

PROJET 4a : le modèle SIR – RMD

12 décembre, 2020

Table of Contents

I. Consigne et objectifs	1
II. Résolution du projet.....	2
1. Présentation et compréhension du problème	2
1.1. Le Modèle SIR : dynamique des épidémies.....	2
1.2. Le système d'équations différentielles.....	2
1.3. Définition du coefficient R_0	3
2. Simulation de l'épidémie avec R.....	4
2.1. Package	4
2.2. Résolution du système numériquement.....	4
2.3. Calculs.....	5
2.4. Initialisation	5
2.5 Courbes	6
3. Simulations et analyses	7
3.1 Taux de guérison γ	7
3.2 Taux de transmission β	8
3.3 Taux de guérison γ et taux de transmission β	9
III. Réalisation d'une étude par simulation d'une quantité caractéristique	10
Seuil de tolérance des hopitaux.....	11
Replicate()	12
IV. Complexification et réalisme du modèle SIR.....	17
1. Réalisme.....	17
2. Modèle SEIR.....	17
3. Résolution et courbes	18

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 *guéris* $R(t)$. Dans ce modèle, on considère les personnes guéries comme immunisées, c'est pourquoi on différencie les deux sous-populations $S(t)$ et $R(t)$.

Le modèle SIR est donc un modèle permettant de modéliser 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 .

β γ $S \rightarrow I \rightarrow R$

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 guéries.

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 :

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

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 R_0

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 guéries au début de l'épidémie)

A $t=0$ on peut écrire :

$$\begin{cases} S'(0) = -\beta S(0)I(0) \\ I'(0) = \beta S(0)I(0) - \gamma I(0) \\ R'(0) = \gamma I(0) \end{cases}$$

Nous allons définir le **taux de reproduction** R_0 comme le nombre moyen de cas secondaires produits par un individu infectieux au cours de sa période d'infection. La valeur que prend R_0 détermine donc le nombre de personnes que va infecter une personne déjà malade.

Reprenons l'équation 2 :

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

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

C'est ce coefficient là que les gouvernement tentent de maîtriser 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 évaluer 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 détermine le signe de $\gamma I'(t)$.

- Si $R_0 s_0 < 1$ alors $I'(0) < 0$, ce qui veut dire que $I(t)$ *décroît*, l'épidémie prend fin.
- Si $R_0 s_0 > 1$ alors $I'(0) > 0$, ce qui veut dire que $I(t)$ *croît*, et atteint une valeur maximale : c'est le **pic de l'épidémie**.

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. Package

Installation du package créé 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 :

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

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 réalisons 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	$-\beta * s_0 * i_0 = -0.045$	$\beta * s_0 * i_0 - \gamma * i_0 = 0.042$	$\gamma * i_0 = 0.0003$
1	$s_0 - \beta * i_0 * s_0 * \Delta t = 0.89955$	$i_0 + (\beta * i_0 * s_0 - \gamma * i_0) * \Delta t = 0.10042$	$0 + \gamma * i_0 * \Delta t = 3.10^{-5}$			
2						
3						

Ce qui va nous intéresser, c'est de pouvoir réaliser ces calcul 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 guéries 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)
- a. Fractions initiales affichées (à t=0) de N qui sont Sains, Infectés et Guéris :

```
init(0.9)
```

```
## [1] 0.9 0.1 0.0
```

- b. Paramètres variables β et de γ , qu'on suppose ne pas connaître :

Nous avons choisi de modéliser le R_0 par une loi normale.

```

beta <- tirageBeta(0.8,1)
print(beta)

## [1] 2.184889

gamma <- tirageGamma(0.02,1)
print(gamma)

## [1] 0.6391331

R0 <- beta/gamma
print(R0)

## [1] 3.418519

```

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
- Durée de la simulation : 50jours
- $\Delta t = 0.001$

