Projet MF01 : Simulation du modèle de Lundberg Ruine en Assurance

Réalisé par :

Yannick ZIRIHI (MF01)
Adnane EL KASMI (MF01)
Aboubacar KOUROUMA (MF01)

Plan de Travail

- Cahier des charges
- Presentation du modele de risque Classique (Cramer-Lundberg)
- Les lois qui regissent le modele et leurs simulations
- Calcule et simulation de la probabilité de la ruine
- le travail en équipe

Objectifs du projet

Simulation Mont-Carlo
Échantillonnage préférentiel

Évaluer la probabilité de ruine à horizon fini et infini d'une compagnie d'assurance

Simulation Mont-Carlo naïf

Etude Mathématique

Simulation grace aux méthodes de Monte Carlo

Simulation avec des données réelles

Le cahier des charges

➔Etude théorique :

- Etude théorique des lois de probabilité qui sous-tendent le modèle de Lundberg (loi gamma ,loi exponentielle , loi de Pareto ,loi exponentielle composée).

- Etude de théorique de la transformation d'Esscher pour le calcul de la probabilité de la ruine.

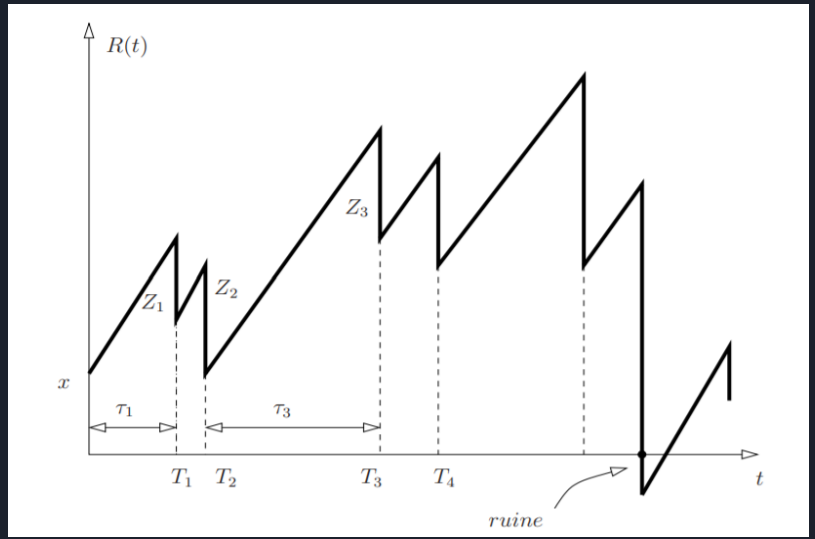
➔Simulations :

– Simulation des différentes lois par la méthodes de Monte-Carlo.

- Simulation de la probabilité de la ruine.

Le modèle du risque Classique (Cramer-Lundberg)

$$R(t) = x + c.t - X(t) = x + c.t - \sum_{k=1}^{N_t} Z_k$$



- $R(t)$: la réserve à l instant t.
- x: la réserve à l'instant de départ.
- c: la prime d'assurance.
- $X(t)$: la somme des sinistres a l'instant t.
- N_t: le nombres de réclamations sur [0, t].
- Z_k: le montant des sinistres.

Les lois qui regissent le modele et leurs simulations

$\{N(t), t \geq 0\}$: le processus de comptage du nombre de réclamation sur [0, t]

