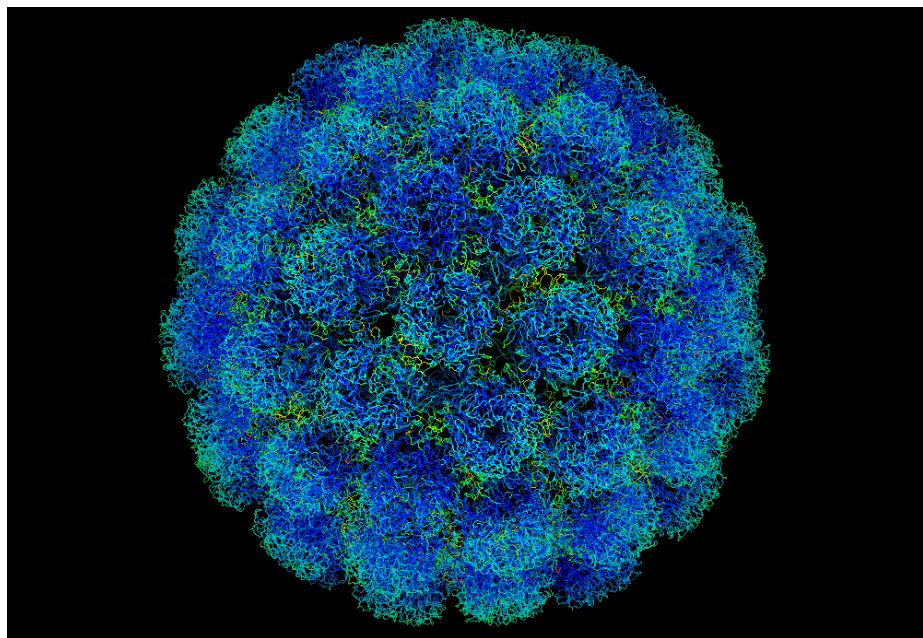


PROJET DE SIMULATION

ISFA

Propagation d'une épidémie



réalisé par

Victor Guillermin, Florian Robinet, Quentin Guignard

Table des matières

1	Introduction	2
2	Modélisation de la propagation d'une épidémie	2
2.1	Modèle	2
2.1.1	Loi continue	2
2.1.2	Loi discrète	3
2.2	Variables aléatoires étudiées	7
3	Simulation et résultats	8
3.1	Résultats de convergence	8
3.2	Réduction de la variance	10
3.3	Élargissement de la population	12
4	Conclusion	15
	Annexes	15

1 Introduction

Problématiques

Quelle est l'espérance du temps de disparition d'une épidémie au sein d'une population ?

Quelle est l'espérance de la proportion de décès dans une population touchée par une épidémie ?

Comment ces espérances varient-elles en fonction de la densité de population ?

Nous approcherons ces espérances à l'aide d'une méthode de Monte-Carlo couplée à une méthode de réduction de variance.

2 Modélisation de la propagation d'une épidémie

2.1 Modèle

2.1.1 Loi continue

La répartition de la population dans l'espace se fait à l'aide de lois mélanges de deux lois normales. L'une sera projetée sur l'axe des abscisses et l'autre sur l'axe des ordonnées afin d'obtenir la distribution initiale de l'échantillon. Cette méthode nous permet de constituer des régions plus ou moins denses représentant des agglomérations.

Nous travaillerons sur une loi de probabilité mélange de ce type :

$$f(x) = \sum_{i=1}^2 \alpha_i f_i(x)$$

où

- les α_i sont des constantes positives (dont la somme est égale à un) tirées aléatoirement
- les fonctions de densité f_i seront des densités de lois normales de moyennes 2 et -2 et de variance 0,5

Nous avons donc :

$$f(x) = \alpha \frac{1}{\sqrt{\pi}} e^{-(x+2)^2} + (1 - \alpha) \frac{1}{\sqrt{\pi}} e^{-(x-2)^2}$$

