Projet - Simulation, Inférence MCMC
M1 M&A
Probabilité de ruine en assurance

Mounir TALBI

Résumé du sujet

Ce projet aura pour objectif d'approximer par une méthode de Monte-Carlo la probabilité de ruine à un an pour une compagnie qui propose des contrats d'assurance incendie.
Celle-ci disposera d'un niveau de réserve $u \in \mathbb{R}^+$ et d'un nombre d'assuré $M \in \mathbb{N}$. Nous choisirons de modéliser une année en temps comme une succession de $T = 52$ semaines. De plus, pour simplifier on considére un modèle homogéne dans lequel toutes les habitations assurées ont la même valeur.
Enfin, celle-ci perçoit des primes à un taux $c \in]0, 1]$ par unité de temps et par assuré, et rembourse des sinistres dont les montants sont aléatoires.

On notera alors, $(R_t, t = 0, 1, ..., T)$ le processus aléatoire de richesse de la compagnie, défini sur un espace de probabilité (Ω, F, P) par

$$R_t = uM + ctM - \sum_{i=1}^{t} \sum_{j=1}^{N_i} X_{i,j}, \; \forall t = 0, 1, ..., T$$

Avec

- $(X_{i,j})_{i,j \geq 1}$ une suite de variable aléatoire indépendante identiquement distribuées, qui représente le montant du j-ième sinistre survenu à la période i
- N_i une variable aléatoire représentant le nombre de sinistres de tous les assurés survenus durant la période i.
- $(X_{i,j})_{i,j \geq 1}$ et $(N_1, ..., N_T)$ sont indépendantes.

On notera P_X la loi commune des variables $(X_{i,j})$.

Contents

Programmation en R

1 Loi du montant des sinitres

Soit Z une variable aléatoire définie sur (Ω, F, P), de loi $N(m, \sigma^2)$, avec $m \in \mathbb{R}$ et $\sigma^2 > 0$. Soit μ la mesure de probabilité sur $\mathbb{R}^+$, définie comme la mesure image de P par l'application mesurable $\omega \rightarrow exp(Z(\omega))$.

Soit Y une variable aléatoire définie sur (Ω, F, P), dont la loi est décrite par $P(Y > x) = (\frac{s}{x})^\alpha$ si $x \geq s$ et $P(Y > x) = 1$ si $x \leq s$, où$s > 0$ est un seuil fixé et $\alpha > 0$. On note ν la loi de la variable aléatoire Y .

Soient $(p_1, p_2, p_3) \in]0, 1[$ tels que $p_1, p_2, p_3 = 1$. On définit P_X par

$$P_X = p_{\delta_0} + p_2\mu + p_3\nu$$

Nous allons maintenant simuler P_X la loi commune des variables $(X_{i,j})_{i,j\geq 1}$, qui représente le montant du j-ième sinistre survenu à la période i.
Pour ce faire, nous allons programmer une fonction 'MontantSinistre' qui se stucture en 3 étapes :

- La simulation de la loi log-normale (Z) : En prenant l'exponentielle d'une gaussienne que l'on aura simuler
- La simulation de la loi de Pareto (Y) : En inversant la fonction de répartition on obtient la fonction $f(x) = (\frac{s}{x})^{\frac{1}{\alpha}} \cdot \mathbb{1}_{(x \geq s)} + \mathbb{1}_{(x < s)}$, que l'on évalura en une uniforme sur $[0, 1]$
- La simulation de la loi du montant des sinistres (X) :
  En posant $p_1 \cdot 0 + p_2 \cdot Y + p_3 \cdot Z$

On a programmé la fonction R suivante :

```
# Loi montant sinistre
MontantSinistre=function(M,m,sigma2,s,alpha,p1,p2,p3){
Gauss=runif(M,m,sigma2) # On simule la loi Gaussienne
Z=exp(Gauss) # Loi Log-Normale
Y=as.numeric(Gauss>=s)*(s/Gauss^(1/alpha))+as.numeric(Gauss<s) # Loi
de Pareto
p1*0+p2*Y+p3*Z # Loi du montant des sinistres }
```

Test de la fonction :

```
> MontantSinistre(10,0.5,1,1,1,0.3,0.2,0.5)
 [1] 1.293465 1.353431 1.035155 1.097383 1.043816
 [6] 1.206792 1.386964 1.133518 1.338912 1.293740
```

2 Loi du nombres de sinistres

Nous allons maintenant décrire la loi du nombre de sinitres $(N_1, ..., N_T)$ survenus à chaque période.