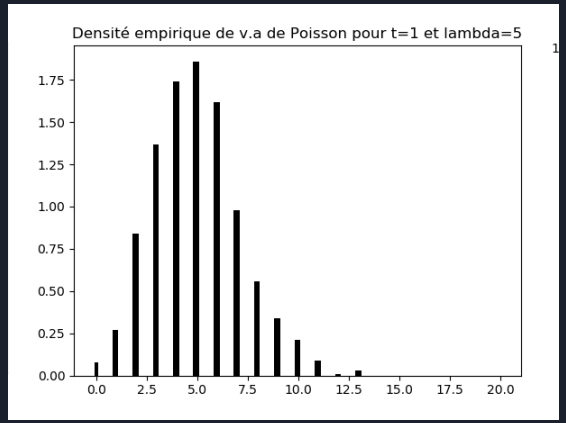
- τ_i l'instant d'arrivée d'un sinistre $\tau \hookrightarrow poisson(\lambda)$.
- $T_k = \sum_{i=0}^{k} \tau_i \qquad avec\ n > 1\ T_n$: l'inter-occurrence d'un sinister.
- $N_t = \sum_{k=1}^{\infty} 1_{\{T_k \leq t\}} \hookrightarrow poisson\ composée\ (\lambda t)$.

Les lois qui regissent le modele et leurs simulations

- Algorithme de simulation de v.a Exponentielle :
  - **function**$[\tau] = V\_A\_Exponentielle(\lambda)$
    - $U = rand()$
    - **set** $\tau = -\frac{1}{\lambda} . ln(1 - U)$
  - **endfunction**

- Algorithme de simulation de N_t :
  - **function**$[N_t] = V\_A\_Processus\_comptage(\lambda, t, N)$
    - $N_t = 0$
    - $T = 0$
    - **for** k = 1: N
      - **for** i = 1: k
      - $\tau(i) = V\_A\_Exponentielle(\lambda)$
      - $T = T + \tau(i)$
      - **endfor**
    - **if** $T \leq t$
    - $N_t = N_t + 1$
    - *endif*
    - *endfor*
  - **endfunction**

Ici N représente un grand nombre pour remplacer l'infini.



Les lois qui regissent le modele et leurs simulations

Z_k : le montant des sinistres

- Processus à queue Fine: $Z_k \hookrightarrow gamma(\alpha, \beta)$ (accident de la route, visite chez le medecin)

• Algorithme de simulation de loi $Gamma(\alpha = n, \beta)$: $n \epsilon N$

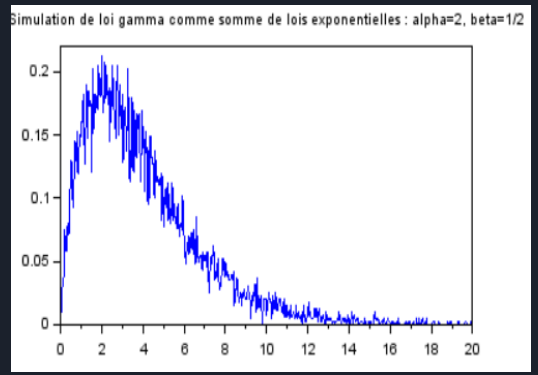
- **function**$[X] = V\_A\_Gamma(n, \beta)$
  - $Set\ Y = V\_A\_Exponentielle(\lambda)$
  - $U = rand()$, $i = 1$
  - While i < n
    - $Y = Y - \frac{1}{\lambda} . ln(1 - U)$
    - $i = i + 1$
  - endwhile
  - $Set\ X = Y$
- endfunction

• Algorithme de simulation de loi $Gamma(\alpha, \beta)$ par la méthode de Rejet :

- **function**$[X] = V\_A\_Gamma\_Rejet(\alpha, \beta)$
  - $U = rand()$
  - $Y = V\_A\_Gamma(n, \delta)$
  - if $U \leq \frac{f_X(Y)}{C . g_Y(Y)}$
    - Set X = Y
  - endif
- endfunction

• Algorithme de fonction de densité de loi $Gamma(\alpha, \beta)$:

- **function**$[f] = f\_Gamma(x, \alpha, \beta)$
  - $f_X(x) = \frac{\beta^{\alpha}}{\Gamma(\alpha)} . x^{\alpha - 1} . e^{-\beta x}$
- endfunction