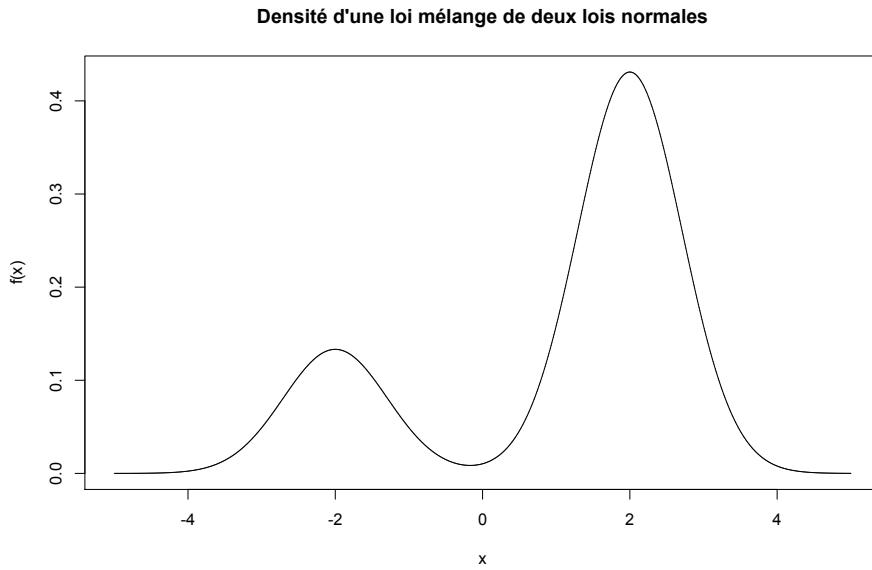
Il sera possible de simuler la loi de densité de f grâce à l'algorithme de simulation d'une loi mélange. Ci dessous, le code R nous permettant de simuler les coordonnées des individus :

```

epidemie=function(n){
  f=function (x){           #fonction de densité de la loi mélange de deux lois normales.
    coef=runif(1)          #On tire le "poids" des deux lois normales aléatoirement
    coef*(1/sqrt(2*pi*0.5))*exp(-(x+2)^2) + (1-coef)*(1/sqrt(2*pi*0.5))*exp(-(x-2)^2)}
  F=function (x){
    coef=runif(1)
    coef*(1/sqrt(2*pi*0.5))*exp(-(x+2)^2) + (1-coef)*(1/sqrt(2*pi*0.5))*exp(-(x-2)^2)}
  x=seq(-5,5,by=0.0001)
  X=c()                  #vecteur abscisse
  Y=c()                  #vecteur ordonnée
  X=sample(x,n,rep=TRUE,prob=f(x))      #On simule n abscisses selon notre loi de densité
  Y=sample(x,n,rep=TRUE,prob=F(x))      #On simule n ordonnées selon notre loi de densité
}

```

Voici un exemple de la densité de f :



2.1.2 Loi discrète

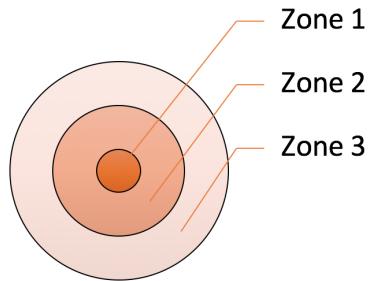
La propagation de l'épidémie au sein d'une population de n individus est modélisée à l'aide de n processus markoviens non homogènes notés $(X_{it})_{t>0}$ représentant les états de santé des individus.

Nous prendrons le jour comme unité de mesure du temps.

L'espace d'état pour chaque individu comporte quatre états :

- l'état 1 dit état "Sain"
- l'état 2 dit état "Infecté"
- l'état 3 dit état "Immunisé"
- l'état 4 dit état "Décédé"

A chaque temps t et pour chaque individu une simulation est réalisée afin de déterminer l'évolution de l'état de santé de celui-ci. La probabilité $p_{i,t}$ pour un individus sain d'être infecté dépend du nombre d'individus infectés autour de lui. On définit alors trois zones pour chaque individu :



La zone 1 est assimilable à une zone de très grande proximité, qui pourra représenter un quartier. La zone 2 est plus large, elle représente une région de proximité moyenne, telle une ville. Enfin la zone 3 représente le reste de l'échantillon.

On définit $p_{i,t}$ comme suit :

$$p_{i,t} = 80\% \frac{n_1(i,t)}{N_1(i)+1} + 15\% \frac{n_2(i,t)}{N_2(i)+1} + 5\% \frac{n_3(i,t)}{N_3(i)+1}$$

On notera :

$N_1(i)$ = Nombre d'individus dans la zone 1

$N_2(i)$ = Nombre d'individus dans la zone 2

$N_3(i)$ = Nombre d'individus dans la zone 3

$n_1(i,t)$ = Nombre d'individus malade à la date t dans la zone 1

$n_2(i,t)$ = Nombre d'individus malade à la date t dans la zone 2

$n_3(i,t)$ = Nombre d'individus malade à la date t dans la zone 3

Cette probabilité traduit le déplacement des individus au sein de la population.

Loi initiale :

Nous supposons que 10% de la population est naturellement immunisée contre le virus.

Les probabilités initiales sont donc $\mathbb{P}(X_{i_0} = 4) = 10\%$ et $\mathbb{P}(X_{i_0} = 1) = 90\%$

De plus, un “Patient Zéro” sera désigné aléatoirement dans la population.

Matrice de transition :

On définit notre matrice de transition non homogène :

$$M = \begin{pmatrix} (1 - p_{i,t}) & p_{i,t} & 0 & 0 \\ 0 & a & b & c \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

avec $a + b + c = 1$

Les paramètres a, b et c ont été choisis de manière arbitraire. Nous avons pris des valeurs qui nous semblaient cohérentes, cependant elles ne reposent sur aucune étude statistique.

· $a = 0,75$

Nous avons choisi de prendre un seuil de 75% concernant le risque de rester malade, une fois que l'on est infecté.

· $b = 0,2$

Nous avons choisi de prendre un seuil de 20% concernant la possibilité de guérir, une fois que l'on est infecté.

· $c = 0,05$

Nous avons choisi de prendre un seuil de 5% concernant le risque de décéder à la suite d'une infection.