2.1 Conditions Météo

On modélise la météo observée chaque semaine par une chaîne de Markov $(Hi)_{0 \leq i \leq T}$ à trois états notés 1, 2 et 3 que l'on peut par exemple associer à "temps pluvieux", "temps normal" et "canicule". On suppose que cette chaîne est dans un état initial H_0 déterministe la semaine précédant l'année étudiée, et on note Q sa matrice de transition.
Pour ce faire, nous allons programmer une fonction 'MarkovChain' :

```
#Chaine de MARKOV (Météo)
MarkovChain = function(Q, H0,T){
res = 1:T
res[1] = sample( x = 1:length(Q[1,]),size = 1, replace = TRUE, prob = Q[H0,]
)
if(T > 1){
 for(i in 2:T){
  res[i] = sample( x = 1:length(Q[1,]),size = 1, replace = TRUE, prob =
Q[res[i-1],] )
}
}
res
}
```

Test de la fonction :

```
> MarkovChain(matrix( c(0.35,0.6,0.05,0.1,0.7,0.2,0.1,0.4,0.5), 3, byrow = TRUE),1,52)
 [1] 1 1 1 1 2 2 3 1 2 3 3 3 2 2 2 3 3 3 1 3 2 2 3 2 2 2 2 2 3 3 3 3 2 2 2 2 3 3 1 2 3 2 1 2 2 2 2 2 3 3 2 3
```

2.2 Occurence des sinistres

Nous supposons que le nombre de sinistres survenant une semaine donnée dépend de la météo de la semaine en question.
Soient $(\lambda_1, \lambda_2, \lambda_3) \in [0, 1]^3$. On suppose que pour chaque semaine i, conditionnellement à H_i , un assuré donné est victime d'un sinistre avec probabilité λ_{H_i} . Ceci permet de déterminer le nombre total N_i de sinistres survenant à la semaine i.
Pour ce faire, nous allons programmer une fonction 'NombreSinistre' qui va associer à chaque élèments du vecteur obtenu (grâce à la fonction MarkovChain) : la probabilité d'accident par rapport au à la météo observée (λ_i) par le nombre d'assuré M.

```
#Loi du nombres de sinistres
NombreSinistre =function(Q,H0,T,M){
Meteo=matrix(MarkovChain(Q,1,T),1)
N=matrix(rep(0,T),1)
for(i in 1:T){
N[1,i]=lambda[1,Meteo[1,i]]*M #Nombre de sinistres la semaine i
}
return(N) # Nombre de sinistre durant l'année 2022 }
```

Test de la fonction :

```
> NombreSinistre(matrix( c(0.35,0.6,0.05,0.1,0.7,0.2,0.1,0.4,0.5), 3, byrow = TRUE),1,52,100)
     [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13] [,14] [,15] [,16] [,17] [,18] [,19] [,20]
[1,]  0.1  0.5    1  0.1  0.1  0.5  0.5  0.5    1   0.5   0.5   0.1   0.5   0.5   0.5   0.5   0.5   0.5     1   0.1
     [,21] [,22] [,23] [,24] [,25] [,26] [,27] [,28] [,29] [,30] [,31] [,32] [,33] [,34] [,35] [,36] [,37] [,38] [,39]
[1,]   0.5   0.5   0.5   0.5   0.5   0.5   0.5   0.5   0.1   0.1   0.1   0.5   0.5   0.5   0.5     1   0.5   0.5     1
     [,40] [,41] [,42] [,43] [,44] [,45] [,46] [,47] [,48] [,49] [,50] [,51] [,52]
[1,]     1   0.5   0.5   0.5   0.5   0.5     1   0.5   0.5   0.5   0.5   0.5   0.5
```

3 Probabilité de ruine

Nous allons finalement approximer la probabilité de ruine a un an pour la compagnie d'assurance définie par

$$V(u,T) =: P(\min_{1\leq t\leq T}\{R_t\} < 0)$$

Pour ce faire, nous allons programmer trois fonctions :

- Une fonction 'Simule' qui va simuler le comportement de la réserve d'argent de la compagnie chaque semaine de l'année.
- Une fonction 'Ruine' qui va dire si l'entreprise est en faillite à un moment donnée de l'année.
- Une fonction 'ProbaRuine' qui va approximer $V(u,T)$.

