

TP avec le logiciel 
M1 BEE@Lyon
UE "Écologie et Évolution Quantitatives"
Modélisation en Épidémiologie

Isabelle AMAT & Christelle LOPES
isabelle.amat@univ-lyon1.fr
christelle.lopes@univ-lyon1.fr

15 janvier 2019

Table des matières

1	Introduction	2
2	Utiliser  avec le logiciel  Studio	2
3	Le modèle de Kermack & McKendrick	3
4	Le modèle <i>SIRS</i> densité-dépendant	6
5	La vaccination	12
6	Le théorème du moustique	13
6.1	Les variables et paramètres	13
6.2	Le système dynamique	14
6.3	Le taux de reproduction de base	14
6.4	Simulations : à vous de jouer!	14
6.5	Reparamétrisation	15
7	Vers d'autres modèles	16
8	Pour aller plus loin	16

1 Introduction

L'objectif de ce TP est de vous apprendre à simuler des modèles épidémiologiques simples à l'aide du logiciel **R** d'explorer plus avant les différents modèles présentés en cours. Cela nécessitera d'apprendre à utiliser **R** pour simuler des équations différentielles ordinaires (EDO). Dans cette optique, deux bibliothèques (*packages*) vous seront nécessaires :

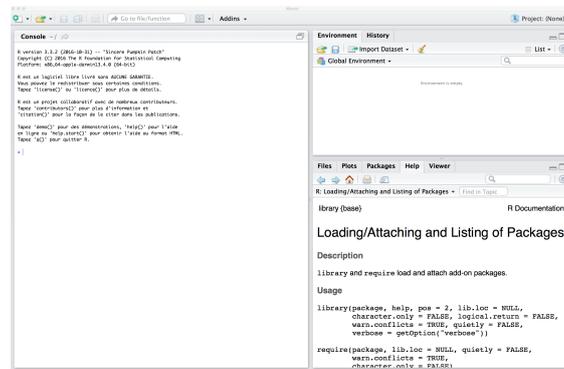
deSolve et en particulier la fonction `lsoda()` que l'on utilisera pour simuler des systèmes d'EDO lorsque l'on aura plus de deux variables ; par exemple les modèles *SIR* et *SIRS* ;

phaseR qui est exclusivement dédié à la simulation de systèmes dynamiques en dimensions 1 et 2 et que l'on utilisera pour représenter les portraits de phase et les chroniques.

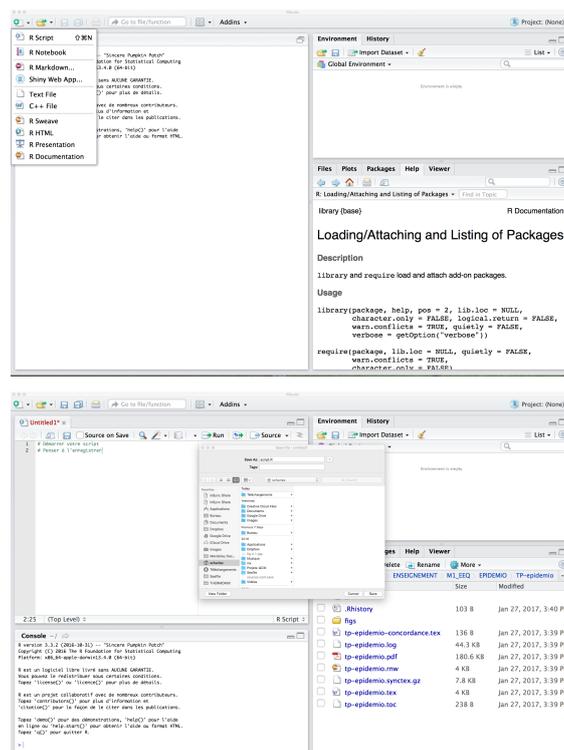
Ce TP sera aussi l'occasion d'apprendre quelques fonctionnalités graphiques de **R** comme les fonctions `curve()` et `points()`. L'ensemble des documents relatifs à ce TP ainsi que le diaporama du cours correspondant sont accessibles via <https://lc.cx/gezf>.

2 Utiliser **R** avec le logiciel **R Studio**

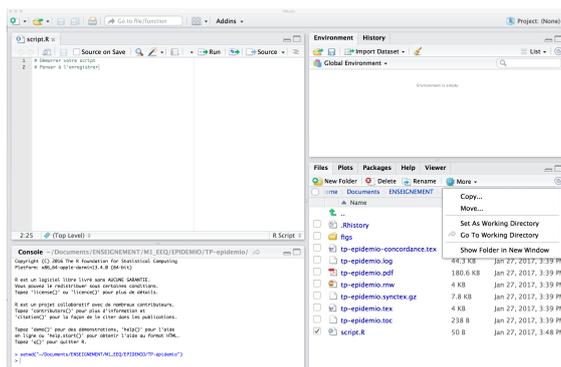
Pour utiliser **R**, lancer le logiciel **R Studio** depuis le menu **Applications** de votre poste de travail. Il s'agit d'un environnement de développement multiplateforme pour **R**, qui propose notamment la coloration syntaxique, bien pratique lorsque l'on écrit du code informatique.



