


TD5 ÉVOLUTION DÉPENDANTE DU TEMPS

Le TD est en lien avec le **problème 2** de la matière « Introduction à la modélisation en SVT ». Les questions avec  sont notées uniquement pour les étudiants de la licence de mathématiques et d'informatique.

Un champ de pommes de terres est infesté de doryphores et pour en contrôler la population on alterne avec la culture de céréales. La population de doryphores à l'année t est donnée en fonction de celle de l'année précédente par :

$$n(t) = r(t) \times n(t-1) \quad (6)$$

avec

$$r(t) = s_a + f s_{ol}(1 - \varepsilon(t)) \text{ et } \varepsilon(t) = \begin{cases} 0 & \text{pour une culture de pommes de terre} \\ \varepsilon_p & \text{pour une culture de céréales} \end{cases} \quad (7)$$

Si la période est n , cela signifie que c'est à la $n^{\text{ième}}$ année que l'on ne plante pas de pommes de terre. Les questions suivantes permettent d'étudier l'impact de l'alternance des cultures sur l'évolution de la population de doryphores.

Exercice 5.1 Fonction efficacité

5.1.1 Implémenter la fonction $\varepsilon(t)$ qui représente l'efficacité en fonction des années et qui prend aussi en compte la périodicité de l'alternance des cultures. Cette fonction se nomme `fun_efficacite` et prend en arguments :

➡ le temps `annee` et

➡ la périodicité `periode`.

Cette fonction `fun_efficacite` ➡ retourne l'efficacité de l'année : 0 ou $\varepsilon_p = 0.99$. Cette valeur indique le taux de réduction des larves s'il s'agit d'une culture de céréales. Si l'année est égale à 0, la valeur retournée doit être de `NA`.

5.1.2 Calculer et afficher la valeur de l'efficacité pour des années allant de 1 à 4 sachant que il y a des céréales les années paires et des pommes de terre sinon.

Correction 5.1

```
1 fun_efficacite <- function(annee, periode) {
2   e_p <- 0.99 # Réduction des larves en absence de pommes de terre
3   if (annee == 0){
4     resultat = NA
5   } else if (annee %% periode == 0) {
6     resultat <- e_p # Année à céréales
7   } else {
8     resultat <- 0 # Année à pommes de terre
```

```
9      }  
10     return(resultat)  
11   }  
12  
13   annee_1 <- fun_efficacite(annee = 1, periode = 2)  
14   annee_2 <- fun_efficacite(annee = 2, periode = 2)  
15   annee_3 <- fun_efficacite(annee = 3, periode = 2)  
16   annee_4 <- fun_efficacite(annee = 4, periode = 2)
```

Exercice 5.2 Calcul des valeurs du modèle à processus dépendant du temps

Considérons les valeurs des paramètres suivantes :


- $s_a = 0.2$, la survie des adultes ;
- $f = 800$, la fertilité ;
- $s_o = 0.02$, survie des oeufs ;
- $s_l = 0.1$ survie des larves.

L'objectif est de reproduire le tableau de données obtenu lors du TD de modelisation à la question 4(a).

5.2.1 Affecter les valeurs aux variables correspondantes.

5.2.2 Construire le tibble `donnee_1` suivant en utilisant la fonction `rep`

annee	efficacite
0	NA
1	0
2	0.99
3	0
4	0.99
5	0
6	0.99
7	0
8	0.99

5.2.3  Construire le tibble `donnee_1` en faisant appel à la fonction `fun_efficacite`. Indiquer en commentaire la différence entre `map` et `map_dbl`.

5.2.4 Ajouter une variable `raison` au tibble `donnee_1` en calculant la raison du modèle (7) et affecter le tibble obtenu à la variable `donnee_2`.

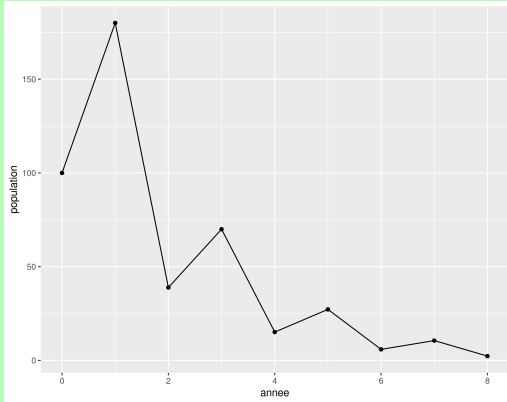
5.2.5 Ajouter une variable `population` au tibble `donnee_2` qui contient 100 pour la population initiale et des `NA` pour les autres années, affecter le tibble obtenu à la variable `donnee_3`.

5.2.6 Parcourir le tibble `donnee_3` pour remplir les cases de la variable `population` en respectant l'équation (6) décrivant le modèle.