3.1 Simulation de la trésorerie de la compagnie

Pour simuler le comportement de la trésorerie de la compagnie d'assurance nous nous servirons de la formule

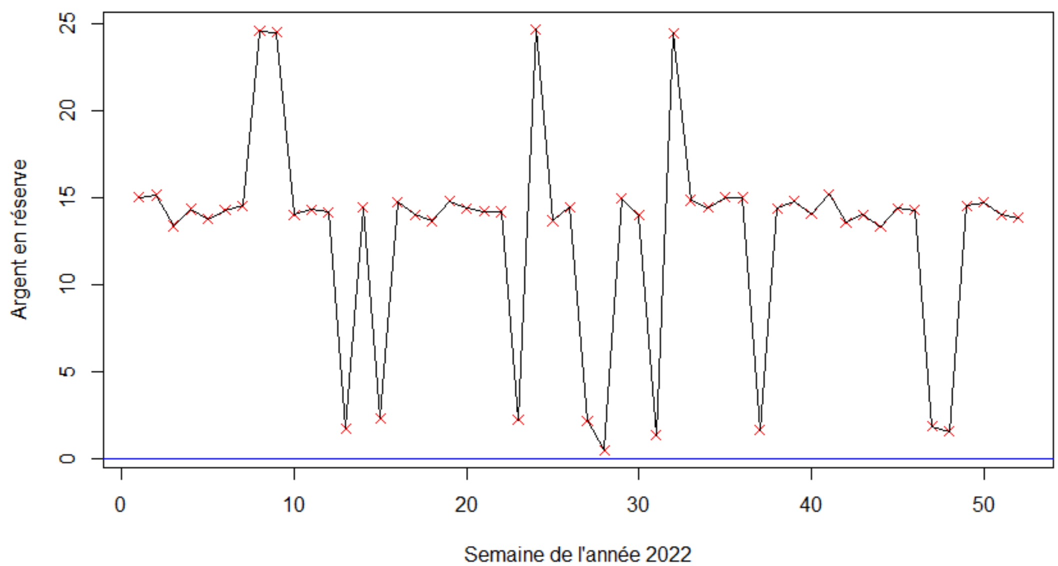
$$R_t = uM + ctM - \sum_{i=1}^{t}\sum_{j=1}^{N_i} X_{i,j},\ \forall t = 0,1,...,T$$

et des fonctions 'NombreSinistre' et 'MontantSinistre' défini précèdemment. Ainsi, la fonction 'NombreSinistre' nous donnera un vecteur qui dénombrera le nombre de sinistre survenu chaque semaine. Nous créerons alors une variable 'cout' qui calculera le montant des réparations d'une semaine donnée, et un vecteur 'plan' qui enregistrera l'évolution de la trésorerie de la compagnie. Nous tracerons enfin cette simulation.

```
# Simulation de la variation de l'argent en réserve durant une année
Simule=function(u,m,M,s,alpha,p1,p2,p3,lambda,H0,Q,c){
N=NombreSinistre(Q,H0,T,M)
plan=rep(c(0),T) # Pour garder les infos chaque semaine
for (i in 1:T){
cout=MontantSinistre(N[1,i],m,sigma2,s,alpha,p1,p2,p3)
plan[i]=u-sum(cout)+c*M
}
plot(plan,xlab="Semaine de l'année 2022",ylab="Argent en réserve",pch=4,
col = "red")
lines(plan, col = "black")
abline(h=0, col="blue") # Ligne de ruine
return(plan) # Argent qu'il reste à l'entreprise chaque semaine
}
```

