PROJET 4a : le modèle SIR – RMD

12 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 *guéris* . Dans ce modèle, on considère les personnes guéries comme immunisées, 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 guéries.

#### 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 interressantes 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 guéries 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 produits par un individu infectieux au cours de sa période d’infection. 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 gouvernement 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 contamminations.

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**.

En effet, si un malade peut contaminer plus d’une personne () la maladie va flamber.

### 2. Simulation de l’épidémie avec R

#### 2.1. Package

Installation du package crée pour ce projet :

#devtools::install\_github("ZoeGerber/ModeleSIR")  
library(ModeleSIR)

Dans ce pakage ce trouve 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 guéries 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)

1. Fractions initiales affichées (à t=0) de N qui sont Sains, Infectés et Guéris :

init(0.9)

## [1] 0.9 0.1 0.0

1. Paramètres variables et de , qu’on suppose ne pas connaitre :

Nous avons choisi de modéliser le par une loi normale.

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

## [1] 1.446534

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

## [1] 0.7213564

R0 <- beta/gamma  
print(R0)

## [1] 2.005298

#### 2.5 Courbes

Pour cet exemple, choisissons :

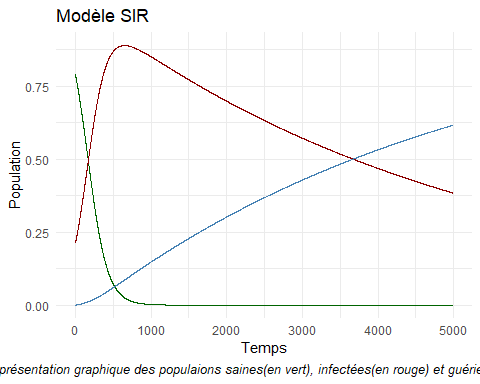
* Taux de transmission
* Taux de guérison
* Proportion de personnes saines au début de l’épidémie
* Durée de la simulation : 50jours

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(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))



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 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.

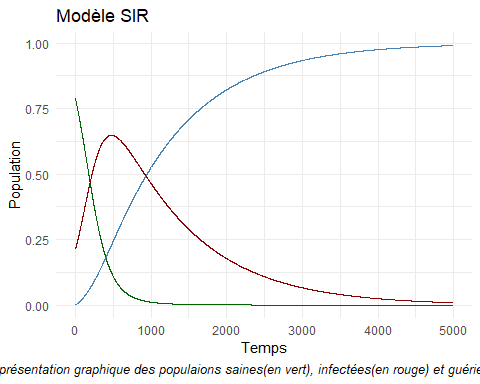
### 3.1 Taux de guérison

*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 : 50 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 guéries(en bleu).")+ theme(plot.caption = element\_text(hjust = 0.5, face = "italic", size =10))



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.

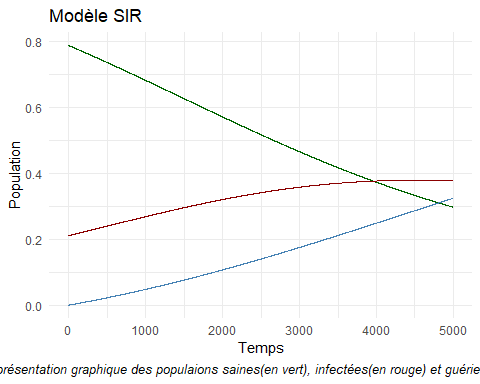
### 3.2 Taux de transmission

*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 : 50 jours

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))



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).

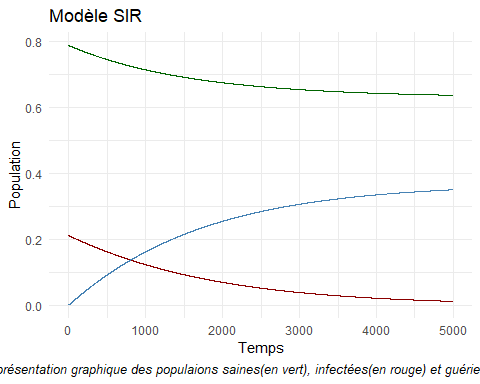
### 3.3 Taux de guérion et taux de transmission

*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 : 50jours

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))



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 R0 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.

### 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(0.8,1)  
print(beta)

## [1] 0.08129828

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

## [1] 0.7579527

R0 <- beta/gamma  
print(R0)

## [1] 0.1072604

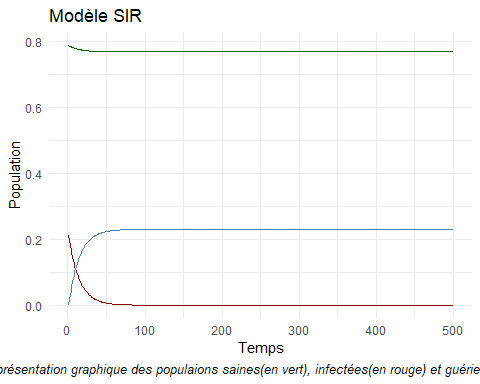
s <- sir(50,0.1,0.7888999,beta,gamma)  
summary(s)