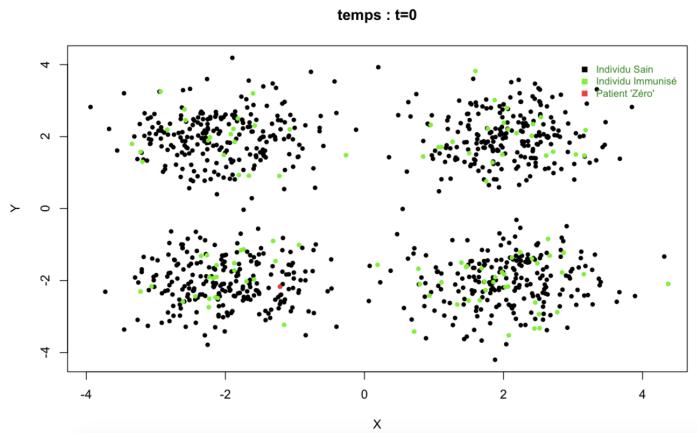
```

epidemie=function(n){
  f=function (x){ }
  F=function (x){ }
  x=seq(-5,5,by=0.0001)
  X=c() #vecteur abscisse
  Y=c() #vecteur ordonnée
  X=sample(x,n,rep=TRUE,prob=f(x))
  Y=sample(x,n,rep=TRUE,prob=F(x))
  Markov=c(0,0.75,0.2,0.05)
  E=matrix(nrow=n,ncol=500)
  E[,1]=1
  E[1,2:500]=0
  IMN=sample(n,floor(n/10))
  E[IMN,1:500]=3
  P0=sample(1:n,1)
  E[P0,1]=2
  POS=matrix(c(X,Y),nrow=n)
  DISTALL=matrix(as.matrix(dist(POS)),nrow=n)
  Z1=apply(DISTALL,1,function(x) sum(x<0.4))
  Z2=apply(DISTALL,1,function(x) sum(x>2))
  Z3=apply(DISTALL,1,function(x) sum(x>=2))

  #On simule n abscisses selon notre loi de densité
  #Chaine de Markov pour les individus infectés (état 2)
  #Matrice des états de chaque individu, à chaque instant t.
  #On initialise tous les individus à l'état 1
  #On remplit les autres colonnes avec des 0 (cf ligne41)
  #Un dizième de la pop est immunisée naturellement
  #On applique l'état 3 à cette partie de la population jusqu'à la fin.
  #On sélectionne le patient 0
  #On ajoute P0 à l'état infecté au temps 1
  #Matrice des coordonnées
  #Matrice des distances
  #On compte le nombre d'individus dans la zone 1 de chaque individu
  #On sélectionne le patient 0
}

```

Voici un exemple de simulation de départ :



```

epideme=function(n){
  f=function(x) { #fonction de densité de la loi mélange de deux lois normales.
    coef=runif(1) #On tire le "poids" des deux lois normales aléatoirement
    coef*(1/sqrt(2*pi*.5))*exp(-(x+2)^2) + (1-coef)*(1/sqrt(2*pi*.5))*exp(-(x-2)^2)
  }
  F=function(x){
    coef=runif(1)
    coef*(1/sqrt(2*pi*.5))*exp(-(x+2)^2) + (1-coef)*(1/sqrt(2*pi*.5))*exp(-(x-2)^2)
  }
  X=C() #vecteur abcisse
  Y=C() #vecteur ordonnée
  X=sample(X,n,rep=TRUE,prob=F(x))
  Y=sample(Y,n,rep=TRUE,prob=F(x))
  Markov=c(0,0.75,0,2,0,0.05)
  E=matrix(c, nrow=n, ncol=500)
  E[,1]=1
  E[,2:500]=0
  INN=sample(0:n, floor(n/10))
  E[INN,1:500]=3
  POS=sample(1:n,1)
  E[POS,1]=2
  POS=matrix(c(X,Y),nrow=n)
  DISTALL=matrix(as.matrix(dist(POS)),nrow=n)
  Z1=apply(DISTALL,1,function(x) sum(x<0.4))
  Z2=apply(DISTALL,1,function(x) sum(x>2))
  Z3=apply(DISTALL,1,function(x) sum(x>2))
  T=1
  while (Z3>0){#on déroule la chaîne tant qu'il y a encore des malades.
    SA=c(which(E[,T]==1)) #On isole les individus à l'état 1 au temps T dans un vecteur.
    IN=c(which(E[,T]==2))
    DIST=as.matrix(DISTALL[SA,IN])
    ZONE=matrix(nrow=length(SA),ncol=6)
    ZONE[,1]=apply(DIST,1,function(x) sum(x<0.4)) #Nb d'individus infectés zone 1
    ZONE[,2]=Z1[SA] #Nb d'individus zone 1
    ZONE[,3]=apply(DIST,1,function(x) sum(x>2)) #Nb d'individus infectés zone 2
    ZONE[,4]=Z2[SA] #Nb d'individus zone 2
    ZONE[,5]=apply(DIST,1,function(x) sum(x>2)) #Nb d'individus zone 3
    ZONE[,6]=Z3[SA] #Nb d'individus sains
    p=0.8*ZONE[,1]/(ZONE[,2]+1)+0.15*ZONE[,3]/(ZONE[,4]+1)+0.05*ZONE[,5]/(ZONE[,6]+1) #Vecteur des probabilités de devenir infecté pour chaque individu sain
    E[,T+1]=sapply(p, function(x) sample(1:2,1,prob=c(1-x,x))) #Chaîne de Markov pour les individus sains
    E[,IN,T+1]=sapply(E[,IN,T+1],function(x) x+sample(1:4,1,prob=Markov)) #Chaîne de markov pour les individus infectés
    E[,which(E[,T+1]==3),T+2]=3 #On applique l'état 3 jusqu'à la fin pour les individus qui viennent d'entrer dans cet état.
    E[,which(E[,T+1]==4),T+2]=4 #On applique l'état 4 jusqu'à la fin pour les individus qui viennent d'entrer dans cet état.
    next
  }
  return(cc[T,100*length(which(E[,T]==4))/n,100*length(which(E[,T]==3))/n])
}

