

Faculté des sciences

Modélisation actuarielle de la plateforme EUSOH

Auteur : Tchapnga Tchetchui Leonel Arol
Promoteur : Michel Denuit
Lectrice : Cindy Courtois
Année académique 2023 - 2024



École de statistique, biostatistique
et sciences actuarielles

Institut de Statistique, Biostatistique et Actuariat

Modélisation actuarielle de la plateforme Eusoh

Promoteur :
Prof. Denuit Michel
UCLouvain

Mémoire présenté en vue de
l'obtention du mastère
en sciences actuarielles

par :
Tchapnga Tchetchui Leonel Arol

Louvain-La-Neuve
Janvier 2024

Résumé

Les aléas du quotidien auxquels nous faisons face n'épargnent pas nos fidèles compagnons animaliers et les dépenses liées à la santé de ces derniers peuvent rapidement devenir considérables. Il serait donc judicieux de se prémunir contre ces éventualités afin de minimiser les risques de devoir supporter des coûts de santé élevés pour nos précieux animaux de compagnie. L'une des solutions à ce besoin est l'assurance P2P ou assurance collaborative, ce nouveau modèle est une forme d'assurance sans assureur utilisant les nouvelles technologies d'internet et d'intelligence artificielle pour créer des plateformes permettant de partager le risque au sein d'un groupe d'individus ayant des besoins similaires. Ce mémoire a pour objectif de proposer une modélisation actuarielle de la plateforme américaine Eusoh qui est une plateforme offrant des couvertures pour coûts élevés de soins vétérinaires pour chiens et chats. Cette modélisation explore les mécanismes permettant d'obtenir la distribution du coût total des réclamations au sein d'un pool. Une fois cette distribution connue, il sera question de partager la charge totale des réclamations entre les membres du pool en utilisant une méthode novatrice, à savoir la *Conditional Mean Risk Sharing (CMRS)* qui confère au pool des propriétés d'équité ; d'attractivité et de transparence. La mise en place d'un système d'assurance P2P avec un (ré)assureur, développé dans ce mémoire, permet d'absorber les chocs potentiels menaçant la stabilité des contributions au sein du pool. Ce modèle novateur offre ainsi une solution personnalisée aux besoins variés des individus, tout en réduisant financièrement les coûts d'assurance.

Abstract

The daily uncertainties we face spare not our loyal animal companions, and expenses related to their health can quickly become substantial. It would be wise to protect against these eventualities to minimize the risks of bearing high healthcare costs for our precious pets. One solution to this need is P2P insurance, or collaborative insurance. This innovative model is a form of insurance without a traditional insurer, utilizing internet and artificial intelligence technologies to create platforms that share risk among a group of individuals with similar needs. This thesis aims to propose an actuarial modeling of the American platform Eusoh, which offers coverage for high veterinary care costs for dogs and cats. The modeling explores mechanisms to obtain the distribution of the total claims cost within a pool. Once this distribution is known, the focus will be on sharing the total claims burden among pool members using an innovative method, namely Conditional Mean Risk Sharing (*CMRS*), which imparts fairness, attractiveness, and transparency to the pool. The establishment of a P2P insurance system with a (re)insurer, developed in this thesis, allows for absorbing potential shocks that threaten the stability of contributions within the pool. This innovative model offers a personalized solution to diverse individual needs while reducing insurance costs financially.

Remerciements

La réalisation de ce mémoire a été possible grâce au concours de plusieurs personnes à qui je voudrais témoigner toute ma gratitude.

Je tiens à exprimer ma profonde reconnaissance envers mon promoteur, le Professeur Michel Denuit, pour son encadrement exceptionnel tout au long de la réalisation de mon mémoire. Votre expertise, vos conseils éclairés, votre patience et votre soutien inestimable ont été des sources d'inspiration essentielles dans la réussite de ce travail.

Un merci particulier au corps enseignant et au personnel de la LSBA pour la qualité de la formation offerte.

À toi, mon pilier de force, Ngouonwe Nwambo, je veux exprimer ma gratitude pour ton soutien indéfectible tout au long de cette aventure. Ta compréhension, ton amour inconditionnel et ta patience infinie ont été mes plus grands encouragements. À mes filles Tchamegni et Lili, merci pour votre compréhension et votre adorable présence qui m'ont donné la force de me lever tous les matins. Cette réalisation est le fruit de notre unité et de notre amour.

J'exprime mes sincères remerciements à ma famille et à mes amis, particulièrement à Enkack Tientcheu Jacques Juste, pour le soutien incroyable que vous m'avez apporté tout au long de la rédaction de mon mémoire. Vos encouragements, vos conseils judicieux et votre présence constante ont été des éléments déterminants dans la réussite de ce projet. Merci pour le soutien, la motivation et de m'avoir rappelé que chaque étape accomplie était une victoire partagée.

Table des matières

Résumé	i
Abstract	i
Remerciements	ii
Liste des tableaux	vi
Liste des figures	vii
Introduction	1
1 Description de la Plateforme Eusoh	3
1.1 Introduction	3
1.2 Différents types de modèles dans les assurances <i>P2P</i>	4
1.2.1 Modèle de courtage P2P "Broker model"	4
1.2.2 Le modèle "Carrier"	5
1.2.3 Modèle autonome "Self-governing"	5
1.3 Présentation de la plateforme Eusoh	7
1.3.1 Offre de couverture	7
1.3.2 Règle d'adhésion	7
1.3.3 Les restrictions du contrat d'adhésion	7
1.3.4 Tarification du plan	8
1.4 Conclusion	9
2 Modélisation actuarielle	10
2.1 Introduction	10
2.2 Distribution de la fréquence et la sévérité des réclamations	11
2.2.1 Les facteurs de risques	11
2.2.2 Choix des paramètres de fréquence-sévérité	11
2.2.3 Etude des facteurs de risques sur un jeu de données	15
2.2.4 Restriction du contrat d'adhésion	16
2.3 Loi du Coût total de réclamations par police	17
2.3.1 Distribution des X_i	17
2.3.1.1 Fonction de répartition des X_i	18
2.3.1.2 Discrétisation des C_{ik}	18

2.3.1.3	Masse de probabilité des X_i	19
2.3.2	Distribution des Y_i	20
2.4	Distribution du coût total des réclamations	21
2.5	Conclusion	23
3	Règle de partage de risque	24
3.1	Introduction	24
3.2	Règle de partage de risques (Risk Sharing Rule)	24
3.3	Conditional mean risk sharing : CMRS	28
3.3.1	Caractéristiques d'un pool	29
3.4	Contribution individuelle	30
3.4.1	Système sans (ré)-assureur	31
3.4.2	Système avec (ré)-assureur	31
3.5	Conclusion	34
4	Étude de cas	36
4.1	Profil de risque de chaque sous groupe	36
4.2	Distribution du montant total de réclamations à la charge du pool	37
4.2.1	Distribution de la fréquence et du coût de réclamation	37
4.2.2	Distribution du Coût total de sinistre par participant	38
4.2.3	Distribution du montant total des réclamations par participant	38
4.3	Contributions individuelles : Système sans (ré)-assureur	39
4.3.1	Sensibilité de la CMRS à la taille du pool	40
4.3.2	Sensibilité de la CMRS à la franchise	41
4.3.3	Sensibilité de la CMRS au remboursement annuel	42
4.4	Contributions individuelles : Système avec (ré)-assureur	44
4.5	Conclusion	46
	Conclusion Générale	47
A	PROGRAMMATION R	I
A.1	Analyse des covariables	I
A.1.1	Influence de l'âge sur la fréquence de sinistre sur la période de couverture	II
A.1.2	Influence du poids sur la période de couverture	III
A.1.3	Influence des plans de couverture B et C sur la sévérité des réclamations	IV
A.2	Étude des distributions du coût total de réclamations	VI
A.2.1	Analyse descriptive	VI
A.2.2	Analyse de la fréquence des réclamations	IX
A.2.3	Étude la Sévérité des réclamations	XV
A.3	CONDITIONAL MEAN RISK SHARING CODE	XIX
A.3.1	Choix des paramètres des distributions de fréquences et de sévérités	XIX
A.3.2	Distribution du coût total individuel	XXV
A.3.3	Calcul de la distribution de la charge S_n	XXVII
A.3.4	Calcul de l'espérance conditionnelle	XXIX
A.4	Étude de la sensibilité des contributions : Taille ; Franchise ; Plafond	XXXIV
A.4.1	Sensibilité à la taille du pool	XXXIV
A.4.2	Sensibilité au plafond de réclamation	XXXIX

A.5	Système avec réassurer	LI
A.6	Variabilité des cotisations dans tranche inférieure $[0, \omega_n]$	LIII
A.6.1	Comparaison avec une variation du nombre de participant	LIII

Liste des tableaux

2.2.1 Description des différentes variables explicatives	12
2.2.2 Estimation des paramètres du modèle de fréquence de réclamation extraite de l'étude de PAI, SHAND et WANG 2006	12
2.2.3 Évolution de la fréquence de réclamation en fonction de l'âge du chien	13
2.2.4 Évolution de la fréquence de réclamation en fonction du poids du chien	13
2.2.5 Probabilité de soumettre des réclamations sur la période de couverture pour chiens âgés de 12 mois ; 60 mois et 120 mois	13
2.2.6 Probabilité de soumettre des réclamations sur la période de couverture pour chiens pesant 5kg ; 20kg et 35kg	14
2.2.7 Estimation des paramètres du modèle de sévérité extraite de l'étude de PAI, SHAND et WANG 2006	14
2.2.8 Évolution du coût de sinistre en fonction de l'âge avec une quote part de 80%	15
2.2.9 Différentes variables de la base de données	16
4.1.1 Profil de risques des participants d'un pool de taille $n = 800$	37
4.4.1 Résultat de la contribution de chaque participant pour chaque tranche	44
A.1.1 Paramètres de la sévérité.	IV
A.1.2 Paramètres de la sévérité	V
A.1.3 Probabilité d'avoir des coûts de sinistres positifs	V
A.1.4 Probabilité d'avoir des coûts de sinistres positifs	VI
A.2.1 Paramètre de la distribution de la fréquence des réclamations à l'aide GAM .	XIV
A.2.2 Paramètre de la distribution du coût des réclamations à l'aide GAM	XIX
A.5.1 Composition du pool	LII

Table des figures

4.2.1	Densité probabilité du nombre moyen de sinistres par an	37
4.2.2	Fonction de répartition du coût de sinistre pour chaque groupe	37
4.2.3	Masse de probabilité du coût total par individu de chaque groupe	38
4.2.4	Calcul direct Vs Approximation collective	39
4.2.5	Contributions individuelles de chaque groupe	39
4.3.1	Variabilité de $h_{i,n}^*(S_n)$ en fonction de la taille du pool	40
4.3.2	Convergence des cotisations pour chaque groupe	41
4.3.3	Influence du niveau de franchise sur la variabilité des contributions	41
4.3.4	Évolution des contributions en fonction de la franchise	42
4.3.5	Influence des montants de réclamation maximale sur les contributions	43
4.3.6	Évolution des contributions en fonction du plafond de réclamation annuel	43
4.4.1	Répartition des contributions dans la tranche inférieur en presence d'un (ré)- assureur	45
4.4.2	Effet sur les contributions du transfert de la tranche supérieure des réclamations à un (ré)-assureur	45
A.1.1	Évolution de la fréquence de réclamation en fonction de l'âge	II
A.1.2	Évolution de la fréquence de réclamation en fonction du poids	IV
A.2.1	Proportion de chaque espèce dans la base de données	VII
A.2.2	Répartition de la fréquence et du coût de réclamation dans le portefeuille	VIII
A.2.3	Importance des variables explicatives pour la fréquence des réclamations	X
A.2.4	Interaction entre les variables explicatives à l'aide de la statistique de Friedman	XI
A.2.5	Effet de l'âge des chiens sur la fréquence des réclamations	XII
A.2.6	Importance des variables explicatives pour le coût des réclamations	XV
A.2.7	Interaction entre les variables explicatives à l'aide de la statistique de Friedman	XVI
A.2.8	Effet de l'âge des chiens sur le coût des réclamations	XVII
A.6.1	Répartition des contributions dans la tranche supérieure	LVI

Introduction

De nos jours, le concept d'assurance sans assureur, communément appelé, en anglais, "*peer to peer*" ou *P2P*, est de plus en plus présent. Il s'agit d'une forme d'autofinancement impliquant uniquement les assurés qui décident de s'assurer entre eux par le biais des plateformes internet et sans l'intermédiaire d'un assureur. Avec l'émergence de ces nouvelles formes d'assurance, l'industrie de l'assurance a été témoin d'une révolution marquée par une approche collaborative inédite répondant à des besoins personnalisés. C'est dans cette vision que notre étude se concentre sur l'aspect actuariel de l'économie collaborative dans le secteur de l'assurance, en mettant particulièrement un accent sur l'assurance pour soins vétérinaires pour chiens et chats.

En premier plan de cette transformation, on retrouve Eusoh, une plateforme américaine pour soins vétérinaires aux chiens et aux chats. Son modèle novateur permet aux propriétaires de compagnons animaliers de réduire les coûts de soins vétérinaires pour chiens et chats. L'évolution rapide de l'économie collaborative via des plateformes d'assurances comme celle de Eusoh marque un changement significatif dans la manière dont les propriétaires d'animaux de compagnie souhaitent gérer les coûts liés aux soins vétérinaires. Cependant, une question fondamentale est de savoir comment la modélisation actuarielle peut-elle être appliquée dans le cadre des plateformes d'assurance pour chiens et chats afin d'optimiser la stabilité financière et la viabilité à long terme de ce système tout en préservant les principes clés de l'assurance P2P, à savoir l'équité, la transparence et la confiance au sein de la communauté des membres.

C'est dans cet élan que ce mémoire offre une contribution significative en proposant un modèle innovant répondant à des besoins de plus en plus personnalisés. notre travail vise donc à examiner dès le premier chapitre le paysage actuel de l'assurance collaborative, les différents modèles existants et leurs particularités, un aperçu sur les raisons de l'échec d'envol de certaines start-ups suivi d'une présentation détaillée du fonctionnement de la plateforme d'assurance collaborative "Eusoh". Par la suite, sur base du fonctionnement de la plateforme Eusoh, il sera question de présenter dans le chapitre suivant une modélisation actuarielle explorant ainsi les mécanismes permettant de déterminer la distribution du coût total de réclamations au sein du pool. Au-delà de la simple analyse du coût, le chapitre suivant, présente les mécanismes de partage des risques, en présentant une méthode novatrice de redistribution du coût total de réclamations basée sur la règle *Conditional Risk Mean Sharing*. Une étude de sensibilité approfondie sera réalisée pour évaluer l'impact de paramètres clés tels que le nombre de participants, la franchise et le montant maximal de réclamation sur la stabilité des contributions des membres du pool.

Pour conclure, nous explorerons une approche alternative qui intègre l'association d'un (ré) — assureur à la plateforme, offrant ainsi une solution pour gérer les charges qui pourraient dépasser la capacité du pool. En combinant ces différentes approches, cette étude offre des perspectives sur la manière dont la stabilité financière des plateformes d'assurance P2P peut être renforcée tout en préservant des principes fondamentaux tels que l'équité, la transparence et la confiance au sein de la communauté des membres afin d'assurer la pérennité d'un tel système.

Chapitre 1

Description de la Plateforme Eusoh

1.1 Introduction

Les désagréments de la vie et les divers dangers que nous rencontrons dans notre quotidien n'épargnent pas nos compagnons animaliers (chiens, chats, etc...). Les frais vétérinaires (opérations, rééducation...) qui en découlent sont dès lors très importants et il serait de ce fait intéressant de se couvrir afin de réduire les risques de se retrouver avec des coûts de santé élevés pour nos animaux de compagnie. Cependant, avant de prendre une assurance pour son compagnon, il faut en effet définir ses besoins. Il n'est pas forcément nécessaire de prendre la formule la plus complète au cas où son compagnon est en bonne santé et rarement malade. En définissant au plus vite ce qu'on attend de son contrat d'assurance, on se donne la possibilité de choisir la formule la plus adaptée en termes de rapport qualité/prix. On note de nos jours, un besoin chez les individus à vouloir appartenir à des groupes qui partagent des intérêts (sociaux, culturels, économiques) similaires, ce besoin se retrouve aussi dans le secteur de l'assurance où les individus se mettent ensemble pour payer des primes d'assurance. Afin de trouver des solutions à cette nouvelle demande de plus en plus croissante, de nouveaux paradigmes se sont développés tels que les assurances peer-to-peer (*P2P*). L'une des solutions à ce besoin est la mise sur pied de plateformes d'assurance collaborative dont le but est principalement d'aider les participants à échapper aux coûts énormes des services d'assurance, en particulier ceux relatifs à l'assurance pour soins vétérinaires aux chiens et aux chats. Les modèles d'assurance P2P diffèrent d'une forme à l'autre selon les modalités de fonctionnement et de gouvernance de la plateforme. On peut citer le modèle de courtage encore appelé "Broker model" qui est une forme d'assurance collaborative qui utilise des plateformes afin de regrouper des individus autour d'un intérêt commun. À ce modèle, il est greffé une entreprise d'assurance. Nous avons aussi le modèle "Carrier" dont le fonctionnement est similaire à celui des compagnies d'assurances traditionnelles, seulement les politiques de souscription et de règlement sont totalement digitalisées. Et enfin, nous avons le modèle autonome, encore appelé modèle "Self governing" où les participants prennent toutes les décisions concernant les questions d'assurances sans l'intervention d'une société d'assurance en appliquant un système communautaire de partage des coûts. Le but de chapitre est de présenter brièvement chacun de ces modèles tout en présentant un exemple pour chacun d'eux. Ensuite de décrire de façon détaillée le fonctionnement d'une de ces plateformes au nom de Eusoh qui est une plateforme

d'assurance collaborative permettant aux participants de faire face aux coûts énormes des soins de santé des animaux de compagnie (chiens et chats).

1.2 Différents types de modèles dans les assurances *P2P*

L'assurance P2P est un réseau de partage des risques dans lequel un groupe d'individus rassemble leurs primes pour s'assurer contre un risque.¹ Le fonctionnement des assurances de type P2P (Peer-to-Peer) présente des similitudes avec les mutuelles, car toutes deux ont pour objectif principal de mutualiser les risques afin de réduire les coûts d'assurance. Cependant, il existe des distinctions importantes.

Contrairement aux mutuelles, où les assurés sont également les propriétaires de la compagnie, les assurances P2P sont généralement des plateformes en ligne qui facilitent la mise en relation d'individus partageant des intérêts similaires. Les mutuelles peuvent utiliser des technologies, mais elles ont souvent des structures plus traditionnelles et sont basées sur les communautés, tandis que les assurances P2P sont étroitement liées à la technologie. Les plateformes en ligne connectent les participants ayant des intérêts similaires grâce au développement de la technologie et de l'intelligence artificielle. Les bénéfices des mutuelles sont redistribués sous forme de ristournes ou de dividendes, tandis qu'ils sont souvent utilisés à des fins caritatives dans les assurances P2P. Les assurances P2P offrent ainsi plusieurs avantages par rapport aux assurances traditionnelles telles que la transparence, la responsabilité des assurés, l'équité. Grâce à son fonctionnement, on distingue trois principaux modèles d'assurance P2P.

1.2.1 Modèle de courtage P2P "Broker model"

Le modèle de courtage ou « Broker model » se développe par la mise en place d'une plateforme qui a pour rôle de permettre aux participants de former des groupes et verser des contributions, qui servent d'une part à alimenter un fond commun et d'autre part à payer des primes d'assurance ou de réassurance. Au cours de la période de couverture, si le montant des prestations dépasse de la capacité du fond commun, alors la compagnie d'assurance ou de réassurance prend en charge l'excédent de sinistres, mais dans le cas contraire, le surplus est reversé à une œuvre caritative au choix des membres du pool ou redistribué aux participants.

La start-up anglaise Bought By Many² en est un exemple. Cette plateforme permettait à des individus ayant des besoins spécifiques de se retrouver au sein de mêmes groupes et la plateforme avait pour rôle de négocier de meilleurs prix pour eux avec les assureurs. Bought By Many proposait une couverture pour traitement vétérinaire pour les accidents et les maladies subis par les chiens et les chats, elle fournissait aussi une couverture pour les conditions préexistantes.

Grâce au nombre important de participants dans chaque groupe, Bought By Many détenait de ce fait un pouvoir important de négociation auprès des assureurs, car cela permettait d'une part de réduire les frais d'acquisitions chez la start-up et d'autre part améliorer la segmentation et le périmètre de souscription chez l'assureur. Face à son succès, elle lança

1. Jake Frankenfield (2021). Peer-to-Peer (P2P) Insurance

<https://www.investopedia.com/terms/p/peertopeer-p2p-insurance.asp>.

2. Société créée en 2012 par Steven Mendel et Guy Farley:https://www.Wikitia.com/wiki/bought_By_Many

en 2017, ses propres polices d'assurances au Royaume-Unis qui propose des assurances pour les animaux de compagnie (chiens et chats). De nos jours, Bought By Many a été rebaptisé ManyPets³ et continue à offrir des couvertures pour soins vétérinaires pour chiens et chats

1.2.2 Le modèle "Carrier"

Ce modèle est bien adapté au secteur de l'assurance traditionnelle. Son fonctionnement est similaire à celui des sociétés d'assurances traditionnelles, mais les politiques de souscription et de règlement de sinistre sont totalement digitalisées. Le fonctionnement de cette plateforme est le suivant : la société récupère un pourcentage de primes sur les polices d'assurances vendues comme frais de gestion et l'autre partie sert à régler les sinistres. Si le nombre de réclamations est faible pendant une période de couverture, un surplus peut être généré. Ce surplus peut être redistribué aux participants du groupe à la fin de la période où être versé à une association à but non lucratif choisie par les membres du groupe.