Créer un nouveau fichier de type **script** dans lequel vous écrirez votre code R. Sauvegardez ce fichier dans un dossier de travail.



Déclarez ce dossier de travail comme l'environnement de travail de votre session .



Les bibliothèques `deSolve` et `phaseR` ne font pas partie de la distribution de base de . Il faut donc d'abord les installer dans son environnement de travail. Vous pouvez le faire en ligne de commande dans la partie Console, cadran inférieur gauche de  :

```
install.packages("deSolve")
install.packages("phaseR")
```

Il est également possible d'utiliser le menu du haut `Tools` puis `install packages...`, ou alors l'onglet `Packages` cadran inférieur droit de .

Une fois installées, les bibliothèques doivent être appelées pour pouvoir être utilisées dans la session en cours.

Retranscrivez les lignes de code ci-dessous dans votre script R puis faites tourner avec  :

```
library(deSolve)
library(phaseR)
```

En savoir plus sur  :

ftp://cran.r-project.org/pub/R/doc/contrib/Paradis-rdebut_fr.pdf

3 Le modèle de Kermack & McKendrick

On s'intéresse dans un premier temps au modèle de Kermack & McKendrick, c'est-à-dire au tout premier modèle *SIR* avec une force d'infection **densité-dépendante** :

$$\begin{cases} \frac{dS(t)}{dt} = -\beta \times I(t) \times S(t) \\ \frac{dI(t)}{dt} = \beta \times I(t) \times S(t) - \nu I(t) \\ \frac{dR(t)}{dt} = \nu I(t) \end{cases}$$

L'objectif de cette section est de simuler ce modèle pour des valeurs de paramètres qui correspondent à l'épidémie de grippe dans le pensionnat anglais pour garçons, telle que rapportée par le *British Medical Journal* en 1978. L'unité de temps est alors le jour.

1. Utilisez la fonction ci-dessous pour définir le modèle en fonction de ses paramètres.

L'objet `time` correspond à la variable t , `y` et `parameters` sont des objets de type vecteurs.

À quelles variables correspondent les objets `y[i]`, $i = 1, 2, 3$?

À quels paramètres correspondent les quantités `parameters[i]`, $i = 1, 2$?

```
modelKMK <- function(time,y,parameters){
  dy1 <- - parameters[1]*y[1]*y[2]
  dy2 <- parameters[1]*y[1]*y[2] - parameters[2]*y[2]
  dy3 <- parameters[2]*y[2]
  list(c(dy1,dy2,dy3))
}
```

2. Affectez les valeurs 0.00225 et 0.5 aux paramètres β et ν , respectivement.

```
beta <- 0.00225  
nu <- 0.5
```

3. Créez l'objet `temps` comme une suite de valeurs entre 0 et 20, régulièrement espacées de 0.1.

```
temps <- seq(0,20,by=0.1)
```

4. Créer un objet `init` contenant la condition initiale.

```
init <- c(762,1,0)
```

5. À l'aide de la fonction `lsoda()` (bibliothèque `deSolve`), simuler le modèle sur une période de 20 jours.

```
simuKMK <- lsoda(y=init, times=temps, func=modelKMK, parms=c(beta,nu))
```

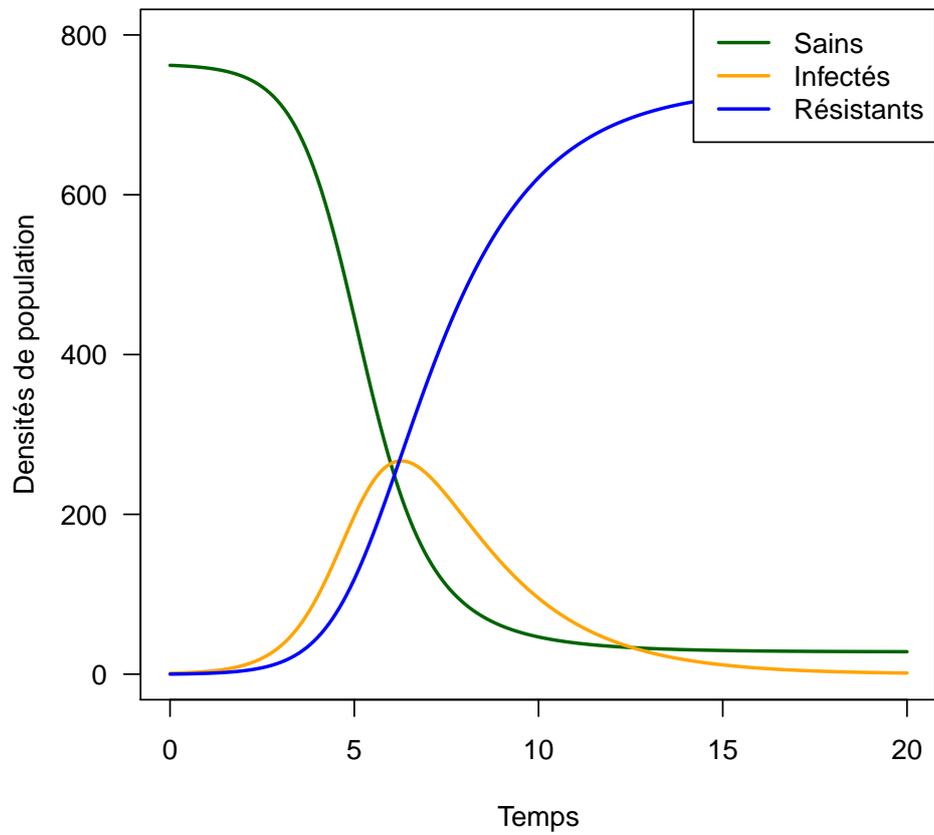
6. De quel type est l'objet `simuKMK`? De quelle(s) dimension(s) est-il? Justifiez.