Test de la fonction :

```
> Simule(u,m,M,s,alpha,p1,p2,p3,lambda,H0,Q,c)
 [1] 14.9945906 15.1299337 13.3626855 14.3272392 13.7679729 14.2676506 14.5097763 24.5503973 24.4872176 14.0106495
[11] 14.2995490 14.1633448  1.7464464 14.4301543  2.3160794 14.7482924 14.0151735 13.6677423 14.7791594 14.3746632
[21] 14.1948123 14.1722950  2.2175244 24.6607669 13.6557671 14.4462650  2.1820414  0.4735986 14.9201637 13.9830875
[31]  1.3447352 24.4479213 14.8442079 14.4084087 15.0059650 14.9898030  1.6432683 14.3745362 14.7797322 14.0557240
[41] 15.1839385 13.5564194 14.0028380 13.3297479 14.3841761 14.2641043  1.8421033  1.5566045 14.5216182 14.7268107
[51] 13.9951349 13.8171200
```



où :

$u = 25$; $M = 2000$; $m = 0.5$; $sigma2 = 1$; $s = 1$; $alpha = 1$; $p1 = 0.3$; $p2 = 0.2$; $p3 = 0.5$; $lambda = [0.001, 0.005, 0.01]$; $H_0 = 1$; $Q = \begin{matrix} 0.35 & 0.6 & 0.05 \\ 0.1 & 0.7 & 0.2 \\ 0.1 & 0.4 & 0.5 \end{matrix}$; $c = 0.001$; $T = 52$

3.2 Faillite de la compagnie ?

Pour savoir si la compagnie sera en faillite, nous allons utiliser un vecteur 'plan' défini comme précèdemment. Puis nous allons créer une variable 'faillite':= 0 qui prendra la valeur 1 si il existe une composante du vecteur plan qui sera inférieur ou égale à 0.

```
# Fonction qui indique la faillite ou non de l'entreprise durant une simulation

Ruine=function(plan,T){
faillite=0 # Variable qui enregistre la faillite de la boite
test=rep(FALSE,T)
for (i in 1:T){
test[i]=(plan[i]<=0)
  if(test[i] ==TRUE){
  faillite=1
  }
}
return(faillite) }
```

Test de la fonction :

```
> Ruine(rep(c(0),52),52)
[1] 1
> Ruine(rep(c(1),52),52)
[1] 0
```

3.3 Approximation de $V(u, T)$

Finallement, nous allons approximer la probabilité de ruine recherchée. Pour cela nous ferons tourner 'NbSimul' fois la fonction 'Simule', puis on va dénombrer le nombre de faillite parmis ces simulations grâce à la fonction 'Ruine'.

```
# Approximation à un an de la probabilité de ruine de la compagnie d'assurance
ProbaRuine=function(u,m,M,s,alpha,p1,p2,p3,lambda,H0,Q,c,nbSimul){
nbRuine=0
  for (i in 1:nbSimul){
  plan=Simule(u,m,M,s,alpha,p1,p2,p3,lambda,H0,Q,c)
  faillite=Ruine(plan,T)
  nbRuine=nbRuine+faillite # On garde en mémoire toute les faillites rencontrées
  }
proportion=nbRuine/nbSimul
return(proportion) }
```

Test de la fonction :

```
> ProbaRuine(u,m,M,s,alpha,p1,p2,p3,lambda,H0,Q,c,500)
[1] 0.14
> ProbaRuine(u,m,M,s,alpha,p1,p2,p3,lambda,H0,Q,c,500)
[1] 0.122
> ProbaRuine(u,m,M,s,alpha,p1,p2,p3,lambda,H0,Q,c,500)
[1] 0.108
```

Approfondissements

4 Influence des paramètres sur la probabilité de ruine

Pour analyser l'effet de l'un des paramètres sur la probabilité de ruine, il faudra simuler un grand nombre de ces test (on choisira de simuler 10000 tests par contrainte technique, même si on préfererait un plus grand nombre de test) et faire varier un paramètre pour voir l'évolution de la probabilité de ruine.