Fondée en 2016, la start-up française Luko⁴ propose des produits d'assurance dommage à l'instar de l'assurance habitation, trottinette électrique, propriétaire non occupant, habitation, colocation... Elle se prélève une commission fixe de 30% sur les cotisations de ses membres, ce qui représente les frais de gestion de la plateforme et 70% des cotisations seront mutualisées entre les membres du pool ayant souscrit à la même offre et sont destinées au remboursement des sinistres. Les cotisations non utilisées lors d'un exercice comptable servent aux financements des associations caritatives au choix du pool ou redistribuées chez les membres du pool. Ayant accumulé un niveau de dette élevée et faisant face aux difficultés financières, les opérations de Luko en France sont achetées par le groupe britannique Admiral⁵ qui conserve la marque.

1.2.3 Modèle autonome "Self-governing"

Développé totalement en dehors du cadre traditionnel du secteur de l'assurance, le modèle "Self-governing" fonctionne sans l'intermédiaire d'un assureur ou d'autre forme de tiers centralisateur, les fonctions d'assurance sont gérées par les participants. La plateforme est formée de groupes appelés communautés ou pools. Chaque pool est constitué de membres qui partagent des intérêts communs, par exemple des intérêts sur des objets assurés (voitures, maisons, vélos), sur des événements assurés (hospitalisation, santé des animaux de compagnies, franchise en RC auto), des affinités sociales ou professionnelles, ou encore par rapport au lieu de résidence (vivre dans la même ville, travailler dans le même bureau). Ici, les contributions servent à constituer un pot commun qui servira à régler les réclamations de l'année. Les contributions sont déterminées par des règles transparentes, chaque participant, dès son adhésion à la plateforme, est informé de la règle de partage de risque ; du montant de la franchise ; du montant maximal d'une réclamation. Avec ce modèle, il n'y a pas d'excédent à redistribuer en fin d'année, car les membres contribuent ex-post pour régler les réclamations à la charge du pool au courant de l'année.

Cette forme d'assurance collective s'est développée très tôt dans l'humanité. Nous pouvons

3. <https://www.manypets.com/uk/brand/>

4. Luko est une start-up française qui distribue des produits d'assurance dommage. Fondée le 6 décembre 2016 par Raphaël Vullierme et Benoît Bourdel. [Wikipedia.org/wiki/Luko_\(entreprise\)](https://fr.wikipedia.org/wiki/Luko_(entreprise))

5. www.usine-digitale.fr/article/clap-de-fin-pour-luko-rachete-par-le-britannique-admiral-group.N2143992

citer entre autres : les chameliers de Mésopotamie⁶ en -3 000 ans avant J.C. D'après les historiens, les chameliers se sont vite aperçus que lorsqu'ils participaient à une caravane dans le désert, tous les propriétaires de chameaux étaient soumis à un même risque ou à plusieurs risques : les tempêtes de sable, la perte de chameaux. Alors pour se couvrir de ces différents risques, ils formaient des groupes de tailles variables et s'engageaient à contribuer régulièrement à un fond commun. Lorsqu'un chamelier subissait une perte de chameaux, alors le fond commun constitué servait à indemniser la perte. Cependant, si un chamelier avait plusieurs sinistres, il ne recevait qu'une indemnisation pour la perte la plus importante. Cette technique a permis aux chameliers de se couvrir contre les différents risques lors de leurs caravanes et à renforcer leur solidarité et leur cohésion sociale.

De nos jours, avec l'évolution des technologies de l'internet, nous observons l'émergence d'une multitude de sociétés startups qui révolutionne le monde de l'assurance grâce au concept de P2P. Créée en 2014, la startup britannique Guevara⁷ en assurance automobile, permet à ses membres de se couvrir contre les coûts énormes de l'assurance automobile. La plateforme est constituée de plusieurs communautés dont les membres partagent des intérêts similaires (habitants d'un même village, club de propriétaires de Porsches). Chaque communauté est constituée soit par initiative des membres d'une communauté en fonction de leur besoin, soit par l'initiative de Guevara⁸ qui a pour rôle de segmenter les risques. L'adhésion à un pool se fait par simple remplissage d'un formulaire en ligne et elle permet les échanges entre les membres d'un pool afin d'améliorer l'offre de service. Ainsi, les contributions versées par les membres constituent un fond commun qui permet d'indemniser les sinistres subis au cours de l'année. Lorsque aucun sinistre ne s'est produit, le fond n'est pas engagé et aucune contribution n'est réclamée sauf la contribution mensuelle. Les contributions individuelles versées par les membres sont plafonnées, elles ne peuvent jamais dépasser leur niveau initial. À tout moment, si le besoin d'indemnisation excède la capacité d'un groupe, alors un fond spécial ("Waterloo Fund") constitué par Guevara prend en charge l'excédent. Par ailleurs, le modèle ne pouvant produire des effets positifs que si les communautés constituées sont suffisamment homogènes, pour cela, Guevara a prévu une politique d'exclusion des membres (mauvais participants) par voie de vote anonyme, sinon Guevara lui trouvera un autre groupe d'accueil, potentiellement moins favorable. Tous les événements tels que les contributions, le règlement de sinistres, les causes d'une exclusion sont connues de tous les membres de la communauté dès l'adhésion, car la règle de transparence est l'une des règles la plus importante du système. Il faut noter toutefois que la start-up Guevara arrête définitivement de fonctionner en 2017⁹.

Lancée en 2017, Eusoh¹⁰ est une plateforme qui offre une couverture contre le coût énorme des soins vétérinaires pour les chiens et chats, basée sur la communauté de pairs, le financement se fait par un modèle de partage des coûts.

6. "Cours des Droit des assurances, UCL 2023"- Professeur Bernard Dubuisson

7. <https://www.lassuranceenmouvement.com/2016/06/30/guevara-la-revolution-tant-attendue/>

8. <https://www.heyguevara.com/>

9. <https://blog.cestpasmonidee.fr/2017/10/guevara-est-mort-vive-lassurance-p2p.html>

10. Jacqueline Demarco (2022)

<https://www.investopedia.com/eusoh-pet-insurance-review-5080444>

1.3 Présentation de la plateforme Eusoh

Fondée en 2017 aux USA par Allen Kamrava, Eusoh¹¹ est une plateforme qui fonctionne sans l'aide d'une société d'assurance, la plateforme permet le partage des coûts de soins vétérinaires entre pairs afin de rendre la couverture plus abordable. Son mécanisme de gestion entre dans la catégorie des modèles autonomes ("Self-governing").

1.3.1 Offre de couverture

La première des choses à faire lorsqu'on souscrit à une police d'assurance et de savoir si l'offre satisfait à nos besoins. L'offre de la plateforme comprend un plan complet pour chiens et chats couvrant les accidents, la maladie, le bien-être, les médicaments et des consultations à distance. Mais certaines conditions ne sont pas couvertes telles que les médicaments de routine, le nettoyage dentaire, le toilettage, la nourriture et certaines conditions préexistantes.

1.3.2 Règle d'adhésion

La plateforme est destinée aux propriétaires de chiens et de chats qui adhèrent à des groupes auxquels les membres partagent les mêmes intérêts sociaux ; culturels et économiques. Elle est composée de plusieurs groupes homogènes et chaque groupe est appelé communauté. Dès l'admission de votre animal de compagnie, vous devenez membre d'un groupe de partage de coûts de soins vétérinaires. Lorsqu'il y a des réclamations au sein d'une communauté, le pool les agrège et les répartage entre les membres de la communauté. L'admission dans une communauté se fait par voie de vote des membres existants. Il n'y a pas de limite d'âge pour les animaux de compagnies qui demandent à rejoindre la communauté. Cependant, si un animal de compagnie est plus âgé, ou un animal de compagnie est susceptible de demander plus de soins au vu de ses antécédents, les membres de la communauté existante peuvent ne pas l'admettre.

1.3.3 Les restrictions du contrat d'adhésion

- **Stage d'attente et retrait d'adhésion**

Tous les membres admis dans l'une des communautés, doivent se soumettre à une période d'attente de trente jours. Durant cette période, le membre ne sera pas en mesure de soumettre des demandes de remboursement. Cette politique est mise en place afin d'éviter la fraude et l'antisélection car cela empêcherait des membres d'adhérer à la plateforme peu de temps avant une dépense vétérinaire importante, puis de se retirer une fois le remboursement soit effectué. Néanmoins, le membre peut se retirer à tout moment. Ainsi, il est possible d'annuler son adhésion dans les trente premiers jours suivant son adhésion et cela se fait sans aucune pénalité et en récupérant la totalité de vos dépôts initiaux. Passé cette période, si le membre souhaite se retirer au courant de la première année, alors, il perd son dépôt initial. Et au-delà de cette première année, pour toute annulation, le membre récupère son dépôt initial et les contributions non utilisées. En cas de décès de l'animal de compagnie, les mêmes conditions sont appliquées.

11. <https://www.investopedia.com/eusoh-pet-insurance-review-5080444#toc-available-plans>

- **Franchise annuelle et limite de remboursement**

Le montant annuel de la franchise s'élève à 250 \$. Il s'agit du montant que vous devez payer chaque année de votre poche, et non par incident, avant de pouvoir recevoir un remboursement. L'année ici est comptée à partir de la date à laquelle le membre adhère au pool.

Lorsqu'un animal est en consultation chez le vétérinaire, vous réglez toutes les dépenses à l'avance et soumettez les factures et les photos de reçus à la plateforme. Par la suite, la communauté s'engage à 80% sur les réclamations faites. Au cours d'une année, le montant maximal que la communauté peut vous rembourser pour les frais vétérinaires est de 8.500\$.

- **Réclamations**

Le montant des soins vétérinaires dépend de plusieurs facteurs : la nature de la maladie ou de l'affection de l'animal, la durée du traitement, le type d'animal à soigner, son poids et sa taille et dans certaines mesures l'expérience du vétérinaire, son niveau de compétences et les commentaires portant sur la qualité des soins et services, mais également la localisation du cabinet vétérinaire, et évidemment le type d'intervention à réaliser. Par exemple, en Belgique¹², la consultation de base chez un vétérinaire coûte en moyenne entre 30€ et 40€. Ce tarif ne concerne que le rendez-vous chez le praticien et non les actes qu'il réalisera au cours du rendez-vous. Le coût des actes réalisés s'ajoute donc au prix de la consultation. Ses frais vétérinaires peuvent être plus élevés, qu'on soit en zone urbaine comme Bruxelles-Capitale ou en zone rurale comme la Famenne, ceci à cause d'une part des coûts d'exploitation qui pourraient se refléter sur les tarifs. Lorsqu'un participant se rend chez le vétérinaire, il paie tous les services à l'avance. Ensuite, ce dernier soumet à sa communauté une photo de la facture et des reçus pour remboursement et une fois le cycle de facturation mensuel terminé le 28 de chaque mois, le remboursement approuvé est déposé dans son portefeuille Eusoh.

1.3.4 Tarification du plan

À l'inverse des sociétés d'assurances traditionnelles, il n'y a pas de primes mensuelles ou annuelles à payer. Dès l'acceptation d'un nouveau membre, le montant à verser chaque mois se divise en deux parties à savoir :

- Un montant fixe qui représente l'abonnement mensuel à la plateforme d'une valeur de 17\$, qui ne varie pas en fonction des facteurs comme l'âge, la race de l'animal ou le montant de la franchise, il est le même pour les chiens et les chats.
- Un montant variant entre 0 à 48\$ qui représente la contribution pour les réclamations à la charge de la communauté, qui sont déposés par les membres dans leurs portefeuilles numériques sur lesquels des montants seront prélevés au courant du mois.

Lorsque des réclamations arrivent au sein d'un groupe, ils sont agrégés et le total des réclamations au sein de ce groupe est partagé équitablement entre les membres du groupe, ce qui diminue le montant de 48\$, que chaque membre devra compléter le mois suivant et si le montant total des réclamations excède la capacité de la communauté alors pour réduire les coûts à supporter par chaque membre de la communauté, on fait appel à d'autres communautés.

12. <https://www.matmut.fr/assurance/chien-chat/conseils/cout-consultation-veterinaire>

Il faut aussi noter que certain mois, il peut arriver qu'aucune facture n'arrive dans le groupe et dans ce cas, il n'y aura pas de frais de partage au sien du groupe, voire au sein de la plateforme. L'engagement est alors reporté pour le mois suivant. Donc, le montant total qu'un membre paierait à la fin d'un mois sera au maximum de 65\$ de frais et au minimum 17\$.

Malgré son succès, Eusoh¹³ n'a pas été à l'abri de défis réglementaires importants qui ont conduit à sa chute. En effet, la start-up a rencontré des problèmes avec le régulateur d'un état aux États-Unis, où l'autorité a mené une enquête approfondie sur son modèle commercial. Bien qu'Eusoh ait coopéré et ait été reconnu comme conforme, la menace de poursuites judiciaires coûteuses a contraint Eusoh à quitter cet état. Néanmoins, cette décision a déclenché des enquêtes supplémentaires dans d'autres états. Malgré les tentatives de négociation avec les régulateurs et la recherche d'un partenaire pour accueillir leurs membres, des contraintes financières croissantes et des obstacles dans la recherche d'un partenaire ont finalement conduit à la fermeture d'Eusoh

1.4 Conclusion

L'assurance collaborative, aussi connue sous le nom d'assurance participative, connaît une forte émergence grâce aux plateformes en ligne et aux technologies telles que l'intelligence artificielle. Le succès de cette nouvelle forme d'assurance repose sur des principes comme la transparence, l'équité, la communication entre les membres et s'adapte aux besoins changeant de la communauté. Le secteur de l'assurance s'adapte à ce besoin émergent en constante évolution. De nombreuses startups d'assurance P2P voient le jour, à l'instar de la plateforme Eusoh. Cependant, il reste encore beaucoup de défis à relever, notamment en ce qui concerne les aspects réglementaires et conformité. Le secteur de l'assurance est soumis à une réglementation stricte par les autorités. Qu'en est-il des plateformes d'assurance P2P ? La plateforme Eusoh, point de départ de notre étude, bien voulant se greffer à une entreprise d'assurance traditionnelle, a dû cesser ses activités pour des raisons de réglementations, judiciaires et de conformité. Du point de vue technique, il sera donc question dans la suite de notre travail d'élaborer un modèle actuariel répondant à ce nouveau besoin en matière d'assurance collaborative, en général, et plus particulièrement dans le secteur de la santé animale.

13. <https://www.petinsuranceguidepodcast.com/2022/10/the-real-reason-eusoh-shutdown-its-pet-healthcare-coverage.html>

Chapitre 2

Modélisation actuarielle

2.1 Introduction

L'assurance P2P, ou assurance collaborative, est une forme d'assurance qui utilise les nouvelles technologies de l'internet et l'intelligence artificielle afin créer des plateformes de partage de risques, permettant de connecter ainsi des individus ayant des intérêts similaires au sein de groupes appelés pools ou communautés. Le fonctionnement du pool se résume ainsi : les membres d'un même pool versent des cotisations qui alimentent un fond commun, et ce fond sert ensuite à régler les réclamations à la charge du pool. Le secteur de la santé animale est un candidat idéal pour évoluer vers cette nouvelle forme d'assurance, offrant de ce fait de nombreux avantages aux propriétaires d'animaux de compagnie par rapport aux assurances traditionnelles. Premièrement, en se regroupant au sein d'une communauté de partage des coûts de soins de santé animale, l'assurance P2P permet aux participants de réaliser des économies par rapport aux primes d'assurance traditionnelles. De plus, soucieux de transparence et de communication entre les participants, l'assurance P2P permet aux membres de la communauté d'exercer un meilleur contrôle sur leurs dépenses et sur celles des autres participants grâce aux échanges entre participants via les plateformes. Néanmoins, il est crucial de bien comprendre les règles de la communauté (limitations de couverture et exclusions) et de s'assurer que tous les membres participent de manière équitable. Dans ce chapitre, en partant du coût d'une visite chez le vétérinaire pour soins de santé par animal (chiens ou chats), nous proposerons une modélisation actuarielle du coût total des réclamations (causé par la maladie) à la charge du pool. Nous supposerons que le nombre de visites chez le vétérinaire pour un chien i est une variable aléatoire noté N_i , suivant une loi de Poisson. Ensuite, nous ferons l'hypothèse que chaque visite chez le vétérinaire entraîne une réclamation (ou une perte positive) d'un montant C_{ik} qui est une variable aléatoire suivant une loi Log-Normale. Et enfin, nous appliquerons les différentes clauses (franchise, quote-part, plafond) afin d'obtenir la distribution du coût total de réclamation par participant. À partir du coût total des réclamations par participant, nous déterminerons la distribution du coût total des réclamations à la charge par le pool.

2.2 Distribution de la fréquence et la sévérité des réclamations

La plateforme de partage de risques est composée de plusieurs communautés dont les membres ont des intérêts communs, à savoir des questions relatives à la nutrition, à la santé et au bien-être de leur animal de compagnie. Considérons un groupe de n propriétaires de chiens ou de chats (dans la suite de notre travail, nous n'intéresserons qu'aux soins vétérinaires pour chiens). Soit N_i la variable aléatoire représentant le nombre de visites pour soins de santé d'un chien i chez le vétérinaire au cours de la période de couverture (12 mois). Chaque réalisation de la variable aléatoire N_i vaut un montant C_{ik} , $k = 1, 2, \dots, N_i$. Notons par X_i , la variable représentant le coût total de soins de santé pour un chien i . Les variables X_i , $i = 1, 2, \dots, n$ sont des variables aléatoires non négatives représentant le montant total des réclamations par participant du pool telle sorte que $P[X_i = 0] > 0$.

2.2.1 Les facteurs de risques

Les paramètres de fréquence et de sévérité des réclamations que nous utiliserons dans ce mémoire ont été extraits de l'étude menée par PAI, SHAND et WANG 2006 sur les réclamations d'assurance de chiens et de chats sur un marché canadien pendant une période d'observation de 8 ans. En effet, les données utilisées dans cette étude portent sur 91 743 réclamations sur un effectif de 15 933 animaux de compagnies assurés. Chaque animal a des caractéristiques propres à lui, à savoir les antécédents médicaux, l'âge, le territoire, le poids, le lieu de résidence, la race et le type de couverture. Lors de cette étude, on observe que les assurés ayant des antécédents médicaux ont des coûts d'assurance élevés. Certaines races sont plus prédisposées à certaines maladies, car elles sont considérées comme plus "fragiles" et ont un régime alimentaire plus coûteux et par ricochet peuvent avoir des coûts d'assurance plus élevés. Sur les 283 races répertoriées lors de l'étude, 191 races ont été classifiées comme ayant des coûts élevés et on peut les catégoriser en classe 1, classe 2 et classe 3. Les chiens de la classe 1 sont : Bulldogs, Great Danes, Dobermans, Schnauzers, Boxers et Dalmatiens. Certaines races vieillissent très vite, elles sont catégorisées en classe 2 qui contient : Newfoundlands, St. Bernards, Poodles, German Shepherds, Rottweilers et Greyhounds. Et la classe 3 contient toutes les autres races qui ne sont ni de la classe 1 ou de la classe 2. Comme les biens et services, les coûts de soins vétérinaires diffèrent d'une région à l'autre du pays. Certaines polices offrent plus d'options de couverture que d'autres, elles sont classées de la moins coûteuse à la plus coûteuse en Plan A, B, C et D. Le plan de couverture D est le plus complet, il comprend en plus une offre des soins dentaires. Le tableau 2.2.1 résume la description les différentes classes de risques pouvant affecter la sinistralité d'un participant.

2.2.2 Choix des paramètres de fréquence-sévérité

a) Le nombre de réclamations par participant : N_i

L'étude menée par PAI, SHAND et WANG 2006 utilise un modèle de Cox permettant de prendre en compte à la fois les variables explicatives et le temps depuis l'adhésion à un pool jusqu'à l'évènement où le temps entre un évènement et le suivant, ce modèle de Cox se définit par l'équation 2.2.1.

$$\lambda(t, Z_1, \dots, Z_m) = \lambda_0(t) \exp \left(\sum_{j=1}^m \beta_j Z_j \right) \quad (2.2.1)$$

Variable	Description
Z_1	antécédents médicaux(0 si premier , 1 sinon)
Z_2	Âge en mois
Z_3	zone d'habitation (1 si en région, 0 sinon)
Z_4	Classe 2 (1 Si le chien est de la race de classe 2, 0 sinon)
Z_5	Classe 1 (1 Si le chien est de la race de classe 1, 0 sinon)
Z_6	Poids (en kilogramme)
Z_7	Plan B (1 si le chien a un couverture Plan B ; 0 sinon)
Z_8	Plan C (1 si le chien a un couverture Plan C ; 0 sinon)
Z_9	Plan D (1 si le chien a un couverture Plan D ; 0 sinon)

TABLE 2.2.1 – Description des différentes variables explicatives

Où $\beta = (\beta_1, \beta_2, \dots, \beta_m)$ est vecteur des paramètres à estimer et $Z = (Z_1, Z_2, \dots, Z_m)$ un vecteur de covariables. $\lambda_0(t)$ correspond au risque de base, c'est-à-dire le risque de l'individu de référence. Sur une période de couverture de 12 mois, on suppose $\lambda_0(t)$ vaut 12λ . Les variables Z_1 et Z_2 représentant respectivement les antécédents médicaux et l'âge, elles sont supposées inchangées sur la période de couverture. Ainsi, pour la modélisation du risque collectif sur une seule période de couverture, le nombre de sinistres N_i suit une distribution de Poisson de paramètre :

$$\lambda_i = 12\lambda \exp\left(\sum_{j=1}^m \beta_j Z_j^i\right) \quad \text{avec } \lambda = \exp(-\alpha) \quad (2.2.2)$$

Le résultat de l'estimation des paramètres β_j obtenus dans l'article est présenté dans le tableau 2.2.2.