7. Que représente l'objet `simuKMK[,1]`? A quel autre objet est-il identique?

8. Représentez sur un même graphique l'évolution au cours du temps des variables $S(t)$, $I(t)$ et $R(t)$. Pour cela, vous complétez la ligne de code ci-dessous pour obtenir le graphique ci-après. Essayez avec les fonctions `plot()` puis `lines()`.

Qu'est-ce qu'une fonction graphique de deuxième niveau?

```
plot(simuKMK[,1], simuKMK[,2])
```



9. Créez un objet `data` à partir du fichier `data-flu.txt` sur l'épidémie de grippe. À quoi sert l'option `header` qui peut être égale à `FALSE` ou `TRUE`?

```
data <- read.table("data-flu.txt")
```

10. Superposez ces données expérimentales au graphique précédent. Pour cela, utilisez la fonction `points()`.

11. Faites varier les paramètres β et ν . Qu'observez-vous ?

4 Le modèle *SIRS* densité-dépendant

On complexifie le modèle précédent pour tenir compte du fait qu'une partie des individus infectés a perdu son immunité pour redevenir sensible (Figure 1).

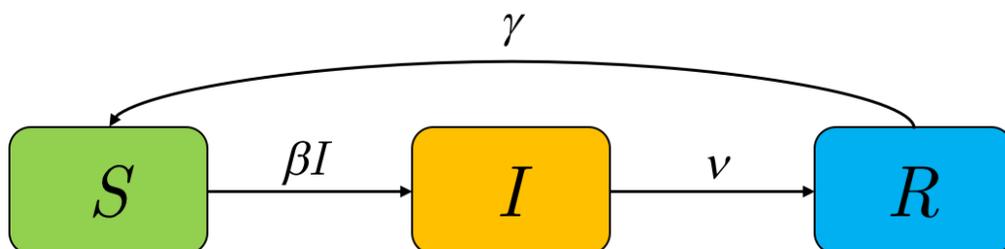


FIGURE 1 – Schéma de principe d'un modèle *SIRS*.

12. Ecrire le système qui régit la dynamique de ce système.

13. Montrez que l'on peut ré-écrire ce système en dimension 2 sous la forme :

$$\begin{cases} \frac{dS(t)}{dt} = -\beta S(t) I(t) + \gamma (N - S(t) - I(t)) \\ \frac{dI(t)}{dt} = \beta S(t) I(t) - \nu I(t) \end{cases}$$

14. Que représente N ?

15. Créez le modèle dans **R** en fonction de la valeur de ses paramètres.

```
modelSIRS <- fonction(time,y,parameters){  
  dy1 <- - parameters[1]*y[1]*y[2] + parameters[3]*(parameters[4]-y[1]-y[2])  
  dy2 <- parameters[1]*y[1]*y[2] - parameters[2] * y[2]  
  list(c(dy1,dy2))  
}
```

16. Cherchez analytiquement les coordonnées des points d'équilibre

17. Créez une fonction `R` qui renvoie les points d'équilibre en fonction de la valeur des paramètres sous la forme d'une matrice à deux colonnes et un nombre de ligne égal au nombre de points d'équilibre.

```
equilibres <- fonction(beta,nu,gamma,N){  
  pt.eq <- matrix(NA, ncol=2, nrow=2)  
  pt.eq[1,] <- c(N,0)  
  pt.eq[2,] <- c(nu/beta, gamma*(N-nu/beta)/(gamma+nu))  
  return(pt.eq)  
}
```

18. À l'aide de la fonction précédente, créez un objet `pts.eq` qui contient tous les points d'équilibre. Vous prendrez les mêmes valeurs de paramètres que pour le modèle de Kermack & McKendrick, avec $N = 763$ et $\gamma = 0.5$.
19. Donnez les coordonnées du point d'équilibre endémique.