Les lois qui regissent le modele et leurs simulations

Z_k : le montant des sinistres

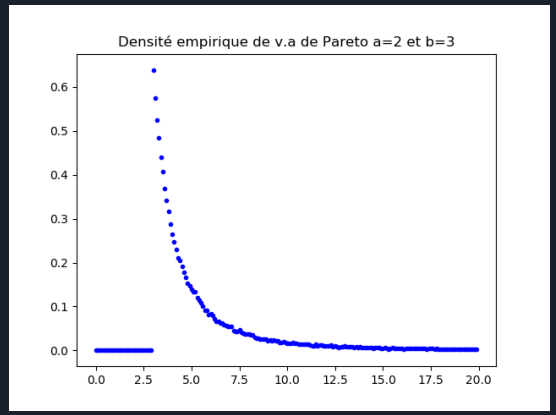
- Processus à queue Lourdes: $Z_k \hookrightarrow Pareto(\alpha, \beta)$ (les tremblements de Terre, La COVID-19)

• Algorithme de simulation de loi Pareto(a,b) :

- **function**[X] = $V\_A\_Pareto(a, b)$
  - $U = rand()$
  - **set** X = $b.(1-u)^{\frac{-1}{a}}$
  - **endfunction**

• Algorithme de simulation d'une chaine de v.a Pareto(a,b) :

- **function**[X] = $Chaine\_V\_A\_Pareto(a, b)$
  - **for** n = 1: N_{mc}
  - $X(n) = V\_A\_Pareto(a, b)$
  - *endfor*
  - **endfunction**



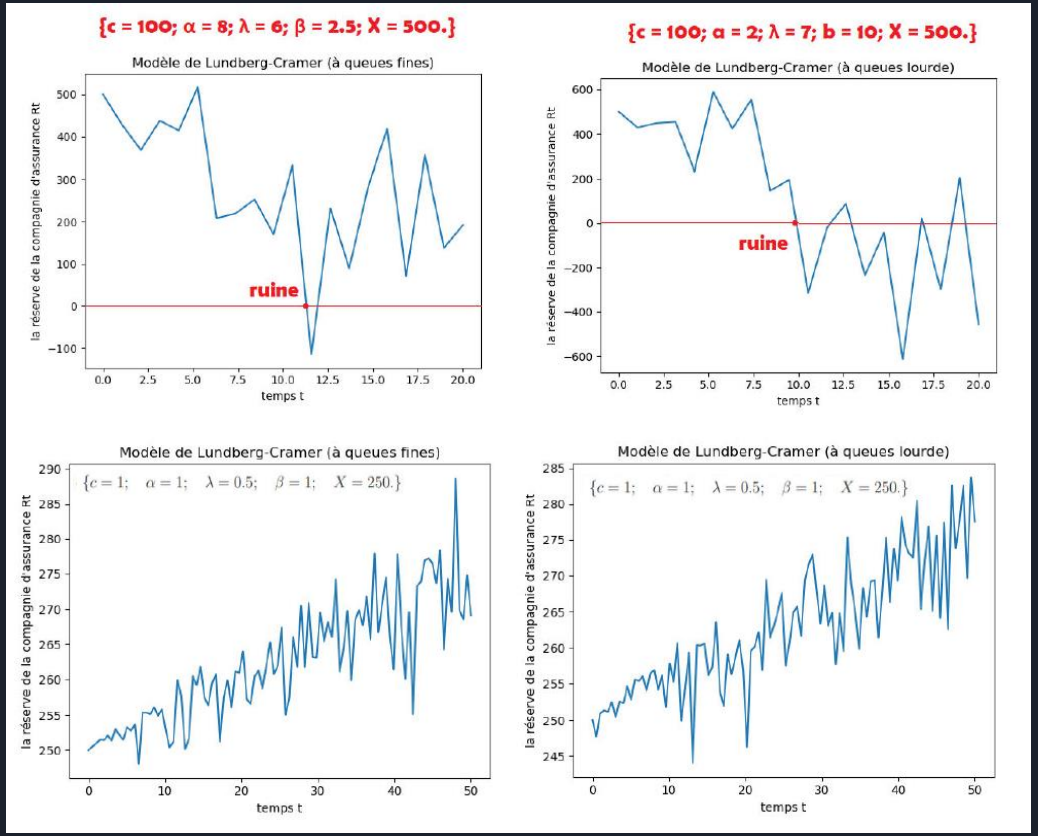
Simulation de la reserve

- Algorithme de simulation de R_t à queues fines :