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.

```

df <- sir(50,0.01,0.7888999,0.8,0.02)

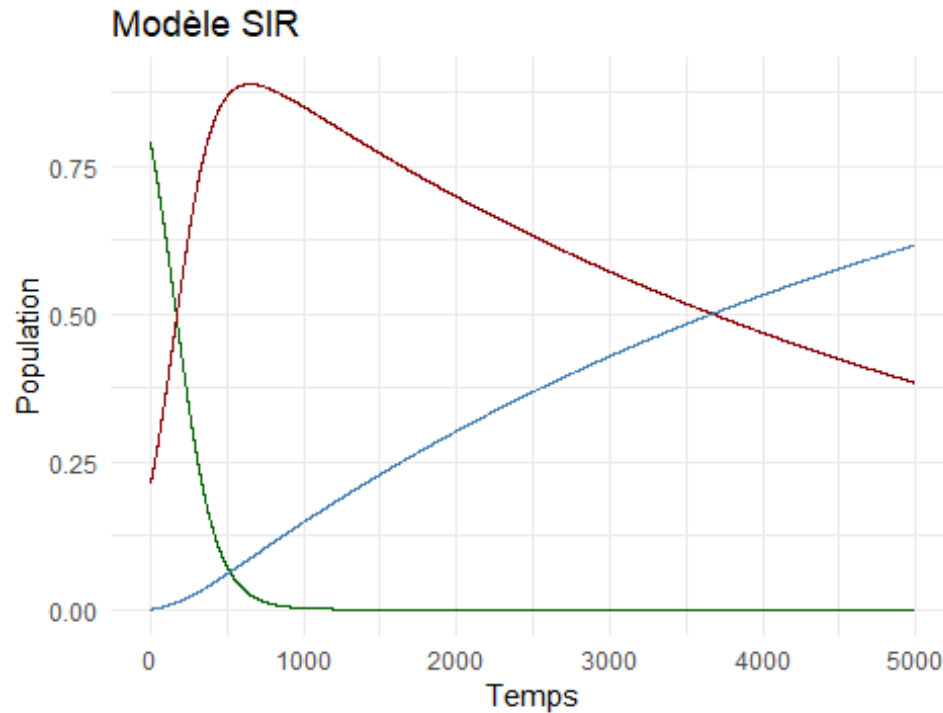
head(df)

##      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

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 guéries(en bleu).")+
theme(plot.caption = element_text(hjust = 0.5, face = "italic", size =10))

```



présentation graphique des populaions saines(en vert), infectées(en rouge) et guéries

En rouge la courbes des personnes **Infectées**, en bleu celle des **Guéries** 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%)) et se trouve aux alentours du 10ème jour. 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 amener des premières conclusions à ce modèle.

3.1 Taux de guérison γ

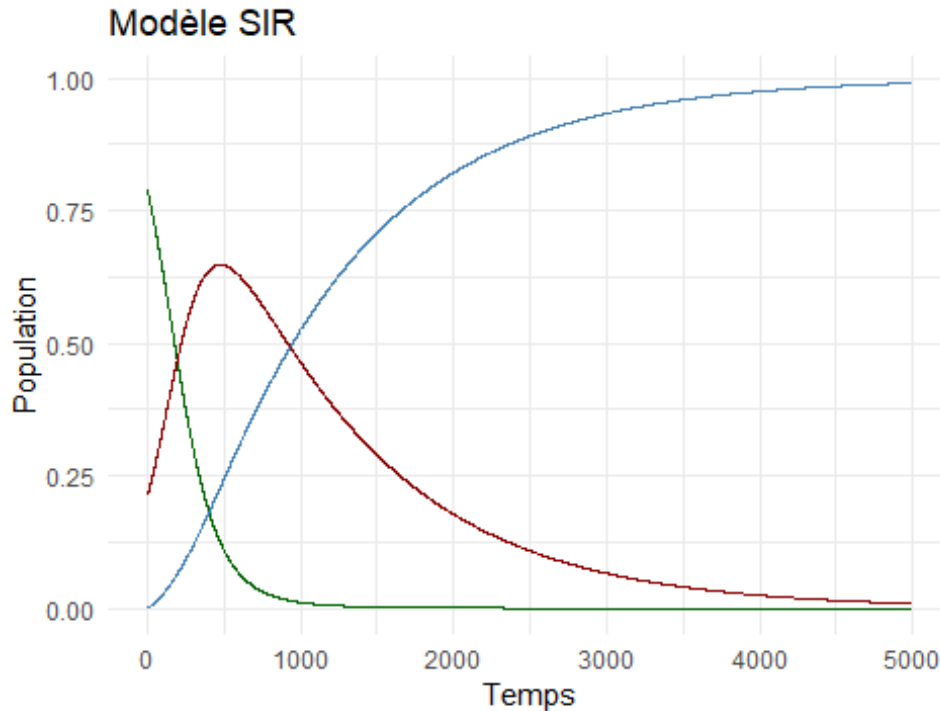
Hypothèse 1 : Nous pouvons agir sur la guérison des personnes malades (médicament ou système de santé performant par exemple)

