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* , *infectés* et *retirés* . Dans ce modèle, on considère les personnes retirées comme immunisées ou mortes, c’est pourquoi on différentie les deux sous-populations et .

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:

Précisons que l’état du système à un instant t donné est défini par les trois nombres , , 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 et qui nous permettront de définir le **taux de transmission**  et le **taux de guérison** .

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

Le taux de transmission est donc le passage des personnes saines à infectées et le taux de guérison 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 ,, 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 :

De plus,on note :

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) :

* (valeur très proche de 1)
* (valeur très proche de 0)
* (on considère aucune personne morte ou immunisée au début de l’épidémie)

A t=0 on peut écrire :

Nous allons définir le **taux de reproduction**  comme le nombre moyen de cas secondaires produit par un individu infectieux au cours de sa période contagieuse. La valeur que prend 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 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 . En effet, nous savons que est forcément positif, donc c’est bien la valeur de qui determine le signe de .

* Si alors , ce qui veut dire que *décroit*, l’épidémie prend fin.
* Si alors , ce qui veut dire que *croit*, et atteint une valeur maximale : c’est le **pic de l’épidemie**.

On constate donc que pour fixé, plus 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 () 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 , le nombre d’individus sains passe de à , et la variation de peut s’écrire :

Nous voulons isoler , et afin de former une suite :

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

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

Ce qui va nous interresser, c’est de pouvoir réaliser ces calcul avec R.

#### 2.4. Initialisation

Au début de l’épidémie (t=0) on a personnes saines, et personnes infectées.

Il n’y a aucune personne retirées au début de l’épidémie.

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
* Le taux de guérison
* Par extension de ces deux paramètres, le
* La proportion de personnes 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 paramètres aléatoires de et de (qu’on suppose ne pas connaitre) :

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

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

## [1] 0.4961657

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

## [1] 0.6028417

R0 <- beta/gamma  
print(R0)

## [1] 0.8230447

#### 2.5 Courbes

Pour cet exemple, choisissons :

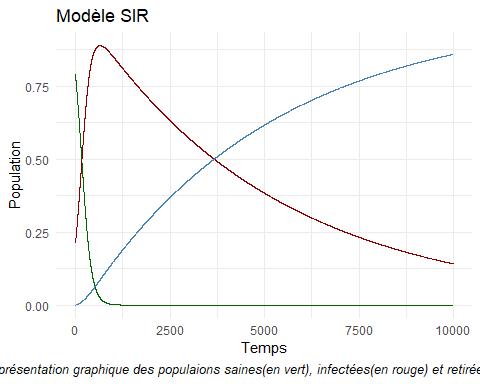
* Taux de transmission
* Taux de guérison
* Proportion de personnes saines au début de l’épidémie (choix arbitraire)
* Durée de la simulation : 100jours

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

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

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))



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%)) et se touve aux alentour 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 ammener des premières conclusions à ce modèle.

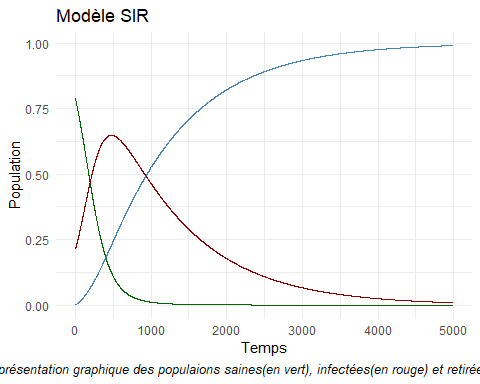
### 3.1 Taux de guérison

*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 :

* Taux de transmission
* Taux de guérison
* Proportion de personnes saines au début de l’épidémie
* Durée de la simulation : 100 jours

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))



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

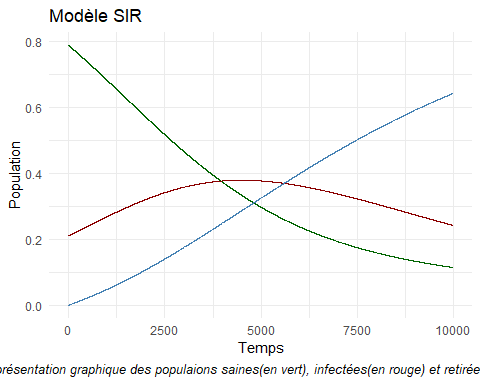
### 3.2 Taux de transmission

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

Diminuons donc le taux de transmission :

* Taux de transmission
* Taux de guérison
* Proportion de personnes saines au début de l’épidémie
* Durée de la simulation : 100 jours

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))



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).