- **function**$[Rt] = Rt\_fines(c, x, \lambda, \alpha, \beta, t)$
  - $N_t = V\_A\_Poisson\_Compose(\lambda, t)$
  - $X = 0$
  - **for** $k = 1 : N_t$
    - $Z_k = V\_A\_Gamma(\alpha, \beta)$
    - $X = X + Z_k$
  - **endfor**
  - **set** $Rt = x + c.t - X$
  - **endfunction**

- Algorithme de simulation de R_t à queues lourde :

- **function**$[Rt] = Rt\_lourde(c, x, \lambda, a, b, t)$
  - $N_t = V\_A\_Poisson\_Compose(\lambda, t)$
  - $X = 0$
  - **for** $k = 1 : N_t$
    - $Z_k = V\_A\_Pareto(a, b)$
    - $X = X + Z_k$
  - **endfor**
  - **set** $Rt = x + c.t - X$
  - **endfunction**



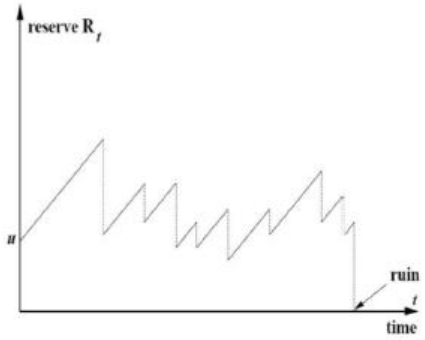
Calcul et simulation de la probabilité de la ruine

- Probabilité de ruine en temps infini :

$$\psi(x) = P(inf_{t\geqslant 0}\, R(t) < 0 \mid R(0) = x)$$

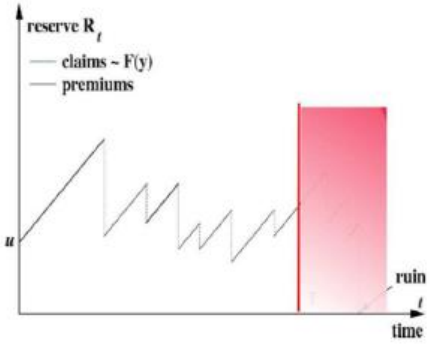
  - L'instant de ruine :

$$T_x = \inf\{t > 0,\ R(t) < 0\}$$



- Probabilité de ruine en temps fini :

$$\psi(x,T) = P(inf_{0\leq t\leq T}\, R(t) < 0 \mid R(0) = x)$$



Calcul et simulation de la probabilité de la ruine

Simulation Monte-Carlo naïf

Probabilité de ruine à horizon fini T:

$$\mathbb{p}(X_T > a) = \mathbb{E}_p\left[\mathbb{I}_{X_T>a}\right] = \frac{1}{Nmc}\sum_{n=0}^{Nmc}\mathbb{I}_{X_T>a}$$

- Algorithme de simulation de Probabilité de ruine :
  - **function** $[Proba] = Proba\_Ruine(c, x, \lambda, T, \alpha, \beta)$
    - $Counter = 0$, $a = x + c.T$
    - **for** $n = 1 : N_{mc}$
      - $X(n) = V\_A\_X(c, x, \lambda, T, \alpha, \beta)$
        - **if** $X(n) > a$
          - $Counter = Counter + 1$
        - **endif**
    - **endfor**
    - **set** $Proba = \frac{Counter}{N_{mc}}$
  - **endfunction**

Pour : $\{ c = 100,\ x = 50,\ T = 1,\ \lambda = 6,\ \alpha = 8,\ \beta = 2.5 \}$ pour $N_{mc} = 10^6$ on trouve :

```
>>> Proba_ruine_fines(100,50,1,6,8,2.5,1000000)
0.264787

>>> Proba_ruine_lourde(100,50,1,6,8,2.5,1000000)
0.0
```