On va maintenant utiliser la librairie `phaseR` pour faire l'analyse qualitative du modèle *SIRS* en dimension 2 et dresser son portrait de phase.

20. À l'aide de la fonction `stability()`, on peut déterminer la nature des points d'équilibre.

```
stab1 <- stability(modelSIRS, ystar=pts.eq[1,], parameters=c(beta,nu,gamma,N), summary=FALSE)  
stab1$classification
```

```
[1] "Saddle"
```

Quelle est la nature du point d'équilibre endémique ?

21. Si vous êtes curieux et/ou que vous avez suivi des UE de modélisation par le passé, dites quelles autres informations sont fournies par la fonction `stability()`.

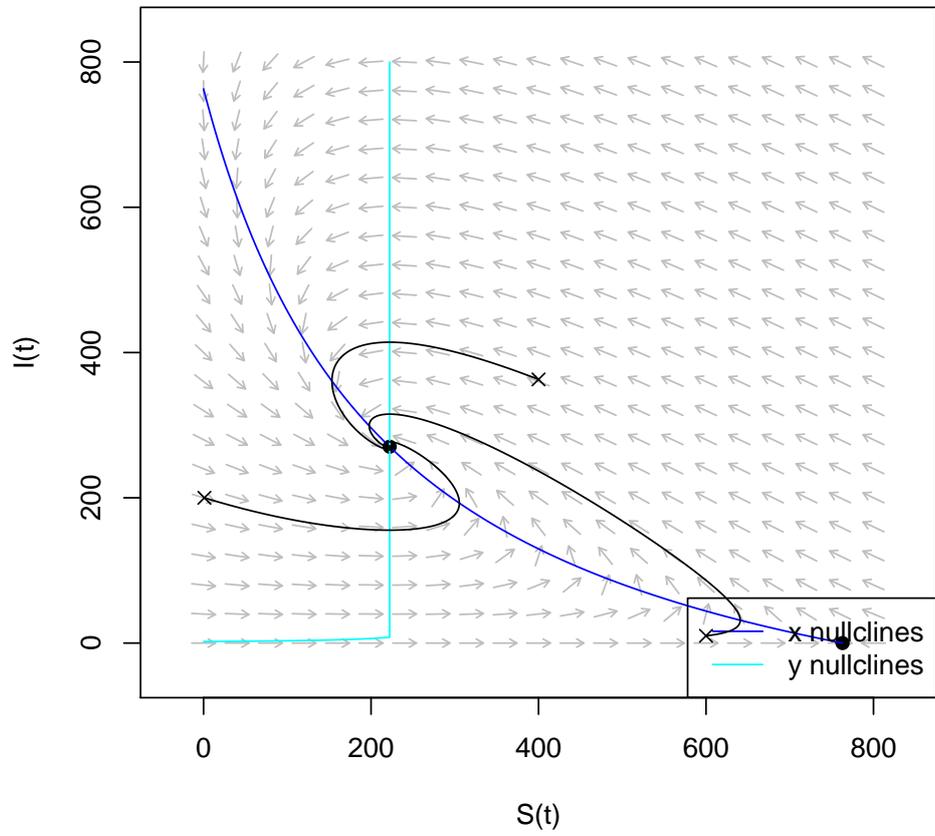
22. Pour dresser le portrait de phase dans le plan (S, I) , utilisez la fonction `phasePlaneAnalysis()`, qui, selon votre sélection, trace successivement le champ de vecteurs, positionne les points d'équilibre, trace les isoclines nulles horizontales et verticales et quelques trajectoires pour des conditions initiales choisies.

Notez que cette fonction utilise de manière "cachée" des fonctions spécifiques : `flowField()` pour représenter le champ de vecteurs; `nullclines()` pour ajouter les isoclines nulles horizontales et verticales; et `trajectory()` pour tracer quelques trajectoires pour des conditions initiales choisies.

Que pouvez-vous conclure quant à la dynamique de l'épidémie ?

```
phasePlaneAnalysis(deriv=modelSIRS, xlim=c(0,800), ylim=c(0,800),tend=100,parameters=c(beta,nu,gamma,N))
```

```
flowField(modelSIRS, xlim=c(0,800), ylim=c(0,800), parameters=c(beta,nu,gamma,N), add=FALSE,
          xlab="S(t)", ylab="I(t)")
points(pts.eq, pch=19)
nullclines(modelSIRS, xlim=c(0,800), ylim=c(0,800), parameters=c(beta,nu,gamma,N), add=TRUE,
           colour=c("red","blue"))
trajectory(modelSIRS, y0=c(600,10), parameters=c(beta,nu,gamma,N), tlim=c(0,50), pch=4)
trajectory(modelSIRS, y0=c(400,N-400), parameters=c(beta,nu,gamma,N), tlim=c(0,50), pch=4)
trajectory(modelSIRS, y0=c(1,200), parameters=c(beta,nu,gamma,N), tlim=c(0,50), pch=4)
```



Vous pouvez éventuellement réfléchir à un moyen de ne faire apparaître le portrait de phase que pour des valeurs de S et de I qui vérifient $S + I = N$.

