PROJET 4a : le modèle SIR – RMD

14 décembre, 2020

Table of Contents

I. Consigne et objectifs

Nous allons étudier une problématique biologique (au sens large) par des simulations avec R.

Pour ce faire nous allons proposer:

- un code fonctionnel dans un package R
- une présentation créée avec Rmarkdown
- le partage du code et de la présentation avec github

Le projet que nous allons traiter est le **Projet 4 : le modèle SIR**.

II. Résolution du projet

1. Présentation et compréhension du problème

1.1. Le Modèle SIR : dynamique des épidémies

Le **modèle SIR** propose de représenter une épidémie en compartimentant les individus d'une population N constante (on néglige la natalité et la mortalité) en sous populations dynamiques au cours du temps t : *sains* $S(t)$, *infectés* $I(t)$ et *retirés* $R(t)$. Dans ce modèle, on considère les personnes retirées comme immunisées ou mortes, c'est pourquoi on différentie les deux sous-populations $S(t)$ et $R(t)$.

Le modèle SIR est donc un modèle permettant de modeliser une épidémie, c'est-à-dire de prédire la transmission d'un pathogène entre les individus, les infections, et qu'il ne prend pas en compte la prédiction de la mortalité de l'épidémie.

On sait que:

$$N = S(t) + I(t) + R(t) = 1$$

Précisons que l'état du système à un instant t donné est défini par les trois nombres $S(t)$, $I(t)$, $R(t)$ sont des fractions de la population et qu'on suppose qu'il y a beaucoup de personnes au sein d'une population et qu'on peut donc oublier qu'on a des nombres entiers, c'est-à-dire qu'il faudra considérer S,I et R comme des variables continues.

Introduisons deux variables β et γ qui nous permettront de définir le **taux de transmission** β et le **taux de guérison** γ.

$$\beta\ \ \ \ \ \ \ \gamma\\ S \longrightarrow I \longrightarrow R$$

Le taux de transmission β est donc le passage des personnes saines à infectées et le taux de guérison γ est le passage des personnes infectées à retirées.

1.2. Le système d'équations différentielles

Nous allons donc étudier l'évolution des sous populations en supposant que la variation de $S(t)$,$I(t)$,$R(t)$ à un instant donné t est une fonction simple de la situation à ce même instant, c'est-à-dire que l'évolution est régie par trois équations différentielles non linéaires à trois inconnues.

Elles représentent un taux d'accroissement par rapport au temps :

$$\begin{cases} S'(t) = \dfrac{dS(t)}{dt} = -\beta S(t)I(t) \\ I'(t) = \dfrac{dI(t)}{dt} = \beta S(t)I(t) - \gamma I(t) \\ R'(t) = \dfrac{dR(t)}{dt} = \gamma I(t) \end{cases}$$

De plus,on note :

$$S'(t) + I'(t) + R'(t) = 0$$

Ces 3 équations nous permettent d'obtenir des informations qualitatives intéressantes sur la façon dont l'épidémie se propage.

L'objectif du projet est d'estimer le temps du pic des infectés par des simulations avec R.

1.3. Définition du coefficient R0

Dans notre étude, nous considérons que le nombre de personnes infectées tend vers 0, c'est-à-dire que l'épidémie prend fin et les populations se stabilisent.

Dans les conditions initiales, on donne S(0), I(0) et R(0) :

- $0 \leq S(0) = s_0 \leq 1$ (valeur très proche de 1)
- $0 \leq I(0) = i_0 \leq 1$ (valeur très proche de 0)
- $R(0) = r_0 = 0$ (on considère aucune personne morte ou immunisée au début de l'épidémie)

A t=0 on peut écrire :

$$\begin{cases} S'(0) = -\beta S(0)I(0) \\ I'(0) = \beta S(t)I(0) - \gamma I(0) \\ R'(0) = \gamma I(0) \end{cases}$$

Nous allons définir le **taux de reproduction** R_0 comme le nombre moyen de cas secondaires produit par un individu infectieux au cours de sa période contagieuse. La valeur que prend R_0 determine donc le nombre de personnes que va infecter une personne déja malade.

Reprenons l'équation 2 :

$$ I'(t) = \beta S(t) I(t) - \gamma I(t) \\= \gamma I (t)(\frac{\beta I(t) S(t)}{\gamma I(t)} -1) \\= \gamma I(t) (\frac{\beta S(t)}{\gamma } -1) \\ =\gamma I(t) (R_0 S(t) -1)$$

On identifie $R_0 = \frac{\beta}{\gamma}$ comme le *coefficient de contact*.

C'est ce coefficient là que les gouvernements tentent de maitriser lors de la propagation d'une épidémie. Notamment avec les mesures de confinement actuelles pour limiter le contact entre les personnes saines et malades, et ainsi, réduire les contaminations.

Nous allons pouvoir evaluer le comportement de I'(t) grâce à ce coefficient R_0. En effet, nous savons que γI est forcément positif, donc c'est bien la valeur de $R_0 S(t) - 1$ qui determine le signe de $\gamma I'(t)$.

- Si $R_0 < 1/S_0$ alors $I'(0) < 0$, ce qui veut dire que $I(t)$ *décroit*, l'épidémie prend fin.
- Si $R_0 > 1/S_0 = 1/p$ alors $I'(0) > 0$, ce qui veut dire que $I(t)$ *croit*, et atteint une valeur maximale : c'est le **pic de l'épidemie**.

On constate donc que pour R_0 fixé, plus p est grand (plus il y a déjà de malades), moins il est probable de voir un pic, l'épidémie sera totalement sous contôle. En effet, si un malade peut contaminer plus d'une personne ($R_0 > 1$) la maladie va flamber.

2. Simulation de l'épidémie avec R

2.1. Packages

Installation du package crée pour ce projet :

```
#devtools::install_github("ZoeGerber/ModeleSIR")
library(ModeleSIR)
```

Dans ce package se trouvent toutes les fonctions nécessaires pour réaliser ces simulations du modèle SIR et prédire le pic des infectés.

Nous nous servirons également du package ggplot2 pour tracer nos courbes :

```
library(ggplot2)

## Warning: package 'ggplot2' was built under R version 4.0.3
```

2.2. Résolution du système numériquement

On veut passer d'un modèle continu au discret.

Pendant une unité de temps Δt, le nombre d'individus sains passe de $S(t)$ à $S(t + \Delta t)$, et la variation de $S(t + \Delta t) - S(t)$ peut s'écrire :

$$\begin{cases} \frac{dS(t)}{dt} = \frac{S(t+\Delta t) - S(t)}{\Delta t} \\ \frac{dI(t)}{dt} = \frac{I(t+\Delta t) - I(t)}{\Delta t} \\ \frac{dS(t)}{dt} = \frac{R(t+\Delta t) - R(t)}{\Delta t} \end{cases}$$

Nous voulons isoler $S(t+\Delta t)$, $I(t+\Delta t)$ et $R(t+\Delta t)$ afin de former une suite :

$$\begin{cases} S(t+\Delta t) - S(t) = -\beta S(t) I(t) \Delta t \\ I(t+\Delta t) - I(t) = \beta S(t) I(t) - \gamma I(t) \Delta t \\ R(t+\Delta t) - R(t) = \gamma I(t) \Delta t \end{cases}$$

$$\begin{cases} S(t+\Delta t) = -\beta S(t) I(t) \Delta t + S(t) \\ I(t+\Delta t) = [\beta S(t) I(t) - \gamma I(t)] \Delta t + I(t) \\ R(t+\Delta t) = \gamma I(t) \Delta t + R(t) \end{cases}$$

Nous pouvons donc calculer les valeurs de S,I et R pour chaque valeurs de t données, avec Δt, fixé.

2.3. Calculs

Si nous réalisions les calcul à la main nous pourrions donner le terme suivant grâce au terme précédent, comme suit :

Fixons pour l'exemple $\beta = 0.5$, $\gamma = 0.03$ et $\Delta t = 0.01$.

t(j)	S(t)	I(t)	R(t)	S'(t)	I'(t)	R'(t)
0	0.9	0.1	0	-β * s0 * i0 = **-0.045**	β * s0 * i0 - γ * i0 = **0.042**	γ * i0 = **0.0003**
1	s0-β * i0 * s0 * dt = **0.89955**	i0+(β * i0 * s0 - γ * i0) * dt = **0.10042**	0+ γ * i0 * dt = **3.10^-5**			
2						
3						

Ce qui va nous interresser, c'est de pouvoir réaliser ces calculs avec R.