Problème ? (Ruine rare)

On veut simuler la probabilité de ruine d'une ruine rare pour les jeux de paramètres :

$\{ c = 100,\ x = 50,\ T = 1,\ \lambda = 7,\ \alpha = 4,\ \beta = 1.5 \}$ Pour $N_{mc} = 10^2$, normalement si on calcule cette probabilité avec la simulation de Monte-Carlo naïf on obtient une probabilité nulle :

```
>>> Proba_ruine_fines(100,50,1,7,4,1.5,100)
0.0
```

Calcul et simulation de la probabilité de la ruine

Simulation Monte-Carlo : Échantillonnage préférentiel

Probabilité de la ruine avec la transformation d' Esscher(permet de calculer de petite probabilité):

$$\mathbb{P}(X_t > a) = \mathbb{E}_p\left[\mathbb{I}_{X_t>a}\right] = \mathbb{E}_Q\left[\frac{\mathbb{I}_{X_Q} d\mathbb{P}}{d\mathbb{Q}}\right] = \mathbb{E}_Q\left[\mathbb{I}_{X_T^Q>a} e_T^{-\theta X^Q + \Gamma(\theta)}\right]$$ et par Monte-Carlo : $$\mathbb{P}(X_t > a) = \frac{1}{Nmc}\sum_{n=0}^{Nmc} \mathbb{I}_{X_T^Q>a} e_T^{-\theta X^Q + \Gamma(\theta)}$$

- Algorithme de simulation de X_T^Q :

- **function**$[X] = V\_A\_X\_Q(c, x, \lambda, T, \alpha, \beta)$
  - $\theta = \beta - \beta\left(\frac{\beta.(x + c.T)}{\alpha\lambda T}\right)^{\frac{-1}{\alpha+1}}$
  - $\lambda^Q = \lambda.\left(\frac{\beta}{\beta - \theta}\right)^{\alpha}$
  - $N_T = V\_A\_Poisson\_Compose(\lambda^Q, T)$
  - $X = 0$
  - *for* $k = 1 : N_T$
    - $Z_k^Q = V\_A\_Gamma(\alpha, \beta - \theta)$
    - $X = X + Z_k^Q$
  - *endfor*
  - *set* X
- **endfunction**

- Algorithme de simulation de Probabilité de ruine Q:

- **function**$[Proba] = Proba\_Ruine\_Q(c, x, \lambda, T, \alpha, \beta)$
  - $\theta = \beta - \beta\left(\frac{\beta.(x + c.T)}{\alpha\lambda T}\right)^{\frac{-1}{\alpha+1}}$
  - $\Gamma(\theta) = \lambda T\left(\left(\frac{\beta}{\beta - \theta}\right)^{\alpha} - 1\right)$
  - $Counter = 0, \quad a = x + c.T$
  - *for* $n = 1 : N_{mc}$
    - $X^Q(n) = V\_A\_X\_Q(c, x, \lambda, T, \alpha, \beta)$
    - *if* $X^Q(n) > a$
      - $Counter = Counter + e^{-\theta.X^Q(n) + \Gamma(\theta)}$
    - *endif*
  - *endfor*
  - *set* $Proba = \frac{Counter}{N_{mc}}$
- **endfunc on**

Problème (Ruine rare) ?

On veut simuler la probabilité de ruine d'une ruine rare pour les jeux de paramètres :

$\{ c = 100,\ x = 50,\ T = 1,\ \lambda = 7,\ \alpha = 4,\ \beta = 1.5 \}$ Pour $N_{mc} = 10^2$, normalement si on calcule cette probabilité avec la simulation de Monte-Carlo naïf on obtient une probabilité nulle :

```
>>> Proba_ruine_fines(100,50,1,7,4,1.5,100)
0.0
```

Solution

On applique cette méthode pour les jeux de paramètres précédents dont on a trouvé que la probabilité de ruine est nulle par la méthode Monte-Carlo naïf et on trouve :

```
>>> Proba_ruine_Q(100,50,1,7,4,1.5,100)
1.4546707830951162e-22
```

Donc on a trouvé que $P(X_T > a) = 1{,}45.10^{-22}$ ce qui est impossible d'obtenir avec Monte-Carlo naïf.