```

2.2 Variables aléatoires étudiées

Les variables aléatoires réelles dont nous allons approximer les espérances sont les suivantes.

- T le temps de disparition de l'épidémie, c'est le temps d'arrêt suivant :

$$T = \inf\{t \geq 1 \mid \sum_{i=1}^n \mathbb{1}_{\{X_{i_t}=2\}} = 0\}$$

- τ_D le pourcentage de décès dans la population à la fin de l'épidémie :

$$\tau_D = \frac{100}{n} \sum_{i=1}^n \mathbb{1}_{\{X_{i_T}=4\}}$$

Nous utiliserons une méthode de Monte-Carlo. Elle consiste à approcher la valeur d'une espérance par une moyenne de plusieurs simulations. Pour cela, on

considère donc la variable aléatoire réelle T et on en simule n réalisations T_i où les $(T_i)_{i \leq n}$ sont indépendantes et identiquement distribuées. D'après la loi des grands nombres on a :

$$\lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=1}^n T_i = \mathbb{E}[T]$$

Nous utiliserons également l'estimateur de la variance $\hat{\sigma}^2 = \frac{1}{n-1} \sum_{i=1}^n (T_i - \bar{T})^2$ pour construire un intervalle de confiance et ainsi contrôler la précision de la méthode. On choisira l'intervalle de confiance à 95% tel que $IC=[\bar{T} \pm z_{0,975} \frac{\hat{\sigma}}{\sqrt{n}}]$ avec ici $z_{0,975} = 1,96$. Nous procéderons de même pour la variable τ_D .

```

MC=function(n,t){
  deb=system.time()
  Ep=c()
  X=c()
  M=c()
  Y=c()
  m=c()
  S=c()
  s=c()
  Ic=c()
  IC1=c()
  IC2=c()
  IM1=c()
  IM2=c()
}

for (i in 1:n){
  Ep=epidemie(t)
  X[i]=Ep[1] #Vecteur des temps de disparitions de l'épidémie
  Y[i]=Ep[2] #Vecteur du pourcentage d'individus décédés
  M[i]=mean(X[1:i]) #Vecteur des espérances des temps de disparitions de l'épidémie
  m[i]=mean(Y[1:i]) #Vecteur des espérances du pourcentage d'individus décédés
  S[i]=sd(X[1:i]) #Vecteur des écarts-types des temps de disparitions de l'épidémie
  s[i]=sd(Y[1:i]) #Vecteur des écarts-types du pourcentage d'individus décédés

  IC1[i]=M[i]-1.96*s[i]/sqrt(i) #borne inf de l'intervalle de confiance pour l'espérance du temps de disparition
  IC2[i]=M[i]+1.96*s[i]/sqrt(i) #borne sup de l'intervalle de confiance pour l'espérance du temps de disparition
  IM1[i]=m[i]-1.96*s[i]/sqrt(i) #borne inf de l'intervalle de confiance pour l'espérance du pourcentage d'individus décédés
  IM2[i]=m[i]+1.96*s[i]/sqrt(i) #borne sup de l'intervalle de confiance pour l'espérance du pourcentage d'individus décédés
}

Ic=IC2[n]-IC1[n] #Taille de l'intervalle de confiance pour l'espérance du temps de disparition
IM=IM2[n]-IM1[n] #Taille de l'intervalle de confiance pour l'espérance pourcentage d'individus décédés
par(mfrow=c(2,2))

matplot(1:n, matrix(c(M[1:n], IC1[1:n], IC2[1:n]), ncol=3), ylim=c(0,60), col=c("black","red","red"), type='l',
        main="Convergence du temps de disparition de l'épidémie", xlab="Nombre de simulations", ylab="Temps de disparition moyen")
matplot(1:n, matrix(c(m[1:n], IM1[1:n], IM2[1:n]), ncol=3), ylim=c(0,40), col=c("black","red","red"), type='l',
        main="Convergence du pourcentage de décès", xlab="Nombre de simulations", ylab="Pourcentage de décès moyen")

hist(X,main="Histogramme du temps de disparition",xlab="Temps de disparition",ylab="Fréquence")
hist(Y,main="Histogramme du pourcentage de décès",xlab="Pourcentage de décès",ylab="Fréquence")
fin=system.time()
temps=fin-deb
return(c(M[n],m[n],ICt,IM,temps))
}