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 : 50 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 guéries(en bleu).")+
theme(plot.caption = element_text(hjust = 0.5, face = "italic", size =10))
```



présentation graphique des populaions saines(en vert), infectées(en rouge) et guéries(en bleu).

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), mais est toujours au niveau du 10eme jour.

3.2 Taux de transmission β

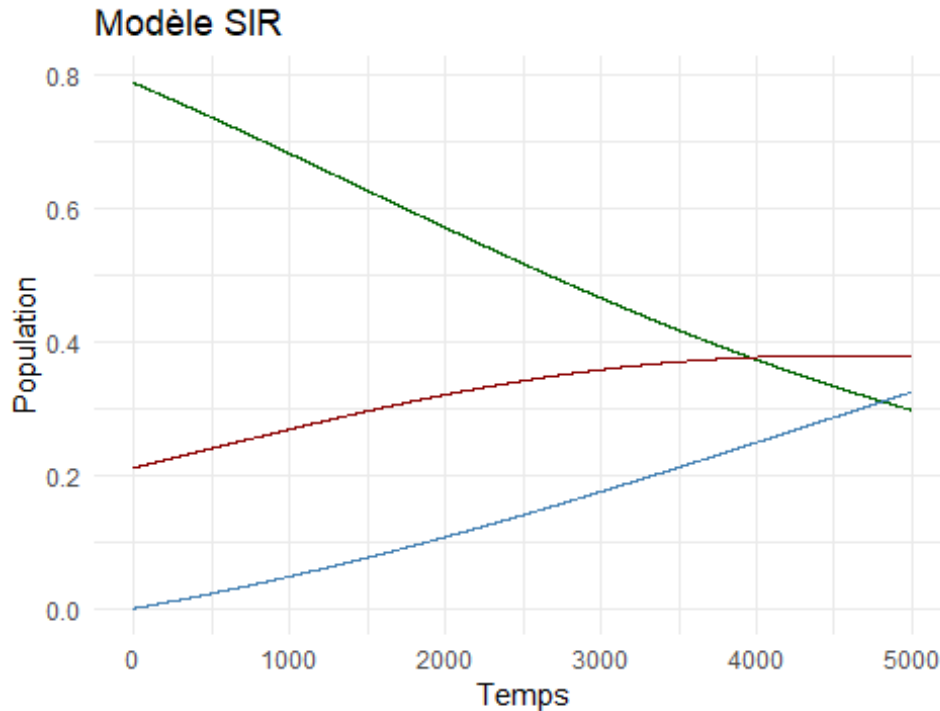
Hypothèse 2 : Nous pouvons agir sur l'infection des personnes saines (confinement, vaccin ou distanciation sociale par exemple)

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 : 50 jours
- $\Delta t = 0.01$


```
df3 <- sir(50,0.01,0.78885555,0.06,0.02)
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 guéries(en bleu).")+
theme(plot.caption = element_text(hjust = 0.5, face = "italic", size =10))
```



Représentation graphique des populaions saines(en vert), infectées(en rouge) et guérie

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 guérison γ et taux de transmission β

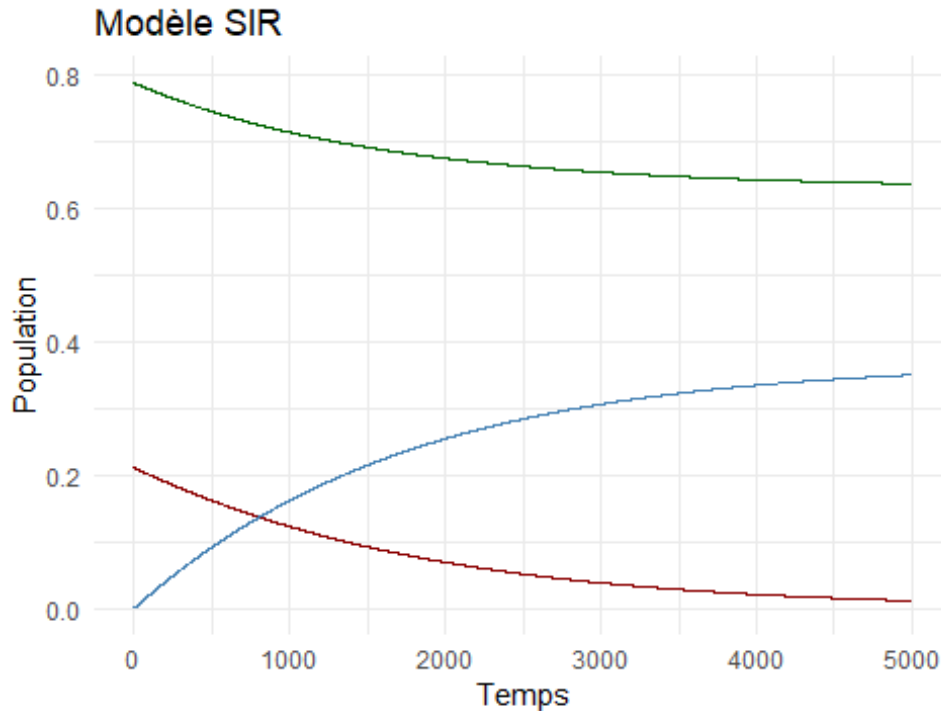
Hypothèse 3 : Nous pouvons agir les deux paramètres en même temps.

Diminuons donc le taux de transmission β et augmentons le taux de guérison 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 : 50jours
- $\Delta t = 0.01$