Variabes	Paramètres	Est	Sd	p-value
	α	2.67253	0.0387	0.000
Antécédents médicaux	β_1	1.23070	0.0130	0.000
Âge en mois	β_2	0.00026	0.0002	0.096
Zone d'habitation	β_3	0.02957	0.0116	0.011
Classe 2	β_4	0.04708	0.0193	0.015
Classe 1	β_5	0.11882	0.0248	0.000
Poids	β_6	-0.00058	0.0005	0.200
Plan B	β_7	-0.24512	0.0372	0.000
Plan C	β_8	-0.16021	0.0363	0.000
Plan D	β_9	0.42197	0.0358	0.000

TABLE 2.2.2 – Estimation des paramètres du modèle de fréquence de réclamation extraite de l'étude de PAI, SHAND et WANG 2006

Au vu des résultats du tableau 2.2.2, toutes les variables sont significatives au seuil de 5%, à l'exception des variables représentant le poids et l'âge. Il est probable que l'influence de ces variables sur la variable dépendante ne soit pas suffisamment importante pour expliquer les variations du nombre de sinistres au cours de la période de couverture. Cependant, elles n'ont pas été exclues de l'étude, car bien qu'elles ne soient pas statistiquement significatives, l'âge et

le poids pourraient avoir un effet qui est potentiellement atténué par d'autres covariables qui sont peut-être plus importantes et corrélées avec l'âge ou le poids. Pour le vérifier, considérons les deux illustrations suivantes :

• **Illustration 1**

Considérons un groupe de participants ayant des chiens dont les caractéristiques sont les suivantes : antécédents médicaux existants ; de la classe 1 ; vivant en zone urbaine ; qui a un poids de $10kg$; ayant souscrit au plan B. Et analysons la fréquence de réclamation en fonction de l'âge et du poids.

Age (mois)	12	24	36	48	60	72	84	96
λ_i	2.562047	2.562713	2.563380	2.564046	2.564713	2.565380	2.566047	2.566714

TABLE 2.2.3 – Évolution de la fréquence de réclamation en fonction de l'âge du chien

Poids (kg)	5	10	15	20	25	30	35	40
λ_i	2.575508	2.568049	2.560613	2.553198	2.545804	2.538432	2.531081	2.523752

TABLE 2.2.4 – Évolution de la fréquence de réclamation en fonction du poids du chien

En examinant les résultats de la fréquence de réclamation des chiens de ce groupe, nous constatons que cette fréquence augmente avec l'âge de façon exponentielle (Figure A.1.1). Cependant, lorsque nous observons des chiens âgés de 12 mois, 60 mois et 120 mois, nous remarquons que les probabilités de soumettre des réclamations sont quasiment équivalentes pour les trois tranches d'âge (Tableau 2.2.5).

Réclamation	$P[N_1 = k]$	$P[N_2 = k]$	$P[N_3 = k]$
0	0.07715	0.07694	0.07643
2	0.25320	0.25305	0.25268
4	0.13850	0.13871	0.13923
6	0.03030	0.03041	0.03069
8	0.00355	0.00357	0.00362

TABLE 2.2.5 – Probabilité de soumettre des réclamations sur la période de couverture pour chiens âgés de 12 mois ; 60 mois et 120 mois

Nous pouvons conclure de ce fait que l'effet de l'âge sur la fréquence de réclamation peut être atténué par d'autres facteurs, tels que les antécédents médicaux des animaux. En résumé, bien que l'âge puisse avoir un effet significatif sur la fréquence de réclamation chez les chiens de ce groupe, il se pourrait que cet effet soit probablement atténué par d'autres covariables potentiellement plus importantes et corrélées avec l'âge.

Par la suite, la figure A.1.2, nous suggère que la fréquence de réclamations diminue de façon exponentielle, mais très faiblement lorsque le poids du chien augmente. Cela implique que le poids a un effet non négligeable sur la fréquence de réclamations. Cependant, nous remarquons également que la probabilité de soumettre de réclamations est quasiment égale pour les chiens pesant 5 kg, 20 kg et 40 kg (Tableau 2.2.6). Comme pour l'effet de l'âge, cela

Réclamations	$P[N_1 = k]$	$P[N_2 = k]$	$P[N_3 = k]$
0	0.07600	0.07772	0.07946
2	0.25236	0.25361	0.25481
4	0.13966	0.13793	0.13619
6	0.03092	0.03001	0.02912
8	0.00367	0.00350	0.00333

TABLE 2.2.6 – Probabilité de soumettre des réclamations sur la période de couverture pour chiens pesant 5kg ; 20kg et 35kg

pourrait indiquer que l'effet du poids sur la fréquence de réclamations est atténué par d'autres facteurs plus importants. En somme, étant dans l'impossibilité d'avoir accès aux données d'analyse, nous sommes incapables de réaliser des tests statistiques approfondis.

b) Le coût de sinistre : C_{ik}

Le coût de sinistre C_{ik} est modélisé à l'aide de la loi Log-Normale définie par :

$$\log C_{ik} = \mu + \theta^T Z^{(i)} + \sigma W \quad \text{avec } W \sim \mathcal{N}(0, 1) \quad (2.2.3)$$

La communauté prend à sa charge un pourcentage ρ du montant des soins, le coût de sinistre devient alors : $\tilde{C}_{ik} = \rho C_{ik}$. La variable aléatoire \tilde{C}_{ik} reste une distribution Log-Normale notée

$$\log \tilde{C}_{ik} = \log \rho + \mu + \theta^T Z^{(i)} + \sigma W \quad (2.2.4)$$

Variables	Paramètres	Est	St	p-value
	μ	4.90284	0.0386	0.000
Antécédents médicaux	θ_1	-0.10522	0.0137	0.000
âge en mois	θ_2	0.00227	0.0002	0.000
Zone d'habitation	θ_3	0.12641	0.0120	0.000
Classe 2	θ_4	0.08862	0.0200	0.000
Classe 1	θ_5	0.14922	0.0256	0.000
Poids	θ_6	0.00355	0.0005	0.000
Plan B	θ_7	0.02991	0.0384	0.435
Plan C	θ_8	0.02971	0.0375	0.428
Plan D	θ_9	-0.90829	0.0370	0.000

TABLE 2.2.7 – Estimation des paramètres du modèle de sévérité extraite de l'étude de PAI, SHAND et WANG 2006

Avec $W \sim \mathcal{N}(0, 1)$. Les résultats de l'estimation des paramètres de ce modèle tirés de l'article sont résumés dans le tableau 2.2.7, l'estimation de σ est 1.03375, toutes les variables sont significatives au seuil de 5%, à l'exception des plans de couvertures Plan B et plan C. Ainsi, il se pourrait que l'influence de ces variables sur la variable dépendante ne soit pas assez suffisante pour expliquer les variations des coûts sur la période de couverture. Néanmoins, elles n'ont pas été retirées de l'étude, car bien qu'elles ne soient pas statistiquement significatives, les différents plans (B et C) de couverture pourraient avoir une importance pratique sur le coût de réclamations dans la réalité.

Les caractéristiques de la variable aléatoire \tilde{C}_{ik} s'obtiennent de la manière suivante :

$$\begin{aligned} E[\tilde{C}_{ik}] &= E\left[\rho \exp\{\mu + \theta^T Z^{(i)} + \sigma W\}\right] \\ &= \rho \exp\left\{\mu + \theta^T Z^{(i)}\right\} E(\exp\{\sigma W\}) \\ &= \rho \exp\left\{\mu + \theta^T Z^{(i)}\right\} \exp\left\{\frac{1}{2}\sigma^2\right\} \\ E(\tilde{C}_{ik}) &= \rho \exp\left\{\mu + \theta^T Z^{(i)} + \frac{1}{2}\sigma^2\right\} \end{aligned}$$

$$\begin{aligned} Var[\tilde{C}_{ik}] &= E[\tilde{C}_{ik}^2] - [E(\tilde{C}_{ik})]^2 \\ &= E\left[\rho^2 \exp\left\{2\mu + 2\theta^T Z^{(i)} + 2\sigma W\right\}\right] - \rho^2 \exp\left\{2\mu + 2\theta^T Z^{(i)} + \sigma^2\right\} \\ &= \rho^2 E\left[\exp\left\{2\mu + 2\theta^T Z^{(i)} + 2\sigma W\right\}\right] - \rho^2 \exp\left\{2\mu + 2\theta^T Z^{(i)} + \sigma^2\right\} \\ &= \rho^2 \exp\left\{2\mu + 2\theta^T Z^{(i)}\right\} E[\exp\{2\sigma W\}] - \rho^2 \exp\left\{2\mu + 2\theta^T Z^{(i)} + \sigma^2\right\} \\ Var[\tilde{C}_{ik}] &= \rho^2 \exp\left\{2\mu + 2\theta^T Z^{(i)} + \sigma^2\right\} (e^{\sigma^2} - 1) \\ \sigma_{\tilde{C}_{ik}}^2 &= Var[\tilde{C}_{ik}] \end{aligned}$$

• **Illustration 2 :**

Reprenons le groupe de participants de l'illustration 1, dont l'âge varie de 12 à 96 moins et le poids est de 10kg, ayant souscrit au plan B.

Age	12	24	36	48	60	72	84	96
$E(\tilde{C}_{ik})$	233.2489	233.7789	234.3102	234.8427	235.3704	235.9113	236.4474	236.9848
$\sigma_{\tilde{C}_{ik}}$	322.4757	323.2085	323.9431	324.6792	325.4171	326.1566	326.8979	327.6408

TABLE 2.2.8 – Évolution du coût de sinistre en fonction de l'âge avec une quote part de 80%

Pour les animaux de cette catégorie, on peut observer le coût moyen d'une visite chez le vétérinaire augmente avec l'âge, une analyse supplémentaire est faite à la section A.1.3.

2.2.3 Etude des facteurs de risques sur un jeu de données

Dans le souci de réaliser de pousser l'analyse sur les facteurs de risques susceptibles d'influencer la sinistralité chez les chiens et les chats, nous effectuons une étude similaire à celle PAI, SHAND et WANG 2006. À cette fin, les données utilisées pour cela proviennent de GitHub¹, cette base est constituée de deux espèces d'animaux de compagnie (chiens et chats). Chaque police a été observée pendant une période de deux ans et renferme les caractéristiques de risques décrites dans le tableau 2.2.9.

Notons que 83,96% de la base de données se compose de chiens. Par conséquent, notre étude se concentre exclusivement à cette catégorie, qui englobe 318 races uniques. Les âges des animaux de la base de données varient de 0 à 13 ans. En ce qui concerne la franchise, nous

1. <https://github.com/stevenrhart/pet-insurance-customer-segmentation/tree/master/data>

Type	Variable	Label
Variables réponses	NumClaimsYr1	Le nombre de réclamations au cours de la période de couverture
	AmtClaimsYr1	Montant de réclamation au cours de la période de couverture
Variables explicatives	AgeYr1	Âge de l'animal durant la période de couverture
	Deductible	Franchise individuelle
	Species	Différentes espèces de l'étude (Chiens ou Chats)
	Breed	Différentes races de l'étude
	annee	Année de survenance de la réclamation

TABLE 2.2.9 – Différentes variables de la base de données

utiliserons le niveau le plus représentatif. Pour des raisons pratiques, nous avons nettoyé la base de données en considérant que les cas avec un nombre de réclamations inférieur à 30 et un coût de réclamation inférieur à 15 000, ce qui représente 99,44% de la base de données des chiens. Le code et les résultats cette analyse sont présentés à l'annexe A.2.

En utilisant un GAM, nous remarquons un comportement inattendu, sur la figure A.2.5, le nombre de réclamations augmente avec l'âge, mais après 10 ans, le nombre de réclamations chute. En outre, bien que la variable "Breed" soit la plus importante d'après l'analyse faite par le GBM, l'analyse à l'aide du GAM nous montre qu'aucun des niveaux de cette variable n'est significatif. De ce fait, nous regroupons les niveaux de cette variable en 4 classes en utilisant la méthode Kmeans.

Ensuite, nous analysons l'effet de l'interaction entre la variable "AgeYr1" et "Breed" sur la fréquence des réclamations. Cette analyse nous permet de constater qu'au seuil de 5%, les animaux de la classe 1 et de la classe 4 sont significatifs, de même que certaines races. En ce qui concerne le coût des réclamations par participants, sur la figure A.2.8, il est à noter que le coût des réclamations augmente de façon croissante avec l'âge, plus un animal est âgé, alors il nécessitera des soins de santé plus coûteux. Comme précédemment, nous analysons l'effet de l'interaction entre l'âge et la race sur les coûts de réclamations et nous observons qu'avec un seuil de 5% seuls, les chiens de la classe 1 ont un effet significatif sur les coûts des réclamations.

En résumé, il est crucial de prendre en considération l'interaction entre la variable "AgeYr1" et la variable "Race" lors de l'évaluation de la fréquence et des coûts des réclamations liées aux soins vétérinaires. Cette interaction complexe souligne l'importance de déterminer judicieusement les regroupements de races pour une analyse plus précise. Cependant, la véritable complexité réside dans la détermination optimale des regroupements des différentes races.

2.2.4 Restriction du contrat d'adhésion

Sur une période de couverture de 12 mois, considérons N_i visites chez le vétérinaire (pour cause de maladie), chacune ayant une valeur \tilde{C}_{ik} . Alors, le montant total des réclamations par

participant est défini par :

$$X_i = \sum_{k=1}^{N_i} \tilde{C}_{ik} \quad i = 1, 2, \dots, n \quad (2.2.5)$$

Les variables \tilde{C}_{ik} sont iid et indépendantes à N_i .

a) La franchise annuelle

Dans le cadre de la santé des animaux de compagnie, la franchise prévue dans le contrat d'assurance est le montant que l'assuré doit payer chaque année de sa propre poche, et non par réclamation, avant de pouvoir recevoir un remboursement. Supposons que le montant de la franchise est noté d qui est la même pour tous les individus, le coût total des soins vétérinaires du membre i devient :

$$X_i^+ = \max\{X_i - d; 0\} = \begin{cases} 0 & \text{si } X_i \leq d \\ X_i - d & \text{si } X_i > d \end{cases} \quad (2.2.6)$$

b) Le plafond annuel d'intervention

L'intervention annuelle du groupe sur le montant total des soins vétérinaires pour un membre i est limitée à un montant M . Il résulte que le montant total de la réclamation à la charge du pool est donné par :

$$Y_i = \min\{X_i^+; M\} \quad (2.2.7)$$

2.3 Loi du Coût total de réclamations par police

Pour un membre i , le montant total de la réclamation à la charge d'une communauté se définit par :

$$Y_i = \begin{cases} 0 & \text{si } X_i \leq d \\ X_i - d & \text{si } d < X_i < M + d \\ M & \text{si } X_i \geq M + d \end{cases} \quad (2.3.1)$$

2.3.1 Distribution des X_i

Pour tout $i = 1, 2, \dots, n$, la variable aléatoire X_i représente le coût total de soins vétérinaires pour le participant i . On a $N_i \sim \mathcal{Poi}(\lambda_i)$. Alors :

$$X_i \sim \mathcal{CP}(\lambda_i, F_{X_i}) \quad \text{et} \quad F_{X_i}(x) = P[X_i \leq x] \\ E[X_i] = \rho \lambda_i E[C_{i1}] \quad \text{et} \quad \text{Var}[X_i] = \rho^2 \lambda_i E[C_{i1}^2]$$

2.3.1.1 Fonction de répartition des X_i

Notons par $F_{C_{ik}}(x) = P(C_{ik} \leq x)$ la fonction de répartition des C_{ik} et par $F_{C_{ik}}^{*n}$ le produit de convolution d'ordre n de $F_{C_{ik}}$ définie par :

$$F_{C_{ik}}^{*n}(x) = P(C_{i1} + C_{i2} + \dots + C_{in} \leq x), \quad x \in \mathbb{R}^+$$

Ainsi, on a :

$$\begin{aligned} F_{X_i}(x) &= P[X_i \leq x] \\ &= P(\rho C_{i1} + \rho C_{i2} + \dots + \rho C_{iN_i} \leq x) \\ &= P(C_{i1} + C_{i2} + \dots + C_{iN_i} \leq x/\rho) \\ &= \sum_{n=0}^{\infty} P(C_{i1} + C_{i2} + \dots + C_{in} \leq x/\rho | N_i = n) P(N_i = n) \\ &= \sum_{n=0}^{\infty} P(C_{i1} + C_{i2} + \dots + C_{in} \leq x/\rho) P(N_i = n) \\ F_{X_i}(x) &= \sum_{n=0}^{\infty} F_{C_{ik}}^{*n}(x/\rho) P(N_i = n) \end{aligned} \quad (2.3.2)$$

Cependant, un calcul direct de $F_{X_i}(x)$ est assez lourd, car il nécessite le calcul de n produits de convolution tout en tenant compte de la somme sur n . Afin de résoudre ce problème de temps de calcul, nous pouvons utiliser l'algorithme de Panjer qui nous permettra d'obtenir la masse de probabilité des variables aléatoire X_i de façon récursive mais cela nécessite une discrétisation de la loi de la variable aléatoire C_{ik} .

2.3.1.2 Discrétisation des C_{ik}

Il existe différentes méthodes de discrétisation de la loi de la variable aléatoire C_{ik} . Pour cela, il est nécessaire de définir la distribution C_{ik} pour des valeurs discrètes $0; t; 2t; \dots$ pour un pas donné t . Soit $F_{C_{ik}}(x)$ la fonction de répartition de la distribution à discrétiser sur un intervalle (a, b) et $\tilde{f}_{C_{ik}}(x)$ la masse de probabilité des variables C_{ik} en x . Notons par $E[X \wedge x]$ la valeur attendue de X limitée à x , définie par :

$$E[X \wedge x] = E[\min(X; x)]$$

La discretisation de la fonction de répartition $F_{C_{ik}}(x)$ peut donc être faite par plusieurs méthodes. (GOULET 2008, p15)

a) Discrétisation supérieure ou différence en avant de $F_{C_{ik}}(x)$

Cette méthode consiste à utiliser la différence en avant de la fonction de répartition $F_{C_{ik}}(x)$ de l'échantillon pour déterminer les probabilités associées à chaque point de discrétisation, c'est-à-dire :

$$\tilde{f}_{C_{ik}}(x) = F_{C_{ik}}(x+t) - F_{C_{ik}}(x) \quad \text{Pour } x = a; a+t, \dots, b-t \quad (2.3.3)$$

La fonction de répartition discrète est toujours au-dessus de celle continue

b) Discrétisation inférieure ou différence en arrière de $F_{C_{ik}}(x)$

Cette méthode consiste à utiliser la différence en arrière de la fonction de répartition $F_{C_{ik}}(x)$ de l'échantillon pour déterminer les probabilités associées à chaque point de discrétisation, c'est-à-dire :

$$\tilde{f}_{C_{ik}}(x) = \begin{cases} F_{C_{ik}}(a), & x = a \\ F_{C_{ik}}(x) - F_{C_{ik}}(x-t), & x = a+t; \dots; b \end{cases} \quad (2.3.4)$$

La fonction de répartition discrète est toujours en dessous de celle continue

c) Méthode du point médian

Cette méthode consiste à diviser l'intervalle de la distribution continue en intervalles de largeur égale, et on associe à chaque intervalle, le milieu de cet intervalle comme point de discrétisation les plus proches pour obtenir la version discrétisée de la variable aléatoire

$$\tilde{f}_{C_{ik}}(x) = \begin{cases} F_{C_{ik}}(a + \frac{t}{2}), & x = a \\ F_{C_{ik}}(x + \frac{t}{2}) - F_{C_{ik}}(x - \frac{t}{2}), & x = a+t; \dots; b-t \end{cases} \quad (2.3.5)$$

La fonction de répartition continue passe exactement au milieu de deux pas de la fonction de répartition discrète.

d) Méthode sans biais ou correspondance locale du premier moment

Cette méthode consiste à discrétiser une distribution continue en déterminant les masses de probabilités de la manière suivante :

$$\tilde{f}_{C_{ik}}(x) = \begin{cases} \frac{E[C_{ik} \wedge a] - E[C_{ik} \wedge (a+t)]}{t} + 1 - F_{C_{ik}}(a), & x = a \\ \frac{2E[C_{ik} \wedge x] - E[C_{ik} \wedge (x-t)] - E[C_{ik} \wedge (x+t)]}{t}, & x = a+t, \dots, b-t \\ \frac{E[C_{ik} \wedge b] - E[C_{ik} \wedge (b-t)]}{t} - 1 + F_{C_{ik}}(b), & x = b \end{cases} \quad (2.3.6)$$

Cette méthode offre l'avantage de parvenir à une distribution discrète présentant une probabilité totale équivalente à celle de la distribution continue. En plus, l'espérance de la distribution discrétisée est égale à la moyenne de la distribution continue sur (a, b) .

2.3.1.3 Masse de probabilité des X_i

Le coût de sinistre étant discrétisé, supposons à présent que C_{i1}, C_{i2}, \dots prennent des valeurs dans $0, 1, 2, 3, \dots$. On pose $l_j = P(C_{1k} = j)$ alors : $l_0 = P(C_{1k} = 0) = 0$ et $N_i \sim \mathcal{P}(\lambda_i)$, en utilisant l'algorithme de Panjer, dont la famille de Panjer est constituée des variables de comptages tels que :

$$P[N_i = n_i] = p_{n_i} = \left(a + \frac{b}{n_i}\right) \times p_{n_i-1}, \quad n_i \geq 1$$

Les valeurs de a et b doivent satisfaire : $a + b \geq 0$. Comme N_i est une distribution de Poisson, on choisit $\frac{p_{n_i}}{p_{n_i-1}} = \frac{\lambda_i}{n_i}$ alors $a = 0$ et $b = \lambda_i$. On obtient la distribution discrète des X_i de façon itérative de la manière suivante :

$$P(X_i = k) = \begin{cases} \exp(-\lambda_i) & \text{si } k = 0 \\ \frac{\lambda_i}{k} \sum_{j=1}^k j l_j P(X_i = k - j) & \text{si } k \geq 1 \end{cases} \quad (2.3.7)$$

2.3.2 Distribution des Y_i

Chaque membre soumet au pool, un montant Y_i , défini par :

$$Y_i = \begin{cases} 0 & \text{si } X_i \leq d \\ X_i - d & \text{si } d < X_i \leq M + d \\ M & \text{si } X_i > M + d \end{cases} \quad (2.3.8)$$

où d désigne la franchise ordinaire annuelle commune à tous les participants et M le montant maximal individuel de la réclamation annuel que le pool peut prendre en charge.