Calcul et simulation de la probabilité de la ruine

Simulation Monte-Carlo

Probabilité de ruine à horizon infini :

- Algorithme de simulation de T_x à queues fines :
  - **function** $[t] = Tx\_fines(c, x, \lambda, \alpha, \beta, \Delta)$
    - $R_t = Rt\_fines(c, x, \lambda, \alpha, \beta, t)$
    - $t = 0$
    - ***while*** $R_T > 0$
      - $t = t + \Delta$
    - ***endwhile***
    - ***set t***
  - **endfunction**

- Algorithme de simulation de T_x à queues lourde :
  - **function** $[t] = Tx\_lourde(c, x, \lambda, \alpha, \beta, \Delta)$
    - $R_t = Rt\_lourde(c, x, \lambda, \alpha, \beta, t)$
    - $t = 0$
    - ***while*** $R_T > 0$
      - $t = t + \Delta$
    - ***endwhile***
    - ***set t***
  - **endfunction**

➢ L'instant de ruine :

$T_x = \inf\{t > 0,\ R(t) < 0\}$

Pour les jeux de paramètres $\{ c = 100,\ x = 50,\ \lambda = 6,\ \alpha = 8,\ \beta = 2.5 \}$ on obtient avec $\Delta = 0.01$:

```
>>> Tx=Tx_fines(50,100,6,8,2.5,0.01)

>>> Tx
0.3200000000000001
```

Et la probabilité de ruine pour l'instant T_x pour $N_{mc} = 10^6$ on trouve :

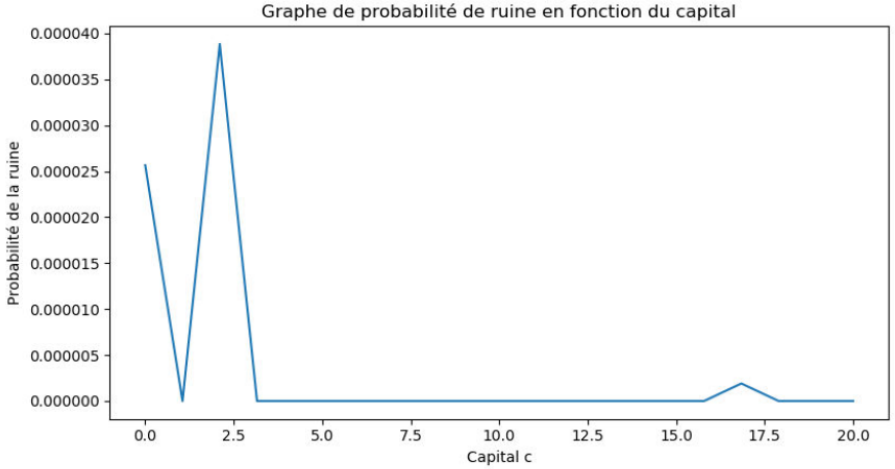
```
>>> Tx=Tx_fines(50,100,6,8,2.5,0.01)

>>> Tx
0.3200000000000001

>>> Proba_ruine_fines(50,100,Tx,6,8,2.5,1000000)
0.015125
```