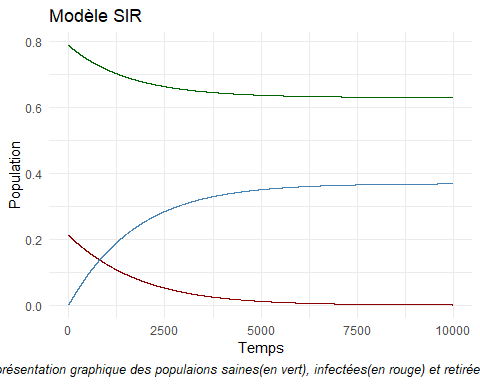
### 3.3 Taux de guérion et taux de transmission

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

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

* Taux de transmission
* Taux de guérison
* Proportion de personnes saines au début de l’épidémie
* Durée de la simulation : 100jours

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))



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

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

En effet, la connaissance précise du 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 et , et regardons comment le nous permet de prédire le pic.

Si on suppose 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 et données, on veut trouver quand et de combien réduire le 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] 0.8505226

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

## [1] 0.1611024

R0 <- beta/gamma  
print(R0)

## [1] 5.279392

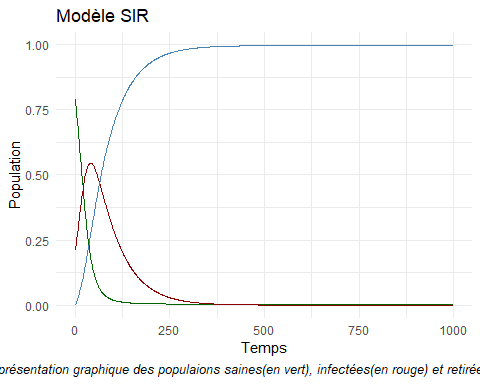
s <- sir(100,0.1,0.7888999,beta,gamma)  
summary(s, simplify=T)

## j resS resI resR   
## Min. : 1.0 Min. :0.003780 Min. :0.0000002 Min. :0.0000   
## 1st Qu.: 250.8 1st Qu.:0.003780 1st Qu.:0.0000100 1st Qu.:0.9672   
## Median : 500.5 Median :0.003791 Median :0.0005348 Median :0.9957   
## Mean : 500.5 Mean :0.027031 Mean :0.0618377 Mean :0.9111   
## 3rd Qu.: 750.2 3rd Qu.:0.004406 3rd Qu.:0.0283749 3rd Qu.:0.9962   
## Max. :1000.0 Max. :0.788900 Max. :0.5455182 Max. :0.9962

head(s)

## j resS resI resR  
## 1 1 0.7888999 0.2111001 0.000000000  
## 2 2 0.7747356 0.2218636 0.003400873  
## 3 3 0.7601163 0.2329085 0.006975148  
## 4 4 0.7450589 0.2442138 0.010727360  
## 5 5 0.7295833 0.2557550 0.014661702  
## 6 6 0.7137130 0.2675050 0.018781976

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))



picI(beta,gamma,s)

## PicI datePicI R0pic  
## 1 0.5455182 41 5.279392

## 2. Replicate()

list\_res <- data.frame(replicate(50,picI2(10,0.01,0.7888999,0.01,0.99), simplify=T))  
print(list\_res)

## X1 X2 X3 X4 X5 X6 X7  
## PicI 0.2715939 0.5519546 0.2308034 0.3204599 0.77006 0.2689237 0.2111001  
## datePicI 656 586 183 263 583 253 1  
## R0pic 1.995879 5.505324 1.613701 2.425058 15.09804 1.97138 0.5826928  
## X8 X9 X10 X11 X12 X13 X14  
## PicI 0.2111001 0.2111001 0.2111001 0.2111001 0.4052794 0.2111001 0.2111001  
## datePicI 1 1 1 1 917 1 1  
## R0pic 0.2783275 0.4258153 0.9450405 0.1119651 3.278394 0.5828375 0.4555015  
## X15 X16 X17 X18 X19 X20  
## PicI 0.2111001 0.2111001 0.5099296 0.2111001 0.2111001 0.2111001  
## datePicI 1 1 380 1 1 1  
## R0pic 0.8471875 0.4197167 4.716954 0.06199566 0.3596948 0.06040078  
## X21 X22 X23 X24 X25 X26 X27  
## PicI 0.2111001 0.2116863 0.2111001 0.2900128 0.2255805 0.3912663 0.2111001  
## datePicI 1 21 1 215 167 325 1  
## R0pic 0.3934802 1.31679 0.2252816 2.154946 1.555087 3.120614 0.431884  
## X28 X29 X30 X31 X32 X33 X34  
## PicI 0.575942 0.2111001 0.2111001 0.6223285 0.2111001 0.2111001 0.2756418  
## datePicI 433 1 1 444 1 1 359  
## R0pic 6.028269 0.5769253 1.017821 7.258574 1.058549 0.9472931 2.030694  
## X35 X36 X37 X38 X39 X40 X41  
## PicI 0.2111001 0.5292718 0.2111001 0.2111001 0.2111001 0.2111001 0.2111001  
## datePicI 1 351 1 1 1 1 1  
## R0pic 0.9005151 5.058027 0.7634915 0.8010047 0.6074224 1.079509 0.9609743  
## X42 X43 X44 X45 X46 X47 X48  
## PicI 0.2111001 0.2111001 0.2565502 0.269002 0.2111001 0.2111001 0.7108928  
## datePicI 1 1 646 270 1 1 572  
## R0pic 0.4001582 0.6713976 1.862565 1.972177 0.2748833 1.067672 10.87949  
## X49 X50  
## PicI 0.2111001 0.2539969  
## datePicI 1 668  
## R0pic 0.5019448 1.839504