## j resS resI resR   
## Min. : 1.0 Min. :0.7696 Min. :0.000e+00 Min. :0.0000   
## 1st Qu.:125.8 1st Qu.:0.7696 1st Qu.:0.000e+00 1st Qu.:0.2303   
## Median :250.5 Median :0.7696 Median :0.000e+00 Median :0.2304   
## Mean :250.5 Mean :0.7702 Mean :6.079e-03 Mean :0.2237   
## 3rd Qu.:375.2 3rd Qu.:0.7696 3rd Qu.:2.636e-05 3rd Qu.:0.2304   
## Max. :500.0 Max. :0.7889 Max. :2.111e-01 Max. :0.2304

head(s)

## j resS resI resR  
## 1 1 0.7888999 0.2111001 0.00000000  
## 2 2 0.7875460 0.1964536 0.01600039  
## 3 3 0.7862882 0.1828212 0.03089064  
## 4 4 0.7851195 0.1701329 0.04474763  
## 5 5 0.7840336 0.1583235 0.05764289  
## 6 6 0.7830244 0.1473325 0.06964307

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))



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

## PicI datePicI R0pic  
## 1 0.2111001 1 0.1072604

## 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 0.2111001 0.417067 0.2111001 0.2111001 0.4760439 0.3877665 0.738528  
## datePicI 1 1000 1 1 171 213 209  
## R0pic 0.2508455 4.248454 0.1670284 0.08888293 4.175282 3.080745 12.53489  
## X8 X9 X10 X11 X12 X13 X14  
## PicI 0.2111001 0.3798572 0.2850345 0.212403 0.942027 0.2600566 0.6823258  
## datePicI 1 213 87 18 507 103 256  
## R0pic 0.8467383 2.997446 2.107778 1.341749 90.53759 1.890841 9.441882  
## X15 X16 X17 X18 X19 X20 X21  
## PicI 0.4057421 0.2111001 0.2111001 0.2111001 0.4401418 0.2111001 0.2132908  
## datePicI 236 1 1 1 260 1 38  
## R0pic 3.278517 0.1154699 0.188928 0.2595451 3.691171 0.7464862 1.366895  
## X22 X23 X24 X25 X26 X27 X28  
## PicI 0.7472568 0.4139762 0.5963182 0.2111001 0.2846783 0.2443707 0.2946872  
## datePicI 474 441 311 1 434 68 139  
## R0pic 13.20295 3.37611 6.524408 0.1947151 2.109755 1.746572 2.194188  
## X29 X30 X31 X32 X33 X34 X35  
## PicI 0.2111001 0.2976539 0.3378355 0.7947439 0.2111001 0.4467656 0.2111001  
## datePicI 1 187 113 412 1 500 1  
## R0pic 0.06312381 2.221296 2.579924 17.66026 1.207328 3.780463 1.109864  
## X36 X37 X38 X39 X40 X41 X42  
## PicI 0.2111001 0.3023677 0.7019645 0.7147886 0.7439657 0.2464455 0.2111001  
## datePicI 1 104 260 223 433 75 1  
## R0pic 0.8806877 2.259793 10.37925 11.06118 12.95824 1.766411 0.5346417  
## X43 X44 X45 X46 X47 X48 X49  
## PicI 0.4323659 0.2899837 0.2522685 0.2111001 0.2518501 0.3270865 0.3809523  
## datePicI 215 123 168 1 428 144 264  
## R0pic 3.592055 2.152652 1.822276 1.047997 1.819668 2.482619 3.010102  
## X50  
## PicI 0.2111001  
## datePicI 1  
## R0pic 0.6177668

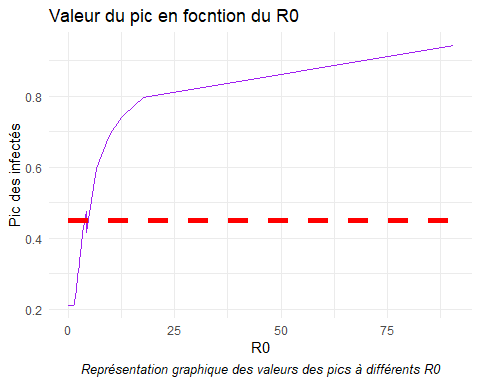
concat <- do.call(rbind.data.frame,list\_res)  
print(concat)