Soit $F_{Y_i}(y) = P(Y_i \leq y)$

- Si $y < 0$ comme $Y \geq 0$ alors $P(Y_i \leq y) = 0$;
- Si $y = 0$ alors $P(Y_i \leq 0) = P(Y_i = 0) = P(X_i \leq d) = F_{X_i}(d)$
- Si $0 < y < M$ alors :

$$\begin{aligned} P(Y_i \leq y) &= P(Y_i \leq y) \\ &= P(X_i - d \leq y) \\ &= P(X_i \leq y + d) \\ &= F_{X_i}(y + d) \end{aligned}$$

- Si $y \geq M$ alors $P(Y_i \leq y) = 1$;

La fonction de répartition des variables Y_i est donnée par :

$$F_{Y_i}(y) = \begin{cases} 0 & \text{si } y < 0 \\ F_{X_i}(d) & \text{si } y = 0 \\ F_{X_i}(y + d) & \text{si } 0 < y < M \\ 1 & \text{si } y \geq M \end{cases} \quad (2.3.9)$$

2.4 Distribution du coût total des réclamations

Dans cette section, il s'agit de déterminer la distribution du montant total des réclamations S_n à la charge d'un pool de n participants sur une période de couverture (généralement 12 mois).

$$S_n = Y_1 + Y_2 + \dots + Y_n \quad (2.4.1)$$

où les $Y_i, i = 1, 2, \dots, n$ sont définis par l'équation 2.3.1 et sont indépendantes. Notons $\gamma_i = E(Y_i)$; $F_{Y_i}(y) = P[Y_i \leq y]$ et par $F_{S_n}(s) = P[S_n \leq s]$. La distribution de S_n peuvent s'obtenir de manière directe de la manière suivante : Notons par $F_Y^{*(n)}$ le produit de convolution d'ordre n de F_Y définie par :

$$\begin{aligned} F_{S_n}(s) &= P[S_n \leq s] = P[Y_1 + Y_2 + \dots + Y_n \leq s] \\ &= \sum_{n=0}^{\infty} P[Y_1 + Y_2 + \dots + Y_N \leq s | N = n] P[N = n] \\ &= \sum_{n=0}^{\infty} F_Y^{*(n)}(s) P[N = n] \end{aligned}$$

Cette approche est très lourde en temps de calcul, une autre approche serait de faire un calcul direct en comptant les réclamations qui effectivement à la charge du pool. A cet effet, pour chaque participant i , on associe une indicatrice I_i telle que

$$I_i = \begin{cases} 0 & \text{Le coût total annuel des réclamations est inférieur à la franchise } (X_i \leq d) \\ 1 & \text{Le coût total annuel des réclamations est supérieur à la franchise } (X_i > d) \end{cases}$$

Le coût total des réclamations par participant au sein d'un groupe devient : $Y_i = I_i \tilde{Y}_i$. Ainsi, le nombre de réclamations au groupe est donné par : $I_{\bullet} = I_1 + I_2 + \dots + I_n$ et \tilde{Y}_i le montant total effectif des réclamations du participant i à la charge du groupe, sachant que le participant i a déclaré au moins un sinistre. Les variables aléatoires $I_1, I_2, \dots, I_n, \tilde{Y}_1, \tilde{Y}_2, \dots, \tilde{Y}_n$ sont indépendantes et on définit \tilde{Y}_i par : $\tilde{Y}_i =_d Y_i | Y_i > 0$. Alors, pour tout $i = 1, 2, \dots, n$, on peut réécrire le coût total des réclamations par participant :

$$Y_i = I_i \tilde{Y}_i = \begin{cases} \tilde{Y}_i > 0 & \text{si } I_i = 1 \\ 0 & \text{si } I_i = 0 \end{cases}$$

En remplaçant Y_i par $I_i \tilde{Y}_i$ avec $P[I_i = 1] = P[Y_i > 0]$, nous comptons ici tous les membres dont la réclamation est prise en charge par le pool alors $\sum_i Y_i$ a la même distribution que $\sum_i I_i \tilde{Y}_i$, dont la distribution est une Binomiale composée avec $P[I_i = 1]$ qui varie entre les membres du pool. D'autre part, on définit la fonction de répartition de \tilde{Y}_i , noté par $B_i(y)$ avec $y \geq 0$

$$B_i(y) = P(\tilde{Y}_i \leq y) = P(Y_i \leq y | Y_i > 0)$$

On définit par $q_i = P[Y_i > 0]$ et $p_i = 1 - P[Y_i > 0] = P[Y_i = 0]$. Soit $y \geq 0$, alors

$$\begin{aligned} F_{Y_i}(y) &= P[Y_i \leq y] \\ &= P[Y_i \leq y | Y_i > 0] P[Y_i > 0] + P[Y_i \leq y | Y_i = 0] P[Y_i = 0] \\ &= B_i(y) P[Y_i > 0] + P[Y_i = 0] \\ B_i(y) &= \frac{F_{Y_i}(y) - P[Y_i = 0]}{1 - P[Y_i = 0]} = \frac{F_{Y_i}(y) - F_{Y_i}(0)}{1 - F_{Y_i}(0)} \end{aligned}$$

En utilisant l'expression 2.3.3, on obtient :

$$B_i(y) = \begin{cases} 0 & \text{si } y = 0 \\ \frac{F_{X_i}(y+d) - F_{X_i}(d)}{1 - F_{X_i}(d)} & \text{si } 0 < y < M \\ 1 & \text{si } y \geq M \end{cases} \quad (2.4.2)$$

Le modèle individuel est très utile pour une tarification sur base individuelle, cependant, il devient moins pratique lorsque les calculs se font à l'échelle d'un groupe de participants. Une approche plus globale consiste à effectuer une approximation collective du modèle individuelle à l'aide d'un algorithme de Panjer. On note

$$S_n^{coll} = \sum_{i=1}^n Y_i =_d \sum_{k=1}^{N_\bullet} V_k \sim \mathcal{C}Poi(\lambda_\bullet, G) \quad (2.4.3)$$

Avec $N_\bullet \sim Poi(\lambda_\bullet)$ et $\lambda_\bullet = \sum_{i=1}^n \lambda_i^*$ et les variables aléatoires V_k sont indépendantes de fonction de répartition G définie par :

$$G(y) = \frac{1}{\lambda_\bullet} \sum_{i=1}^n \lambda_i^* P[Y_i \leq y | Y_i > 0] \quad (2.4.4)$$

Cette approche nécessite un choix du paramètre λ_i^* qui dépend de l'objectif à atteindre par le modèle collectif :

- **Choix 1** : $\lambda_i^* = q_i$

$$E[I_\bullet] = \sum_{i=1}^n q_i = \sum_{i=1}^n \lambda_i^* = \lambda_\bullet = E[N_\bullet]$$

On constate qu'en moyenne; le nombre de réclamations dans le modèle individuel est identique à celui dans le modèle collectif. Comme les variables Y_i sont indépendantes, on a

$$E[S_n] = \sum_{i=1}^n E[Y_i] = \sum_{i=1}^n q_i E[\tilde{Y}_i] = E[S_n^{coll}]$$

De même, ce choix permet en moyenne de conserver le coût moyen des réclamations.

$$\begin{aligned} Var[S_n] &= \sum_{i=1}^n Var[Y_i] = \sum_{i=1}^n \left(E[Var[Y_i | I_i]] + Var[E[Y_i | I_i]] \right) \\ &= \sum_{i=1}^n \left(q_i Var[\tilde{Y}_i] + q_i(1 - q_i) E[\tilde{Y}_i]^2 \right) \\ &= \sum_{i=1}^n \left(q_i [E[\tilde{Y}_i^2] - E[\tilde{Y}_i]^2] + q_i(1 - q_i) E[\tilde{Y}_i]^2 \right) \\ &= \sum_{i=1}^n \left(q_i E[\tilde{Y}_i^2] - q_i^2 E[\tilde{Y}_i]^2 \right) \leq \sum_{i=1}^n q_i E[\tilde{Y}_i^2] = Var[S_n^{coll}] \end{aligned}$$

En résumé, les espérances étant égales, on obtient une approximation du modèle individuel par un modèle qui a une variance élevée, ainsi l'approximation par le modèle collectif est une stratégie prudente.

- **Choix 2 :** $\lambda_i^* = -\ln(1 - q_i) \iff \exp(-\lambda_i^*) = 1 - q_i > 1 - \lambda_i^*$ il vient que $\lambda_i^* > q_i$, alors,

$$E[S_n] = \sum_{i=1}^n q_i E[\tilde{Y}_i] < \sum_{i=1}^n \lambda_i E[\tilde{Y}_i] = E[S_n^{coll}]$$

$$\forall i, \quad \exp(-\lambda_i^*) = 1 - q_i \iff P[N_i = 0] = P[Y_i = 0].$$

$$\text{De même, } P[N_\bullet = 0] = P[I_\bullet = 0] \iff P[S_n^{coll} = 0] = P[S_n = 0].$$

Ce choix permet d'avoir un modèle collectif dont les probabilités qu'aucun participant ne soumette de réclamations sont les mêmes que dans le modèle individuel et par ricochet, on retrouve ce même résultat au niveau du pool. De même, la moyenne et la variance dans le modèle collectif sont plus élevés.

En choisissant $\lambda_i^* = q_i$, on conserve la moyenne et la fonction de répartition de S_n^{coll} devient :

$$G(y) = \frac{1}{\lambda_\bullet} \sum_{i=1}^n q_i B_i(y) \quad (2.4.5)$$

La méthode de discrétisation sans biais permet d'avoir la distribution des variables Y_k . Alors, la distribution du coût total à la charge du groupe dans l'approximation collective S_n^{coll} est obtenue à l'aide de l'algorithme de Panjer. On note par :

$$p_k = P[N_\bullet = k] \text{ et } g_i = P[S_n^{coll} = i]$$

La famille de Panjer est constitué des variables de comptages tels que :

$$P[N_\bullet = k] = p_k = \left(a + \frac{b}{k}\right) \times p_{k-1}, \quad k \geq 1$$

Les valeurs de a et b doivent satisfaire : $a + b \geq 0$. Comme N_\bullet est une distribution de poisson alors $a = 0$ et $b = \lambda_\bullet$. On note par $f_i = P[\tilde{Y}_1 = i]$ alors :

$$f_0 = P[\tilde{Y}_1 = 0] = 0 \implies P[S_n^{coll} = i] = \begin{cases} \exp(-\lambda_\bullet) & \text{si } i = 0 \\ \frac{\lambda_\bullet}{i} \sum_{j=1}^i j f_j g_{i-j} & \text{si } i \geq 1 \end{cases} \quad (2.4.6)$$

2.5 Conclusion

En supposant que le nombre de visites chez le vétérinaire suit une loi de Poisson et que la distribution du coût d'une visite suit une loi Log-normale, nous avons pu déterminer la distribution du montant total des réclamations d'un participant. Cela nous a permis d'obtenir la distribution du coût total des réclamations à la charge du groupe. Dès à présent, se pose la question de trouver une méthode de partage transparente, optimale et équitable de cette charge entre les participants du pool, de manière à ce qu'il soit plus attractif de rejoindre un pool que de payer une prime d'assurance commerciale.

Chapitre 3

Règle de partage de risque

3.1 Introduction

Afin de proposer de nouveaux produits d'assurance qui permettraient aux assurés de faire partie d'une communauté de partage de primes d'assurance, aidant ainsi à réaliser des économies sur les prix de ces produits, des solutions en phase avec l'économie collaborative se développent de plus en plus. Les plateformes d'assurance P2P sont l'une de ces solutions, mises en place grâce au développement des technologies de l'intelligence artificielle, permettant ainsi de connecter les personnes souhaitant adhérer à une communauté de partage. Pour les assurances pour chiens et chats, l'émergence de cette nouvelle forme d'assurance répond à la nécessité de lutter contre les coûts énormes des soins de santé pour animaux. En pratique, la plupart de ces plateformes proposent un mode de fonctionnement qui se résume ainsi : lorsque le pool reçoit des réclamations, les assurés versent une contribution à une entité centrale. Cette entité prend un pourcentage de la contribution pour couvrir ses frais de gestion, tandis que l'autre partie sert à régler les réclamations. Il devient alors crucial de déterminer une méthode de partage des réclamations au sein de la communauté qui soit optimale, équitable et transparente, en prenant en compte le montant total des réclamations à la charge de la communauté, de sorte qu'aucun participant ne se sente lésé ou avantagé. Malheureusement, nous constatons que la plupart des startups qui utilisent l'assurance P2P sont soit rachetées par les compagnies d'assurance, soit tout simplement fermées. Le but de ce chapitre est, dans un premier temps, de définir la notion de règle de partage de risque au sein d'une communauté et d'énumérer quelques exemples et propriétés de cette règle. Ensuite, nous présenterons la règle de partage de la perte conditionnelle moyenne, introduite par DENUIT et DHAENE 2012, puis nous illustrerons son fonctionnement sur un groupe réduit de participants."

3.2 Règle de partage de risques (Risk Sharing Rule)

Considérons un pool de n propriétaires de chiens ou de chats. Sur une période de couverture de 12 mois, chaque animal est susceptible d'être atteint d'une maladie qui peut causer des coûts de soins vétérinaires. Le montant total des réclamations d'un participant i est noté $Y_i \geq 0$, de moyenne $\gamma_i = E(Y_i)$ et de variance $\delta_i^2 = Var(Y_i)$. On appelle pool tout vecteur aléatoire de réclamations $\mathbf{Y}_n = (Y_1, Y_2, \dots, Y_n)$. Le montant total des réclamations à la charge

d'un pool de n participants est noté :

$$S_n = Y_1 + Y_2 + \dots + Y_n \quad (3.2.1)$$

Définition 3.2.1. *Partage de risque (**Risk sharing**)*

Le partage de risque au sein d'un pool $\mathbf{Y}_n = (Y_1, Y_2, \dots, Y_n)$ est un processus à deux étapes.

► Une étape *ex-ante*, au temps 0, avant que les réclamations n'arrivent dans le pool. Les réclamations aléatoires à la charge du pool \mathbf{Y}_n sont réallouées en transformant \mathbf{Y}_n en un autre vecteur aléatoire $\mathbf{h}_{i,n}(\mathbf{Y}_n)$ de même dimension, le premier indexage désigne le participant i et le second la taille n du pool tel que :

$$\mathbf{h}_{i,n}(\mathbf{Y}_n) = \left(h_{1,n}(\mathbf{Y}_n), h_{2,n}(\mathbf{Y}_n), \dots, h_{n,n}(\mathbf{Y}_n) \right) \quad (3.2.2)$$

avec tous les $h_{i,n}(\mathbf{Y}_n) \geq 0$, et tel que la propriété d'allocation totale "**full allocation**" soit satisfaite i.e :

$$\sum_{i=1}^n h_{i,n}(\mathbf{Y}_n) = \sum_{i=1}^n Y_i = S_n \quad (3.2.3)$$

► L'étape *ex-post*, qui a lieu au temps 1, au moment où la réalisation $\mathbf{y}_n = (y_1, y_2, \dots, y_n)$ du pool \mathbf{Y}_n est observée. À ce moment, chaque participant du pool reçoit la réalisation y_i de sa réclamation Y_i . En retour, chaque participant contribue au pool, la réalisation $h_{i,n}(y_n)$ de sa réclamation ré-attribuée $h_{i,n}(\mathbf{Y}_n)$.

Définition 3.2.2. *Règle de partage de risque (**Risk sharing Rule**)*

La règle de partage de risque est une application $\mathbf{h}_{i,n}$ qui transforme tout pool \mathbf{Y}_n en un autre vecteur aléatoire de même dimension, noté

$$\mathbf{h}_{i,n}(\mathbf{Y}_n) = \left(h_{1,n}(\mathbf{Y}_n), h_{2,n}(\mathbf{Y}_n), \dots, h_{n,n}(\mathbf{Y}_n) \right)$$

avec tous les $h_{i,n}(\mathbf{Y}_n) \geq 0$, et tel que la condition 3.2.3 soit satisfaite. Par la suite, on notera

$$\mathbf{h}_{i,n}(\mathbf{Y}_n) = \mathbf{h}_{i,n}(S_n)$$

DENUIT, DHAENE et ROBERT 2022 ont proposé plusieurs exemples de règles de partage du montant total des réclamations entre les participants au sein d'un pool, on peut citer entre autres :

Exemple 3.2.2.1. *La règle de partage uniforme*

Le montant total des réclamations est partagé de façon uniforme entre tous les participants du pool.

$$\mathbf{h}_{i,n}(S_n) = \frac{S_n}{n} \quad i = 1, 2, \dots, n$$

Bien que la condition 3.2.3 soit satisfaite, ce partage pourrait être équitable et acceptable à condition que tous les membres du pool aient des besoins et des préoccupations similaires, et soumettent donc des réclamations équivalentes, il n'est pas toujours évident de former un tel groupe. Cependant, dans un groupe non homogène, ce partage ne prend pas en compte les différences individuelles dans le besoin d'assurance et la capacité de contribution.

Exemple 3.2.2.2. La règle de partage des risques proportionnels moyens

Chaque participant accepte de prendre une part fixe du montant total des réclamations au sein du pool d'assurance. La taille de cette part est fonction du montant espéré des réclamations à la charge du pool par rapport au montant total des réclamations. Ce partage se définit par :

$$h_{i,n}(S_n) = \frac{E(Y_i)}{E(S_n)} S_n$$

On constate que cette règle de partage respecte bien la condition 3.2.3. En effet ;

$$\begin{aligned} \sum_{i=1}^n h_{i,n}(S_n) &= \sum_{i=1}^n \frac{E(Y_i)}{E(S_n)} S_n \\ &= \frac{S_n}{E(S_n)} \sum_{i=1}^n E(Y_i) = \frac{S_n}{E(S_n)} E\left(\sum_{i=1}^n Y_i\right) \\ \sum_{i=1}^n h_{i,n}(S_n) &= \sum_{i=1}^n Y_i \end{aligned}$$

En moyenne, pour des réclamations plus importantes, les contributions seront aussi importantes car la contribution moyenne est proportionnelle au montant total de la réclamation. Ce qui semblerait être équitable. Ainsi, il serait de ce fait nécessaire de définir la notion d'équité de la règle de partage.

Propriété 3.2.1. Équité actuarielle "Actuarial fairness"

Une règle de partage h_{in} satisfait à la propriété "Actuarial fairness" si pour pool Y_n .

$$E[h_{i,n}(S_n)] = E[Y_i] \text{ pour tout } i = 1, 2, \dots, n \quad (3.2.4)$$

L'équité actuarielle est une règle importante pour le partage des réclamations totales entre les participants d'un pool, car elle garantit que chaque participant contribue en moyenne à hauteur de la réclamation espérée s'il choisissait de ne pas adhérer au pool. Ainsi, cette règle permet aux participants d'un pool de ne ni gagner ni perdre en moyenne du partage de risque, en assurant une répartition équitable des coûts entre les membres du pool.

On constate que la règle de partage proportionnel vérifie bien cette propriété

Définition 3.2.3. Dominance stochastique

Deux variables aléatoires X et Y sont ordonnées dans le sens de la dominance stochastique, noté $X \preceq_{st} Y$, si

$$E[u(-X)] \geq E[u(-Y)] \quad (3.2.5)$$

Pour toute fonction u non décroissante. Si $X \preceq_{st} Y$ alors la réclamation X est préférée à Y et par conséquent remplacer la réclamation X par Y est une stratégie prudente.

Propriété 3.2.2. Ordre convexe Les variables aléatoires X et Y sont ordonnées au sens de l'ordre convexe et on note $X \preceq_{cx} Y$ si :

$$\text{Pour toute fonction convexe } g, \text{ on a } : E[g(X)] \leq E[g(Y)] \quad (3.2.6)$$

L'ordre convexe donne les préférences d'un agent économique averse au risque entre les réclamations ayant les mêmes moyennes. Ainsi, X et Y étant de mêmes espérances, on peut remplacer la réclamation X par la réclamation moins attractive Y

Définition 3.2.4. Règle de partage Pareto optimal

Une règle de partage de risque $\mathbf{h}_{i,n}^*(S_n)$ est dite **Pareto-optimal**, si on ne peut pas trouver une autre règle de partage de risque $\mathbf{h}_{i,n}(S_n)$ telle que, on a :

$$\mathbf{h}_{i,n}(S_n) \preceq_{cx} \mathbf{h}_{i,n}^*(S_n) \quad (3.2.7)$$

Pour $i = 1, 2, \dots, n$ et dont au moins l'une des inégalités d'ordre convexe étant stricte.

Cette définition traduit le fait que le partage ne peut pas améliorer la situation d'un participant du pool (au sens de l'ordre convexe) sans toutefois dégrader celle d'au moins un autre.

Définition 3.2.5. Ensemble Comonotonique

• Un ensemble A est dit **comonotonique** si pour tout vecteur $X = (x_1, x_2, \dots, x_n)$ et $Y = (y_1, y_2, \dots, y_n)$ de A , on a :

$$x_i \leq y_i \quad \text{ou} \quad x_i \geq y_i \quad \text{pour tout} \quad i = 1, 2, \dots, n$$

• Un Vecteur aléatoire X est dit **comonotonique** s'il possède un support comonotonique.

Travailler dans un ensemble comonotonique présente plusieurs avantages. L'évolution dans le même sens des réclamations et des contributions individuelles permet d'éviter les conflits entre participants. De plus, cet espace permet d'avoir une vision claire des scénarios les plus défavorables, ce qui facilite la définition des règles de partage prenant en compte les préférences de tous les participants dans ces situations extrêmes.