```

3 Simulation et résultats

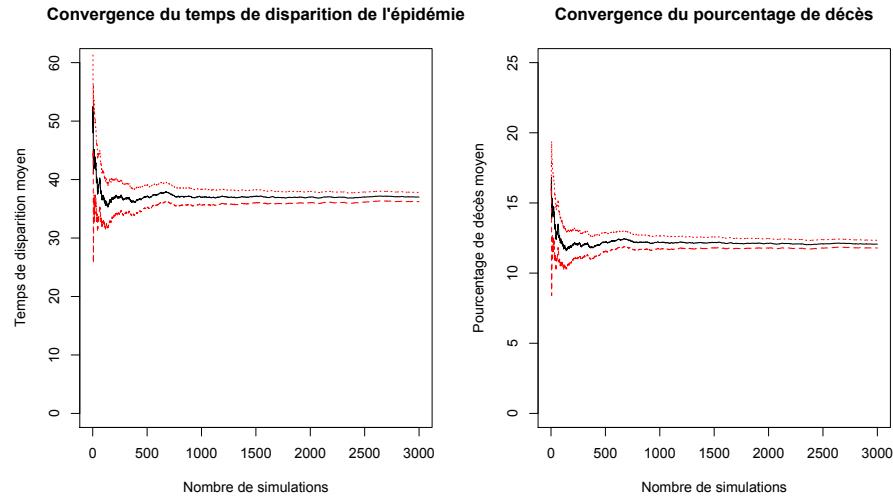
3.1 Résultats de convergence

Nous avons réalisé dans un premier temps 10 000 simulations afin de comparer la taille des intervalles de confiance :

Nombre de Simulations	Temps de disparition T	
	Intervalle de confiance	Taille de l'intervalle
100	[35.136 , 42.984]	7.848
1000	[36.780 , 39.444]	2.665
3000	[36.488 , 38.041]	1.553
5000	[36.581 , 37.789]	1.208
10000	[36.865 , 37.722]	0.856
Nombre de Simulations	Pourcentage de décès τ_D	
	Intervalle de confiance	Taille de l'intervalle
100	[11.760 , 14.539]	2.778
1000	[12.075 , 12.999]	0.924
3000	[12.001 , 12.545]	0.544
5000	[12.000 , 12.412]	0.422
10000	[12.031 , 12.329]	0.298

Le temps de fonctionnement pour 10 000 simulations étant relativement long, nous nous limiterons à présent à 3000 simulations afin de répondre à notre problématique. En effet, nous remarquons dans ces tableaux que la taille de nos 2 intervalles diminuent fortement jusqu'à 3000 simulations, puis que leur réductions est ensuite très ralenti, compte tenu du temps de calcul.

Analysons à présent les convergences de nos variables au bout de 3000 simulations :



Résultats

Temps de disparition T	Pourcentage de décès τ_D
$\bar{T} = 37.00$	$\bar{\tau}_D = 12.06\%$
IC = [36.21,37.79]	IC = [11.79,12.33]
$\hat{\sigma}^2 = 484.01$	$\hat{\sigma}^2 = 58.36$

Sur les graphiques ci-dessus nous avons choisi de représenter les bornes de l'intervalle de confiance en pointillés rouges et les variables que nous étudions en trait plein noir. Sur la figure de gauche, on remarque que le temps moyen de disparition de l'épidémie est convergent, et converge vers 37,00. Sur la figure de droite, on remarque que le pourcentage moyen de décès des suites de l'épidémie est convergent, et converge vers 12,06%.

De plus, les variances calculées nous permettent d'établir des intervalles de confiance pour les espérances.

Nous pourrions encore réduire ces intervalles de confiance en augmentant le nombre de simulations. Cependant, le temps de calcul étant déjà important, il est préférable d'utiliser une technique de réduction de variance pour améliorer sa précision.

3.2 Réduction de la variance

Afin d'affiner au mieux les résultats, c'est à dire réduire la taille des intervalles de confiance sans avoir à augmenter le nombre de simulations, nous pouvons utiliser une méthode de réduction de variance.

Nous allons donc procéder à une telle méthode pour la variable aléatoire τ_D . Pour cela, nous construisons une variable antithétique, et de même loi que τ_D , à l'aide de la variable aléatoire τ_I qui représente le pourcentage d'individus immunisés à la fin de l'épidémie. Notons Y cette variable :

$$Y = \frac{c}{b}(\tau_I - 10) = \frac{1}{4}(\tau_I - 10)$$

En effet, la probabilité pour un individu de rentrer dans l'état "Immunisé" est, dans notre modèle, quatre fois plus importante que de rentrer dans l'état "Décédé". De plus, d'après nos conditions initiales, il y a 10% de la population qui est déjà immunisée. D'après notre modèle, à un temps T fixé, τ_D et τ_I sont négativement corrélées. En effet ces deux états sont absorbants et cela signifie directement que si τ_D augmente, à T fixé, alors τ_I diminue. On a donc :

$$\text{cov}(\tau_D, Y) < 0$$

D'après ce qui précède, nous avons d'une part :

$$\mathbb{E}\left[\frac{\tau_D+Y}{2}\right] = \mathbb{E}[\tau_D]$$

Et d'autre part :

$$Var\left(\frac{\tau_D+Y}{2}\right) = \frac{1}{2}Var(\tau_D) + cov(\tau_D, Y) < \frac{1}{2}Var(\tau_D)$$