4.1 Influence du taux des primes

Pour commencer, et même si cela semble trivial, on va chercher à vérifier si le taux des primes à une influence sur la probabilité de ruine.
Données Receuillis :(Les variables autres que c restent les mêmes que précèdement)

c	ProbaRuine(u,m,M,s,alpha,p1,p2,p3,lambda,H0,Q,c,10000)
1e-04	0.999
2e-04	0.9962
3e-04	0.9892
4e-04	0.9611
5e-04	0.9059
6e-04	0.7899
7e-04	0.6077
8e-04	0.4108
9e-04	0.236
0.001	0.1248
0.002	0

Sans surprise, on remarque que plus le taux des primes est bas plus notre probabilité de ruine est grande.

4.2 Influence du nombre d'assuré

Il est plus intéressant de se demander quelle est l'influence du nombre d'assuré sur la probabilité de ruine. En effet, un nombre plus grands d'assurés augmente le montant des primes reçues mais aussi le nombre d'accident et donc le montant des sinistres à rembourser.

Données Receuillis :(Les variables autres que M restent les mêmes que précèdement)

M	ProbaRuine(u,m,M,s,alpha,p1,p2,p3,lambda,H0,Q,c,10000)
1900	3e-04
1925	2e-04
1950	2e-04
1975	0
2000	0.1258
2025	0.1129
2050	0.1001
2075	0.0897
2100	0.9608
2125	0.9538
2150	0.9482
2175	0.9424
2200	0.9999

On remarque ainsi que le nombre de d'assuré a une influence sur la probabilité de ruine. Il semble alors exister un nombre maximale d'assuré que l'on pourrait desservir sans pour autant augmenter notre risque de faillite.

5 Comparaison avec le modèle météo constante

On cherche maintenant à montrer l'impact de la Météo sur la probabilité de ruine. Pour cela, nous allons programmer un modèle dans lequel la météo sera constante. Il suffira alors de poser une matrice de transition :

$$Q = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Et de lancer le programme 'ProbaRuine'. On obtient alors :

```
> Q
     [,1] [,2] [,3]
[1,]    1    0    0
[2,]    0    1    0
[3,]    0    0    1
> ProbaRuine(u,m,M,s,alpha,p1,p2,p3,lambda,H0,Q,0.0001,10000)
[1] 0
> Q=matrix( c(0.35,0.6,0.05,0.1,0.7,0.2,0.1,0.4,0.5), 3, byrow
 = TRUE) #Matrice de transition météo
>
> ProbaRuine(u,m,M,s,alpha,p1,p2,p3,lambda,H0,Q,0.0001,10000)
[1] 0.9986
```

On remarque alors que l'impact de la météo sur la probabilité de ruine est observable.

6 Critiques et améliorations du modèle

Parmi les critiques que l'on pourrait attribuer à ce modèle est que celui-ci ne prends pas en compte que :

- Toutes les habitations assurée n'ont pas la même valeur.
- La météo dépend aussi de la période de l'année à laquelle on se trouve (Il y'a peu de chance qu'on observe une canicule en hiver).
- Le nombre de clients peut aussi varier pendant l'année.
- L'existence d'incendies volontaires qui ont des conséquences financières plus importantes,notamment parce que l'incendiaire recherche la destruction totale d'un bien.

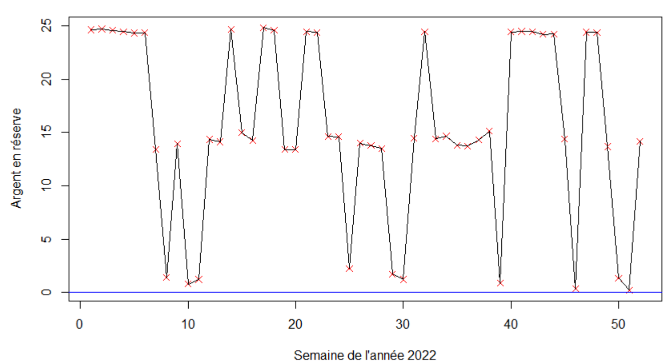
On pourrait en conséquence améliorer ce modèle en :