```
df4 <- sir(50,0.01,0.78885555,0.06,0.099)
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 guéries(en bleu).")+
theme(plot.caption = element_text(hjust = 0.5, face = "italic", size =10))
```



Représentation graphique des populaions saines(en vert), infectées(en rouge) et guérie

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 études précédente, nous avons modélisé différentes courbes avec des valeurs du taux de reproduction R_0 maitrisées, car on donnait 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 déterminer. 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 restrictif 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.

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 s_0 et i_0 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 restrictif ou peu restrictif)

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

```
beta <- tirageBeta(0.8,1)
print(beta)

## [1] 1.462373

gamma <- tirageGamma(0.02,1)
print(gamma)

## [1] 0.4740336

R0 <- beta/gamma
print(R0)

## [1] 3.084956

s <- sir(50,0.1,0.7888999,beta,gamma)
summary(s)

##           j           resS           resI           resR
## Min.      : 1.0      Min.      :0.03818      Min.      :0.0000000      Min.      :0.0000
## 1st Qu.:125.8      1st Qu.:0.03818      1st Qu.:0.0000002      1st Qu.:0.9522
## Median :250.5      Median :0.03819      Median :0.0000412      Median :0.9618
## Mean    :250.5      Mean    :0.06681      Mean    :0.0405800      Mean    :0.8926
## 3rd Qu.:375.2      3rd Qu.:0.03934      3rd Qu.:0.0084701      3rd Qu.:0.9618
## Max.    :500.0      Max.    :0.78890      Max.    :0.3947308      Max.    :0.9618

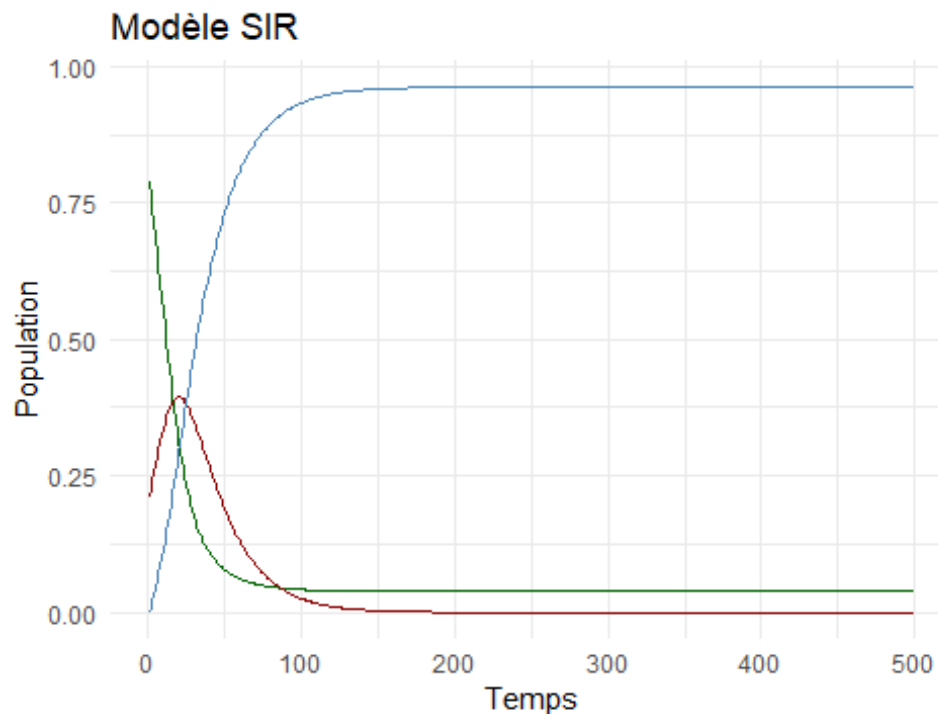
head(s)

