

  ## Argument : modalité prend "bache", "enh.ras" ou "enh.haut".
  ## Par défaut NULL renvoie pour le bloc entier

  if (!is.null(modalite)){
    piegeA %<>% filter(trait==modalite)
    piegeAB %<>% filter(trait==modalite)
  }

  ### Initialisation data
  nb.arbre.bloc <- 153
  nb.arbre.bache <- 45
  nb.arbre.ras <- 49
  nb.arbre.haut <- 59
  larves <- rep(NA, 20)
  inflos_vivantes <- rep(NA, 20)
  inflos_mortes <- rep(NA, 20)
  larves_inflos <- rep(NA, 20)

  ### Avant 2017-08-17 (un seul piège)

  ## Création d'une variable calculant pour chaque observation
  ## nb.larves * nb.inflo / nb.inflo.piege
  ## qui est égale à 0 si nb.inflo.piege = 0
  piegeA %<>% mutate(larves.arbre = nb.larves * nb.inflo / nb.inflo.piege)
  piegeA$larves.arbre[which(is.na(piegeA$larves.arbre) |
                             is.infinite(piegeA$larves.arbre))] <- 0

  ## Dates de relevés avant la pose du second piège
  dateA <- unique(piegeA$date)

  ## Calcule pour chaque date les valeurs voulues
  for (day in 1:length(dateA)){
    aux <- piegeA %>% filter(date==dateA[day])
    larves[day] <- sum(aux$larves.arbre)
    inflos_vivantes[day] <- sum(aux$nb.inflo)
    inflos_mortes[day] <- sum(aux$nb.inflo.morte)
  }

  ### Après 2017-08-17 (deux pièges)

  ## Dates de relevés après la pose du second piège
```

```

dateAB <- unique(piegeAB$date)

## Séparation des données selon le piège
piegeAB_A <- piegeAB %>% filter(piege=="A")
piegeAB_B <- piegeAB %>% filter(piege=="B")

## Addition des nb.larves et nb.inflos.piege des pièges A et B pour chaque arbre
piegeAB_A %<>% bind_cols(nb.larves_B = piegeAB_B$nb.larves,
                        nb.inflo.piege_B = piegeAB_B$nb.inflo.piege)
piegeAB_A %<>% mutate(nb.larvesAB = nb.larves+nb.larves_B,
                     nb.inflo.piegeAB = nb.inflo.piege+nb.inflo.piege_B) %>%
  mutate(larves.arbre = nb.larvesAB * nb.inflo / nb.inflo.piegeAB)
piegeAB_A$larves.arbre[which(is.na(piegeAB_A$larves.arbre) |
                             is.infinite(piegeAB_A$larves.arbre))] <- 0

## Calcule pour chaque date les valeurs voulues
for (day in 1:length(dateAB)){
  aux <- piegeAB_A %>% filter(date==dateAB[day])

  larves[day+length(dateA)] <- sum(aux$larves.arbre)
  inflos_vivantes[day+length(dateA)] <- sum(aux$nb.inflo)
  inflos_mortes[day+length(dateA)] <- sum(aux$nb.inflo.morte)
}

### Calcul du coeficient n / n_p en fonction de la modalité de couverture du sol
arbre.piege <- dim(piegeA %>% filter(date=="2017-07-18"))[1]
if (is.null(modalite)){
  coef <- nb.arbre.bloc/arbre.piege
} else {
  if (modalite == "bache")
    coef <- nb.arbre.bache/arbre.piege
  if (modalite == "enh.ras")
    coef <- nb.arbre.ras/arbre.piege
  if (modalite == "enh.haut")
    coef <- nb.arbre.haut/arbre.piege
}

### Résultats
larves <- coef * larves
inflos_vivantes <- coef * inflos_vivantes
inflos_mortes <- coef * inflos_mortes
larves_inflos <- larves / inflos_vivantes

df <- as_tibble(cbind(date = c(dateA, dateAB), larves = larves,
                      inflos_vivantes = inflos_vivantes, inflos_mortes,
                      larves_inflos = larves_inflos))
df$date <- as_date(df$date)

write.csv2(df, paste0("Bloc1_",modalite,".csv"))
return(df)
}

```

Second bloc

Le raisonnement est entièrement analogue à celui du bloc 1.

```
bad_ind <- which(piege2$piege=="B")[which(which(piege2$piege=="B") %% 2 == 1)]
piege2[bad_ind,]$piege <- "A"

piege2[which(is.na(piege2$nb.inflo.piege)), ]$nb.inflo.piege <- 0.84
piege2[which(is.na(piege2$nb.larves) & piege2$nb.inflo.piege == 0), ]$nb.larves <- 0
piege2[which(is.na(piege2$nb.larves)), ]$nb.larves <- 0

piege2A <- piege2 %>% filter(date < "2017-08-17")
piege2AB <- piege2 %>% filter(date > "2017-08-17")

estimation_piege_bloc2 <- function(modalite = NULL){
  ## Renvoie un fichier .csv contenant les estimations du nombre de larves dans
  ## la parcelle, le nombre d'inflorescences vivante dans la parcelle, le nombre
  ## d'inflorescences mortes et le rapport de larves/inflo_vivantes. Les estimations
  ## sont données pour chaque date.