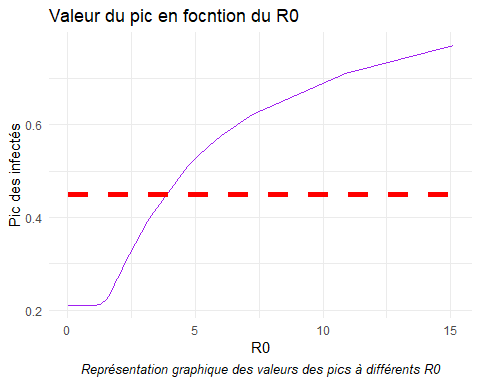
concat <- do.call(rbind.data.frame,list\_res)  
head(concat)

## PicI datePicI R0pic  
## X1 0.2715939 656 1.995879  
## X2 0.5519546 586 5.505324  
## X3 0.2308034 183 1.613701  
## X4 0.3204599 263 2.425058  
## X5 0.7700600 583 15.098035  
## X6 0.2689237 253 1.971380

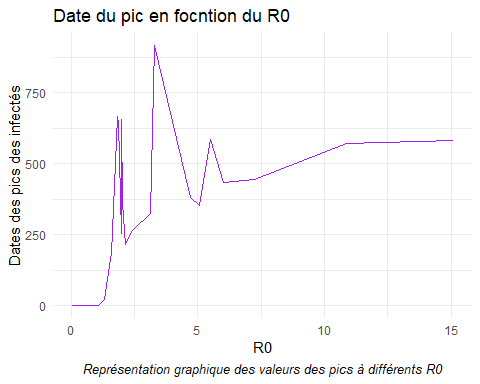
orderConcat <- concat[order(concat[,3],decreasing=F),]  
print(orderConcat)

## PicI datePicI R0pic  
## X20 0.2111001 1 0.06040078  
## X18 0.2111001 1 0.06199566  
## X11 0.2111001 1 0.11196505  
## X23 0.2111001 1 0.22528160  
## X46 0.2111001 1 0.27488331  
## X8 0.2111001 1 0.27832746  
## X19 0.2111001 1 0.35969478  
## X21 0.2111001 1 0.39348024  
## X42 0.2111001 1 0.40015816  
## X16 0.2111001 1 0.41971669  
## X9 0.2111001 1 0.42581533  
## X27 0.2111001 1 0.43188403  
## X14 0.2111001 1 0.45550148  
## X49 0.2111001 1 0.50194481  
## X29 0.2111001 1 0.57692532  
## X7 0.2111001 1 0.58269282  
## X13 0.2111001 1 0.58283749  
## X39 0.2111001 1 0.60742235  
## X43 0.2111001 1 0.67139762  
## X37 0.2111001 1 0.76349147  
## X38 0.2111001 1 0.80100473  
## X15 0.2111001 1 0.84718746  
## X35 0.2111001 1 0.90051510  
## X10 0.2111001 1 0.94504046  
## X33 0.2111001 1 0.94729311  
## X41 0.2111001 1 0.96097431  
## X30 0.2111001 1 1.01782067  
## X32 0.2111001 1 1.05854919  
## X47 0.2111001 1 1.06767163  
## X40 0.2111001 1 1.07950935  
## X22 0.2116863 21 1.31679008  
## X25 0.2255805 167 1.55508709  
## X3 0.2308034 183 1.61370148  
## X50 0.2539969 668 1.83950431  
## X44 0.2565502 646 1.86256465  
## X6 0.2689237 253 1.97137974  
## X45 0.2690020 270 1.97217687  
## X1 0.2715939 656 1.99587851  
## X34 0.2756418 359 2.03069395  
## X24 0.2900128 215 2.15494630  
## X4 0.3204599 263 2.42505826  
## X26 0.3912663 325 3.12061406  
## X12 0.4052794 917 3.27839384  
## X17 0.5099296 380 4.71695394  
## X36 0.5292718 351 5.05802688  
## X2 0.5519546 586 5.50532377  
## X28 0.5759420 433 6.02826859  
## X31 0.6223285 444 7.25857416  
## X48 0.7108928 572 10.87948745  
## X5 0.7700600 583 15.09803505