##    j      resS      resI      resR
## 1 1 0.7888999 0.2111001 0.0000000
## 2 2 0.7645460 0.2254471 0.01000685
## 3 3 0.7393399 0.2399663 0.02069380
## 4 4 0.7133949 0.2545361 0.03206901
```

```
## 5 5 0.6868405 0.2690247 0.04413488
## 6 6 0.6598192 0.2832933 0.05688755
```

```
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 guéries(en bleu).")+
theme(plot.caption = element_text(hjust = 0.5, face = "italic", size =10))
```



présentation graphique des populaions saines(en vert), infectées(en rouge) et guéries(en bleu).

```
p <- picI(beta,gamma,s)
print(p)
```

```
##          PicI datePicI      R0pic
## 1 0.3947308        20 3.084956
```

Replicate()

```
list_res <- data.frame(replicate(50,picI2(10,0.01,0.7888999,0.6,0.02,1,1),
simplify=T))
print(list_res)
```

```
##          X1          X2          X3          X4          X5          X6
X7
## PicI      0.6364677  0.2111001 0.7675024 0.4749209 0.2111001 0.9501379
0.405147
```

## datePicI	245	1	287	195	1	177
624						
## R0pic	7.690111	0.03901167	14.82965	4.16084	0.8223875	107.9551
3.276072						
##	X8	X9	X10	X11	X12	X13
X14						
## PicI	0.2111001	0.2111001	0.2988383	0.2111001	0.2111001	0.2111001
0.2111001						
## datePicI	1	1	81	1	1	1
1						
## R0pic	0.613625	0.9880726	2.227115	0.2011787	0.8522471	0.4865322
0.6843618						
##	X15	X16	X17	X18	X19	X20
X21						
## PicI	0.2111001	0.7299613	0.2111001	0.6621024	0.421437	0.5779546
0.2111001						
## datePicI	1	208	1	554	137	131
1						
## R0pic	0.9854238	11.96761	0.5858173	8.623072	3.454734	6.047925
0.6113974						
##	X22	X23	X24	X25	X26	X27
X28						
## PicI	0.2111001	0.2711638	0.2139448	0.4144181	0.2111001	0.8183913
0.371951						
## datePicI	1	636	22	150	1	431
159						
## R0pic	0.1534436	1.992087	1.380535	3.373613	1.181757	20.87692
2.914072						
##	X29	X30	X31	X32	X33	X34
X35						
## PicI	0.6758268	0.8723272	0.2111001	0.2111001	0.2111001	0.2665196
0.2111001						
## datePicI	535	198	1	1	1	467
1						
## R0pic	9.180026	33.1658	0.2454865	0.2649945	0.564932	1.951024
0.148335						
##	X36	X37	X38	X39	X40	X41
X42						
## PicI	0.2377954	0.2111001	0.2111001	0.2111001	0.2111001	0.3528466
0.2111001						
## datePicI	187	1	1	1	1	241
1						
## R0pic	1.685685	0.4279274	0.08889067	0.3512551	0.6310941	2.72717
1.220081						
##	X43	X44	X45	X46	X47	X48
X49						
## PicI	0.4039932	0.3442771	0.2402465	0.2220946	0.3314925	0.2111001
0.2111001						
## datePicI	560	204	209	58	97	1
1						

```
## R0pic      3.262776   2.64432   1.709905   1.510708   2.519949  0.2382321
0.9127683
##           X50
## PicI      0.2233853
## datePicI   122
## R0pic      1.528255
```

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