2.4. Initialisation

Au début de l'épidémie (t=0) on a p personnes saines, et $1-p$ personnes infectées.

Il n'y a aucune personne retirée au début de l'épidémie.

p est donc la proportion de personnes saines au commencement de l'épidémie. (Cf fonction dans le package)

Les paramètres qui vont pouvoir être modifiés et ajustés au cours des simulations sont :

- Le taux de transmission β
- Le taux de guérison γ
- Par extension de ces deux paramètres, le R_0
- La proportion de personnes p saines au début de l'épidémie
- La durée de la simulation (vecteur donné en jours)

La fonction initSir(p) calcule les fractions initiales (à t=0) de N qui sont Sains, Infectés et Retirés :

```
initSir(0.9)

## [1] 0.9 0.1 0.0
```

Les fonctions de tirage donnent des valeurs aléatoires aux paramètres β et γ (qu'on suppose ne pas connaitre) :

Nous avons choisi de modéliser le R_0 par une loi uniforme (valeurs comprises entre 0.01 et 0.99).

```
beta <- tirageBeta(0.01,0.99)
print(beta)

## [1] 0.8584399

gamma <- tirageGamma(0.01,0.99)
print(gamma)

## [1] 0.9898906

R0 <- beta/gamma
print(R0)

## [1] 0.8672068
```

2.5 Courbes

Pour cet exemple, choisissons :

- Taux de transmission $\beta = 0.8$
- Taux de guérison $\gamma = 0.02$
- Proportion de personnes saines $p = 0.7888999$ au début de l'épidémie (choix arbitraire)
- Durée de la simulation : 100jours
- $\Delta t = 0.001$

On part donc du **postulat** que le virus se transmet facilement, et dont on guérit plutôt mal. La valeur du R_0 est de 40.

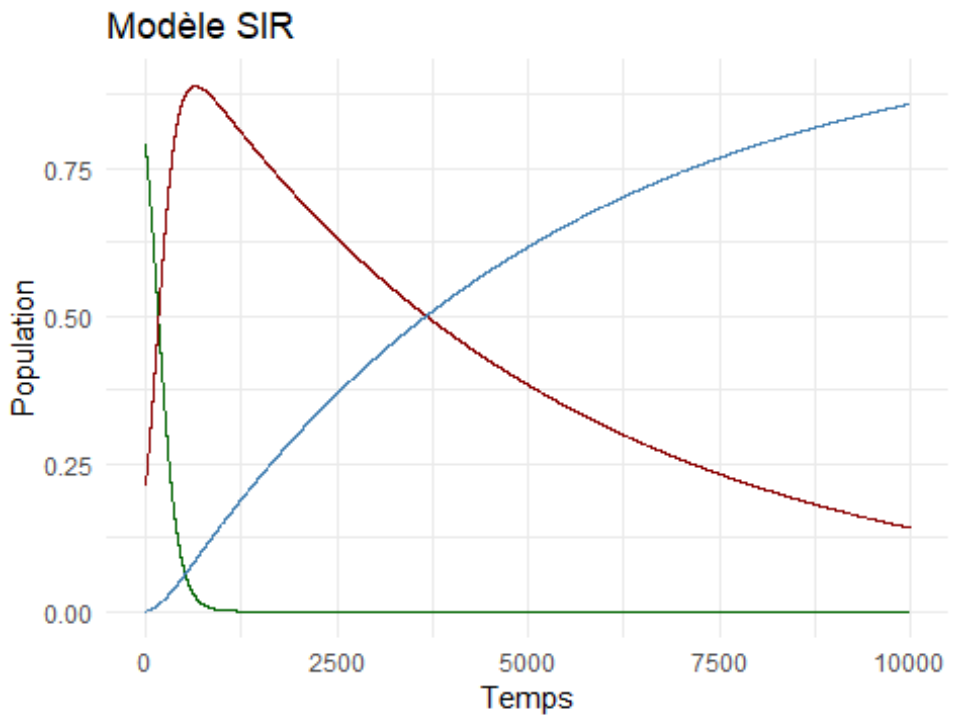
```r
df <- sir(100,0.01,0.7888999,0.8,0.02)

head(df)
```

```
##   j      resS      resI         resR
## 1 1 0.7888999 0.2111001 0.000000e+00
## 2 2 0.7875676 0.2123902 4.222002e-05
## 3 3 0.7862294 0.2136859 8.469805e-05
## 4 4 0.7848854 0.2149872 1.274352e-04
## 5 5 0.7835355 0.2162941 1.704327e-04
## 6 6 0.7821797 0.2176066 2.136915e-04
```

```r
theme_set(theme_minimal())

ggplot(df, aes(x=j)) + geom_line(aes(y = resS), color = "darkgreen") +
geom_line(aes(y = resI),color = "darkred") + geom_line(aes(y = resR),
color="steelblue") + labs(title = "Modèle SIR", x = "Temps", y="Population",
color="L") + labs(caption = "Représentation graphique des populaions
saines(en vert), infectées(en rouge) et retirées(en bleu).")+
theme(plot.caption = element_text(hjust = 0.5, face = "italic", size =10))
```



ɔrésentation graphique des populaions saines(en vert), infectées(en rouge) et retirée

```r
picI(0.8,0.02,df)
```

```
##        PicI datePicI R0pic
## 1 0.8889601      651    40
```

En rouge la courbes des personnes **Infectées**, en bleu celle des **Retirées** et en vert celle des personnes **Saines**

On observe que le pic de l'épidémie dans ces conditions fixées est haut (quasiment toute la population a attrapé le virus (88,7%)).Chaque personne malade contamine 40 personnes, ce qui est énorme.

Cherchons des pistes d'améliorations.

3. Simulations et analyses

Un fois notre graphe fonctionnel, nous pouvons donc nous amuser à modifier la valeur des paramètres et ainsi ammener des premières conclusions à ce modèle.

3.1 Taux de guérison γ

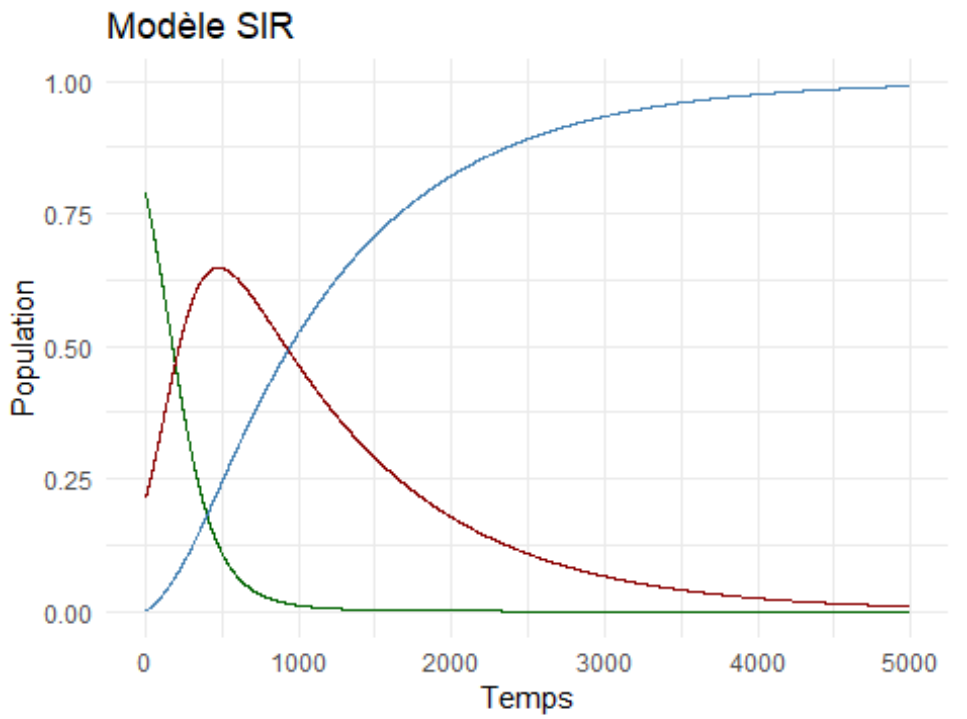
Hypothèse 1 : Nous pouvons agir sur la guérison des personnes malades (médicament ou système de santé performant par exemple)