Définition 3.2.6. Partage de risque comonotonique

Le vecteur aléatoire \mathbf{X} est dit **comonotonique** si pour tout i il existe une fonction continue et non décroissante $g_i : \mathbb{R} \rightarrow \mathbb{R}$ telle que :

$$\forall z \in \mathbb{R}, \quad \sum_{i=1}^n g_i(z) = z \quad \text{et} \quad \mathbf{X} = (g_1(S_n), g_2(S_n), \dots, g_n(S_n))$$

Le partage de risque comonotonique garantit qu'aucune contribution individuelle $h_{in}(S_n)$ ne saurait augmenter plus que le montant total des réclamations. Ainsi, la comonotonicité permet de répartir les réclamations entre participants de manière équitable tout en tenant compte des niveaux de risque individuel ; ce qui a pour effet de réduire l'exposition globale au risque, car si un participant subit une réclamation importante en raison d'un évènement imprévu, cela n'affectera pas les autres participants.

Afin de conférer à un pool les caractéristiques d'attractivité, d'équité et de transparence. La règle de partage de risque devrait respecter plusieurs propriétés décrites dans DENUIT, DHAENE et ROBERT 2022, nous reprenons ici quelques-unes d'entre elles.

Propriété 3.2.3. Propriété de "No-ripoff"

Une règle de partage $h_{i,n}$ respecte la propriété de no-ripoff si pour tout pool \mathbf{Y}_n , on a :

$$h_{i,n}(S_n) \leq F_{Y_i}^{-1}(1) \quad \text{pour tout } i = 1, 2, \dots, n \quad (3.2.8)$$

où $F_{Y_i}^{-1}(1) = \inf\{x \in \mathbb{R} | F_{Y_i}(x) = 1\}$, désigne la plus grande valeur possible de Y_i

En acceptant la règle de partage de risque, chaque participant ne devra contribuer que jusqu'à hauteur de sa réclamation maximale $F_{Y_i}^{-1}(1)$.

Propriété 3.2.4. Volonté d'adhésion "Willingness-to-join"

Une règle de partage $h_{i,n}$ satisfait à la propriété de **Volonté d'adhésion** si, pour tout pool \mathbf{Y}_n :

$$h_{i,n}(S_n) \preceq_{cx} Y_i \quad \text{pour tout } i = 1, 2, \dots, n \quad (3.2.9)$$

Un agent économique averse au risque, acceptera de rejoindre le pool et de participer à la charge totale du pool en payant $h_{i,n}(S_n)$ que de conserver sa position isolée et supporté Y_i .

3.3 Conditional mean risk sharing : CMRS

Afin de redistribuer le montant total des réclamations à la charge du pool entre les participants d'une communauté d'assurance P2P, DENUIT et DHAENE 2012 ont proposé un moyen, permettant d'avoir des contributions équitables tout en tenant compte du montant global des réclamations, du nom de **Conditional Mean Risk Sharing**, noté **CMRS**.

Définition 3.3.1. (Conditional mean risk sharing)

Le partage de risque conditionnel à la moyenne $h_{i,n}^*$ ou "Conditional mean risk sharing (CMRS)" est défini par :

$$h_{i,n}^*(S_n) = E[Y_i | S_n], \quad \text{pour } i = 1, 2, \dots, n \quad (3.3.1)$$

En d'autres termes, connaissant la charge globale du pool, les contributions individuelles correspondent en moyenne au montant que chaque participant soumet au pool.

Cette règle de partage vérifie bien la condition 3.2.3

$$\sum_{i=1}^n h_{i,n}^*(S_n) = \sum_{i=1}^n E[Y_i | S_n] = S_n$$

Propriété 3.3.1. La "Conditional mean risk sharing" $h_{i,n}^*(\mathbf{Y}_n)$ de Y_i minimise :

$$\sum_{i=1}^n E \left[(Y_i - h_{i,n}(S_n))^2 \right] \quad (3.3.2)$$

sur l'ensemble des règles de partage $h_{i,n}$

Parmi toutes les règles de partage de risque, la *CMRS* est celle qui minimise l'écart quadratique moyen entre la réclamation Y_i apporté au pool et la contribution individuelle $h_{i,n}(S_n)$.

Propriété 3.3.2. *Pour tout i , si $h_{i,n}^*(s)$ sont comonotoniques, i.e $s \mapsto E(Y_i|S_n = s)$ sont non décroissante, alors la *CMRS* est Pareto optimale.*

Lorsque des individus font face à des coûts de soins vétérinaires, deux choix s'offrent à eux, gérer les factures individuellement, ou collectivement en adhérant à pool et négocier sur d'une règle de partage de la charge globale. Une règle de partage de risque comme la *CMRS* permet une amélioration de leur situation du point de vue de l'ordre convexe, car en gérant les réclamations collectivement, ils sont prêts à accepter un certain niveau de perte ou de renonciation pour réduire leur exposition au risque.

- La *CMRS* vérifie la propriété de **Volonté d'adhésion**

Soit g une fonction convexe, montrons que $E[g(h_{i,n}^*(S_n))] \leq E[g(Y_i)]$ pour tout $i = 1, 2, \dots, n$

$$\begin{aligned} E[g(Y_i)] &= E[E[g(Y_i)|S_n]] \\ &\geq E[g(E[Y_i|S_n])] \quad \text{Inégalité de Jensen} \\ \text{Alors } E[g(Y_i)] &\geq E[g(h_{i,n}^*(S_n))] \end{aligned}$$

On conclut alors : $h_{i,n}^*(S_n) \preceq_{cx} Y_i$ pour tout $i = 1, 2, \dots, n$

- La *CMRS* vérifie la propriété 3.2.3. En effet, pour tout $i = 1, 2, \dots, n$, les variables aléatoires Y_i sont discrètes et positive. On note par $F_{Y_i}^{-1+}(0) = \sup\{x \in \mathbb{R} | F_{Y_i}(x) = 0\}$ la plus petite valeur possible de Y_i . $\forall i = 1, 2, \dots, n$, $P[Y_i \in [F_{Y_i}^{-1+}(0); F_{Y_i}^{-1}(1)]] = 1$. Alors ;

Soit s une réalisation non nul de S_n , On a :

$$\begin{aligned} h_{i,n}^*(s) &= E(Y_i|S_n = s) = \sum_{k=1}^s kP[Y_i = k|S_n = s] \\ &\leq \sum_{k=1}^s F_{Y_i}^{-1}(1)P[Y_i = k|S_n = s] = F_{Y_i}^{-1}(1) \\ \text{or } \sum_{k=1}^s F_{Y_i}^{-1}(1)P[Y_i = k|S_n = s] &= F_{Y_i}^{-1}(1) \sum_{k=1}^s P[Y_i = k|S_n = s] = F_{Y_i}^{-1}(1) \end{aligned}$$

- La *CMRS* vérifie la propriété 3.2.1 :

$$E[h_{i,n}^*(S_n)] = E(E[Y_i|S_n]) = E(Y_i) \quad i = 1, 2, \dots, n$$

3.3.1 Caractéristiques d'un pool

Le partage de la perte totale à la charge du pool en utilisant la *CMRS* procure au pool les caractéristiques suivantes :

Propriété 3.3.3. *Considérons un participant i avec une fonction d'utilité concave u_i et une richesse initiale w_i , faisant partie d'un pool de taille n*

$$E[u_i(w_i - Y_i)] \leq E \left[u_i \left(w_i - h_{i,n}^*(S_n) \right) \right] \text{ pour tout } i \in \{1, 2, \dots, n\} \quad (3.3.3)$$

Comme dans la théorie de l'utilité, la préférence d'un agent économique averse au risque, sera d'adhérer au pool et contribuer $h_{i,n}^(S_n)$ que d'adopter une position isolée et conserver son risque Y_i .*

Propriété 3.3.4. *DENUIT et ROBERT 2021bp.9&10*

Considérons des pertes indépendantes Y_1, Y_2, \dots, Y_n avec $S_n = Y_1 + Y_2 + \dots + Y_n$, alors l'inégalité stochastique suivante est toujours vérifiée

$$E[Y_i | S_{n+1}] \preceq_{cx} E[Y_i | S_n] \text{ pour tout } i \in \{1, 2, \dots, n\} \text{ et pour tout entier } n \quad (3.3.4)$$

L'utilisation de la CMRS permet l'adhésion de nouveaux membres au pool. Ce qui est toujours bénéfique pour un agent économique averse au risque.

En d'autres termes, il faut souligner qu'il est souhaitable d'avoir un pool donc le nombre de participants est de plus en plus grand, pour autant que les contributions individuelles soient déterminées selon la règle de la CMRS et que les pertes subies par le pool soient totalement indépendantes. Cette propriété permet de démontrer que la dispersion entre les contributions devient de plus en plus faible lorsque le pool devient de plus en plus grand. Sous des conditions énoncées dans l'article DENUIT et ROBERT 2021a p.7, lorsque la taille du pool augmente, les contributions obtenues à l'aide de la CMRS tendent à se stabiliser autour de la prime pure qu'aurait payé chaque participant.

$$\lim_{n \rightarrow \infty} h_{i,n}^*(S_n) = E[Y_i] \text{ avec une probabilité } 1 \quad (3.3.5)$$

La relation 3.3.5 indique lorsque le pool devient très grand (n très grand, mais fini) alors les contributions obtenues grâce à la CMRS semblent se stabiliser autour de la prime pure $E[Y_i]$.

3.4 Contribution individuelle

L'émergence de nouvelles startups marque une véritable révolution dans le domaine de l'assurance, grâce à l'introduction du concept P2P. Dans la pratique, pour la plupart d'entre elles, le mode de fonctionnement se résume comme suit : les membres versent une contribution dans un fond commun. Une entité centrale prend un pourcentage de ce fond pour ses frais de gestion, une autre partie pourrait servir à payer les réclamations courantes, verse une partie de la contribution à un (ré)-assureur qui devrait prendre les réclamations excédentaires, et retourne les surplus aux membres. DENUIT et DHAENE 2012 ont proposé une méthode permettant de déterminer la contribution de chaque participant à la perte totale supportée par le pool.

3.4.1 Système sans (ré)-assureur

Dans ce système, les contributions se font ex-post au moment de la réalisation de la charge totale S_n . Et ces contributions sont destinées à couvrir, S_n et par conséquent, nous n'avons ni d'excédent de charge ; ni de surplus à redistribuer. La distribution de S_n étant approchée par le modèle collectif, alors :

Si la réalisation s de S_n est nulle alors on a : $E(Y_i|S_n = 0) = 0$, pas de contribution venant de la part des membres. Et dans le cas où la réalisation s de S_n est non nul, chaque participant contribue à hauteur de :

$$\begin{aligned} h_{i,n}^*(s) &= E(Y_i|S_n = s) = \sum_{k=1}^s kP(Y_i = k|S_n = s) \\ &= \sum_{k=1}^s k \frac{P(Y_i = k; S_n = s)}{P(S_n = s)} = \sum_{k=1}^s k \frac{P(Y_i = k)P(\sum_{j \neq i} Y_j = s - k)}{P(S_n = s)} \end{aligned} \quad (3.4.1)$$

En posant $S'_n = \sum_{j \neq i} Y_j$, on a $S'_n =_d \sum_{k=1}^{M_i} T_{ik}$ avec $M_i \sim Poi(\lambda_\bullet - \lambda_i)$. Les variables T_{i1}, T_{i2}, \dots sont indépendantes et distribuées comme une variable T_i tel que :

$$P(T_i = k) = \frac{1}{\lambda_\bullet - \lambda_i} \sum_{j \neq i} \lambda_j P(\tilde{Y}_j = k) \quad k = 0, 1, \dots \quad (3.4.2)$$

avec $\tilde{Y}_j =_d Y_j | Y_i > 0$

Dans un contexte auquel les contributions des membres ne sont connues qu'à la fin de chaque période et peuvent connaître des variations significatives d'une période à l'autre, il pourrait avoir des situations dans lesquelles les participants ne désirent pas ou ne sont pas en mesure de contribuer. Dans de telles circonstances, le succès du système dépend largement de la confiance mutuelle entre les participants. À moins que les participants ne soient disposés à verser des cotisations substantielles, la capacité du pool à indemniser un certain niveau de réclamations se trouve naturellement restreinte. Face à ces défis, DENUIT 2020, propose une solution alternative, c'est-à-dire système alternatif intégrant un (ré)-assureur dans le processus.

3.4.2 Système avec (ré)-assureur

Dans ce système, la plateforme d'assurance P2P agit comme un intermédiaire et achète une couverture au nom des participants qui servira à couvrir l'excédent des réclamations. Dans cas, les participants versent une contribution fixe en début de période, avec la possibilité de recevoir un remboursement en cas de résultat favorable. Ainsi, la contribution aléatoire $h_{i,n}^*(Y_n)$ versée en fin de période est remplacée par une contribution fixe en début de période constituée de :

- Une provision en début de période, $\pi_{i,n}$ qui dépend à la fois de la distribution de Y_i et celle de S_n
- Un cash-back $B_{i,n}$ en fin de période, lorsque le montant des contributions effectives est supérieur au montant total des réclamations soumis au pool.

Dans ce modèle, après réception des réclamations de tous les membres du pool, la plateforme souscrit un contrat stop-loss auprès d'un (ré)-assureur de rétention, w_n alors le montant à la charge du pool est :

$$S_n^{P2P} = \min\{w_n; S_n\} \quad (3.4.3)$$

Alors, la contribution maximale fixe payée par chaque participant est $\pi_{i,n}$, qui permettra au pool de prendre à sa charge le montant total des réclamations de la tranche inférieure $[0; w_n]$ et de transférer la tranche supérieure $(w_n; \infty)$ chez un (ré)-assureur. on peut donc décomposer la contribution totale comme

$$\pi_{i,n}^{tot} = \pi_{i,n}^{P2P} + \pi_{i,n}^{SL} \quad (3.4.4)$$

- $\pi_{i,n}^{P2P}$ part de la contribution totale de chaque participant, permettant de financer la tranche inférieure, déterminé comme dans le modèle sans assureur tout en tenant compte du niveau de rétention
- $\pi_{i,n}^{SL}$ part de la contribution totale permettant de financer la contribution stop-loss à la charge d'un (ré)-assureur, dont la gestion devrait engendrer des coûts supplémentaires tel que les chargements de frais de gestion et d'administration.

a) Définition du niveau de rétention

La définition d'un niveau de rétention souhaité est une pierre angulaire dans l'élaboration d'un système d'assurance P2P de type courtier, car c'est le niveau permettant au pool de fonctionner sans dépendre entièrement d'un appui externe, il permet d'offrir des cotisations compétitives tout en garantissant la satisfaction et la confiance des membres. La valeur de la rétention w_n peut être définie de sorte que $P[S_n < w_n]$ soit la plus grande possible de sorte qu'on puisse avoir un surplus à redistribuer entre les participants en fin de période.

Soit β la probabilité de distribuer un surplus aux participants, alors :

$$\beta = F_{S_n}(w_n) \iff w_n = F_{S_n}^{-1}(\beta) \quad (3.4.5)$$

b) Contribution pour la tranche inférieure $[0; w_n]$

Supposons que le groupe est assez large et que la règle de partage utilisée est une fonction non décroissante de la réalisation s . Une fois le niveau de rétention w_n définie tel que $0 < F_{S_n}(w_n) < 1$. On définit les quantiles $w_{1,n}; w_{2,n}; \dots; w_{n,n}$ représentant la contribution maximale d'un participant dans la tranche inférieure :

$$\begin{aligned} \pi_{i,n}^{P2P} &= w_{i,n} \\ &= F_{E[Y_i|S_n]}^{-1}(F_{S_n}(w_n)) \\ &= F_{E[Y_i|S_n]}^{-1}(\beta) \\ &= E[Y_i|S_n = w_n] = h_{i,n}^*(w_n) \end{aligned} \quad (3.4.6)$$

A l'aide de la propriété 3.2.3, on peut s'assurer que le pool détient assez de ressources financière pour couvrir cette tranche; on a :

$$\sum_{i=1}^n w_{i,n} = \sum_{i=1}^n h_{i,n}^*(w_n) = w_n \quad (3.4.7)$$

Comme les fonctions $h_{i,n}^*$ sont continues et non décroissantes, alors on a :

$$S_n \geq w_n \iff h_{i,n}^*(S_n) \geq w_{i,n} = F_{h_{i,n}^*(S_n)}^{-1}(F_{S_n}(w_n)), \text{ pour } i = 1, 2, 3, \dots, n$$

Alors,

$$\begin{aligned} (S_n - w)_+ &= \left(\sum_{i=1}^n \left(h_{i,n}^*(S_n) - F_{h_{i,n}^*(S_n)}^{-1}(F_{S_n}(w_n)) \right) \right)_+ \\ &= \sum_{i=1}^n \left(h_{i,n}^*(S_n) - F_{h_{i,n}^*(S_n)}^{-1}(F_{S_n}(w_n)) \right)_+ \\ &= \sum_{i=1}^n \left(h_{i,n}^*(S_n) - w_{i,n} \right)_+ \end{aligned} \quad (3.4.8)$$

La décomposition de $(w - S_n)_+$ est faite de manière similaire en une somme de variables aléatoires $(w_{i,n} - h_{i,n}^*(S_n))_+$. Permettant ainsi de distribuer de façon appropriée le surplus $(w - S_n)_+$ et part la même occasion de déterminer la contribution individuelle correspondante pour la charge retenue $S_n^{P2P} = \min\{w_n; S_n\}$

$$\begin{aligned} S_n^{P2P} &= w_n - (w_n - S_n)_+ \\ &= \sum_{i=1}^n \left(w_{i,n} - (w_{i,n} - h_{i,n}^*(S_n))_+ \right) \\ &= \sum_{i=1}^n \min\{h_{i,n}^*(S_n), w_{i,n}\} \end{aligned} \quad (3.4.9)$$

Ainsi, pour la tranche inférieure, chaque membre contribue $\min\{h_{i,n}^*(S_n), w_{i,n}\}$ où $w_{i,n}$ est le niveau de rétention individuel.

c) Contribution pour la tranche supérieure $(w_n; \infty)$

L'émergence de nouvelles startups cherchant à s'établir sur le marché de l'assurance ne se traduit pas par l'adoption d'un modèle P2P pur, ni même par la création d'un modèle entièrement nouveau. Au contraire, elles intègrent un modèle existant dans une nouvelle architecture à la pointe de la technologie. En général, leur fonctionnement est le suivant : la communauté assume la responsabilité de la tranche inférieure $(0; w_n)$ et transfère la tranche supérieure $(w_n; \infty)$ à un (ré)-assureur. Toutefois, la question de la répartition de cette prime entre les participants se pose.

Le choix de la méthode dépend des objectifs de la communauté P2P et de son souci d'équité dans la répartition. Ainsi, la prime stop-loss pour la tranche supérieure peut être répartie de manière égale ; chaque participant paierait le même montant pour cette protection, quel que soit son niveau de contribution aux réclamations. Alternativement, chaque participant pourrait la répartir en tenant compte de sa contribution individuelle à la tranche supérieure des réclamations. Par souci d'équité, de transparence et d'attractivité, la seconde approche sera retenue.

Le prix pour la couverture stop-Loss est de la forme : $\pi_{SL} = (1 + \theta_{SL})E[(S_n - w_n)_+]$ avec θ_{SL} qui désigne les chargements pour obtenir la prime demandée par le (ré)assureur

pour la couverture de la tranche supérieure. La relation (3.4.7) nous permet de déterminer la contribution de chaque participant à la prime stop-Loss totale, on a :

$$E[(S_n - w_n)_+] = \sum_{i=1}^n E[(h_{i,n}^*(S_n) - w_{i,n})_+]$$

Et chaque participant contribue au transfert à la tranche supérieure à hauteur de :

$$\pi_{i,n}^{SL} = (1 + \theta_{SL})E[(h_{i,n}^*(S_n) - w_{i,n})_+]$$

d) Redistribution du surplus

Lorsqu'en fin de la période, $S_n \leq w_n$, un bénéfice se dégage et peut soit être attribué aux participants ou donné à une œuvre caritative. Si B_{P2P} désigne le montant de ce bénéfice, alors :

$$B_{P2P} = (w_n - S_n)_+$$

Si le pool décide de distribuer B_{P2P} , alors chaque participant recevra $B_{i,n}$ défini par :

$$B_{i,n} = \begin{cases} 0 & \text{si } S_n \geq w_n = \sum_{i=1}^n \pi_{i,n}^{P2P} \\ \pi_{i,n}^{P2P} - E[Y_i|S_n] & \text{si } S_n \leq w_n \end{cases} \quad (3.4.10)$$

$\pi_{i,n}^{P2P} = E[Y_i|S_n = w_n]$ cela garantit que la contribution de chaque participant sera au plus $\pi_{i,n}^{P2P}$ pour toute réalisation $s \leq w_n$. Cela implique $B_{i,n} \geq 0$, alors chaque participant a droit à une part de l'excédent ou du moins, il n'aura pas besoin de payer une contribution supplémentaire à la fin de la période.

Les variables aléatoires $B_{i,n}$ suivent des distributions augmentées de zéro, c'est-à-dire qu'il y a une probabilité non nulle que certains participants n'aient pas de part d'excédent ($P[B_{i,n} = 0] = P[S_n \geq w_n]$).