Simulons à présent 3000 nouvelles épidémies et comparons les résultats obtenus entre τ_D et notre nouvelle variable $Z = \frac{\tau_D+Y}{2}$.

```

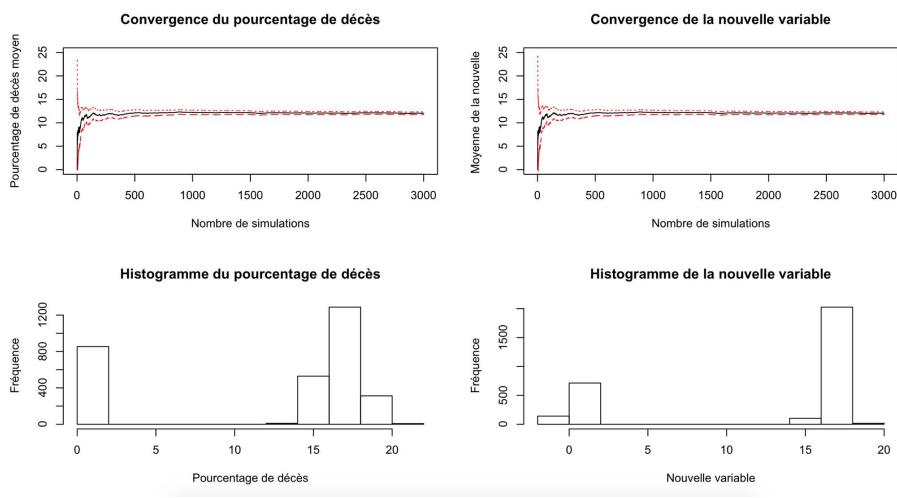
MC=function(n,t){
  deb=Sys.time()
  Ep=c()
  X=c()
  M=c()
  Y=c()
  m=c()
  S=c()
  s=c()
  ICm1=c()
  ICm2=c()
  ICv1=c()
  ICv2=c()
  for (i in 1:n){
    Ep=epidemie(t)
    X[i]=Ep[2] #Vecteur du pourcentage d'individus décédés
    Y[i]=(Ep[2]+0.25*(Ep[3]-10))/2 #Vecteur des valeurs de la nouvelle variable
    M[i]=mean(X[1:i]) #Vecteur des espérances du pourcentage d'individus décédés
    m[i]=mean(Y[1:i]) #Vecteur des espérances de la nouvelle variable
    S[i]=sd(X[1:i]) #Vecteur des écarts-types du pourcentage d'individus décédés
    s[i]=sd(Y[1:i]) #Vecteur des écarts-types de la nouvelle variable
    ICm1[i]=M[i]-1.96*S[i]/sqrt(i)
    ICm2[i]=M[i]+1.96*S[i]/sqrt(i)
    ICv1[i]=m[i]-1.96*s[i]/sqrt(i)
    ICv2[i]=m[i]+1.96*s[i]/sqrt(i)
  }
  ICv=ICv2[n]-ICv1[n] #Taille de l'intervalle de confiance pour l'espérance du pourcentage d'individus décédés
  ICm=ICm2[n]-ICm1[n] #Taille de l'intervalle de confiance pour l'espérance de la nouvelle variable
  par(mfrow=c(2,2))

  matplot(1:n,matrix(c(M[1:n],ICm1[1:n],ICm2[1:n]),ncol=3),ylim=c(0,25),col=c("black","red","red"),type='l',
         main="Convergence du pourcentage de décès",xlab="Nombre de simulations",ylab="Pourcentage de décès moyen")

  matplot(1:n,matrix(c(m[1:n],ICv1[1:n],ICv2[1:n]),ncol=3),ylim=c(0,25),col=c("black","red","red"),type='l',
         main="Convergence de la nouvelle variable",xlab="Nombre de simulations",ylab="Moyenne de la nouvelle ")

  hist(X,main ="Histogramme du pourcentage de décès",xlab="Pourcentage de décès",ylab="Fréquence")
  hist(Y,main ="Histogramme de la nouvelle variable",xlab="Nouvelle variable",ylab="Fréquence")
  fin=Sys.time()
  temps=fin-deb
  return(c(M[n],m[n],ICm,ICv,var(X),var(Y),temps))
}