Augmentons donc le taux de guérison γ :

- Taux de transmission $\beta = 0.8$
- Taux de guérison $\gamma = 0.099$
- Proportion de personnes saines $p = 0.78885555$ au début de l'épidémie
- Durée de la simulation : 100 jours
- $\Delta t = 0.01$

```
df2 <- sir(50,0.01,0.78885555,0.8,0.099)
theme_set(theme_minimal())

ggplot(df2, aes(x=j)) + geom_line(aes(y = resS), color = "darkgreen") +
geom_line(aes(y = resI),color = "darkred") + geom_line(aes(y = resR),
color="steelblue") + labs(title = "Modèle SIR", x = "Temps", y="Population",
color="L") + labs(caption = "Représentation graphique des populaions
saines(en vert), infectées(en rouge) et retirées(en bleu).")+
theme(plot.caption = element_text(hjust = 0.5, face = "italic", size =10))
```



orésentation graphique des populaions saines(en vert), infectées(en rouge) et retirée

```
picI(0.8,0.099,df2)

##        PicI datePicI    R0pic
## 1 0.6475086      477 8.080808
```

On observe directement que le pic d'infectés est beaucoup moins haut (un peu plus de 50% (64,7%) de la population à été malade).

3.2 Taux de transmission β

Hypothèse 2 : Nous pouvons agir sur l'infection des personnes saines (confinement, vaccin ou distantiacion sociale par exemple)

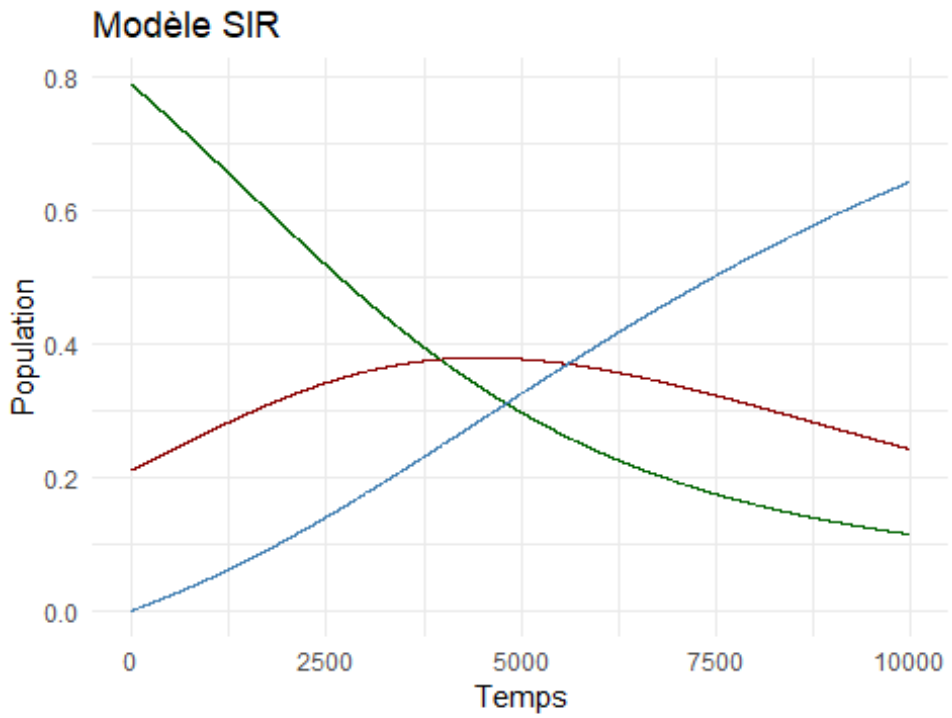
Diminuons donc le taux de transmission β :

- Taux de transmission $\beta = 0.06$
- Taux de guérison $\gamma = 0.02$
- Proportion de personnes saines $p = 0.78885555$ au début de l'épidémie
- Durée de la simulation : 100 jours
- $\Delta t = 0.01$

```
df3 <- sir(100,0.01,0.78885555,0.06,0.02)
theme_set(theme_minimal())

ggplot(df3, aes(x=j)) + geom_line(aes(y = resS), color = "darkgreen") +
geom_line(aes(y = resI),color = "darkred") + geom_line(aes(y = resR),
color="steelblue") + labs(title = "Modèle SIR", x = "Temps", y="Population",
```

```
color="L") + labs(caption = "Représentation graphique des populaions
saines(en vert), infectées(en rouge) et retirées(en bleu).")+
theme(plot.caption = element_text(hjust = 0.5, face = "italic", size =10))
```



résentation graphique des populaions saines(en vert), infectées(en rouge) et retirée

```
picI(0.06,0.02,df3)

##        PicI datePicI R0pic
## 1 0.3795481     4500     3
```

On observe directement que le pic d'infectés est aussi moins haut , et surtout plus étalé (certainement plus facile à gérer médicalement) : un peu moins de 40% (37.9%) de la population à été malade. Il se trouve au niveau du 50eme jour (ce qui laisse le temps de s'organiser pendant une épidémie).

3.3 Taux de guérion γ et taux de transmission β

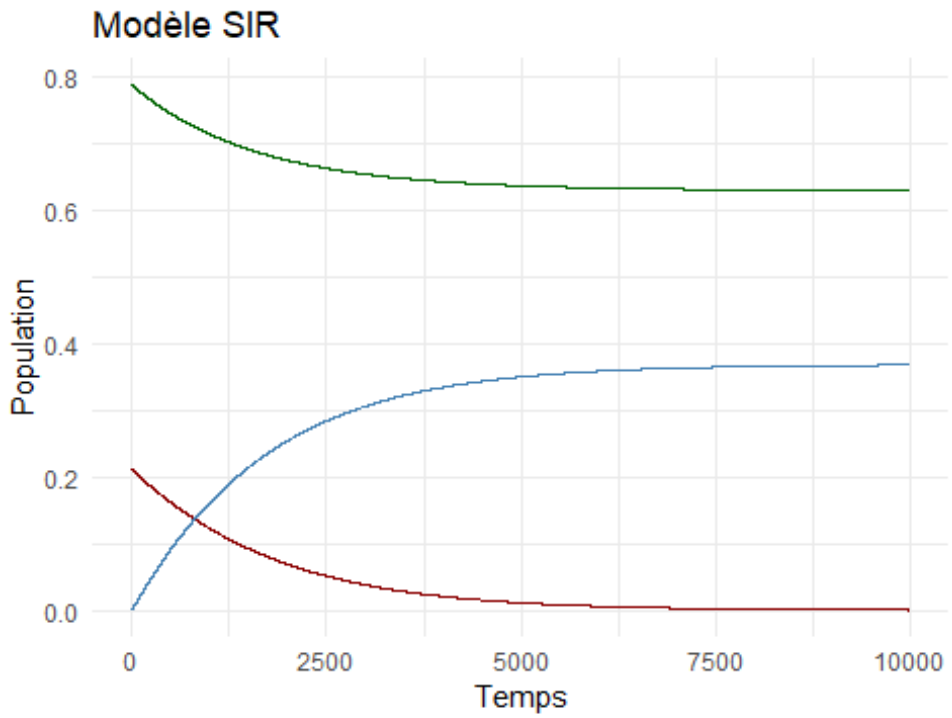
Hypothèse 3 : Nous pouvons agir les deux paramètres en même temps.