## PicI datePicI R0pic  
## X1 0.2111001 1 0.25084549  
## X2 0.4170670 1000 4.24845370  
## X3 0.2111001 1 0.16702838  
## X4 0.2111001 1 0.08888293  
## X5 0.4760439 171 4.17528224  
## X6 0.3877665 213 3.08074524  
## X7 0.7385280 209 12.53489283  
## X8 0.2111001 1 0.84673833  
## X9 0.3798572 213 2.99744555  
## X10 0.2850345 87 2.10777794  
## X11 0.2124030 18 1.34174852  
## X12 0.9420270 507 90.53759362  
## X13 0.2600566 103 1.89084088  
## X14 0.6823258 256 9.44188191  
## X15 0.4057421 236 3.27851741  
## X16 0.2111001 1 0.11546991  
## X17 0.2111001 1 0.18892805  
## X18 0.2111001 1 0.25954510  
## X19 0.4401418 260 3.69117099  
## X20 0.2111001 1 0.74648616  
## X21 0.2132908 38 1.36689499  
## X22 0.7472568 474 13.20295241  
## X23 0.4139762 441 3.37611011  
## X24 0.5963182 311 6.52440794  
## X25 0.2111001 1 0.19471507  
## X26 0.2846783 434 2.10975516  
## X27 0.2443707 68 1.74657184  
## X28 0.2946872 139 2.19418840  
## X29 0.2111001 1 0.06312381  
## X30 0.2976539 187 2.22129614  
## X31 0.3378355 113 2.57992359  
## X32 0.7947439 412 17.66026067  
## X33 0.2111001 1 1.20732756  
## X34 0.4467656 500 3.78046303  
## X35 0.2111001 1 1.10986412  
## X36 0.2111001 1 0.88068774  
## X37 0.3023677 104 2.25979317  
## X38 0.7019645 260 10.37924568  
## X39 0.7147886 223 11.06117795  
## X40 0.7439657 433 12.95823650  
## X41 0.2464455 75 1.76641051  
## X42 0.2111001 1 0.53464173  
## X43 0.4323659 215 3.59205507  
## X44 0.2899837 123 2.15265179  
## X45 0.2522685 168 1.82227626  
## X46 0.2111001 1 1.04799676  
## X47 0.2518501 428 1.81966843  
## X48 0.3270865 144 2.48261854  
## X49 0.3809523 264 3.01010222  
## X50 0.2111001 1 0.61776676

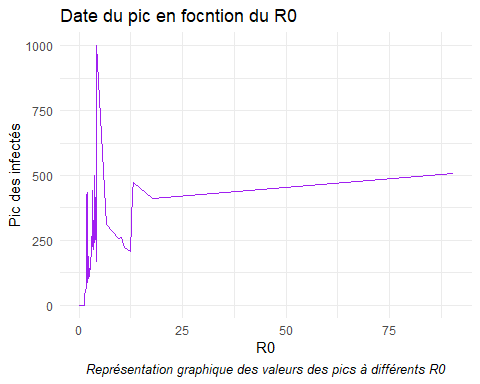
orderConcat <- concat[order(concat[,3],decreasing=F),]  
print(orderConcat)

## PicI datePicI R0pic  
## X29 0.2111001 1 0.06312381  
## X4 0.2111001 1 0.08888293  
## X16 0.2111001 1 0.11546991  
## X3 0.2111001 1 0.16702838  
## X17 0.2111001 1 0.18892805  
## X25 0.2111001 1 0.19471507  
## X1 0.2111001 1 0.25084549  
## X18 0.2111001 1 0.25954510  
## X42 0.2111001 1 0.53464173  
## X50 0.2111001 1 0.61776676  
## X20 0.2111001 1 0.74648616  
## X8 0.2111001 1 0.84673833  
## X36 0.2111001 1 0.88068774  
## X46 0.2111001 1 1.04799676  
## X35 0.2111001 1 1.10986412  
## X33 0.2111001 1 1.20732756  
## X11 0.2124030 18 1.34174852  
## X21 0.2132908 38 1.36689499  
## X27 0.2443707 68 1.74657184  
## X41 0.2464455 75 1.76641051  
## X47 0.2518501 428 1.81966843  
## X45 0.2522685 168 1.82227626  
## X13 0.2600566 103 1.89084088  
## X10 0.2850345 87 2.10777794  
## X26 0.2846783 434 2.10975516  
## X44 0.2899837 123 2.15265179  
## X28 0.2946872 139 2.19418840  
## X30 0.2976539 187 2.22129614  
## X37 0.3023677 104 2.25979317  
## X48 0.3270865 144 2.48261854  
## X31 0.3378355 113 2.57992359  
## X9 0.3798572 213 2.99744555  
## X49 0.3809523 264 3.01010222  
## X6 0.3877665 213 3.08074524  
## X15 0.4057421 236 3.27851741  
## X23 0.4139762 441 3.37611011  
## X43 0.4323659 215 3.59205507  
## X19 0.4401418 260 3.69117099  
## X34 0.4467656 500 3.78046303  
## X5 0.4760439 171 4.17528224  
## X2 0.4170670 1000 4.24845370  
## X24 0.5963182 311 6.52440794  
## X14 0.6823258 256 9.44188191  
## X38 0.7019645 260 10.37924568  
## X39 0.7147886 223 11.06117795  
## X7 0.7385280 209 12.53489283  
## X40 0.7439657 433 12.95823650  
## X22 0.7472568 474 13.20295241  
## X32 0.7947439 412 17.66026067  
## X12 0.9420270 507 90.53759362

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="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))



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

# 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.

### 2. Modèle SEIR

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

On considèrera 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 t (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** (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 : le **taux d’incubation d’une maladie** *n* représente une naissance et *m* un décès.

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

De ce fait, nos équations différentielles seront modifiées :

### 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(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))