23. Compte-tenu de ce portrait de phase, à quoi peut-on s'attendre en termes d'évolution au cours du temps de la dynamique des individus sains et infectés pour la condition initiale $(762, 1)$?

24. Pour vérifier, on peut utiliser la fonction `numericalSolution()`.

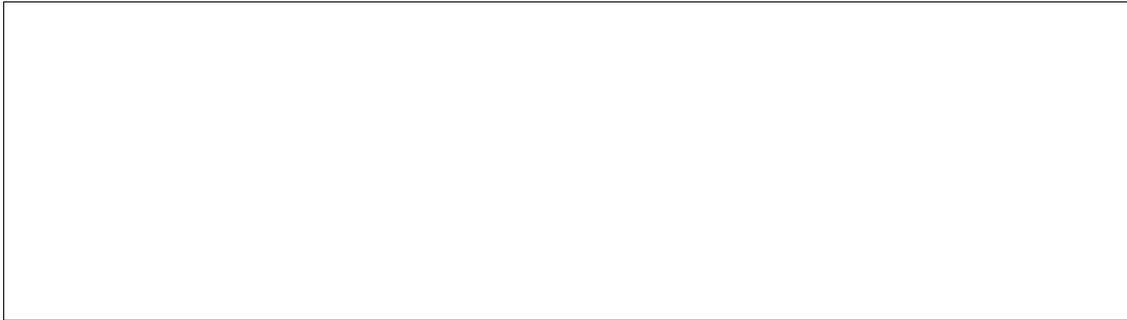
```
numericalSolution(modelSIRS, y0=c(762,1), parameters=c(beta,nu,gamma,N), tlim=c(0,20),
  type="one", col=c("darkgreen","orange"),
  ylim=c(0,763),state.names = c("S", "I"),add.legend=FALSE)
legend("topright", legend=c("Sains","Infectés"), col=c("darkgreen","orange"), lty=1)
```

25. Calculez le taux de reproduction de base.

Sa valeur est-elle cohérente avec vos précédentes observations ? Justifiez.

```
R0 <- beta*N/nu
print(R0)
```

```
[1] 3.4335
```



26. Sous l'hypothèse de ce modèle *SIRS*, proposez des paramètres β et ν permettant d'éradiquer l'épidémie. Représentez graphiquement l'évolution au cours du temps des densités d'individus sains et infectés pour ces valeurs.

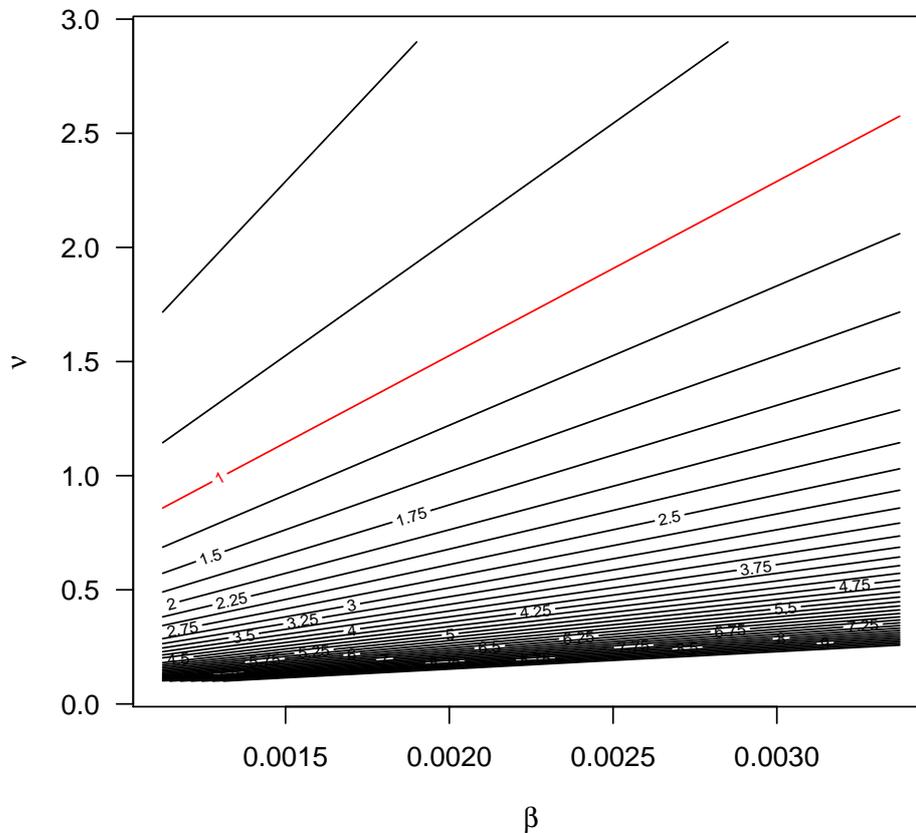
Plusieurs couples de paramètres (β, ν) sont possibles. Pour l'un d'eux par exemple, on obtient la valeur suivante de R_0 :

```
[1] 0.858375
```

27. Il peut être intéressant, pour compléter, de regarder plus globalement R_0 comme une fonction de deux variables β et ν et de faire une représentation graphique des courbes de niveaux de R_0 , le niveau 1 étant celui qui nous intéresse en particulier.