Diminuons donc le taux de transmission β et augmentons le taux de guérion γ :

- Taux de transmission $\beta = 0.06$
- Taux de guérison $\gamma = 0.099$
- Proportion de personnes saines $p = 0.78885555$ au début de l'épidémie
- Durée de la simulation : 100jours
- $\Delta t = 0.01$

```
df4 <- sir(100,0.01,0.78885555,0.06,0.099)
theme_set(theme_minimal())

ggplot(df4, aes(x=j)) + geom_line(aes(y = resS), color = "darkgreen") +
geom_line(aes(y = resI),color = "darkred") + geom_line(aes(y = resR),
color="steelblue") + labs(title = "Modèle SIR", x = "Temps", y="Population",
color="L") + labs(caption = "Représentation graphique des populaions
saines(en vert), infectées(en rouge) et retirées(en bleu).")+
theme(plot.caption = element_text(hjust = 0.5, face = "italic", size =10))
```



résentation graphique des populaions saines(en vert), infectées(en rouge) et retirée

```
picI(0.06,0.099,df4)

##        PicI datePicI     R0pic
## 1 0.2111445        1 0.6060606
```

On observe clairement que les mesures prises ont joué un role dans la gestion de cette épidémie ! Certaines personnes n'ont même jamais entendu parler de ce virus il semblerait ! Les personnes infectées contaminaient moins de 1 personnes (0.6).

III. Réalisation d'une étude par simulation d'une quantité caractéristique

Dans notre étude précédente, nous avons modélisé différentes courbes avec des valeurs du taux de reproduction R_0 maitrisées, car on fixait β et γ.

En réalité, lors d'une épidémie, on ne connaît pas le R_0 précisément, et c'est lui qu'on cherche à determiner.

En effet, la connaissance précise du R_0 nous permet de savoir s'il faut prendre des mesures, et quand les prendre, comme imposer un confinement (très restricif ou pas).

Reprenons des valeurs inconnues de β et γ, et regardons comment le R_0 nous permet de prédire le pic.

Si on suppose R_0 entre 2 bornes (>1 [pas d'épidémie] et 20 [épidémie incontrôlée] par exemple), on pourra être capable de déterminer la valeur du pic et le temps du pic.

1. Seuil de tolérance des hopitaux

On cherche à avoir un pic d'infectés qui ne dépasse par une certaine valeur (là où les hopitaux ne peuvent plus gérer les cas).

Avec des conditions initiales $s0$ et $i0$ données, on veut trouver quand et de combien réduire le R_0 pour éviter de dépasser ce seuil, soit trouver quand imposer un confinement et de quelle nature (très restricif ou peu restrictif)

Définissons arbitrairement notre seuil comme le moment ou les infectés seraient à plus de 45% de la population.

```
beta <- tirageBeta(0.01,0.99)
print(beta)
```

```
## [1] 0.3072283
```

```
gamma <- tirageGamma(0.01,0.99)
print(gamma)
```

```
## [1] 0.3286046
```

```
R0 <- beta/gamma
print(R0)
```

```
## [1] 0.9349481
```

```
s <- sir(100,0.1,0.7888999,beta,gamma)
summary(s, simplify=T)
```

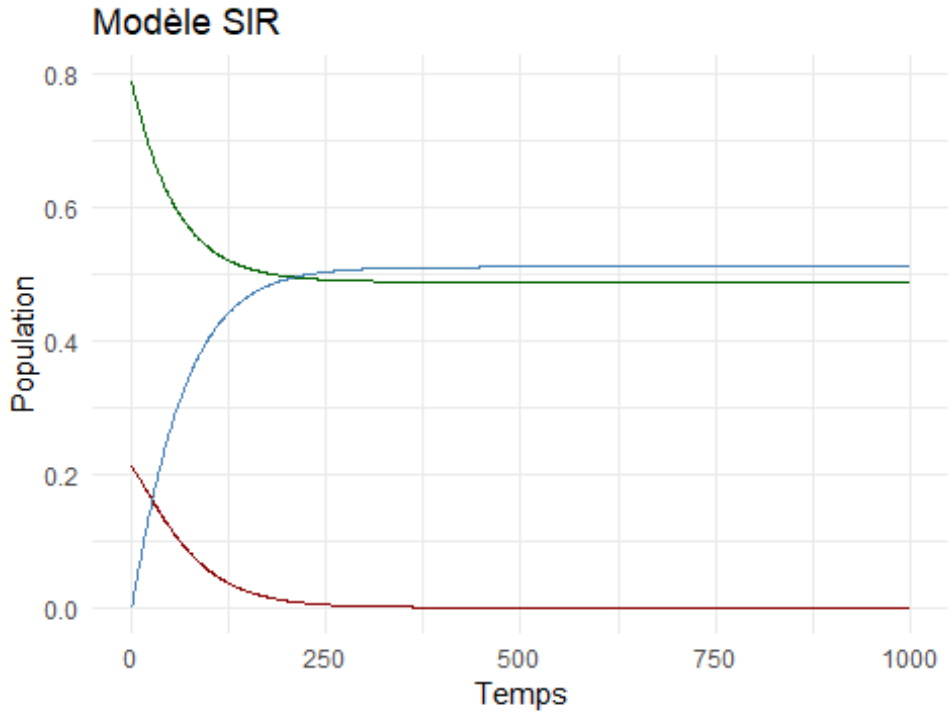
```
##        j               resS             resI                resR        
##  Min.   :   1.0   Min.   :0.4887   Min.   :1.000e-08   Min.   :0.0000  
##  1st Qu.: 250.8   1st Qu.:0.4887   1st Qu.:5.000e-07   1st Qu.:0.5040  
##  Median : 500.5   Median :0.4887   Median :4.450e-05   Median :0.5112  
##  Mean   : 500.5   Mean   :0.5057   Mean   :1.556e-02   Mean   :0.4788  
##  3rd Qu.: 750.2   3rd Qu.:0.4920   3rd Qu.:3.972e-03   3rd Qu.:0.5113  
##  Max.   :1000.0   Max.   :0.7889   Max.   :2.111e-01   Max.   :0.5113
```

```
head(s)
```

```
##   j      resS      resI        resR
## 1 1 0.7888999 0.2111001 0.000000000
## 2 2 0.7837834 0.2092797 0.006936847
## 3 3 0.7787440 0.2074422 0.013813876
## 4 4 0.7737809 0.2055886 0.020630522
## 5 5 0.7688934 0.2037203 0.027386260
## 6 6 0.7640810 0.2018383 0.034080603
```

```
theme_set(theme_minimal())

ggplot(s, aes(x=j)) + geom_line(aes(y = resS), color = "darkgreen") +
geom_line(aes(y = resI),color = "darkred") + geom_line(aes(y = resR),
color="steelblue") + labs(title = "Modèle SIR", x = "Temps", y="Population",
color="L") + labs(caption = "Représentation graphique des populaions
saines(en vert), infectées(en rouge) et retirées(en bleu).")+
theme(plot.caption = element_text(hjust = 0.5, face = "italic", size =10))
```



résentation graphique des populaions saines(en vert), infectées(en rouge) et retirée

```
picI(beta,gamma,s)
```

```
##        PicI datePicI     R0pic
## 1 0.2111001        1 0.9349481
```

2. Replicate()

Le replicate(n,fun) permet de réaliser la simulation de nombreuses fois. La fonction picIsimu() prend en paramètre t, Δt, p et les bornes min et max des parmètres β et γ.

```
list_res <- data.frame(replicate(50,picIsimu(10,0.01,0.7888999,0.01,0.99),
simplify=T))
print(list_res)
```

```
##                X1        X2        X3        X4        X5         X6
X7
## PicI     0.4729376 0.2270557 0.2111001 0.2111001 0.2111001  0.2111001
0.4106224
## datePicI       412       108         1         1         1          1
420
## R0pic     4.139143  1.571619 0.7798903 0.7260329 0.9249562 0.08242176
3.337164
##                X8        X9       X10       X11       X12       X13
X14
## PicI     0.2111001 0.2111001 0.2316643 0.2111001 0.7357832 0.2114545
0.2610685
## datePicI         1         1       337         1       826        21
210
## R0pic    0.9621785 0.3573945  1.623396 0.7216563  12.40456  1.305591
1.901745
##               X15       X16       X17       X18       X19       X20
X21
## PicI     0.2111001 0.2128671 0.6629866 0.2275358 0.3718065 0.2111001
0.2111001
## datePicI         1       110       474       121       388         1
1
## R0pic    0.3396232  1.356995   8.65469  1.577236  2.917272 0.6487954
0.7447063
##               X22       X23       X24        X25       X26       X27
X28
## PicI     0.2111001 0.2111001 0.2111001  0.2111001 0.2111001 0.2111001
0.2111001
## datePicI         1         1         1          1         1         1
1
## R0pic    0.8567552 0.7208636 0.4362372 0.01620386 0.5140468  0.273524
0.4847746
##               X29       X30       X31       X32       X33        X34
X35
## PicI     0.2111001 0.2111001 0.2111001 0.2111001 0.2903852  0.2111001
0.2111001
## datePicI         1         1         1         1       228          1
1
## R0pic    0.2090699 0.5284954 0.1040603 0.5995353  2.158349 0.09156322
0.8747815
##               X36       X37       X38       X39       X40        X41
X42
## PicI     0.2111001 0.2111001 0.6142178 0.2111001 0.2111001  0.2111001
0.3455726
## datePicI         1         1       405         1         1          1
451
```

```
## R0pic    0.8473573 0.2165059  7.018154  1.234472 0.2197083 0.04164869
2.659324
##                X43       X44       X45       X46       X47       X48
X49
## PicI     0.2229967 0.2111001 0.4250528 0.2111001 0.2143285 0.3870407
0.2111001
## datePicI       100         1       689         1        55      1000
1
## R0pic     1.523152 0.1243427  3.508805 0.1262007  1.390372  3.291849
0.1888842
##                X50
## PicI     0.2111001
## datePicI         1
## R0pic     1.026717
```

```
concat <- do.call(rbind.data.frame,list_res)
head(concat)
```

```
##         PicI datePicI      R0pic
## X1 0.4729376      412 4.13914261
## X2 0.2270557      108 1.57161861
## X3 0.2111001        1 0.77989027
## X4 0.2111001        1 0.72603285
## X5 0.2111001        1 0.92495624
## X6 0.2111001        1 0.08242176
```

```
tail(concat)
```