  ## Argument : modalité prend "bache", "enh.ras" ou "enh.haut".
  ## Par défaut NULL renvoie pour le bloc entier

  if (!is.null(modalite)){
    piege2A %<>% filter(trait==modalite)
    piege2AB %<>% filter(trait==modalite)
  }

  ### Initialisation data
  nb.arbre.bloc <- 150
  nb.arbre.bache <- 51
  nb.arbre.ras <- 53
  nb.arbre.haut <- 46
  larves <- rep(NA, 20)
  inflos_vivantes <- rep(NA, 20)
  inflos_mortes <- rep(NA, 20)
  larves_inflos <- rep(NA, 20)

  ### Avant 2017-08-17 (un seul piège)

  ## Création d'une variable calculant pour chaque observation
  ## nb.larves * nb.inflo / nb.inflo.piege
  ## qui est égale à 0 si nb.inflo.piege = 0
  piege2A %<>% mutate(larves.arbre = nb.larves * nb.inflo / nb.inflo.piege)
  piege2A$larves.arbre[which(is.na(piege2A$larves.arbre) |
                             is.infinite(piege2A$larves.arbre))] <- 0

  ## Dates de relevés avant la pose du second piège
  dateA <- unique(piege2A$date)

  ## Calcule pour chaque date les valeurs voulues
  for (day in 1:length(dateA)){
    aux <- piege2A %>% filter(date==dateA[day])
    larves[day] <- sum(aux$larves.arbre)
    inflos_vivantes[day] <- sum(aux$nb.inflo)
```

```

    inflos_mortes[day] <- sum(aux$nb.inflo.morte)
  }

### Après 2017-08-17 (deux pièges)

## Dates de relevés après la pose du second piège
dateAB <- unique(piege2AB$date)

## Séparation des données selon le piège
piegeAB_A <- piege2AB %>% filter(piege=="A")
piegeAB_B <- piege2AB %>% filter(piege=="B")

## Addition des nb.larves et nb.inflos.piege des pièges A et B pour chaque arbre
piegeAB_A %<>% bind_cols(nb.larves_B = piegeAB_B$nb.larves,
                        nb.inflo.piege_B = piegeAB_B$nb.inflo.piege)
piegeAB_A %<>% mutate(nb.larvesAB = nb.larves+nb.larves_B,
                     nb.inflo.piegeAB = nb.inflo.piege+nb.inflo.piege_B) %>%
  mutate(larves.arbre = nb.larvesAB * nb.inflo / nb.inflo.piegeAB)
piegeAB_A$larves.arbre[which(is.na(piegeAB_A$larves.arbre) |
                             is.infinite(piegeAB_A$larves.arbre))] <- 0

## Calcule pour chaque date les valeurs voulues
for (day in 1:length(dateAB)){
  aux <- piegeAB_A %>% filter(date==dateAB[day])

  larves[day+length(dateA)] <- sum(aux$larves.arbre)
  inflos_vivantes[day+length(dateA)] <- sum(aux$nb.inflo)
  inflos_mortes[day+length(dateA)] <- sum(aux$nb.inflo.morte)
}

### Calcul du coeficient n / n_p en fonction de la modalité de couverture du sol
arbre.piege <- dim(piege2A %>% filter(date=="2017-07-18"))[1]
if (is.null(modalite)){
  coef <- nb.arbre.bloc/arbre.piege
} else {
  if (modalite == "bache")
    coef <- nb.arbre.bache/arbre.piege
  if (modalite == "enh.ras")
    coef <- nb.arbre.ras/arbre.piege
  if (modalite == "enh.haut")
    coef <- nb.arbre.haut/arbre.piege
}

### Résultats
larves <- coef * larves
inflos_vivantes <- coef * inflos_vivantes
inflos_mortes <- coef * inflos_mortes
larves_inflos <- larves / inflos_vivantes

df <- as_tibble(cbind(date = c(dateA, dateAB), larves = larves,
                       inflos_vivantes = inflos_vivantes,
                       inflos_mortes, larves_inflos = larves_inflos))
df$date <- as_date(df$date)

```

```
write.csv2(df, paste0("Bloc_",modalite,".csv"))
return(df)
}
```

Différence avec Laurie

Il y a quelques différences avec le travail de Laurie :

- la suppression de l'arbre 36 du bloc 2 ;
- les données manquantes n'ont pas été supprimées mais remplacées par la moyenne des individus ayant les mêmes caractéristiques. Les données manquantes étant peu nombreuses, cela ne change pas grand chose ;
- les dates sélectionnées sont ici les dates de relevés ;
- des différences de valeurs pour certaines modalités/variables dans le bloc 2.