Sur la Figure ci-après, on peut ainsi visualiser la zone du plan (β, ν) qui permet d'obtenir des valeurs de $R_0 < 1$ (au-dessus du niveau 1 en rouge).

```
N <- 763
beta.grid <- seq(0.00225*0.5,0.00225*1.5, 0.00005)
nu.grid <- seq(0.1,2.9,0.01)
R0.grid <- matrix(NA, nrow=length(beta.grid), ncol=length(nu.grid))
for (i in 1:length(beta.grid)){
  for (j in 1:length(nu.grid)){
    R0.grid[i,j] <- beta.grid[i]*N/nu.grid[j]
  }
}
contour(beta.grid,nu.grid,R0.grid, levels=seq(0,10,0.25), las=1,
  xlab=expression(beta), ylab=expression(nu), col=c(rep(1,4),2,rep(1,36)))
```



5 La vaccination

Dans cette section, nous allons tenter d'évaluer l'impact d'une campagne de vaccination sur l'expansion d'une épidémie. On appelle p la fraction de la population à vacciner pour éradiquer une maladie. On choisit donc p telle que :

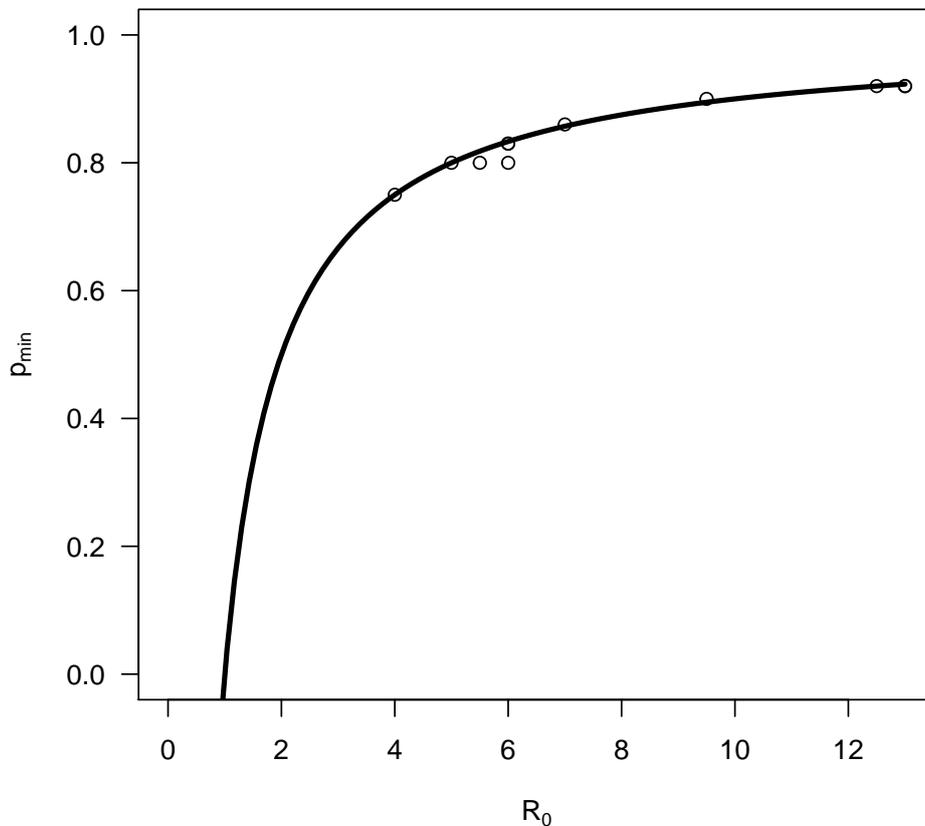
$$(1 - p)R_0 < 1 \Leftrightarrow p > 1 - \frac{1}{R_0}$$

28. Représentez graphiquement la couverture vaccinale minimale p_{min} en fonction de R_0 .

```
curve(1-1/x, from=0, to=13, las=1, lwd=3, ylim=c(0,1),
      xlab=expression(R[0]), ylab=expression(p[min]))
```

29. Vérifiez que cette relation est cohérente avec les données de R. May (1983) et qui se trouvent dans le fichier `data-may-1983.txt`.