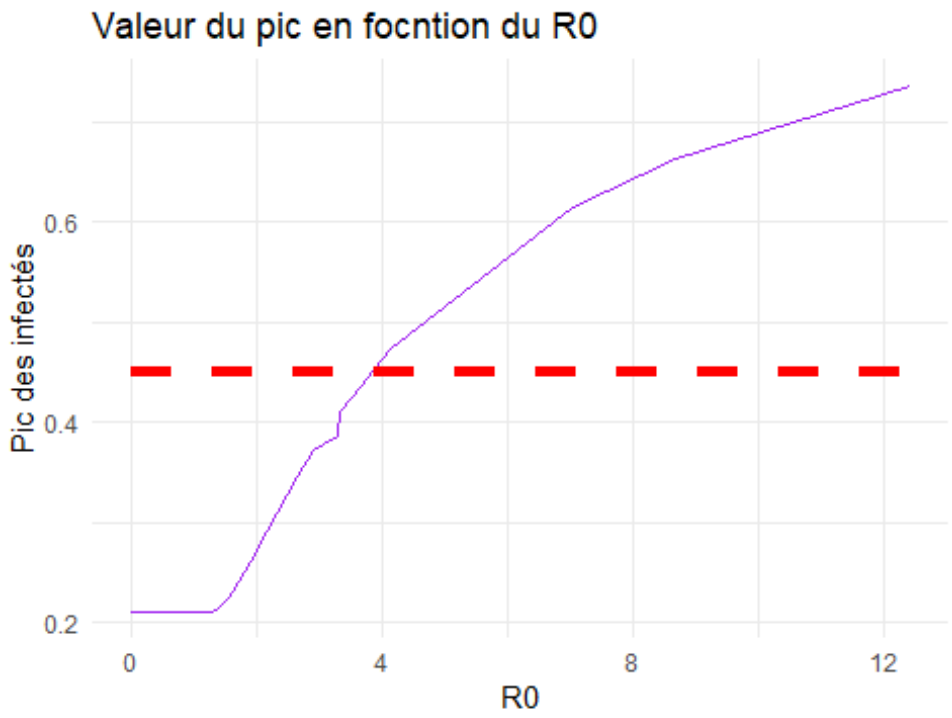
```
##          PicI datePicI     R0pic
## X45 0.4250528      689 3.5088046
## X46 0.2111001        1 0.1262007
## X47 0.2143285       55 1.3903724
## X48 0.3870407     1000 3.2918490
## X49 0.2111001        1 0.1888842
## X50 0.2111001        1 1.0267171
```

```
orderConcat <- concat[order(concat[,3],decreasing=F),]
print(orderConcat)
```

```
##          PicI datePicI       R0pic
## X25 0.2111001        1  0.01620386
## X41 0.2111001        1  0.04164869
## X6  0.2111001        1  0.08242176
## X34 0.2111001        1  0.09156322
## X31 0.2111001        1  0.10406028
## X44 0.2111001        1  0.12434267
## X46 0.2111001        1  0.12620074
## X49 0.2111001        1  0.18888422
## X29 0.2111001        1  0.20906990
## X37 0.2111001        1  0.21650590
## X40 0.2111001        1  0.21970832
```

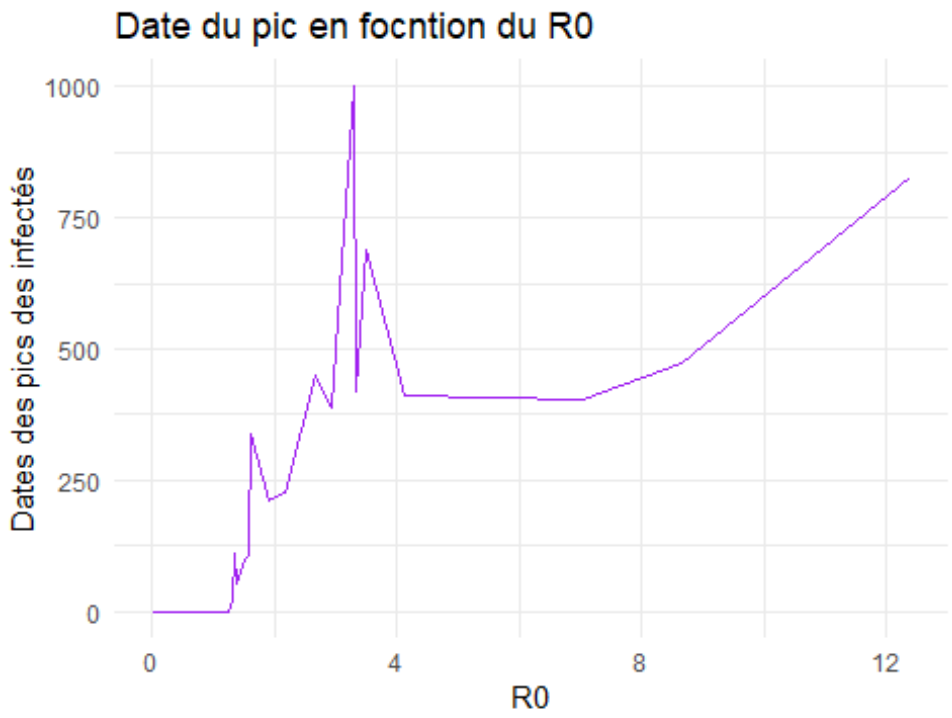
```
## X27 0.2111001        1  0.27352397
## X15 0.2111001        1  0.33962321
## X9  0.2111001        1  0.35739451
## X24 0.2111001        1  0.43623724
## X28 0.2111001        1  0.48477463
## X26 0.2111001        1  0.51404681
## X30 0.2111001        1  0.52849544
## X32 0.2111001        1  0.59953531
## X20 0.2111001        1  0.64879537
## X23 0.2111001        1  0.72086356
## X11 0.2111001        1  0.72165635
## X4  0.2111001        1  0.72603285
## X21 0.2111001        1  0.74470632
## X3  0.2111001        1  0.77989027
## X36 0.2111001        1  0.84735734
## X22 0.2111001        1  0.85675519
## X35 0.2111001        1  0.87478146
## X5  0.2111001        1  0.92495624
## X8  0.2111001        1  0.96217853
## X50 0.2111001        1  1.02671712
## X39 0.2111001        1  1.23447215
## X13 0.2114545       21  1.30559057
## X16 0.2128671      110  1.35699480
## X47 0.2143285       55  1.39037243
## X43 0.2229967      100  1.52315242
## X2  0.2270557      108  1.57161861
## X18 0.2275358      121  1.57723560
## X10 0.2316643      337  1.62339558
## X14 0.2610685      210  1.90174542
## X33 0.2903852      228  2.15834856
## X42 0.3455726      451  2.65932399
## X19 0.3718065      388  2.91727193
## X48 0.3870407     1000  3.29184899
## X7  0.4106224      420  3.33716372
## X45 0.4250528      689  3.50880459
## X1  0.4729376      412  4.13914261
## X38 0.6142178      405  7.01815398
## X17 0.6629866      474  8.65469047
## X12 0.7357832      826 12.40455575
```

```
ggplot(orderConcat, aes(x = R0pic)) + geom_line(aes(y = PicI),
col="purple")+geom_line(aes(y = 0.45), col="red", linetype = "dashed",
size=2)+ labs(title = "Valeur du pic en focntion du R0", x = "R0", y="Pic des
infectés", color="red") + labs(caption = "Représentation graphique des
valeurs des pics à différents R0")+ theme(plot.caption = element_text(hjust =
0.5, face = "italic", size =10))
```