```



Résultats

Pourcentage de décès τ_D

$$\bar{\tau}_D = 12.04\%$$

$$IC = [11.771, 12.318]$$

$$\hat{\sigma}^2 = 58.47$$

Nouvelle variable $Z = \frac{\tau_D + Y}{2}$

$$\bar{Z} = 12.04\%$$

$$IC = [11.773, 12.313]$$

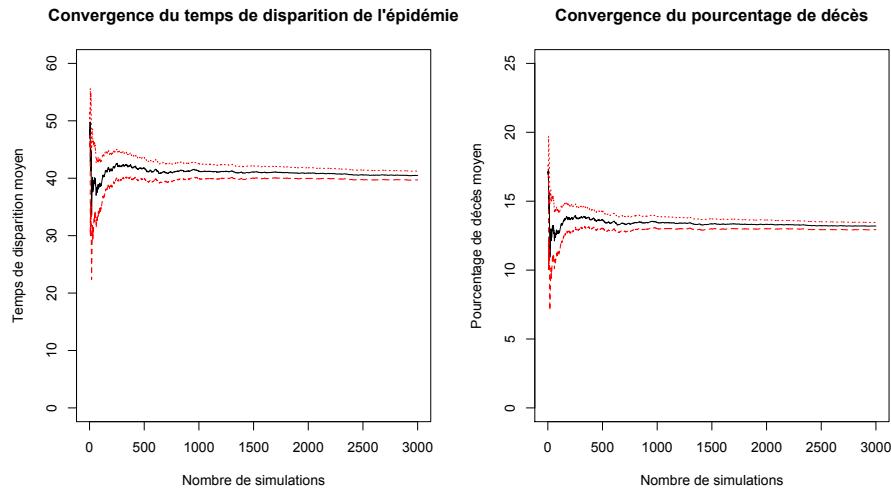
$$\hat{\sigma}^2 = 57.70$$

Nous remarquons dans un premier temps que les deux variables convergent vers la même valeur : 12,04%. Dans un deuxième temps, nous observons bel et bien une réduction de variance et, logiquement, une réduction de l'intervalle de confiance.

3.3 Élargissement de la population

Étudions à présent l'impact d'une telle épidémie sur une population plus importante. Nous étudions la même variable T pour le temps ainsi que notre variable Z utilisée pour réduire la variance de $\bar{\tau}_D$.

Simulons tout d'abord une telle épidémie dans une population deux fois plus grande, avec 2000 individus.



Résultats

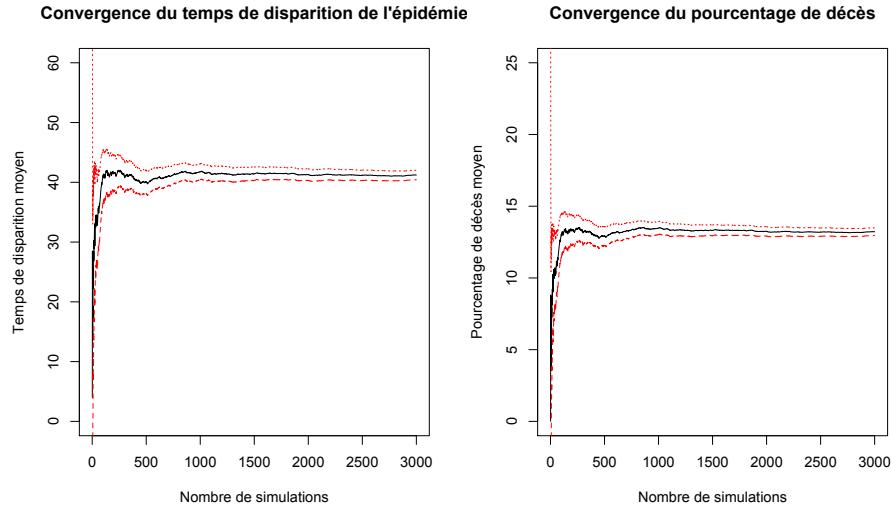
Temps de disparition T

$$\begin{aligned}\bar{T} &= 40.48 \\ \text{IC} &= [39.71, 41.25] \\ \hat{\sigma}^2 &= 459\end{aligned}$$

Pourcentage de décès τ_D

$$\begin{aligned}\bar{\tau}_D &= 13.20\% \\ \text{IC} &= [12.94, 13.46] \\ \hat{\sigma}^2 &= 1.39\end{aligned}$$

Nous remarquons que l'espérance du temps de disparition T augmente de 3 jours et le pourcentage de décès de plus de 1%



Plaçons nous à présent dans une population de 3000 individus :

Résultats

Temps de disparition T

$$\begin{aligned}\bar{T} &= 41.22 \\ \text{IC} &= [40.43, 42.02] \\ \hat{\sigma}^2 &= 489.84\end{aligned}$$

Pourcentage de décès τ_D

$$\begin{aligned}\bar{\tau}_D &= 13.22\% \\ \text{IC} &= [12.96, 13.49] \\ \hat{\sigma}^2 &= 55.33\end{aligned}$$

Nous remarquons ici que l'espérance du temps de disparition augmente de 4 jours et le pourcentage de décès augmente de plus de 1%. Ils se dégagent donc une tendance selon laquelle plus la population est dense, et plus l'impact de l'épidémie est important. Cependant nous remarquons à la fois un temps de convergence plus long et un temps de calcul bien plus important. La précision peut donc encore être améliorer.