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))



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))



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

## 3. Conclusion

Nous avons tenté de démontrer la corrélation entre la valeur aléatoire du et la date ou le nombre d’infectées 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 de places en réanimation par exemple, et si le nombre de personnes necessitant une place était suppérieur 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 acceptable, et s’il s’avère que le la valeur du 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** par exemple, en instaurant une distanciation sociale, ou rendre obligatoire le port des masques dans les espaces clos. Si les scientifiques et les médecins parvennaient à trouver un médicament efficace, cela pourra augmenter le **taux de guérison** .

# 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 négilgées.

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 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 au cours du temps.

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

On considèrera également que les personnes qui meurent, peuvent être dans n’importe quelle sous-population à l’instant (la mort n’est pas toujours liée au virus). C’est le **taux de mortalité** .

Une nouvelle sous-population 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 : 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 :

### 3. Résolution et courbes

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

Nous voulons isoler , , et afin de former une suite :

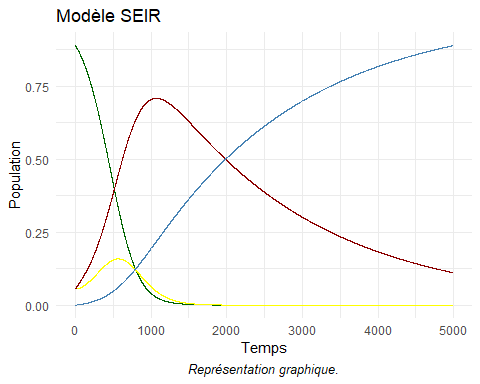
Nous avons cherché à décrire l’évolution de la, population en fonction des naissances et des décès.

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

dfseir <- seir(50,0.01,0.88889999,0.8,0.05,0.75,0,0)  
  
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))



Le pic correspondant au graphe ci dessus :

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

## $PicI  
## [1] 0.7079925  
##   
## $datePicI  
## [1] 1071  
##   
## $R0pic  
## [1] 16

Il faut en plus réaliser un **tirage du taux d’incubation** :

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

## [1] 0.2291469

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

## [1] 0.2520423

tirageAlpha(0.01,0.99)

## [1] 0.1769974

picI2Seir(50, 0.01, 0.788554,0.01,0.99)

## PicI datePicI R0pic  
## 1 0.5758993 854 8.92508

list\_resSeir <- data.frame(replicate(50,picI2Seir(50, 0.01,0.788554,0.01,0.99), simplify=T))  
print(list\_resSeir)

## X1 X2 X3 X4 X5 X6 X7  
## PicI 0.5419349 0.120559 0.105723 0.105723 0.1491786 0.1067057 0.105723  
## datePicI 1463 82 1 1 176 62 1  
## R0pic 5.939743 0.2752732 0.6318659 0.3323571 0.6376991 1.15421 0.9460118  
## X8 X9 X10 X11 X12 X13 X14  
## PicI 0.105723 0.1084765 0.105723 0.105723 0.12389 0.3761687 0.8120859  
## datePicI 1 60 1 1 120 800 1316  
## R0pic 0.08155229 0.6791732 0.9456999 0.7481938 1.132106 4.578269 35.36006  
## X15 X16 X17 X18 X19 X20 X21  
## PicI 0.105723 0.1062836 0.1126142 0.5695687 0.1799298 0.105723 0.131274  
## datePicI 1 16 59 831 324 1 172  
## R0pic 1.747003 0.1931463 0.4134825 8.608734 1.07492 0.6039653 1.222999  
## X22 X23 X24 X25 X26 X27 X28  
## PicI 0.1085069 0.105723 0.5074183 0.1124922 0.105723 0.3011647 0.111408  
## datePicI 33 1 925 87 1 723 51  
## R0pic 0.4639907 0.2701008 9.038521 0.5034297 0.8602206 3.089278 0.6857155  
## X29 X30 X31 X32 X33 X34 X35  
## PicI 0.105723 0.105723 0.4180834 0.2575413 0.105723 0.105723 0.6664986  
## datePicI 1 1 1172 830 1 1 759  
## R0pic 1.560881 0.6409016 4.92045 3.071174 0.7889695 0.7582521 12.62562  
## X36 X37 X38 X39 X40 X41 X42  
## PicI 0.1284457 0.1081782 0.2024262 0.105723 0.4390816 0.1237816 0.1354054  
## datePicI 90 42 670 1 1278 85 114  
## R0pic 0.1758159 0.14038 1.952342 0.4059554 6.83203 0.1656856 0.05976471  
## X43 X44 X45 X46 X47 X48 X49  
## PicI 0.105723 0.1178549 0.105723 0.1185226 0.105723 0.196897 0.105723  
## datePicI 1 454 1 72 1 772 1  
## R0pic 1.328268 1.623945 0.02066293 0.4318905 0.2669008 2.233336 0.4642711  
## X50  
## PicI 0.1369357  
## datePicI 143  
## R0pic 0.9086637