Représentation graphique des valeurs des pics à différents R0

```
 ggplot(orderConcat, aes(x = R0pic)) + geom_line(aes(y = datePicI),
col="purple")+ labs(title = "Date du pic en focntion du R0", x = "R0",
y="Dates des pics des infectés", color="red") + labs(caption =
"Représentation graphique des valeurs des pics à différents R0")+
theme(plot.caption = element_text(hjust = 0.5, face = "italic", size =10))
```



Représentation graphique des valeurs des pics à différents R0

A partir de quelle valeur du R_0 les hopitaux sont-ils débordés ?

On peut observer qu'après de nombreuses répétitions de la fonction, à partir de la valeur de $R0 \approx 3$, le pic atteint la valeur seuil critique.

3. Conclusion

Nous avons tenté de démontrer la corrélation entre la valeur aléatoire du R_0 et la date où le nombre d'infectés sera à son maximun.

L'objectif était d'éviter que le pic de personnes malades ne dépasse une certaine valeur arbitraire (nous avions choisi 45% de la population), car il fallait supposer que les capacités hospitalières ou de prise en charge des malades n'étaient pas extensibles. Ce genre de seuil est valable actuellement avec la pandémie que nous vivons.

En effet, il y a un nombre x de places en réanimation par exemple, et si le nombre de personnes necessitant une place était supérieur à la capacité d'acceuil, cela aurait des conséquences dramatiques.

Il est donc nécessaire de pouvoir prévoir quand devrait arriver le pic, et determiner s'il va depasser la valeur critique seuil. Nous determinons alors le R_0 acceptable, et s'il s'avère que la valeur du R_0 de l'épidémie est au dessus, il faut vite prendre des mesures pour le faire diminuer.

Nous avions vu plus haut dans nos simulations qu'il était possible de diminuer le **taux de transmission** β par exemple, en instaurant une distanciation sociale, ou en rendant

obligatoire le port du masque dans les espaces clos. Si les scientifiques et les médecins parvenaient à trouver un médicament efficace, cela pourra augmenter le **taux de guérison** γ.

IV. Complexification et réalisme du modèle SIR

1. Réalisme

Le modèle SIR est un modèle simple, qui considèrait une population constante au cours du temps, où les naissances et la mortalité étaient négligées.

En effet, pour se rapprocher de la réalité il faudrait prendre en compte ces paramètres.

De plus, lors d'une épidémie, il est possible de mettre en place des mesures pour réduire le nombre de malades ou de morts. On peut par exemple décider de confiner la population si le taux de transmission β du pathogène est élevé. Ou alors lancer une campagne de vaccination pour réduire le nombre de contaminations.

De plus, nous sommes partis du postulats que les personnes retirées étaient soit guéries soit mortes. En réalité, ce n'est pas le cas pour toutes les maladies. Il se pose la question de l'existence d'un nouveau compartiment, ou d'un possible retour des personnes retirées vers sains, qui dans ce cas-là, redeviennent suceptibles d'attraper une fois de plus la maladie.

2. Modèle SEIR

Dans ce modèle, il est possible de prendre en compte la démographie, et donc d'avoir une évolution de $N(t)$ au cours du temps.

On considèrera toujours que les personnes qui naissent sont saines. On introduit alors le **taux de natalité** ν.

On considèrera également que les personnes qui meurent, peuvent être dans n'importe quelle sous-population à l'instant t (la mort n'est pas toujours liée au virus). C'est le **taux de mortalité** μ.

Une nouvelle sous-population $E(t)$ est ajoutée au modèle : les **personnes exposées** c'est-à-dire **les infectées non-contagieuses** (c'est-à-dire les personnes qui ont été en contact avec une personne malade, mais qui ne transmettent pas encore le pathogène).

Cela nous permet de prendre en compte la **durée d'incubation** du pathogène dans l'organisme et d'introduire un nouveau paramètre α : le **taux d'incubation d'une maladie**.

n représente une naissance et *m* un décès.

Voici donc une nouvelle schématisation, qui se rapproche un peu plus de la réalité :

$$ \nu \ \ \ \ \ \ \ \ \ \beta\ \ \ \ \ \ \ \ \alpha \ \ \ \ \ \ \ \ \ \ \ \gamma \ \ \ \ \ \ \ \ \ \mu \\ n \longrightarrow\ S \ \longrightarrow E \longrightarrow\ I \ \longrightarrow R \longrightarrow m\\ \downarrow \mu \ \ \ \ \ \ \ \ \downarrow \mu \ \ \ \ \ \ \ \

\downarrow \mu\ \ \ \ \ \ \ \ \\\ m \ \ \ \ \ \ \ \ \ m \ \ \ \ \ \ \ m \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \$$

De plus, nos équations différentielles du modèle SIR seront compléxifiées avec l'ajout de ces paramètres :

$$\begin{cases} S'(t) = \dfrac{dS(t)}{dt} = -\beta S(t)I(t) + \nu N(t) - \mu S(t) \\ E'(t) = \dfrac{dE(t)}{dt} = \beta S(t)I(t) - \alpha E(t) - \mu E(t) \\ I'(t) = \dfrac{dI(t)}{dt} = \alpha E(t) - \gamma I(t) - \mu I(t) \\ R'(t) = \dfrac{dR(t)}{dt} = \gamma I(t) - \mu R(t) \end{cases}$$

3. Résolution et courbes

De la même manière que pour le modèle SIR, nous pouvons résoudre numériquement ces équations différentielles :

$$\begin{cases} \dfrac{dS(t)}{dt} = \dfrac{S(t+\Delta t) - S(t)}{\Delta t} \\ \dfrac{dE(t)}{dt} = \dfrac{E(t+\Delta t) - E(t)}{\Delta t} \\ \dfrac{dI(t)}{dt} = \dfrac{I(t+\Delta t) - I(t)}{\Delta t} \\ \dfrac{dS(t)}{dt} = \dfrac{R(t+\Delta t) - R(t)}{\Delta t} \end{cases}$$

Nous voulons isoler $S(t+\Delta t)$, $E(t+\Delta t)$, $I(t+\Delta t)$ et $R(t+\Delta t)$ afin de former une suite :

$$\begin{cases} S(t+\Delta t) - S(t) = -\beta S(t)I(t)\Delta t \\ I(t+\Delta t) - I(t) = \beta S(t)I(t) - \gamma I(t)\Delta t \\ R(t+\Delta t) - R(t) = \gamma I(t)\Delta t \end{cases}$$

$$\begin{cases} S(t+\Delta t) = S(t) + (-\beta S(t)I(t) + \nu N(t) - \mu S(t))\Delta t \\ E(t+\Delta t) = E(t) + (\beta S(t)I(t) - \alpha E(t) - \mu E(t))\Delta t \\ I(t+\Delta t) = I(t) + (\alpha E(t) - \gamma I(t) - \mu I(t))\Delta t \\ R(t+\Delta t) = R(t) + (\gamma I(t) - \mu R(t))\Delta t \end{cases}$$

Nous avons cherché à décrire l'évolution de la, population en fonction des naissances et des décès.