```
##           PicI datePicI      R0pic
## X1  0.6364677      245  7.69011124
## X2  0.2111001        1  0.03901167
## X3  0.7675024      287 14.82964842
## X4  0.4749209      195  4.16084014
## X5  0.2111001        1  0.82238746
## X6  0.9501379      177 107.95506430
## X7  0.4051470      624  3.27607211
## X8  0.2111001        1  0.61362498
## X9  0.2111001        1  0.98807257
## X10 0.2988383       81  2.22711536
## X11 0.2111001        1  0.20117875
## X12 0.2111001        1  0.85224709
## X13 0.2111001        1  0.48653223
## X14 0.2111001        1  0.68436183
## X15 0.2111001        1  0.98542378
## X16 0.7299613      208 11.96761042
## X17 0.2111001        1  0.58581730
## X18 0.6621024      554  8.62307185
## X19 0.4214370      137  3.45473402
## X20 0.5779546      131  6.04792531
## X21 0.2111001        1  0.61139743
## X22 0.2111001        1  0.15344357
## X23 0.2711638      636  1.99208687
## X24 0.2139448       22  1.38053539
## X25 0.4144181      150  3.37361339
## X26 0.2111001        1  1.18175670
## X27 0.8183913      431 20.87691648
## X28 0.3719510      159  2.91407175
## X29 0.6758268      535  9.18002576
## X30 0.8723272      198 33.16580141
## X31 0.2111001        1  0.24548648
## X32 0.2111001        1  0.26499453
## X33 0.2111001        1  0.56493203
## X34 0.2665196      467  1.95102356
## X35 0.2111001        1  0.14833501
## X36 0.2377954      187  1.68568452
## X37 0.2111001        1  0.42792739
## X38 0.2111001        1  0.08889067
## X39 0.2111001        1  0.35125508
```

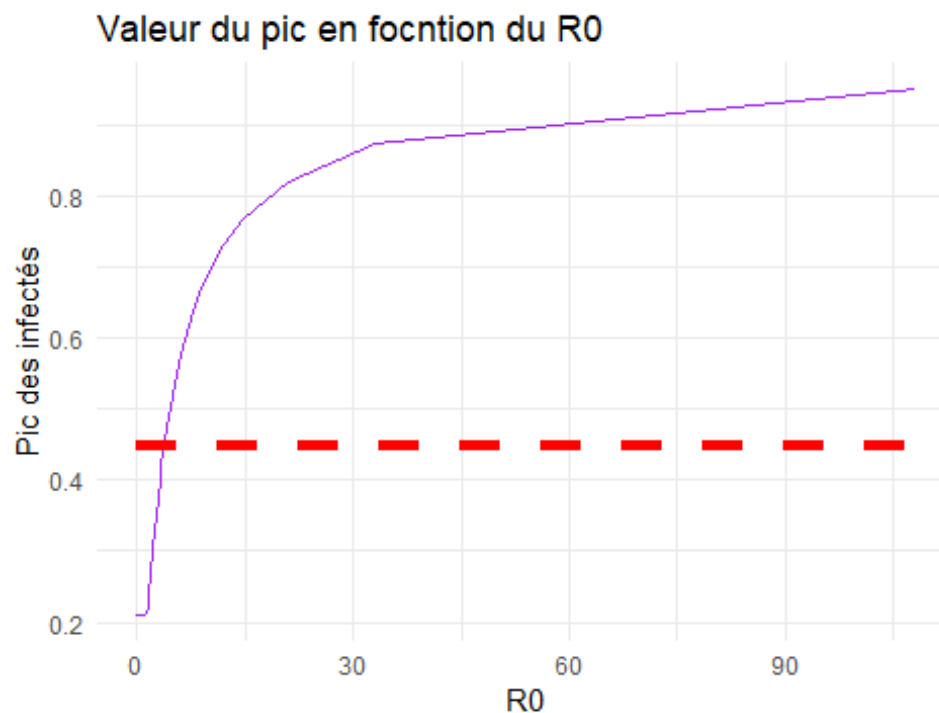
```
## X40 0.2111001      1  0.63109410
## X41 0.3528466    241  2.72717025
## X42 0.2111001      1  1.22008127
## X43 0.4039932    560  3.26277646
## X44 0.3442771    204  2.64432033
## X45 0.2402465    209  1.70990467
## X46 0.2220946     58  1.51070804
## X47 0.3314925     97  2.51994926
## X48 0.2111001      1  0.23823207
## X49 0.2111001      1  0.91276826
## X50 0.2233853    122  1.52825514
```

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