4 Conclusion

Pour améliorer notre étude il serait sans aucun doute nécessaire, avec l'aide de spécialiste du domaine, de complexifier le modèle épidémiologique en prenant en compte d'autres paramètres de contamination. De plus, un avis de professionnels nous permettrait très certainement de mieux définir nos paramètres (a , b , c) dans la matrice de propagation de l'épidémie afin d'obtenir un modèle plus proche de la réalité.

Annexes

Code R

```
epidemie=function(n){  
f=function (x){  
coef=runif(1)  
coef*(1/sqrt(2*pi*0.5))*exp(-(x+2)^2) + (1-coef)*(1/sqrt(2*pi*0.5))*exp(-(x-2)^2}  
F=function (x){  
coef=runif(1)  
coef*(1/sqrt(2*pi*0.5))*exp(-(x+2)^2) + (1-coef)*(1/sqrt(2*pi*0.5))*exp(-(x-2)^2}  
x=seq(-5,5,by=0.0001)  
X=c()  
Y=c()  
X=sample(x,n,rep=TRUE,prob=f(x))  
Y=sample(x,n,rep=TRUE,prob=F(x))  
Markov=c(0,0.75,0.2,0.05)  
E=matrix(nrow=n,ncol=500)  
E[,1]=1  
E[,2:500]=0  
IMN=sample(n,floor(n/10))  
E[IMN,1:500]=3  
P0=sample(1:n,1)  
E[P0,1]=2  
POS=matrix(c(X,Y),nrow=n)  
DISTALL=matrix(as.matrix(dist(POS)),nrow=n)  
Z1=apply(DISTALL,1,function(x) sum(x<0.4))  
Z2=apply(DISTALL,1,function(x) sum(x<2))  
Z3=apply(DISTALL,1,function(x) sum(x>=2))  
  
T=1  
while (2%in%E[,T]){  
  
SA=c(which(E[,T]==1))  
IN=c(which(E[,T]==2))
```

```

DIST=as.matrix(DISTALL[SA,IN])

ZONE=matrix(nrow=length(SA),ncol=6)
ZONE[,1]=apply(DIST,1,function(x) sum(x<0.4))
ZONE[,2]=Z1[SA]
ZONE[,3]=apply(DIST,1,function(x) sum(x<2))
ZONE[,4]=Z2[SA]
ZONE[,5]=apply(DIST,1,function(x) sum(x>=2))
ZONE[,6]=Z3[SA]

p=0.8*ZONE[,1]/(ZONE[,2]+1)+0.15*ZONE[,3]/(ZONE[,4]+1)+0.05*ZONE[,5]/(ZONE[,6]+1)
E[SA,T+1]=sapply(p,function(x) sample(1:2,1,prob=c(1-x,x)))

E[IN,T+1]=sapply(E[IN,T+1],function(x) x+sample(1:4,1,prob=Markov))
E[which(E[,T+1]==3),T+2]=3
E[which(E[,T+1]==4),T+2]=4

T=T+1
next
}
return(c(T,100*length(which(E[,T]==4))/n,100*length(which(E[,T]==3))/n))
}

MC=function(n,t){
deb=Sys.time()
Ep=c()
X=c()
M=c()
Y=c()
m=c()
S=c()
s=c()
ICt1=c()
ICt2=c()
ICm1=c()
ICm2=c()

for (i in 1:n){
Ep=epidemie(t)
X[i]=Ep[1]
Y[i]=(Ep[2]+0.25*(Ep[3]-10))/2
M[i]=mean(X[1:i])
m[i]=mean(Y[1:i])
S[i]=sd(X[1:i])
s[i]=sd(Y[1:i])
ICt1[i]=M[i]-1.96*S[i]/sqrt(i)
}
}

```

```

ICt2[i]=M[i]+1.96*S[i]/sqrt(i)
ICm1[i]=m[i]-1.96*s[i]/sqrt(i)
ICm2[i]=m[i]+1.96*s[i]/sqrt(i)
}

ICt=ICt2[n]-ICt1[n]
ICm=ICm2[n]-ICm1[n]
par(mfrow=c(1,2))

matplot(1:n,matrix(c(M[1:n],ICt1[1:n],ICt2[1:n]),ncol=3),ylim=c(0,60),
        col=c("black","red","red"),type='l',main="Convergence du temps de disparition
        de l'épidémie",xlab="Nombre de simulations",ylab="Temps de disparition moyen")

matplot(1:n,matrix(c(m[1:n],ICm1[1:n],ICm2[1:n]),ncol=3),ylim=c(0,25),
        col=c("black","red","red"),type='l',main="Convergence du pourcentage de décès",
        xlab="Nombre de simulations",ylab="Pourcentage de décès moyen")

hist(X,main="Histogramme du temps de disparition de l'épidémie",xlab="Temps de
      disparition",ylab="Fréquence")

hist(Y,main ="Histogramme du pourcentage de décès",xlab="Pourcentage de décès",
      ylab="Fréquence")

fin=Sys.time()
temps=fin-deb
return(c(mean(X),M[n],m[n],ICt,ICt1[n],ICt2[n],var(X),ICm,ICm1[n],ICm2[n],var(Y),temps))
}

```