3.5 Conclusion

Dès l'adhésion au pool, il est essentiel, par souci de transparence, d'expliquer aux membres la règle de répartition des coûts totaux du pool. Nous venons de présenter deux approches : dans la première, sans partenariat, le pool gère toutes les fonctions d'assurance, et les contributions individuelles, en fin de période, sont calculées en fonction du montant attendu des réclamations qu'un participant pourrait soumettre, tout en prenant en compte le montant total des réclamations. Dans la seconde approche, les contributions sont connues en début de période et le pool achète un contrat de protection (stop-loss) auprès d'un partenaire, et comme le (ré)assureur facture des frais supplémentaires en plus de la prime stop-loss, le système pourrait ne pas être équitable dans son ensemble. Pour garantir l'équité, ces frais supplémentaires ne sont pas pris en compte. On peut donc décomposer l'opération en :

- Un paiement fixe en début de période : $\pi_{i,n}^{P2P} = w_{i,n}$
- Un remboursement en espèces aléatoires : $B_{i,n} = (w_{i,n} - h_{i,n}^*(S_n))_+$

- Une contribution fixe $E[(h_{i,n}^*(S_n) - w_{i,n})_+]$ à la tranche supérieure tout en excluant les chargements
- En échange d'un remboursement de la réclamation Y_i

En moyenne, on a

$$E\left[Y_i - w_{i,n} + \left(w_{i,n} - h_{i,n}^*(S_n)\right) - E\left[h_{i,n}^*(S_n) - w_{i,n}\right]\right] = E\left[Y_i - h_{i,n}^*(S_n)\right] = 0$$

En excluant les chargements, le système d'assurance P2P est donc considéré comme équitable, car en moyenne, la contribution qu'un participant verse est égale au remboursement de la réclamation soumise au pool.

Chapitre 4

Étude de cas

Dès leur adhésion au pool, les participants doivent être préalablement informés des paramètres essentiels, tels que la taille du pool, la franchise, la quote-part et le montant maximal de remboursement annuel, en plus de la règle de partage. Dans cette section, nous mettons en application la méthode de partage présentée dans les chapitres précédents. À cet effet, nous considérons initialement un pool composé de $n = 800$ participants qui se mettent ensemble dans le but de faire face aux coûts énormes de soins vétérinaires de leurs animaux de compagnie. Par la suite, nous subdivisons ce groupe en trois profils de risque distincts. Après avoir déterminé les contributions respectives de chaque participant, à la fois dans un système sans réassurance et dans un système avec réassurance, nous analysons l'évolution de ces contributions en fonction de la taille du pool, de la franchise et du montant maximal de remboursement annuel.

4.1 Profil de risque de chaque sous groupe

Le pool est constitué de propriétaires de chiens ayant les caractéristiques décrites à la section 2.2.1. On peut y retrouver les profils suivants :

Sous-groupe 1 : Ce groupe est constitué des chiens qui ont un premier antécédent médical ; âgés de 12 mois ; vivant en région, de poids $45kg$, de race St. Bernards (Classe 2), qui a un plan de couverture B. Il est considéré comme le moins risqué et représente 50% du nombre total de participants.

Sous-groupe 2 : Ce groupe est constitué des chiens qui ont un premier antécédent médical ; âgés de 60 mois ; vivant en région, ayant un poids de $60kg$ de race Bulldog (Classe 1), qui a un plan de couverture A. Il représente 30% des participants.

Sous-Groupe 3 : Ce groupe est constitué de chiens (pas de la classe 1, ni de classe 2) qui ne sont pas à leur premier antécédent médical, âgé de 120 mois et ne vivant pas en région, de masse 20 kg, ayant une couverture basique complète. Ce groupe est considéré comme présentant le plus risqué, il représente 20% du nombre total de participants.

	Groupe 1	Groupe 2	Groupe 3
Taille	$n_1 = 50\%n$	$n_2 = 30\%n$	$n_3 = 20\%n$
λ_i	0.682541	0.929849	4.28897
$E[\tilde{C}_{ik}]$	276.1588	303.07975	73.27401
σ_{ik}	381.8003	419.0196	101.3042
$E[X_i]$	188.48970	281.81840	314.27003

TABLE 4.1.1 – Profil de risques des participants d'un pool de taille $n = 800$

4.2 Distribution du montant total de réclamations à la charge du pool

Afin de calculer les contributions individuelles de chaque participant, il est impératif de définir la distribution du montant total des réclamations à la charge du pool. Pour ce faire, il est préalablement nécessaire de déterminer la distribution des réclamations émanant de chaque participant.

4.2.1 Distribution de la fréquence et du coût de réclamation

Les figures 4.2.1 et 4.2.2 illustrent respectivement les masses de probabilité de la fréquence de réclamations de chaque participant du pool ainsi que les fonctions de répartition des coûts de sinistre. On peut constater que les groupes 1 et 2 présentent vraisemblablement une probabilité élevée de soumettre moins de deux réclamations comparativement au groupe 3. Afin de déterminer la distribution du montant total des réclamations de chaque membre, il est indispensable de procéder à la discrétisation de la distribution des coûts de réclamation de chaque participant en utilisant l'approche numérique définie à la section 2.3.

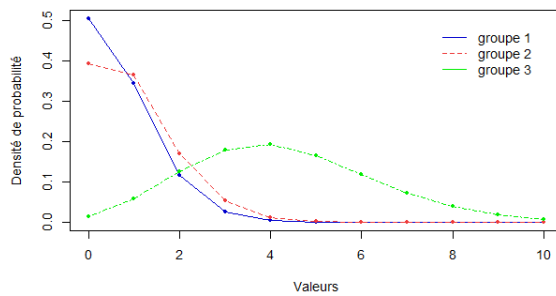


FIGURE 4.2.1 – Densité de probabilité du nombre moyen de sinistres par an

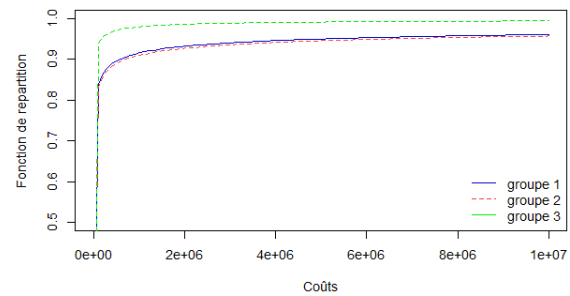


FIGURE 4.2.2 – Fonction de répartition du coût de sinistre pour chaque groupe

Partant de la distribution du coût de soins vétérinaires consécutifs à une visite médicale d'un animal qui est une Log-Normale, nous utilisons une approche numérique décrite par l'équation 2.3.6 permettant de procéder à la discrétisation de cette variable. Cette méthode de discrétisation nous garantit que la moyenne de la distribution discrète et continue est équivalente.

4.2.2 Distribution du Coût total de sinistre par participant

Après avoir obtenu la distribution des coûts des soins vétérinaires, nous avons mis en œuvre l'algorithme de Panjer défini par l'équation 2.3.7 en utilisant la fonction `AggregateDist` du package `Actuar` (GOULET 2008). Cela nous permet ainsi d'obtenir la masse de probabilité du coût total des réclamations par individu dans chaque groupe (Figure 4.2.3).

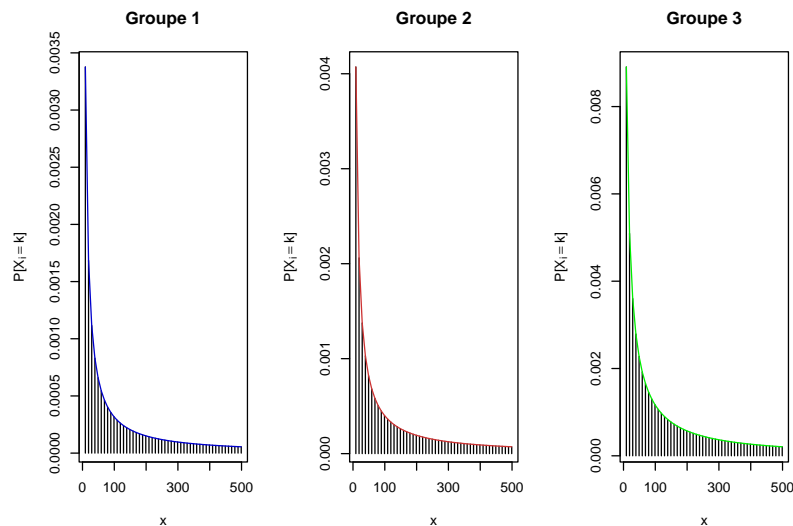


FIGURE 4.2.3 – Masse de probabilité du coût total par individu de chaque groupe

4.2.3 Distribution du montant total des réclamations par participant

Dans le cadre de notre étude, une fois que le coût total de soins vétérinaires est connu, nous appliquons une franchise et un plafond pour obtenir la variable représentant le montant total des réclamations par participant, à la charge du pool. Cette variable est définie par l'équation 2.3.8. Pour notre étude, nous optons pour une franchise de 250, approximativement égale au coût moyen des réclamations par participant dans le pool. Quant au montant maximal de remboursement annuel¹, nous fixons un plafond de 8500 et une quote-part à 80%.

Pour déterminer la distribution du montant total des réclamations à la charge du pool, deux approches peuvent être utilisées. Soit le montant total à la charge du groupe, noté S_n , est obtenu en utilisant l'approximation collective décrite par l'équation 2.4.3, soit nous procédons par un calcul direct présenté à la section 2.4, et dont la distribution est définie par l'équation 2.4.2. On note :

$$S_1 = Y_1 + Y_2 + \dots + Y_{n_1} \text{ le coût total des réclamations soumis par le groupe 1}$$

$$S_2 = Y_1 + Y_2 + \dots + Y_{n_2} \text{ celui du groupe 2}$$

$$S_3 = Y_1 + Y_2 + \dots + Y_{n_3} \text{ celui du groupe 3}$$

1. <https://www.hellosafe.be/assurance-animaux>

En utilisant le calcul direct, pour tout i , $Y_i > 0$, les variables aléatoires S_1, S_2 et S_3 ont respectivement une distribution Binomiale composée.

$$S_1 =_d \sum_{k=1}^{N_1} Z_k^{(1)} \text{ avec } N_1 \sim \text{Binomial}(n_1, q_1)$$

$$S_2 =_d \sum_{k=1}^{N_2} Z_k^{(2)} \text{ avec } N_2 \sim \text{Binomial}(n_2, q_2)$$

$$S_3 =_d \sum_{k=1}^{N_3} Z_k^{(3)} \text{ avec } N_3 \sim \text{Binomial}(n_3, q_3)$$

Pour tout i , $q_i = P[Y_i > 0] = 1 - P[Y_i = 0]$ et $S_n = S_1 + S_2 + S_3$. La comparaison entre le calcul direct et l'approximation collective, comme illustrée par la figure 4.2.4, montre que l'approximation collective surestime les probabilités de réclamations aux niveaux de la queue de la distribution tout en les sous-estimant autour de la moyenne. De ce fait, pour déterminer les contributions de chaque participant de manière prudente, nous poursuivons nos calculs en utilisant la distribution obtenue par le modèle collectif.

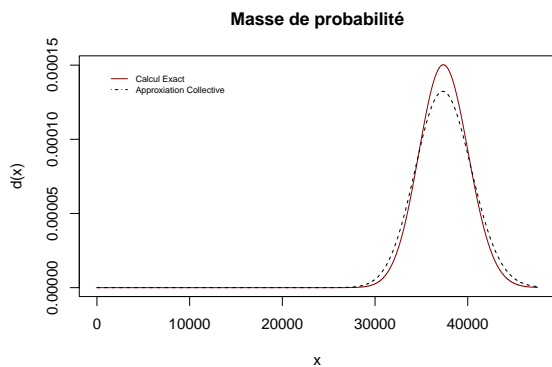


FIGURE 4.2.4 – Calcul direct Vs Approximation collective

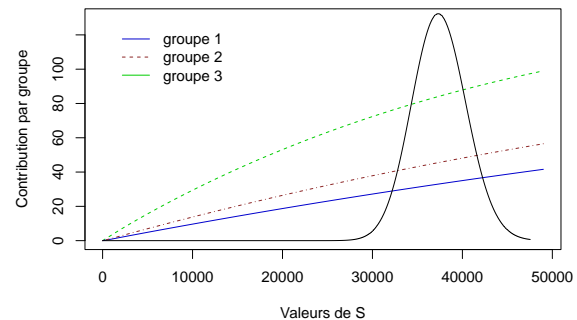


FIGURE 4.2.5 – Contributions individuelles de chaque groupe

4.3 Contributions individuelles : Système sans (ré)-assureur

Connaissant la distribution de la réclamation totale au sein du groupe, nous pouvons désormais déterminer la contribution individuelle de chaque participant en utilisant le résultat 3.4.1. En observant la figure 4.2.5, il est possible d'observer que les contributions de chaque groupe augmentent avec les réalisations s de S_n , ainsi les contributions sont des fonctions non décroissantes des réalisations de S_n . En se basant sur ces observations, nous pouvons conclure que la règle **CMRS** est une règle de partage Pareto-optimal. Elle permet donc d'obtenir une répartition équitable et attractive du montant total des réclamations entre les membres du

pool. Par exemple, les contributions individuelles pour une charge de 35000 sont :

$$E[Y_i|S_n = 35000] = \begin{cases} 31.174475 & \text{pour } i \in \text{Groupe 1} \\ 43.135172 & \text{pour } i \in \text{Groupe 2} \\ 80.550068 & \text{pour } i \in \text{Groupe 3} \end{cases}$$

Une fois que les contributions sont connues, il devient intéressant d'analyser la sensibilité de la règle **CMRS** à divers paramètres, tels que le nombre de participants, la franchise et le montant maximal des réclamations par an. Cette étude a été réalisée à l'aide du code présenté à l'annexe A.4, nous offrant une meilleure compréhension de la manière dont la règle **CMRS** réagit face à ces différents paramètres. Cette analyse nous permet ainsi de choisir une meilleure combinaison de ses paramètres lors de la mise en place d'un système optimal.

4.3.1 Sensibilité de la **CMRS** à la taille du pool

Il est à noter qu'au sein d'un pool de taille réduite, les contributions présentent une grande variabilité, comme observé sur la figure 4.3.1. Cela peut s'expliquer par le fait que chaque réclamation, même petite, a un impact plus important sur la contribution moyenne de chaque participant dans un pool réduit. De ce fait, une seule réclamation importante pourrait entraîner une variabilité importante. La taille du pool limite la diversification, tout comme dans les assurances traditionnelles. Il est également à noter que la probabilité d'obtenir des valeurs extrêmes décroît à mesure que la taille du pool augmente. Cette évolution se manifeste par une distribution de probabilité de plus en plus recentrée et symétrique autour de la moyenne, ce qui peut être avantageux pour une assurance P2P. Une observation similaire peut être faite en analysant la figure 4.3.2.

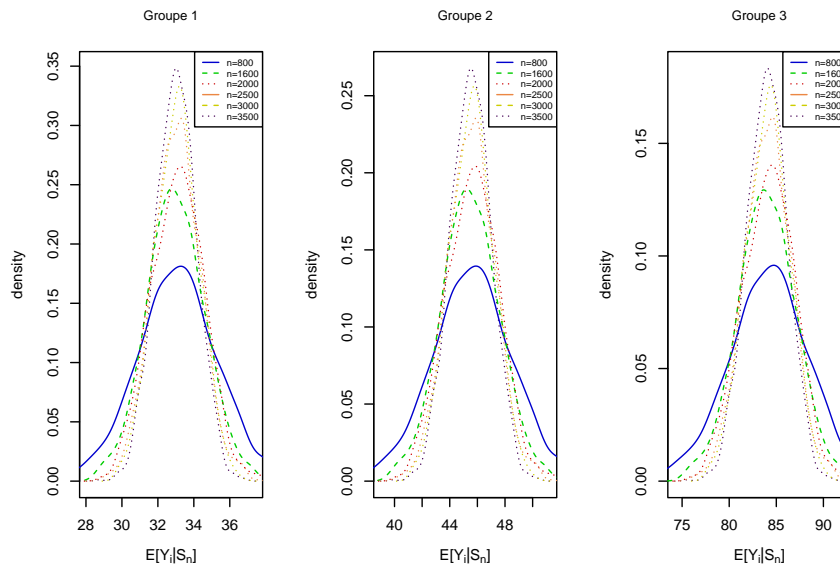


FIGURE 4.3.1 – Variabilité de $h_{i,n}^*(S_n)$ en fonction de la taille du pool

À mesure que la taille du pool augmente, les contributions individuelles ont tendance à se stabiliser et à converger vers la prime pure. Cette convergence suggère qu'il est préférable

d'avoir un pool assez grand pour garantir l'efficacité et l'attrait du système, car cela offre une plus grande prévisibilité et une meilleure gestion des risques dans l'ensemble. Cependant, il est important de trouver un équilibre entre la diversification du risque, qui augmente avec la taille, et la gestion efficace du groupe, afin d'assurer la stabilité et la durabilité du système.

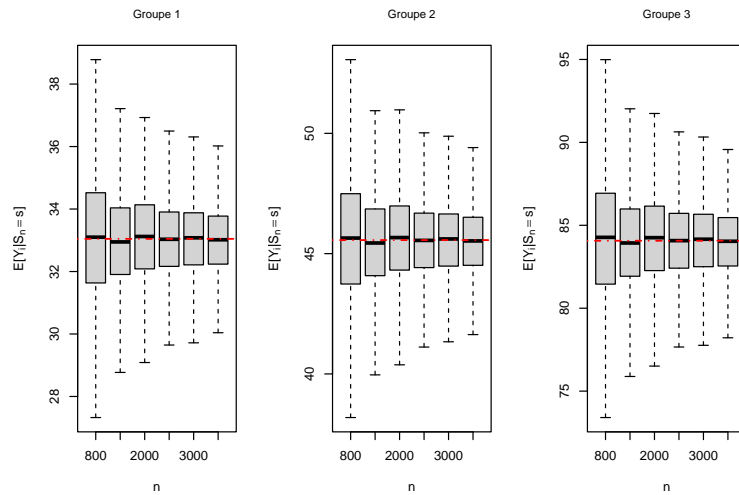


FIGURE 4.3.2 – Convergence des cotisations pour chaque groupe

4.3.2 Sensibilité de la *CMRS* à la franchise

La franchise en assurance joue un rôle mécanique sur les coûts de sinistres, en ce sens qu'en cas de réclamations, elle influence directement les paiements et la tarification. Une franchise élevée implique une prime faible, et vice versa. Notons que le système est conçu de sorte que le même niveau de franchise est défini pour tous les participants du pool. D'une part, en

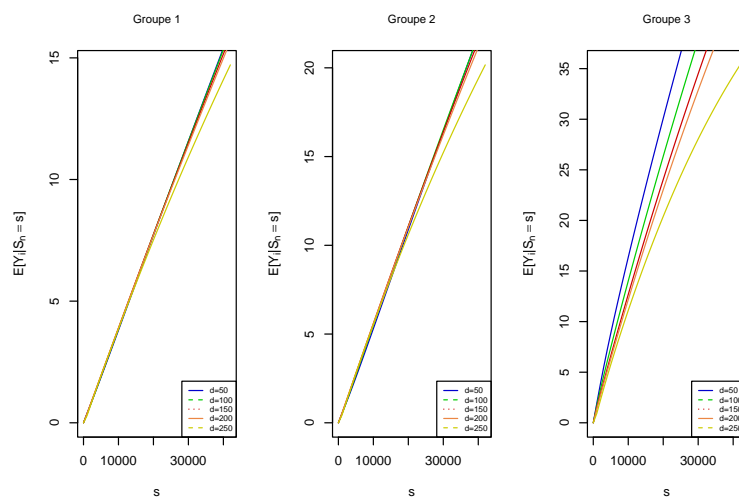


FIGURE 4.3.3 – Influence du niveau de franchise sur la variabilité des contributions

observant la figure 4.3.3, on constate que lorsque la franchise devient de plus en plus élevée, les contributions ont tendance à nettement diminuer pour les participants du troisième groupe. La variation du niveau de la franchise semble avoir un effet significatif sur les contributions dans ce groupe. Dans le groupe 1, cet effet de la variation de la franchise est très peu perceptible sur le niveau des contributions. Bien que les contributions soient plus élevées dans le groupe 2 que dans le groupe 1, l'effet de la franchise est très peu perceptible dans ce groupe.

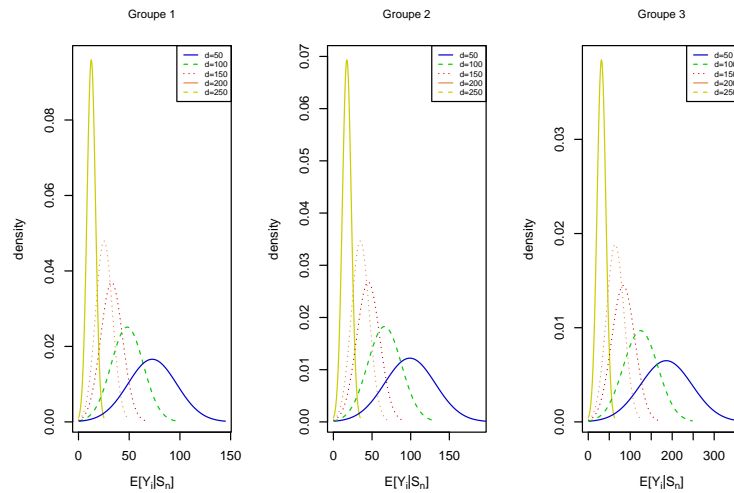


FIGURE 4.3.4 – Évolution des contributions en fonction de la franchise

D'autre part, sur la figure 4.3.4, on constate que pour les trois groupes, la figure montre une distribution normale des contributions de plus en plus effilée et centrée autour de la moyenne lorsque le niveau de franchise devient de plus en plus élevé. Cela pourrait indiquer que, malgré la variation de la franchise, la majorité des contributions des membres se situent autour d'une valeur moyenne.

À l'opposé, la distribution devient plus large lorsque la franchise est plus faible. Cela suggère une variabilité importante des réclamations, et de ce fait, les contributions sont plus élevées pour tenir compte de ce risque.

4.3.3 Sensibilité de la *CMRS* au remboursement annuel

Le niveau maximal de remboursement annuel n'a pratiquement pas d'effet sur la variabilité des cotisations, comme l'illustre la figure 4.3.5. Ceci pourrait s'expliquer par le fait que le risque pris par le pool est plafonné, et tout montant au-dessus de ce plafond retourne à la charge du participant. La stabilité de la variabilité pourrait s'expliquer par le fait que le plafonnement des réclamations permet à la plateforme de maintenir un équilibre global.