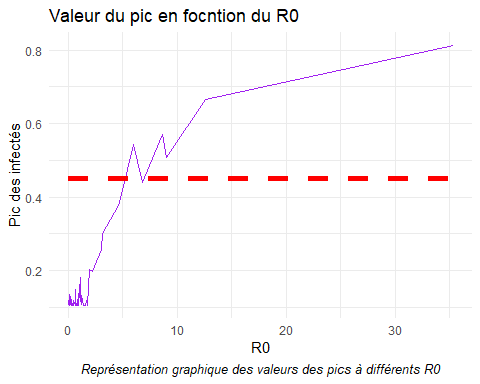
concatSeir <- do.call(rbind.data.frame,list\_resSeir)  
head(concatSeir)

## PicI datePicI R0pic  
## X1 0.5419349 1463 5.9397433  
## X2 0.1205590 82 0.2752732  
## X3 0.1057230 1 0.6318659  
## X4 0.1057230 1 0.3323571  
## X5 0.1491786 176 0.6376991  
## X6 0.1067057 62 1.1542102

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

## PicI datePicI R0pic  
## X45 0.1057230 1 0.02066293  
## X42 0.1354054 114 0.05976471  
## X8 0.1057230 1 0.08155229  
## X37 0.1081782 42 0.14037996  
## X41 0.1237816 85 0.16568555  
## X36 0.1284457 90 0.17581589  
## X16 0.1062836 16 0.19314632  
## X47 0.1057230 1 0.26690075  
## X23 0.1057230 1 0.27010077  
## X2 0.1205590 82 0.27527324  
## X4 0.1057230 1 0.33235711  
## X39 0.1057230 1 0.40595541  
## X17 0.1126142 59 0.41348246  
## X46 0.1185226 72 0.43189055  
## X22 0.1085069 33 0.46399072  
## X49 0.1057230 1 0.46427111  
## X25 0.1124922 87 0.50342970  
## X20 0.1057230 1 0.60396528  
## X3 0.1057230 1 0.63186587  
## X5 0.1491786 176 0.63769911  
## X30 0.1057230 1 0.64090158  
## X9 0.1084765 60 0.67917316  
## X28 0.1114080 51 0.68571553  
## X11 0.1057230 1 0.74819383  
## X34 0.1057230 1 0.75825212  
## X33 0.1057230 1 0.78896955  
## X26 0.1057230 1 0.86022060  
## X50 0.1369357 143 0.90866367  
## X10 0.1057230 1 0.94569989  
## X7 0.1057230 1 0.94601181  
## X19 0.1799298 324 1.07492031  
## X12 0.1238900 120 1.13210632  
## X6 0.1067057 62 1.15421016  
## X21 0.1312740 172 1.22299852  
## X43 0.1057230 1 1.32826750  
## X29 0.1057230 1 1.56088077  
## X44 0.1178549 454 1.62394539  
## X15 0.1057230 1 1.74700339  
## X38 0.2024262 670 1.95234216  
## X48 0.1968970 772 2.23333637  
## X32 0.2575413 830 3.07117361  
## X27 0.3011647 723 3.08927822  
## X13 0.3761687 800 4.57826903  
## X31 0.4180834 1172 4.92045042  
## X1 0.5419349 1463 5.93974325  
## X40 0.4390816 1278 6.83203012  
## X18 0.5695687 831 8.60873396  
## X24 0.5074183 925 9.03852134  
## X35 0.6664986 759 12.62561796  
## X14 0.8120859 1316 35.36006339

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))



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))