Vous devez obtenir la figure suivante :



6 Le théorème du moustique

Ronald Ross peut être considéré comme le père fondateur de la modélisation en épidémiologie actuelle. Il a reçu le prix Nobel en 1902 pour sa preuve que le paludisme (ou malaria), maladie infectieuse due à un parasite du genre *Plasmodium*, était transmis par la piqûre de certaines espèces de moustiques anophèles. C'est lui en 1911 qui a publié le premier modèle dynamique de la transmission du paludisme. Il a prouvé qu'en-dessous d'un certain seuil de population des moustiques, le paludisme disparaissait ¹.

6.1 Les variables et paramètres

Les variables du modèle proposé par Ross sont :

- H , le nombre total d'humains dans la population ;
- V , le nombre total de moustiques dans la population ;
- S_h , le nombre d'humains sains ;
- I_h , le nombre d'humains infectés ;
- S_v , le nombre de moustiques sains ;
- I_v , le nombre de moustiques infectés.

Les paramètres de ce modèle sont les suivants :

- a , le taux de repas sanguin des moustiques sur les humains ;
- b_1 , la proportion de piqûres par des moustiques infectés sur des humains sains, et qui conduisent à une infection effective des humains ;

1. Ross R. 1911. Some Quantitative Studies in Epidemiology. *Nature*, 87 : 466–467.

- b_2 , la proportion de piqûres par des moustiques sains sur des humains infectés, et qui conduisent à une infection effective des moustiques ;
- μ , le taux de mortalité naturelle des moustiques ;
- γ , le taux de guérisons des humains.

D'autre part, le modèle de Ross repose sur un certain nombre d'hypothèses simplificatrices :

- La population totale des humains H est constante ;
- La population totale des moustiques V est constante ;
- On néglige le temps de "guérison" des moustiques (durée de vie courte).

6.2 Le système dynamique

La dynamique du paludisme peut se modéliser de la façon suivante :

$$\begin{cases} \frac{dI_h}{dt} = b_1 a I_v \frac{H - I_h}{H} - \gamma I_h \\ \frac{dI_v}{dt} = b_2 a \frac{I_h}{H} (V - I_v) - \mu I_v \end{cases}$$

Les équations pour S_h et S_v s'obtiennent facilement par $S_h + I_h = H$ et $S_v + I_v = V$.

6.3 Le taux de reproduction de base

Le modèle de Ross admet deux équilibres : $(0, 0)$, sans maladie, et (I_h^*, I_v^*) , l'équilibre endémique pour $R_0 > 1$ avec :

$$I_h^* = \frac{(R_0 - 1)H}{R_0 + \frac{b_2 a}{\mu}} \quad \text{et} \quad I_v^* = \frac{(R_0 - 1)V}{R_0 + \frac{b_1 a V}{\gamma H}}$$

Le taux de reproduction de base est défini par :

$$R_0 = \frac{a^2 b_1 b_2 V}{\mu \gamma H}$$

Le théorème du moustique stipule simplement que si $R_0 \leq 1$, alors la maladie disparaît complètement de la population au bout d'un certain temps, tandis que si $R_0 > 1$, alors la maladie reste endémique dans la population.

6.4 Simulations : à vous de jouer !

```
H <- 100
V <- 10000
a <- 0.28
b1 <- 0.021
b2 <- 0.11
mu <- 0.033
gamma <- 0.05
```

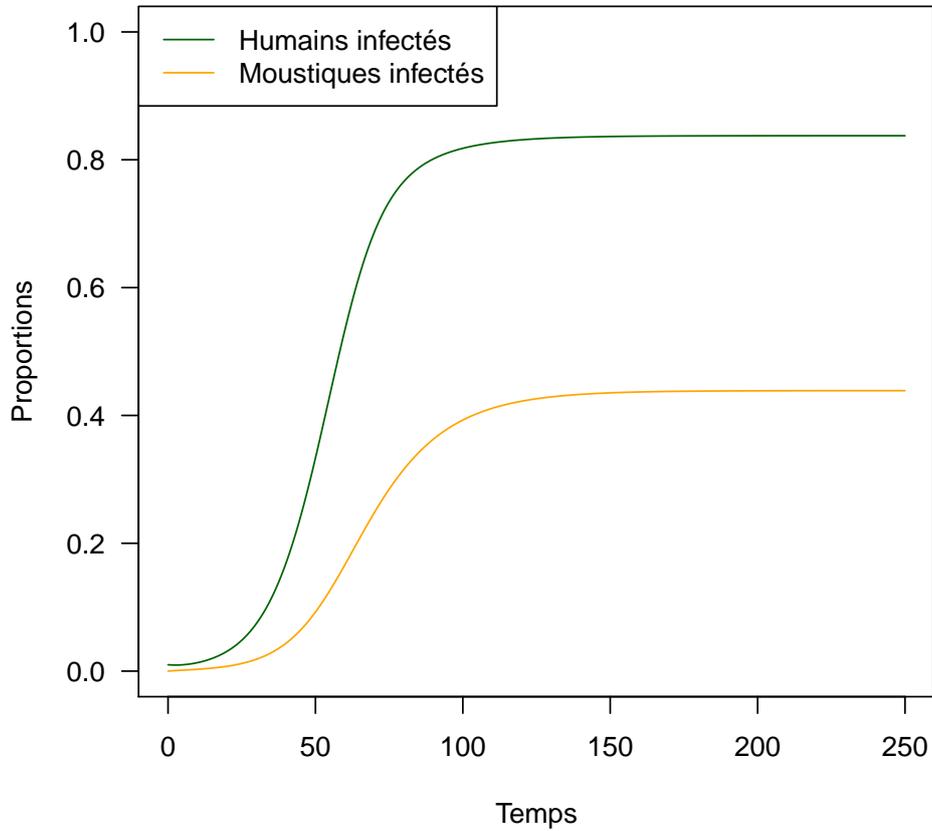
30. Quelle est la valeur de R_0 ?