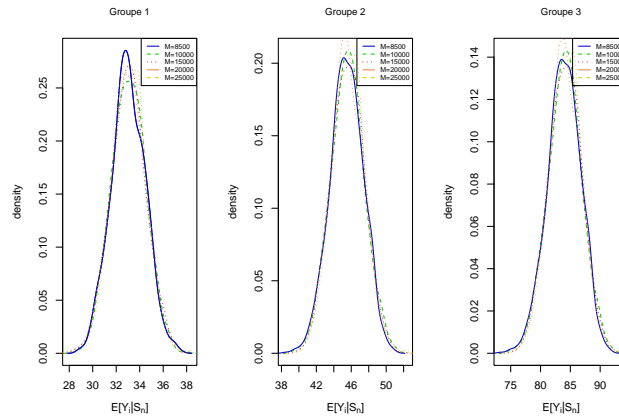


FIGURE 4.3.5 – Influence des montants de réclamation maximale sur les contributions

Toutefois, on constate également que lorsque le niveau de remboursement maximal par an augmente, les cotisations ont tendance à augmenter également. Cela pourrait s'expliquer par le fait que lorsque le niveau de remboursement maximal par an augmente au sein du pool, les contributions individuelles au sein de différents groupes augmentent afin d'assurer la stabilité financière et la capacité du pool à faire face à des dépenses potentiellement plus importantes.

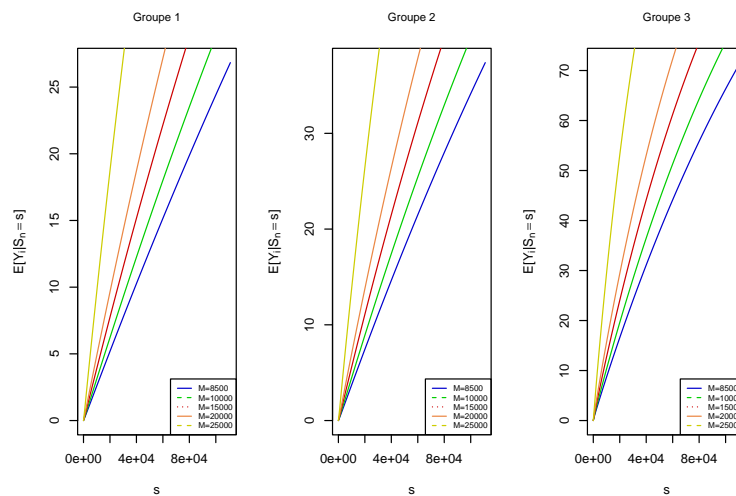


FIGURE 4.3.6 – Évolution des contributions en fonction du plafond de réclamation annuel

En résumé, les observations faites mettent en lumière la complexité du comportement des membres face à des variations de paramètres tels que le nombre de participants, la franchise et le montant maximal de réclamation. Bien définir les paramètres devient une étape cruciale dans la mise sur pied d'un système d'assurance P2P afin de garantir l'équité, l'attractivité, la stabilité, la confiance des membres et la capacité à répondre aux évolutions des besoins de ses membres.

4.4 Contributions individuelles : Système avec (ré)-assureur

Avec un pool de 800 participants, l'objectif est de mettre sur pied un mécanisme visant à accroître l'attractivité du pool en transformant les contributions ex-post aléatoires par des contributions ex-ante fixes. Pour cela, on souhaite distribuer un surplus dans 85% des cas. La charge totale maximale que le pool devra supporter pour atteindre cet objectif est calculée en utilisant la relation 3.4.5

$$F_{S_n}^{-1}(0.85) = 40568$$

Ainsi, avec une probabilité de 0.85, la charge à supporter par la tranche inférieure est de 40568, et la contribution de chaque participant de la tranche inférieure à cette charge est de :

$$E[Y_i|S_n = 40568] = \begin{cases} 35.45397 & \text{pour } i \in \text{Groupe 1} \\ 48.69435 & \text{pour } i \in \text{Groupe 2} \\ 88.63701 & \text{pour } i \in \text{Groupe 3} \end{cases}$$

Cependant, si la charge totale maximale du pool n'atteint pas la somme prédéterminée, le pool redistribuera ce surplus, que ce soit à une œuvre caritative ou à ses membres. En revanche, si la charge totale maximale du pool dépasse cette somme prédéterminée, le pool aura recours à un (ré)-assureur. Dans ce cas, les participants contribueront également pour cette couverture stop-loss. Supposons que des frais de chargement s'élevant à $\theta_{SL} = 0.5$. Alors, les résultats des contributions de chaque membre se résument dans le tableau 4.4.1.

Pool	Age(mois)	N_i	w_i	$E[Y_i]$	π_i^{SL}	π_i	$E[B_i]$
Groupe 1	12	400	35.45397	33.04540	0.2720516	35.72602	2.632828
Groupe 2	60	240	48.69435	45.56313	0.3486331	49.04298	3.419542
Groupe 3	120	160	88.63701	84.06873	0.4818218	89.11883	4.971851

TABLE 4.4.1 – Résultat de la contribution de chaque participant pour chaque tranche

Au regard de nos résultats, il apparaît que chaque participant contribue en moyenne à la hauteur de la charge de réclamation qu'il soumet au pool. De plus, avec une probabilité de 85%, chaque participant reçoit un bonus suivant son profil de risque, ce qui rend le pool plus attractif. Le transfert de la tranche supérieure à un (ré)-assureur a pour effet de réduire la volatilité des contributions, comme le met en évidence la figure 4.4.2.

La prise en charge de la tranche supérieure par un (ré)-assureur contribue de ce fait à stabiliser les contributions individuelles et à limiter les variations extrêmes dans la tranche inférieure, comme on peut le voir sur la figure 4.4.1. Ce qui aide à maintenir un équilibre financier et pourrait accroître l'attractivité du pool, car l'introduction du (ré)-assureur permet aussi une protection renforcée contre des scénarios financiers difficile.

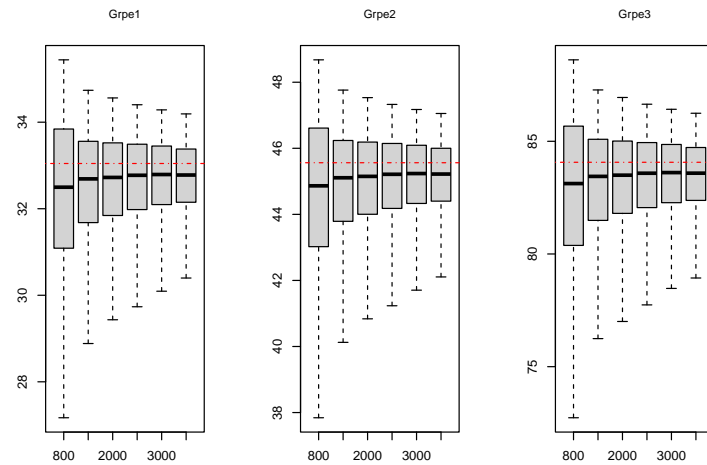


FIGURE 4.4.1 – Répartition des contributions dans la tranche inférieur en présence d’un (ré)-assureur

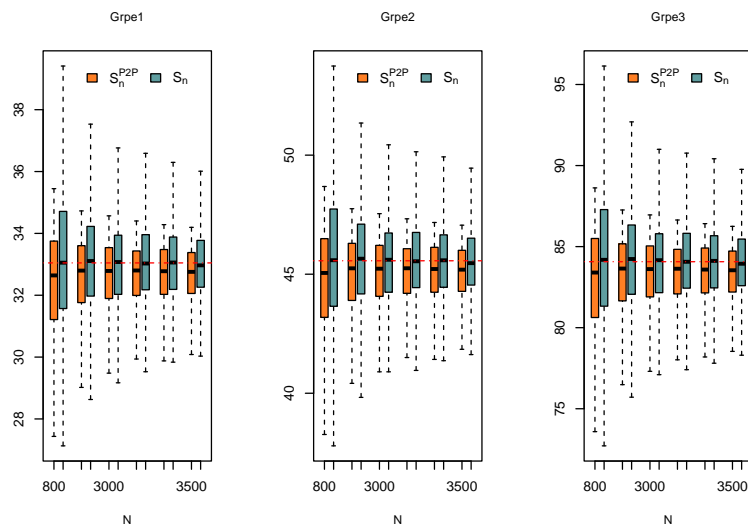


FIGURE 4.4.2 – Effet sur les contributions du transfert de la tranche supérieure des réclamations à un (ré)-assureur

4.5 Conclusion

Cette stratégie de transfert de risque a donc un impact significatif sur la stabilité financière du pool, en limitant les fluctuations extrêmes des contributions des participants, elle favorise l'attractivité du pool. Ces mesures peuvent être cruciales pour la durabilité du système et la préservation des principes clés tels que l'équité et la confiance au sein du pool.

Conclusion Générale

L'émergence de l'économie collaborative a favorisé un esprit communautaire, incitant ainsi les individus à collaborer dans la recherche de solutions à leurs problèmes et au partage des ressources. Cela s'est naturellement étendu au domaine de l'assurance en général, et en particulier à l'assurance pour soins vétérinaires. L'avènement des technologies numériques et de l'intelligence artificielle a participé à la mise en place de nouveaux mécanismes de couverture des risques. Dans ce mémoire, nous avons présenté de manière détaillée le fonctionnement de la plateforme d'assurance pour chiens et chats, Eusoh. À partir d'une présentation globale des différents modèles d'assurance P2P, de leurs particularités et de leurs différences, nous avons ainsi mis en lumière la position d'Eusoh dans ce paysage.

La présentation détaillée de cette plateforme nous a fourni les bases nécessaires pour une modélisation actuarielle que nous avons réalisée par la suite. Centrée sur la distribution du coût et du nombre de visites chez le vétérinaire par membre, cette distribution a été agrégée afin d'obtenir la distribution du coût total par membre. Cela nous a permis d'aboutir à la distribution du coût total des réclamations à la charge du pool. Les résultats obtenus ont été la clé pour la présentation d'une méthode novatrice permettant ainsi de partager de façon équitable le montant total de réclamation au sein du pool.

Ensuite, l'utilisation de la règle de partage *Conditional Risk Mean Sharing* permet de réaliser une redistribution de la charge totale du pool tout en considérant les niveaux de réclamations individuelles. Une étude de sensibilité sur des paramètres tels que le nombre de participants, la franchise et le montant maximal de réclamation, nous a permis de déterminer les paramètres susceptibles d'affecter la stabilité du pool. De manière complémentaire, une méthode alternative impliquant l'association d'un (ré)-assureur à la plateforme a été présentée, offrant ainsi une solution pour gérer toute charge au-dessus de la capacité du pool. En combinant ces deux approches, les résultats suggèrent des possibilités pour renforcer la stabilité financière du pool tout en continuant à conserver des aspects attractifs, notamment de l'équité, la transparence et la communication entre les membres.

Toutefois, il est crucial de noter qu'en tant que concept nouveau, l'assurance collaborative, bien qu'offrant des solutions novatrices et plus adaptées aux besoins changeants des participants, rencontre des défis à surmonter pour atteindre un succès durable. Le premier défi majeur concerne la réglementation, car le secteur de l'assurance est soumis à des normes strictes pour protéger les assurés, garantir la stabilité financière et prévenir les abus. Bien que la plateforme Eusoh ait connu un certain succès, elle a dû faire face à ce défi, entraînant sa fermeture quatre ans après son lancement en 2018. Notons également que la croissance rapide de la communauté

pourrait poser des défis en termes de gestion des risques, de gestion des données et de capacité à répondre à une augmentation du nombre de réclamations. Une autre inquiétude majeure réside dans la gestion de la collectivité et la garantie d'une répartition équitable des coûts afin d'assurer la stabilité du pool et la pérennité du système.

L'utilisation des technologies modernes est l'un des principaux aspects qui distinguent les assurances collaboratives des modèles d'assurance traditionnels. L'utilisation des techniques de modélisation prédictive avancée, l'utilisation de l'apprentissage automatique et de l'intelligence artificielle pourrait être envisagés pour identifier avec précision les variables, repérer les anomalies. Des technologies comme la blockchain pourraient aider à garantir la transparence et la sécurité des transactions, ainsi que pour faciliter la gestion des contrats intelligents. Il serait donc essentiel que les assurances P2P mettent en place des mesures de cybersécurité robustes pour protéger les données sensibles, prévenir les cyberattaques et assurer la continuité des opérations. Une gestion pro-active des risques cybers est nécessaire pour garantir la sécurité et la fiabilité du système, renforçant ainsi la confiance des membres.

En somme, ce travail offre une contribution significative à la compréhension et à l'amélioration des modèles actuariels appliqués aux plateformes d'assurance P2P pour soins vétérinaires telles qu'Eusoh. Les défis identifiés et les solutions proposées fournissent une base solide pour des recherches futures et soulignent l'importance continue de l'innovation dans le domaine de l'assurance collaborative.

Bibliographie

- DENUIT, Michel (2020). « Investing in your own and peers' risks: the simple analytics of P2P insurance ». In : *Eur. Actuar. J.* 2, p. 335-359.
- DENUIT, Michel et Jan DHAENE (sept. 2012). « Convex order and comonotonic conditional mean risk sharing ». In : *Insurance: Mathematics and Economics* 2, p. 265-270.
- DENUIT, Michel, Jan DHAENE et Christian Y. ROBERT (sept. 2022). « Risk-sharing rules and their properties, with applications to peer-to-peer insurance ». In : *J of Risk & Insurance* 3, p. 615-667.
- DENUIT, Michel et Christian Y. ROBERT (2021a). « From risk sharing to pure premium for a large number of heterogeneous losses ». In : *Insurance: Mathematics and Economics*.
- (juin 2021b). « Risk sharing under the dominant peer-to-peer property and casualty insurance business models ». In : *Risk Manage Insurance Review* 2, p. 181-205.
- DHAENE J, Linders D (2022). « Foundations of quantitative risk measurement ».
- FENG, Runhuan et Mao LI (2023). « Distributed Insurance: Tokenization of Risk and Reward Allocation ». In : *Department of Mathematics, University of Illinois at Urbana-Champaign*.
- GOULET, Vincent (2008). « **actuar** : An R Package for Actuarial Science ». In : *Journal of Statistical Software* 25.7.
- PAI, Jeffrey S., Kevin J. SHAND et Xikui WANG (oct. 2006). « Compound Poisson Model with Covariates: A Case Study on Pet Insurance ». In : *North American Actuarial Journal* 4, p. 219-234.
- TURCOTTE, Michel (2017). « L'assurance sans assureur ou le P2P ». In : *Assurances et gestion des risques* 1-2, p. 77.

Annexe A

PROGRAMMATION R

A.1 Analyse des covariables

Cette section présente l'application de l'illustration 1 de la page 17 de notre mémoire. Les valeurs des paramètres σ ; α ; μ ; β et θ sont obtenues de l'article de Pai et al (oct. 2006). « Compound Poisson Model with Covariates : A Case Study on Pet Insurance, p227, p229 »

```
#Choix des paramètres
sigma=1.03375
alpha=2.67253
mu=4.90284
q=0.8#quote part

#Résultat obtenu de l'étude menée par de Jeffrey S Pai et al. (2013)
beta_N=c(1.23070,0.00026 ,0.02957,0.04708,0.11882,-0.00058,-0.24512,
        -0.16021,0.42197)#vecteur de paramètres de beta

theta_C=c(-0.10522 ,0.00227,0.12641,0.08862,0.14922,0.00355,0.02991,
          0.02971,-0.90829)#Vecteur de paramètres theta
```

Nous définissons ici la fonction *param* qui prend en argument le vecteur des variables *covar* et l'*âge* (en mois) de l'animal. Connaissant les caractéristiques de chaque animal, il s'ensuit :

```
#Caractéristique d'un animal
param=function(covar,a){
  val1 = a*12
  #Sévérité et Fréquence
  val2=12*exp(-alpha+(t(beta_N)%*%covar))
  val3=q*exp(mu+(t(theta_C)%*%covar)+(sigma^2)/2)
  val4= q*sqrt((exp(sigma^2)-1)*exp(2*(mu+(t(theta_C)%*%covar))+(sigma^2)))
  return(list(Age=val1, lambda=val2, E_C_ik=val3, Ecart_type=val4))}
```

A.1.1 Influence de l'âge sur la fréquence de sinistre sur la période de couverture

Considérons tout d'abord un groupe de participants ayant des chiens dont les caractéristiques les suivantes : antécédents médicaux existants ; de la classe 1 ; vivant en zone urbaine ; qui a un poids de 10 kg ; ayant souscrit au plan B. $Z(1, \text{Age}, 1, 0, 1, 10, 1, 0, 0)$

```
#différents d'animaux de l'étude
Age=seq(0,10000,1)
groupe=data.frame(Age=Age, lambda=rep(0,length(Age)),
E_C_ik=rep(0,length(Age)),
                  Ecart_type=rep(0,length(Age)))
for (i in 1:length(Age)) {
  #Nous considérons l'age ici en mois
  p=param(c(1, Age[i], 1, 0, 1, 10, 1, 0, 0), Age[i])
  groupe$Age[i]=p$Age
  groupe$lambda[i]=p$lambda
  groupe$E_C_ik[i]=p$E_C_ik
  groupe$Ecart_type[i]=p$Ecart_type
  rm(p)
}
# View(groupe)
plot(groupe$Age,groupe$lambda,type="l",pch=19,ltty=1,
      ylab=TeX(r"(\lambda_{i})"),
      xlab="Age")
```

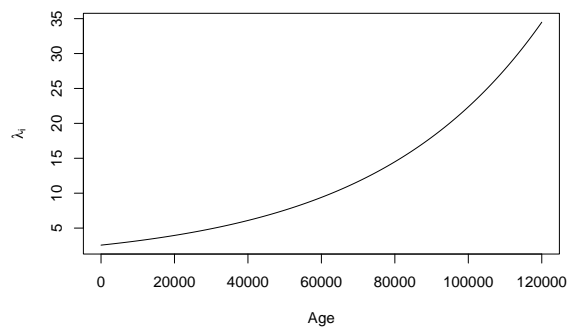


FIGURE A.1.1 – Évolution de la fréquence de réclamation en fonction de l'âge

Sur une échelle très grande, On peut constater que le nombre de réclamations sur la période de couverture augmente de manière exponentielle, ce qui suggère une croissance rapide et non linéaire du nombre de réclamations à mesure que l'âge augmente.

```
#Distribution de la sévérité
```

```
# Nombre de réclamations allant de 0 à 10 au cours de la période de couverture
u=seq(0,10,1)
```

```

# Probabilité de déclarer au moins un sinistre
F1=dpois(u, groupe$lambda[2])#Nombre de sinistre pour un chien âgé de 12mois
F2=dpois(u, groupe$lambda[6])#Nombre de sinistre pour un chien âgé de 60mois
F3=dpois(u, groupe$lambda[16])#Nombre de sinistre pour un chien âgé de 180mois

Resultal.Age=data.frame(Nbre.Sinistres=u,densite.12=round(F1,5),
densite.60=round(F2,5),
densite.120=round(F3,5))
print(Resultal.Age)

```

On constate que les probabilités de soumettre au moins une réclamation présentent des ordres de grandeurs similaires au sein de chaque groupe d'âge.

A.1.2 Influence du poids sur la période de couverture

Considérons tout d'abord un groupe de participants ayant des chiens âgés de 60 mois (5 ans) dont les caractéristiques sont par exemple les suivantes : antécédents médicaux existants ; de la classe 1 ; vivant en zone urbaine ; qui a un poids qui varie ; ayant souscrit au plan B. $Z(1, 60, 1, 0, 1, poids, 1, 0, 0)$

```

#différents d'animaux de l'étude
poids=seq(0,6000,1)
groupe1=data.frame(poids=poids, lambda=rep(0,length(poids)),
E_C_ik=rep(0,length(poids)),
                Ecart_type=rep(0,length(poids)))
for (i in 1:length(poids)) {
  p=param(c(1,10,1,0,1,poids[i],1,0,0),5*12)
  groupe1$lambda[i]=p$lambda
  groupe1$E_C_ik[i]=p$E_C_ik
  groupe1$Ecart_type[i]=p$Ecart_type
  rm(p)
}
# View(groupe1)
plot(poids,groupe1$lambda,type="l",pch=20,lty=4,ylab=TeX(r"($\lambda_{i}$"))

```

Sur une échelle très grande, on peut constater que le nombre de réclamations sur la période de couverture diminue de manière exponentielle, ce qui suggère une décroissance rapide et non linéaire du nombre de sinistres à mesure que le poids augmente.

```

#Distribution de la sévérité
# Nombre de réclamations allant de 0 à 10 au cours de la période de couverture
u=seq(0,10,1)
# Probabilité de déclarer au moins une sinistre
f1=dpois(u, groupe1$lambda[5])#Nombre de sinistre pour un chien de 5kg
f2=dpois(u, groupe1$lambda[20])#Nombre de sinistre pour un chien de 20kg
f3=dpois(u, groupe1$lambda[35])#Nombre de sinistre pour un chien de 35kg

```

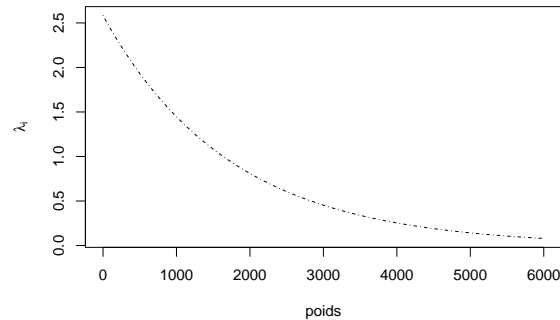


FIGURE A.1.2 – Évolution de la fréquence de réclamation en fonction du poids

```
Resultat.poids=data.frame(Nbre.Sinistres=u,densite.5=round(f1,5),
densite.20=round(f2,5),densite.35=round(f3,5))
print(Resultat.poids)
```

A.1.3 Influence des plans de couverture B et C sur la sévérité des réclamations

À présent, analysons l'effet des plans de couverture B et C sur la sévérité des réclamations mise en évidence dans l'illustration 2 de la page 19 du mémoire.