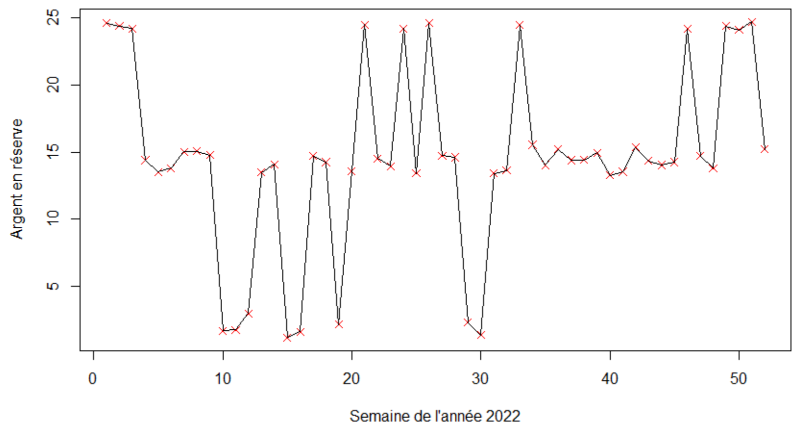
- Modélisant le prix des habitations grâce à une loi gaussienne.
- Décomposant notre période T en 4 (une pour chaque saison) et en associant à chacune une matrice de transition adapté.
- Modélisant une variable aléatoire M qui représente le nombre de clients de l'assurance à chaque semaine de l'année 2022.
- En remplaçant P_X par $P_X = p_{\delta_0} + p_2\mu + p_3\nu + p_4\theta$ où θ est la loi du variable aléatoire T qui va modéliser les prix des incendies volontaires.

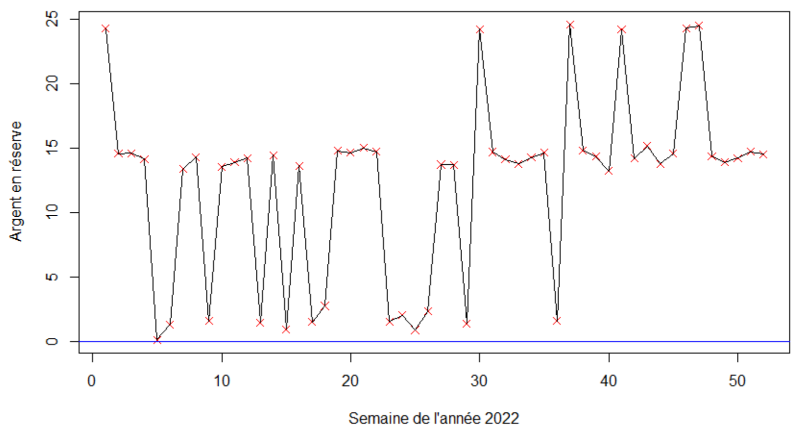
7 Annexe : Quelques exemples de trajectoires obtenues avec 'Simule'

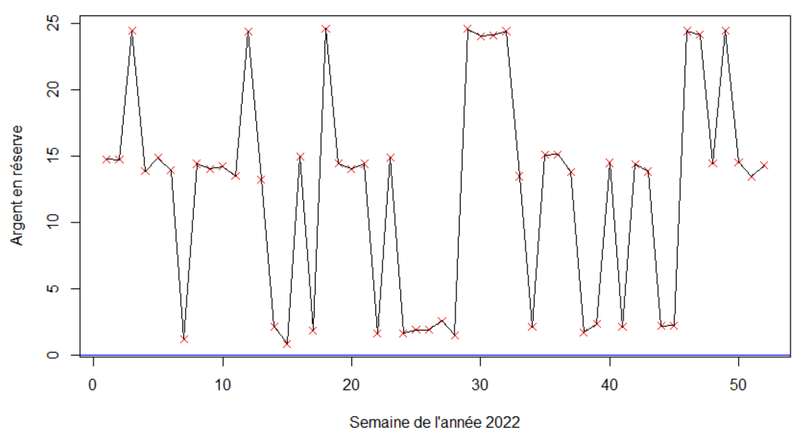
Dans cette partie, on va lancer plusieurs fois une fonction 'Simule' avec à chaque fois les mêmes paramètres pour voir le comportement des simulations.

On obtient alors les graphiques suivants :




Argent en réserve
Semaine de l'année 2022


Argent en réserve
Semaine de l'année 2022


Argent en réserve
Semaine de l'année 2022