Simulation de la probabilité de la ruine

Simulation Monte-Carlo

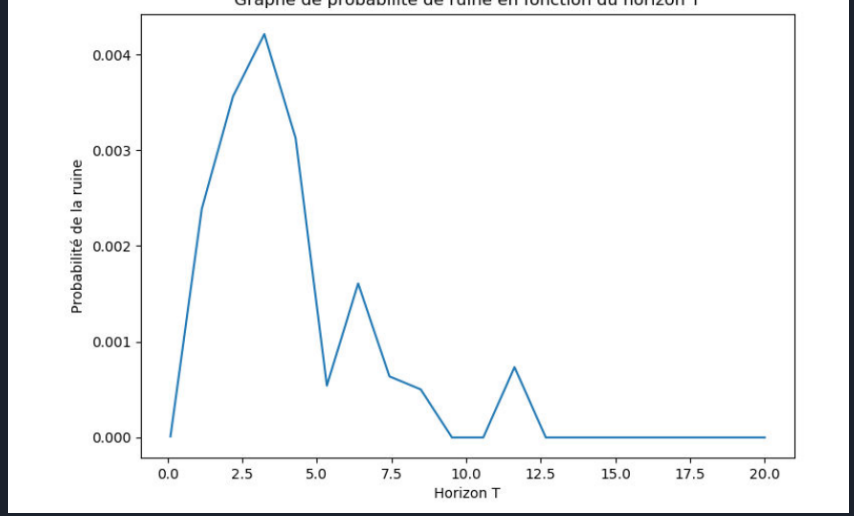
Pour $\{x = 50,\ T = 1,\ \lambda = 2,\ \alpha = 5,\ \beta = 0.75\}$ on trouve avec $N_{mc} = 10^3$:

```
>>> grapheProba_ruine_Q(50,1,2,5,0.75,1000)
```



Pour $\{x = 20,\ c = 14,\ \lambda = 2,\ \alpha = 5,\ \beta = 0.75\}$ on trouve avec $N_{mc} = 10^3$:

```
>>> grapheProba_ruine_Q_T(14,20,2,5,0.75,1000)
```



Présentation du rôle de chaque membre du groupe

Membres\Langages	Mathématiques	Simulation	Rapport & Présentation
Adnane EL KASMI	X	X	X
Yannick ZIHIRI	X	X	X
Aboubacar KOUROUMA	X	X	X

Ce que nous avons réalisé

- ✓ Se familiariser avec des notions mathématiques de l'assurance et avec des bases de la théorie de la ruine : le modèle de Lundberg-Cramer.
- ✓ Montrer théoriquement que le processus de comptage N_t suit la loi de Poisson de paramètre λt.
- ✓ Calculer l'espérance $E[X(t)]$ $avec\ X(t) = \sum_{k=1}^{N_t} Z_k$
- ✓ Calculer une espérance très utile : la fonction génératrice des moments de $X(t)$:

$$M_{X(t)}(\theta) = E\left[e^{\theta.X(t)}\right]$$

- ✓ Simuler par Monte-Carlo le processus de comptage N_t par deux façons différentes et de tracer les graphes de fonction de densité de Poisson.
- ✓ Simuler par Monte-Carlo la loi $Gamma(\alpha = n, \beta)$ avec $n \epsilon N$ et simuler par la méthode de rejet la loi $Gamma(\alpha, \beta)$ avec $\alpha > 0$ puis tracez les graphes de fonction de densité de Gamma pour les différents paramètres et les comparer avec le graphe de Wikipédia.
- ✓ Simuler par Monte-Carlo la loi Pareto(a,b) et tracez les graphes de fonction de densité de Pareto pour les différents paramètres afin de modéliser des dommages causés par COVID-19
- ✓ Estimer la probabilité de ruine au bout d'un an (horizon fini) pour des deux jeux de paramètres par la méthode Monte-Carlo naïf et par échantillonnage préférentiel.

Ce que nous avons réalisé

- ✓ Estimer la probabilité de ruine et le temps de la ruine (horizon infini) pour des deux jeux de paramètres par la méthode Monte-Carlo naïf et par échantillonnage préférentiel.
- ✓ Etudier la méthode de changement de l'espace de probabilité et la transformation d'Esscher.
- ✓ Simuler une ruine rare à l'aide de transformation d'Esscher.
- ✓ Etudier le mécanisme de fonctionnement de compagnie diversifiée.
- ✓ Construire le graphe qui affiche l'évolution de la probabilité de ruine en fonction du capital de départ.
- ✓ Construire le graphe qui affiche l'évolution de la probabilité de ruine en fonction du l'horizon T.

Merci de votre attention