```
##      PicI datePicI      R0pic
## X2  0.2111001      1  0.03901167
## X38 0.2111001      1  0.08889067
## X35 0.2111001      1  0.14833501
## X22 0.2111001      1  0.15344357
## X11 0.2111001      1  0.20117875
## X48 0.2111001      1  0.23823207
## X31 0.2111001      1  0.24548648
## X32 0.2111001      1  0.26499453
## X39 0.2111001      1  0.35125508
## X37 0.2111001      1  0.42792739
## X13 0.2111001      1  0.48653223
## X33 0.2111001      1  0.56493203
## X17 0.2111001      1  0.58581730
## X21 0.2111001      1  0.61139743
## X8  0.2111001      1  0.61362498
## X40 0.2111001      1  0.63109410
## X14 0.2111001      1  0.68436183
## X5  0.2111001      1  0.82238746
## X12 0.2111001      1  0.85224709
## X49 0.2111001      1  0.91276826
## X15 0.2111001      1  0.98542378
## X9  0.2111001      1  0.98807257
## X26 0.2111001      1  1.18175670
## X42 0.2111001      1  1.22008127
## X24 0.2139448     22  1.38053539
## X46 0.2220946     58  1.51070804
## X50 0.2233853    122  1.52825514
## X36 0.2377954    187  1.68568452
## X45 0.2402465    209  1.70990467
## X34 0.2665196    467  1.95102356
## X23 0.2711638    636  1.99208687
## X10 0.2988383     81  2.22711536
## X47 0.3314925     97  2.51994926
## X44 0.3442771    204  2.64432033
```

```
## X41 0.3528466      241  2.72717025
## X28 0.3719510      159  2.91407175
## X43 0.4039932      560  3.26277646
## X7  0.4051470      624  3.27607211
## X25 0.4144181      150  3.37361339
## X19 0.4214370      137  3.45473402
## X4  0.4749209      195  4.16084014
## X20 0.5779546      131  6.04792531
## X1  0.6364677      245  7.69011124
## X18 0.6621024      554  8.62307185
## X29 0.6758268      535  9.18002576
## X16 0.7299613      208  11.96761042
## X3  0.7675024      287  14.82964842
## X27 0.8183913      431  20.87691648
## X30 0.8723272      198  33.16580141
## X6  0.9501379      177  107.95506430
```

```
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 fonction 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

```
g2 <- ggplot(orderConcat, aes(x = R0pic)) + geom_line(aes(y = datePicI),
col="purple")+ labs(title = "Date du pic en fonction du R0", x = "R0", y="Pic
des infectés", color="red") + labs(caption = "Représentation graphique des
```



```
valeurs des pics à différents  $R_0$ ") + theme(plot.caption = element_text(hjust = 0.5, face = "italic", size = 10))
```

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

IV. Complexification et réalisme du modèle SIR

1. Réalisme

Le modèle SIR est un modèle simple, qui considèrerait 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.

2. Modèle SEIR

Dans ce modèle, nous prenons en compte la démographie, et donc nous aurons une évolution de $N(t)$ au cours du temps.

On considèrera 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** (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** et d'introduire un nouveau paramètre α : le **taux d'incubation d'une maladie** n représente une naissance et m un décès.

```


$$\begin{aligned}
& \frac{dN}{dt} = \nu - \mu N \\
& \frac{dS}{dt} = \beta \frac{I}{N} S - \alpha E - \mu S \\
& \frac{dE}{dt} = \alpha E - \mu E \\
& \frac{dI}{dt} = \mu E - \mu I
\end{aligned}$$


```

De ce fait, nos équations différentielles seront modifiées :

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

3. Résolution et courbes

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

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

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'à un instant t , la taille de la population sera :

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

```
dfseir <- seir(100,0.01,0.88889999,0.8,0.05,0.75,0,0)
```

```
head(dfseir)
```

```
##      j      resS      resE      resI      resR resN
## 1 1 0.8889000 0.05555001 0.05555001 0.000000e+00 1
## 2 2 0.8885050 0.05552841 0.05593886 2.777500e-05 1
## 3 3 0.8881073 0.05550956 0.05632735 5.574443e-05 1
```

```
## 4 4 0.8877071 0.05549344 0.05671551 8.390810e-05 1
## 5 5 0.8873044 0.05548001 0.05710335 1.122659e-04 1
## 6 6 0.8868990 0.05546925 0.05749090 1.408175e-04 1

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")+
  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))
```