5.2.7 Tracer, avec lignes et points, l'évolution de la population en fonction des années.

Correction 5.2

```
1 library(tidyverse)
2
3 # Q 5.2.1
4 f <- 800
5 s_a <- 0.2
6 s_o <- 0.02
7 s_l <- 0.1
8
9 # Q 5.2.2
10 donnee_1 <- tibble(annee = 0:8,
11   efficacite = c(NA, rep(c(0, 0.99),4)))
12
13 # Q 5.2.3
14 donnee_1 <- tibble(annee = 0:8) %>%
15   mutate(efficacite = map_dbl(annee, fun_efficacite, periode = 2))
16
17 # alternative non recommandée
18 # rowwise() allows you to compute on a data frame a row-at-a-time.
19 # This is most useful when a vectorised function doesn't exist.
20 # donnee_1 <- tibble(annee = 0:8) %>%
21 #   rowwise() %>%
22 #   mutate(efficacite = fun_efficacite(annee, 2)) %>%
23 #   ungroup()
24
25 # Avec map, c'est une liste qu'on obtient
26 # > glimpse(donnee_1)
27 # Rows: 9
28 # Columns: 2
29 # $ annee      <int> 0, 1, 2, 3, 4, 5, 6, 7, 8
30 # $ efficacite <list> NA, 0, 0.99, 0, 0.99, 0, 0.99, 0, 0.99
31
32 # Q 5.2.4
33 donnee_2 <- donnee_1 %>%
34   mutate(raison = s_a + f * s_o * s_l * (1 - efficacite))
35
36 # Q 5.2.5
37 donnee_3 <- donnee_2 %>%
38   mutate(population = c(100, rep(NA, 8)))
39
40 # Q 5.2.6
41 for (ligne in 2:9){
42   donnee_3[ligne, 4] = donnee_3[ligne-1, 4] * donnee_3[ligne, 3]
43 }
44
45 # Q 5.2.7
46 ggplot(donnee_3) +
47   aes(x = annee,
48     y = population) +
49   geom_point() +
50   geom_line()
51
```

52 `ggsave("tibble.png")`

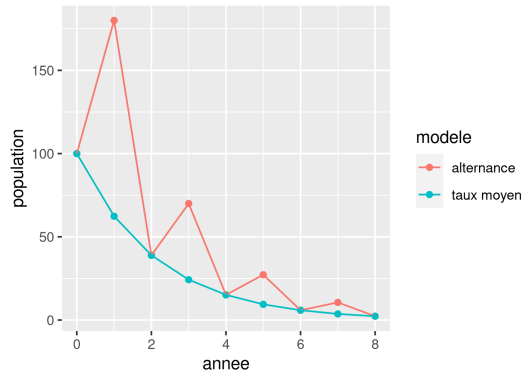
Exercice 5.3 Moyenne géométrique

À partir des données de l'exercice précédent, on calcule la moyenne géométrique des taux sur les 8 premières années avec la formule suivante :

$$\sqrt[8]{R(1) \times R(2) \times \dots \times R(8)} \quad (8)$$

- 5.3.1 En partant du tibble `donnee_3`, sélectionner les lignes des données n'ayant pas de `NA` et enregistrer ce tibble dans `donnee_4`
- 5.3.2 Calculer la valeur de l'équation (8) à partir de `donnee_4`. Pour cela, créer un tibble qui contient un résumé des données pour calculer la moyenne géométrique en utilisant la fonction `prod`. Extraire le contenu du tibble pour obtenir au final un nombre et affecter cette valeur numérique à la variable `raison_huit_ans`. Pour avoir la totalité des points il faut trouver le bon ordre d'utilisation des fonctions suivantes et obtenir la bonne valeur numérique :
- `prod`
 - `pull`
 - `summarise`
- 5.3.3 Créer un vecteur `vecteur_population` de 9 nombres avec `numeric`. Calculer itérativement l'évolution de la population avec la raison calculée à la question précédente et stocker cette évolution dans le `vecteur_population`.
- 5.3.4 À partir des éléments obtenus précédemment, créer un tibble avec trois *variables* :
- `annee`
 - `raison`

— population

affecter ce tibble à la variable `donnee_5`5.3.5 Combiner les tibbles `donnee_4` et `donnee_5` et produire la figure suivante

Correction 5.3

```
1 # Q 5.3.1
2 donnee_4 <- donnee_3 %>%
3   filter(!is.na(raison))
4 # ou
5 donnee_4 <- drop_na(donnee_3)
6
7 # Q 5.3.2 puzzle
8 raison_huit_ans <- donnee_4 %>%
9   summarise(raison_moyenne = prod(raison)^(1/8)) %>%
10  pull(raison_moyenne)
11
12 # Q 5.3.3
13 vecteur_population = numeric(9)
14 vecteur_population[1] <- 100
15 for (annee in 2:9){
16   vecteur_population[annee] = vecteur_population[annee-1] * raison_huit_ans
17 }
18
19 # Q 5.3.4
20 donnee_5 <- tibble(annee = 0:8,
21   raison = rep(raison_huit_ans, 9),
22   population = vecteur_population)
23
24 # Q 5.3.5
25 donnee_4b <- mutate(donnee_3, modele = "alternance")
26 donnee_5b <- mutate(donnee_5, modele = "taux moyen")
27 donnee_plot <- bind_rows(donnee_4b, donnee_5b)
28
29 ggplot(data = donnee_plot) +
30   aes(x = annee,
```

```
31 y = population,  
32   color = modele) +  
33 geom_point() +  
34 geom_line()
```