On sait qu'a un instant t, la taille de la population sera :

$$N(t) = S(t) + E(t) + I(t) + R(t)$$

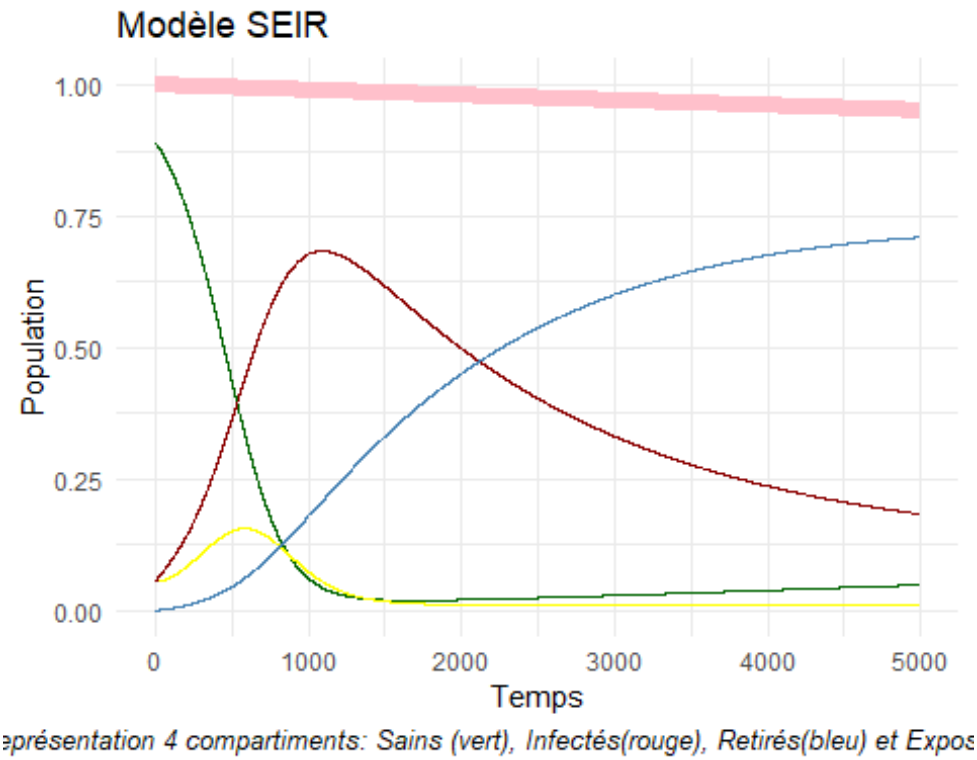
```r
dfseir <- seir(50,0.01,0.8899,0.8,0.05,0.75,0.01,0.009)

head(dfseir)
```

```
##   j      resS       resE       resI         resR    resN
## 1 1 0.8899000 0.05505000 0.05505000 0.000000e+00 1.00000
## 2 2 0.8895091 0.05502353 0.05542984 2.752500e-05 1.00000
## 3 3 0.8891157 0.05499980 0.05580926 5.523717e-05 0.99999
## 4 4 0.8887198 0.05497876 0.05618828 8.313628e-05 0.99998
## 5 5 0.8883215 0.05496041 0.05656690 1.112221e-04 0.99997
## 6 6 0.8879206 0.05494471 0.05694517 1.394944e-04 0.99996
```

```r
theme_set(theme_minimal())

ggplot(dfseir, aes(x=j)) +
geom_line(aes(y = resS), color = "darkgreen") +
geom_line(aes(y = resE), color = "yellow")+
geom_line(aes(y = resI),color = "darkred") +
geom_line(aes(y = resR), color="steelblue")+geom_line(aes(y = resN), color =
"pink",size =3)+ labs(title = "Modèle SEIR", x = "Temps", y="Population",
color="L") +    labs(caption = "Représentation 4 compartiments: Sains (vert),
Infectés(rouge), Retirés(bleu) et Exposés(jaune).")+
theme(plot.caption = element_text(hjust = 0.5, face = "italic", size =10))
```



eprésentation 4 compartiments: Sains (vert), Infectés(rouge), Retirés(bleu) et Expos

Le pic correspondant au graphe ci dessus :

```
graphe <- picISeir(0.8,0.05,0.75,dfseir)
print(graphe)
```

```
## $PicI
## [1] 0.6837233
##
## $datePicI
## [1] 1092
##
## $R0pic
## [1] 16
```

Une question se pose : comment calculer tous ces paramètres $\alpha, \beta, \gamma, \mu$ et ν. Ce travail est effectué en collaboration entre les équipes médicales, les épidémiologistes et virologistes, et les mathématiciens. Les taux de natalité et de mortalité peuvent être obtenus grâce au recensement de population.

Il faut, pour cette étude, réaliser en plus un **tirage du taux d'incubation** α :

```
beta <- tirageBeta(0.1,0.99)
print(beta)
```

```
## [1] 0.7241921
```

```
gamma <- tirageGamma(0.01,0.99)
print(gamma)
```

```
## [1] 0.04146162
```

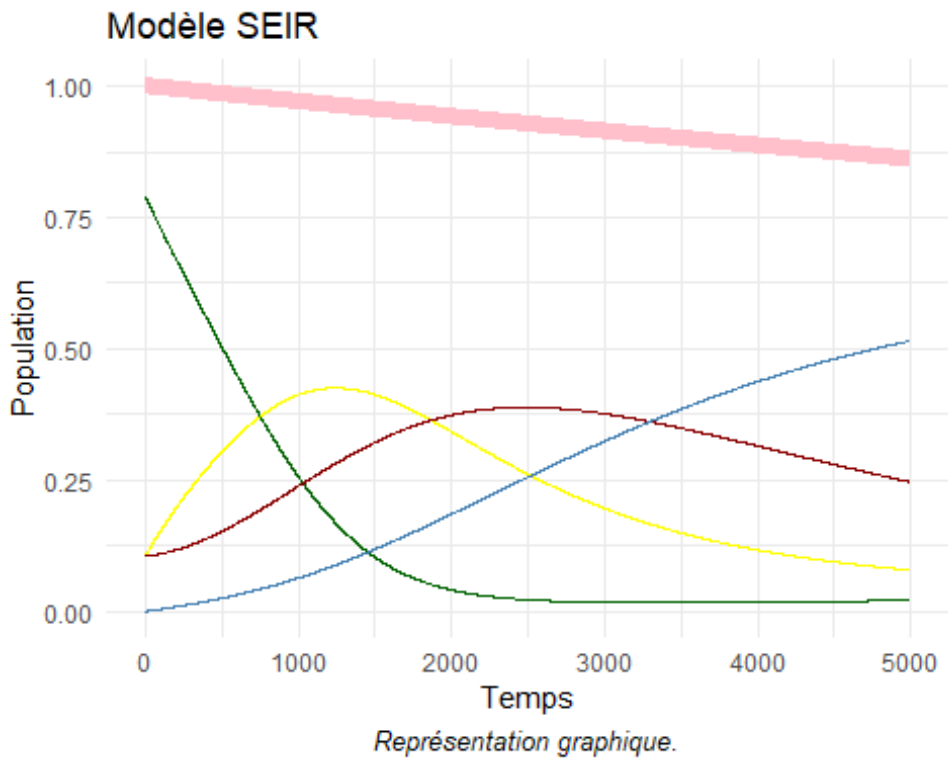
```
alpha <- tirageAlpha(0.01,0.99)
print(alpha)
```

```
## [1] 0.07290937
```

```
dfseirtest <- seir(50,0.01,0.78855,beta,gamma,alpha,0.008,0.005)



ggplot(dfseirtest, aes(x=j)) +
geom_line(aes(y = resS), color = "darkgreen") +
geom_line(aes(y = resE), color = "yellow")+
geom_line(aes(y = resI),color = "darkred") +
geom_line(aes(y = resR), color="steelblue")+ geom_line(aes(y = resN), color =
"pink",size =3)+                                labs(title = "Modèle SEIR", x
= "Temps", y="Population", color="L") +    labs(caption = "Représentation
graphique.")+theme(plot.caption = element_text(hjust = 0.5, face = "italic",
size =10))
```