31. Dessinez le portrait de phase.

32. Dessinez les chroniques pour la condition initiale $(1, 1)$.

33. Représentez les proportions $x = \frac{I_h}{H}$ et $y = \frac{I_v}{V}$ pour cette même condition initiale.

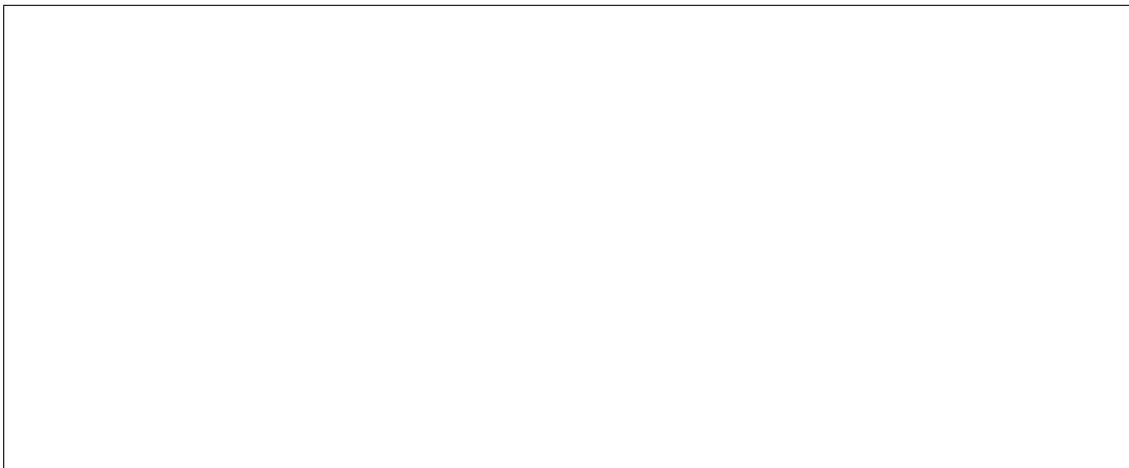
Vous devez obtenir une figure similaire à la suivante :



6.5 Reparamétrisation

On peut ré-écrire le système avec les proportions x et y précédentes, en posant $m = \frac{V}{H}$, $\beta_1 = b_1 a$ et $\beta_2 = b_2 a$. Vérifiez que l'on obtient le système suivant :

$$\begin{cases} \frac{dx}{dt} = \beta_1 m (1-x)y - \gamma x \\ \frac{dy}{dt} = \beta_2 (1-y)x - \mu y \end{cases}$$



34. Quels sont les avantages et inconvénients de cette reparamétrisation ?

35. Simuler le modèle écrit en proportions avec .
Vérifiez que vous obtenez des résultats cohérents avec ceux de la question 32.

7 Vers d'autres modèles

Choisissez un (ou plusieurs!) des modèles présentés en cours mais qui n'a pas été traité dans ce TP pour l'implémenter dans . Commencez par exemple avec le modèle prenant en compte la démographie.

Pour chacun des modèles que vous implémenterez,

- Choisissez plusieurs valeurs plausibles pour les paramètres ;
- Représentez chaque fois le portrait de phase ;
- Représentez chaque fois les chroniques ;
- Calculez chaque fois le R_0 ;
- Discutez chaque fois de vos résultats.

8 Pour aller plus loin

Si vous connaissez déjà  et que vous avez déjà utilisé les bibliothèques `deSolve` et `phaseR`, vous pouvez profiter de ce TP pour :

- Rédiger un compte-rendu détaillé au fur et à mesure des questions, soit avec `R Markdown`, soit avec `R Notebook` ;
- Vous plonger dans l'article de Bonmarin et al. (2008)² et écrire un script qui permette la simulation de leur modèle.

2. Bonmarin I, Santa-Olalla P, Lévy-Bruhl D. 2008. Modélisation de l'impact de la vaccination sur l'épidémiologie de la varicelle et du zona. *Revue Épidémiologique de Santé Publique*, 56 : 323–331.