Représentation graphique.

```
picISeir(beta,gamma,alpha,dfseirtest)
```

```
## $PicI
## [1] 0.3885607
##
## $datePicI
## [1] 2484
##
## $R0pic
## [1] 17.46656
```

```
picIsimuSeir(50, 0.01, 0.788554,0.01,0.99)
```

```
##        PicI datePicI    R0pic
## 1 0.1508232      610 1.838461
```

```
list_resSeir <- data.frame(replicate(20,picIsimuSeir(50,
0.01,0.688554,0.01,0.99), simplify=T))
print(list_resSeir)
```

```
##                X1        X2        X3        X4        X5        X6
X7
## PicI      0.155723  0.155723  0.155723  0.155723 0.2184771 0.3027908
0.1737595
## datePicI         1         1         1         1       246       625
123
## R0pic    0.3341321 0.8862359 0.6861366 0.3345512   1.79327  3.259724
1.036565
```

```
##                 X8        X9       X10       X11        X12       X13
X14
## PicI     0.5032931 0.2031482 0.1968713 0.1781537  0.1562071 0.5164256
0.2339398
## datePicI      1040       450       193        94         12      1284
573
## R0pic     8.471284  1.373852  1.638228 0.2484773 0.08247232  12.95476
2.675021
##                X15       X16       X17       X18       X19       X20
## PicI     0.2765717 0.1904133 0.2234758  0.155723 0.4094602 0.8089207
## datePicI      1019       128       313         1      2575      2393
## R0pic     1.960159 0.9106902  1.713764 0.2992747  3.278205  19.35902
```

```
concatSeir <- do.call(rbind.data.frame,list_resSeir)
head(concatSeir)
```

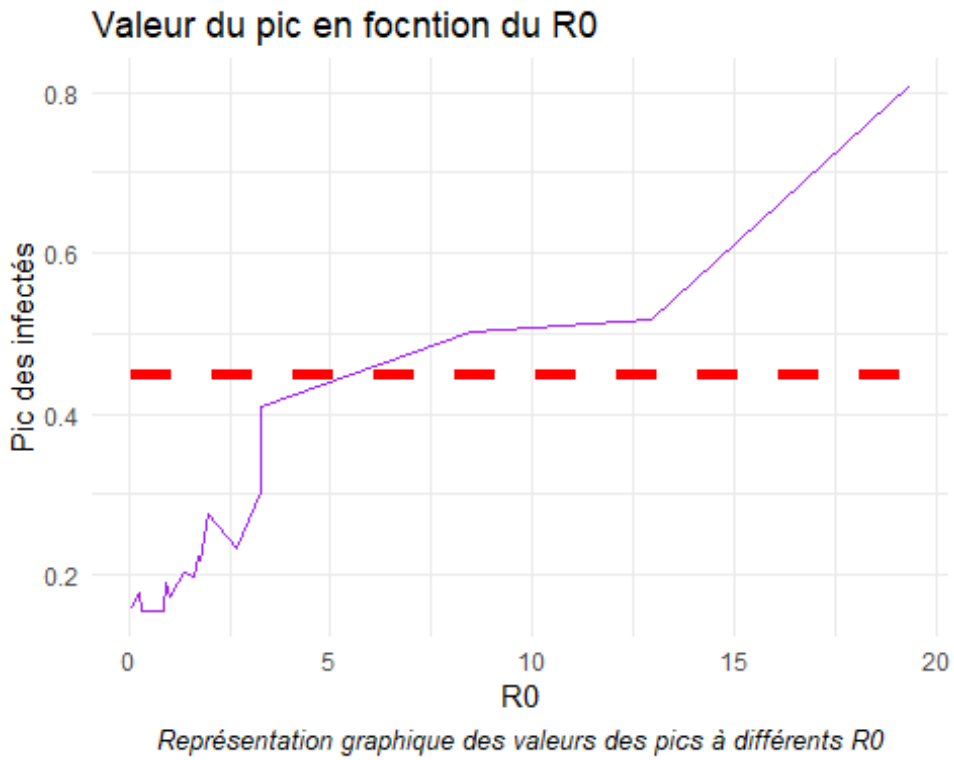
```
##         PicI datePicI     R0pic
## X1 0.1557230        1 0.3341321
## X2 0.1557230        1 0.8862359
## X3 0.1557230        1 0.6861366
## X4 0.1557230        1 0.3345512
## X5 0.2184771      246 1.7932696
## X6 0.3027908      625 3.2597240
```

```
orderConcatSeir <- concatSeir[order(concatSeir[,3],decreasing=F),]
print(orderConcatSeir)
```

```
##          PicI datePicI       R0pic
## X12 0.1562071       12  0.08247232
## X11 0.1781537       94  0.24847732
## X18 0.1557230        1  0.29927465
## X1  0.1557230        1  0.33413212
## X4  0.1557230        1  0.33455121
## X3  0.1557230        1  0.68613664
## X2  0.1557230        1  0.88623587
## X16 0.1904133      128  0.91069020
## X7  0.1737595      123  1.03656498
## X9  0.2031482      450  1.37385159
## X10 0.1968713      193  1.63822763
## X17 0.2234758      313  1.71376441
## X5  0.2184771      246  1.79326956
## X15 0.2765717     1019  1.96015923
## X14 0.2339398      573  2.67502132
## X6  0.3027908      625  3.25972399
## X19 0.4094602     2575  3.27820458
## X8  0.5032931     1040  8.47128417
## X13 0.5164256     1284 12.95476115
## X20 0.8089207     2393 19.35901503
```

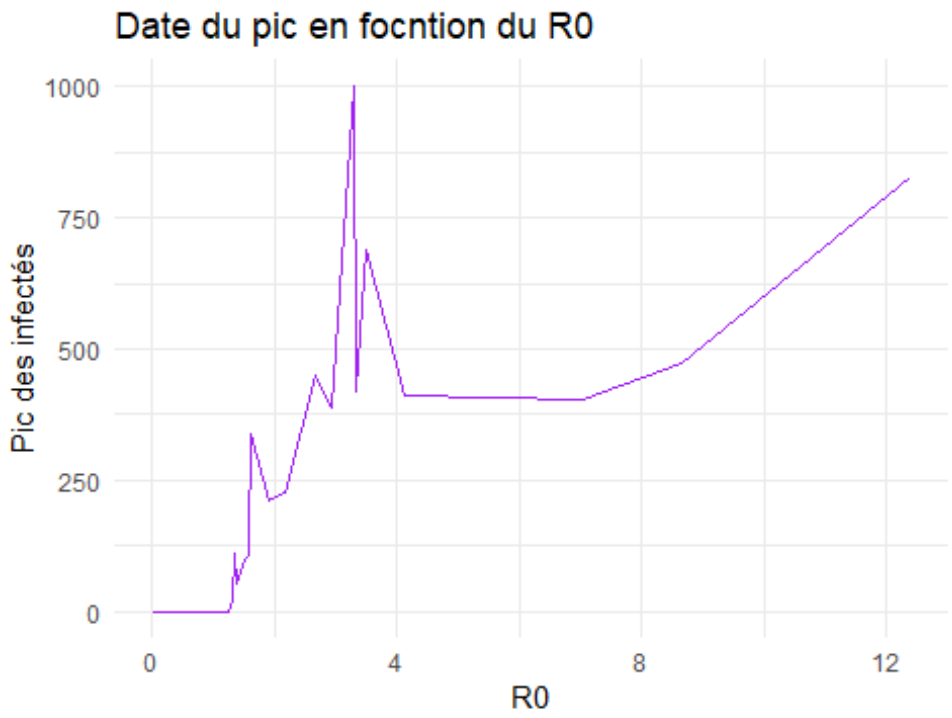
```
ggplot(orderConcatSeir, aes(x = R0pic)) + geom_line(aes(y = PicI),
col="purple")+geom_line(aes(y = 0.45), col="red", linetype = "dashed",
```

```
size=2)+ labs(title = "Valeur du pic en focntion du R0", x = "R0", y="Pic des infectés", color="red") + labs(caption = "Représentation graphique des valeurs des pics à différents R0")+ theme(plot.caption = element_text(hjust = 0.5, face = "italic", size =10))
```



Représentation graphique des valeurs des pics à différents R0

```
ggplot(orderConcat, aes(x = R0pic)) + geom_line(aes(y = datePicI), col="purple")+ labs(title = "Date du pic en focntion du R0", x = "R0", y="Pic des infectés", color="red") + labs(caption = "Représentation graphique des valeurs des pics à différents R0")+ theme(plot.caption = element_text(hjust = 0.5, face = "italic", size =10))
```



Représentation graphique des valeurs des pics à différents R0