Considérons tout abord un groupe de participants ayant des chiens âgés de 60 mois (5 ans) dont les caractéristiques sont par exemple les suivantes : antécédents médicaux existants ; de la classe 1 ; vivant en zone urbaine ; qui a un poids qui varie ; ayant souscrit au plan B ou C. $Z(1, 60, 1, 0, 1, 10, Couverture, 0, 0)$

```
planB=seq(0,1)
plan1=data.frame(planB=c("NON", "OUI"), E_C_ik=rep(0,length(planB)),
Ecart_type=rep(0,length(planB)))
for (i in 1:length(planB)) {
p=param(c(1,10,1,0,1,10,planB[i],0,0),10*12)
plan1$E_C_ik[i]=p$E_C_ik
plan1$Ecart_type[i]=p$Ecart_type
rm(p)}
print(plan1)
```

plan B	$E[C_{ik}]$	$\sigma_{C_{ik}}$
NON	231.0481	319.4331
OUI	238.0632	329.1316

TABLE A.1.1 – Paramètres de la sévérité.

```
planC=seq(0,1)
plan2=data.frame(planC=c("NON", "OUI"),
```

```

      E_C_ik=rep(0,length(planC)),
      Ecart_type=rep(0,length(planC)))
for (i in 1:length(planC)) {
p=param(c(1,10,1,0,1,10,0,planC[i],0),10*12)
  plan2$E_C_ik[i]=p$E_C_ik
  plan2$Ecart_type[i]=p$Ecart_type
  rm(p)}

print(plan2)

```

plan C	$E[C_{ik}]$	$\sigma_{C_{ik}}$
NON	231.0481	319.4331
OUI	238.0155	329.0658

TABLE A.1.2 – Paramètres de la sévérité

Le coût de sinistre par animal suit une loi log-normale de paramètre $\log(E_{C_{ik}}[i])$ et $Ecart - type[i]$

```

# Créer un vecteur de valeurs pour tracer la courbe
val = seq(1000, 10000, by=1000)
# Calculer de la densité de probabilité pour chaque valeur de x
# pour le groupe ayant souscrit au plan de couverture B
dB_1 = dlnorm(val, meanlog = log(plan1$E_C_ik[1]),
sdlog = log(plan1$Ecart_type[1]))
# pour le groupe n'ayant pas souscrit au plan de couverture B
dB_2 = dlnorm(val, meanlog = log(plan1$E_C_ik[2]),
sdlog = log(plan1$Ecart_type[2]))
par(mfrow=c(1,2))
table.B=data.frame(valeur=val,densité.1=dB_1,densité.2=dB_2)
print(table.B)

```

Coût	densité 1	densité 2
2000	3.225119 e-05	3.216923 e-05
5000	1.200314 e-05	1.199131 e-05
7000	8.296916 e-06	8.294018 e-06
9000	6.282955 e-06	6.283895 e-06
10000	5.588490 e-06	5.590527 e-06

TABLE A.1.3 – Probabilité d'avoir des coûts de sinistres positifs

```

# Créer un vecteur de valeurs pour tracer la courbe
val = seq(1000, 10000, by=1000)
# Calculer de la densité de probabilité pour chaque valeur de x
# pour le groupe ayant souscrit au plan de couverture B
dC_1 = dlnorm(val, meanlog = log(plan2$E_C_ik[1]),
sdlog = log(plan2$Ecart_type[1]))

```



```
# pour le groupe n'ayant pas souscrit au plan de couverture B
dC_2 = dlnorm(val, meanlog = log(plan2$E_C_ik[2]),
sdlog = log(plan2$Ecart_type[2]))
par(mfrow=c(1,2))
table.C=data.frame(valeur=val,densité.1=dC_1,densité.2=dC_2)
print(table.C)
```

Coût	Densité 1	Densité 2
2000	3.225119 e-05	3.216979 e-05
5000	1.200314 e-05	1.199139 e-05
7000	8.296916 e-06	8.294040 e-06
9000	6.282955 e-06	6.283890 e-06
10000	5.588490 e-06	5.590515 e-06

TABLE A.1.4 – Probabilité d’avoir des coûts de sinistres positifs

Quel que soit le plan de couverture (B ou C), on observe que les probabilités de soumettre une réclamation ayant une valeur positive sont très faibles et presque égales, que l’on souscrive ou non à un plan.

En conclusion, comme pour l’âge et le poids, les plans de souscription, bien que n’étant pas significatifs au seuil de 5%, n’ont pas été retirés de l’étude, car leur effet a été atténué par d’autres covariables. Cette atténuation de l’effet des plans de couverture par d’autres covariables suggère que d’autres variables explicatives ou des interactions entre les covariables peuvent jouer un rôle dans le résultat observé.

A.2 Étude des distributions du coût total de réclamations

Ce code représente la partie technique de l’analyse faite à la section 2.2.3 de notre mémoire. Pour cela, nous faisons une analyse descriptive du jeu de données présenté dans cette section.

A.2.1 Analyse descriptive

- **Chargement de la base de données**

Nous filtrons la base de données et on choisit les variables explicatives à conserver pour étude.

```
pets_pca_dbscan <-read.csv("~/ACTU2M/Travaux R/data/pets_pca_dbscan.csv", sep=";")
```

Recherchons s’il existe des valeurs manquantes de l’espèce et supprimons les.

```
# which(pets_pca_dbscan$Species %in% "")
base=filter(pets_pca_dbscan, !(Species %in% ""))
```

Grâce à un filtrage de la base de données et on choisit les variables explicatives à conserver pour étude. Les données (fréquence et cout) sont données sur deux ans, l’âge a été incrémenté de 1 an pour prendre cet effet en compte.

```
base1=base[,c("X", "Deductible", "AgeYr1", "AmtClaimsYr1", "NumClaimsYr1", "Species",
"Breed")]
base2=base[,c("X", "Deductible", "AgeYr1", "AmtClaimsYr2", "NumClaimsYr2", "Species",
"Breed")]
base1$annee = 1
base2$annee = 2
colnames(base2) = colnames(base1)
base2$AgeYr1 = base2$AgeYr1 + 1
dataTest = rbind(base1,base2) %>% arrange(X)
dataTest$annee=as.factor(dataTest$annee)
names(dataTest)
str(dataTest)
```

```
ggplot(dataTest) + aes(x = Species) + geom_bar(fill = "#2C96C6") +
  geom_label(stat = "count", aes(label=percent(prop.table(after_stat(count)),
accuracy=0.01)))
  theme_minimal()
```

```
ggplot(dataTest) +
  aes(x = annee, y = NumClaimsYr1) +
  geom_col(fill = "#0080FF") +
  labs(x = "Annee de reclamation", y = "Nombre de sinistres") +
  theme_minimal()
```

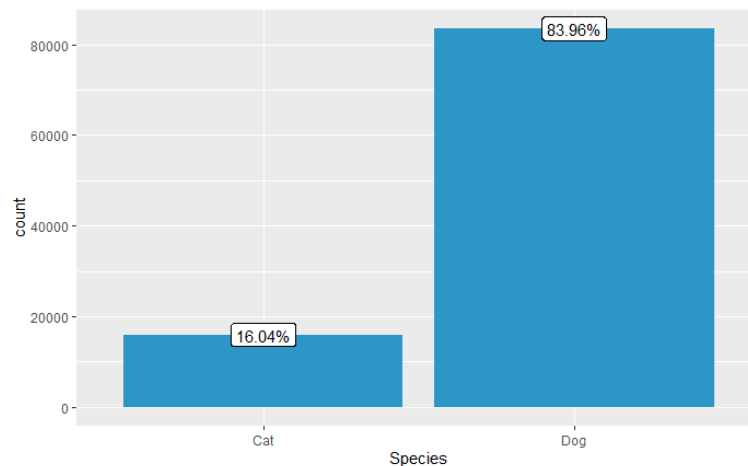


FIGURE A.2.1 – Proportion de chaque espèce dans la base de données

La base donnée étant composé de 83.96% de chiens, nous allons restreintes notre étude à la base de données composée uniquement des chiens.

```
data_dog=filter(dataTest,Species=="Dog")
summary(data_dog)
```

Filtrage en fonction du cout et de la fréquence

```
ggplot(data_dog, aes(x = NumClaimsYr1)) + geom_histogram(bins = 50L, fill="#00AFBB")+
  scale_x_continuous(name = "Number of Claims",
    breaks = seq(0, 116, 5)) + scale_y_continuous(name = "Count",
    labels = label_number())
```

```
AmtClaimsYr1=as.numeric(base$AmtClaimsYr1)
```

```
ggplot(data_dog, aes(x = AmtClaimsYr1)) + geom_histogram(bins = 58L, fill="#E7B800")+
  scale_x_continuous(name = " Claims Amount", breaks = seq(0, 80000, 10000)) +
  scale_y_continuous(name = "Count",
    labels = label_number())
```

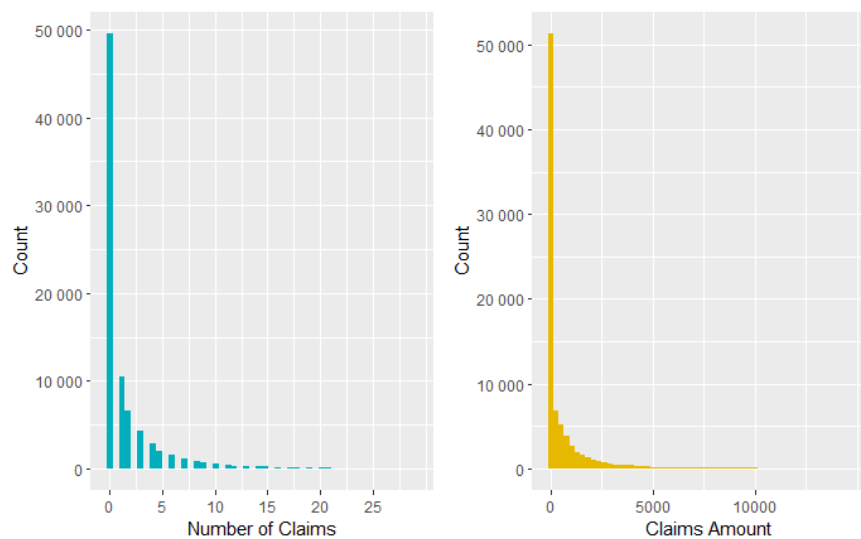


FIGURE A.2.2 – Répartition de la fréquence et du coût de réclamation dans le portefeuille

Au vu de la figure A.2.2 précédente, nous remarquons que les réclamations déclarées vont jusqu'à 116, alors que le quartile à 75% est de 2 pour cela, nous nettoyons de nouveau la base de données. En une année, le nombre de réclamations au-delà de 60 représente 0.012% du portefeuille. Et les réclamations ayant un cout supérieur à 35 000 représente 0.023% du portefeuille.

```
test1=filter(data_dog, NumClaimsYr1>30)
round(nrow(test1)/nrow(data_dog), 7)*100
test2=filter(data_dog, AmtClaimsYr1>15000)
round(nrow(test2)/nrow(data_dog), 7)*100
```

```
Dogdata=filter(data_dog, NumClaimsYr1<30 & AmtClaimsYr1<15000)
ggplot(Dogdata, aes(x = NumClaimsYr1)) + geom_histogram(bins = 50L, fill="#00AFBB")+
  scale_x_continuous(name = "Number of Claims",
    breaks = seq(0, max(Dogdata$NumClaimsYr1), 5)) + scale_y_continuous(name = "Count",
    labels = label_number())
```

```
ggplot(Dogdata, aes(x = AmtClaimsYr1)) + geom_histogram(bins = 58L, fill="#E7B800")+
  scale_x_continuous(name = "Claims Amount",
    breaks = seq(0, max(Dogdata$AmtClaimsYr1), 5000)) +
  scale_y_continuous(name = "Count",
    labels = label_number())
```

```
ggplot(Dogdata) +
  aes(x = Deductible) +
  geom_histogram(bins = 30L, fill = "#2B7899") +
  theme_minimal()
```

Filtrons le jeu de données en tenant compte de la franchise, pour cela nous retiendrons, les franchises les plus représentatives de la base de données.

```
donnee = subset(Dogdata, Deductible==500)
donnee$Breed=as.factor(donnee$Breed)
annee=as.factor(donnee$annee)
deductible=as.factor(donnee$deductible)
str(donnee)
```

A.2.2 Analyse de la fréquence des réclamations

- GBM

```
set.seed(21)
in_training = createDataPartition(donnee$NumClaimsYr1, times = 1, p = 0.78,
  list = FALSE)
training_set = donnee[in_training, ]
testing_set = donnee[-in_training, ]
str(training_set)
```

Effectuons un GBM, sur notre base de données, afin de capter les interactions pertinentes.

```
set.seed(123)
regPois=gbmt(NumClaimsYr1~AgeYr1+Breed+annee,
  data=training_set,
  distribution=gbm_dist("Poisson"),
  train_params=training_params(num_trees=400,
    shrinkage=0.01,
    interaction_depth=4,
    min_num_obs_in_node=500,
    bag_fraction=0.5,
    num_train=0.5*nrow(training_set)),
  is_verbose=FALSE,
  keep_gbm_data=TRUE,
  cv_folds = 5,
  par_details=gbmParallel(num_threads=3))
```

```
plot(gbmt_performance(regPois),method="cv")
```

```
best_iter = gbmt_performance(regPois, method="cv")
best_iter
```

On constate que la variable "BREED" est la variable la plus importante,

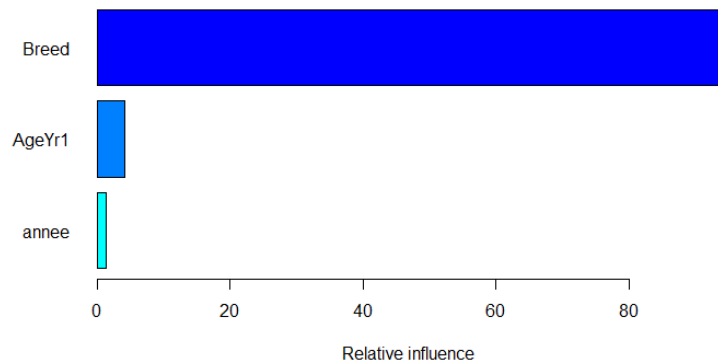


FIGURE A.2.3 – Importance des variables explicatives pour la fréquence des réclamations

```
resultat=summary(regPois, num_trees = best_iter)
```

```
var_names = c("AgeYr1","Breed","annee")
```

```
res = matrix(NA, 3, 3)
```

```
for (i in 1:2) {
```

```
  for (j in (i + 1):3) {
```

```
    res[i, j] = interact(gbm_fit_obj = regPois, data = training_set,
      var_indices = c(i,j), best_iter)
```

```
  }
```

```
}
```

```
diag(res) = 0
```

```
row.names(res) = var_names
```

```
colnames(res) = row.names(res)
```

```
interact_melt <- melt(res, na.rm = TRUE)
```

```
ggplot(data = interact_melt, aes(x = Var1, y = Var2, fill = value)) +
```

```
geom_tile(color = "white") +
```

```
  scale_fill_gradient2(
```

```
    low = "white", mid = "gray", high = "blue", name = "Friedman's\nH-statistic") +
```

```
  theme(axis.text.x = element_text(angle = 45, vjust = 1, size = 12, hjust = 1)) +
```

```
  coord_fixed())
```

Sur la figure A.2.4, La statistique H de Friedman nous montre qu'une forte interaction entre Age et espèce. Ce qui s'explique tout simplement par le fait que l'âge s'incrémente après qu'une année est passé

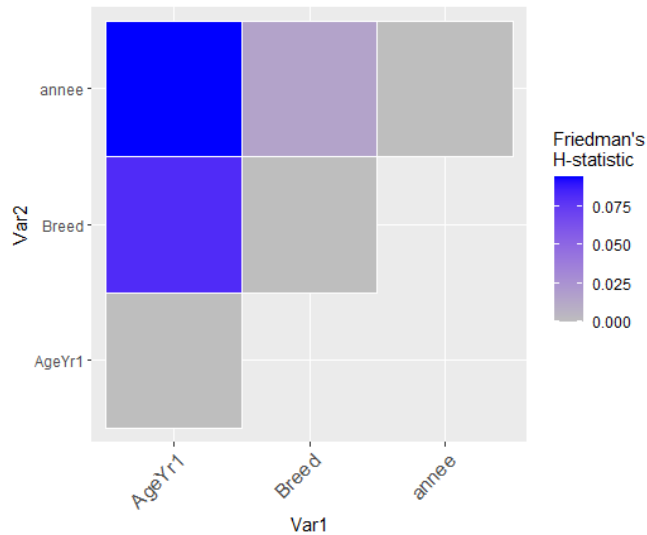


FIGURE A.2.4 – Interaction entre les variables explicatives à l'aide de la statistique de Friedman

- GAM

```
require(parallel)
cl = makeCluster(detectCores()-1) # Number of cores to use, for parallel computing.
ptn_0 = Sys.time()
freq_bam = bam(NumClaimsYr1 ~ s(AgeYr1)+Breed+annee,
               data = training_set,
               select = TRUE,
               family = poisson(), cluster = cl)
stopCluster(cl)
freq_bam
print(Sys.time() - ptn_0)

visreg(freq_bam, xvar="AgeYr1", gg=TRUE, scale="response")+
ylim(c(0,5))+ylab("Multiplicative Effect")+
scale_x_continuous(name = "Age of Dog", limits = c(0,14), breaks = c(0,seq(0,14,2),14))

summary(freq_bam)
```

Visualisation des différents niveaux de la variable Breed

```
viz2=visreg(freq_bam, "Breed")
tmp=viz2[[1]]
```

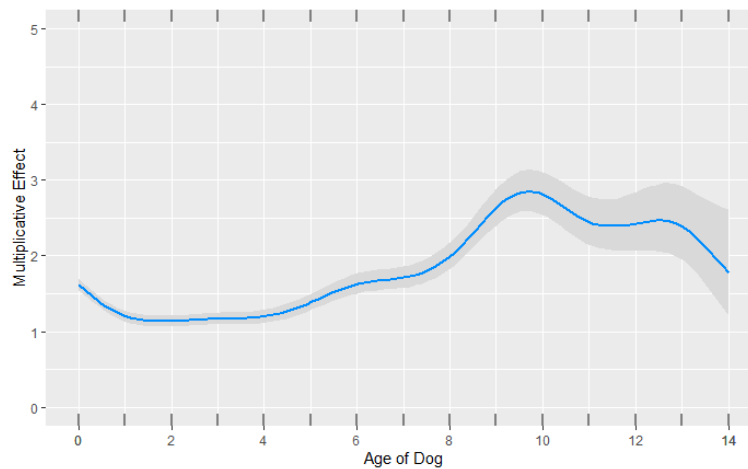


FIGURE A.2.5 – Effet de l'âge des chiens sur la fréquence des réclamations

À l'aide la fonction KMEANS, nous procédons à une classification de notre jeu de données en fonction de la variable BREED. Pour cela, nous utilisons les coefficients de visualisation des niveaux de référence du GAM, pour regrouper les niveaux de la variable BREED.

```
tmp.cont=tmp[,-c(1,3,2,4)]
# groupe=tmp$Breed
#Normalisation
normalize=function(x){
  return((x-min(x))/(max(x)-min(x)))
}
tmp_clean=normalize(tmp.cont)

#Clustering avec KMEANS
set.seed(12)
k_res=kmeans(tmp_clean,centers=4,nstart = 50)
print(k_res)
tmp$class=as.factor(k_res$cluster)
```

Le coefficient de silhouette variant entre $[-1;1]$, On constate la moyenne est proche de 1 alors, on peut conclure que la segmentation est de bonne qualité.

```
#Verifions si le partition a été bon
library(cluster)
si.metric=silhouette(
  k_res$cluster,dist(tmp_clean,"euclidean")
)
print(summary(si.metric))
```

Création des différentes classes de chiens en fonction des races

```
#Merge dans la base training
training_set$classe=training_set$Breed%>%
```

```

fct_collapse(

classe1=c("Airedale Terrierb", "Akita", "Alaskan Malamute", "American Bulldog", "American Bull
classe2=c("Affenpinscher", "Alaskan Klee Kai", "American Hairless Terrier", "Anatolian Shephe
classe3=c("American English Coonhound", "American Eskimo", "Argentine Dogo", "Australian Catt
classe4=c("American Foxhound", "American Water Spaniel", "Barbet", "Bedlington Terrier", "Blac

#Merge dans la base testing
testing_set$classe=testing_set$Breed%>%
fct_collapse(
classe1=c("Airedale Terrierb", "Akita", "Alaskan Malamute", "American Bulldog", "American Bull
classe2=c("Affenpinscher", "Alaskan Klee Kai", "American Hairless Terrier", "Anatolian Shephe
classe3=c("American English Coonhound", "American Eskimo", "Argentine Dogo", "Australian Catt
classe4=c("American Foxhound", "American Water Spaniel", "Barbet", "Bedlington Terrier", "Blac

table(training_set$classe)

require(parallel)
cl = makeCluster(detectCores()-1) # Number of cores to use, for parallel computing.
ptn_0 = Sys.time()
freq_bam1 = bam(NumClaimsYr1 ~ s(AgeYr1)+annee+classe,
               data = training_set,
               family = poisson(),
               select = TRUE ,
               cluster = cl)
stopCluster(cl)
freq_bam1
print(Sys.time() - ptn_0)

summary(freq_bam1)
plot(freq_bam1, trans = exp, scale = 0, shade = TRUE)

require(parallel)
cl = makeCluster(detectCores()-1) # Number of cores to use, for parallel computing.
ptn_0 = Sys.time()
freq_bam2 = bam(NumClaimsYr1 ~ s(AgeYr1, by=classe)+annee,
               data = training_set,
               family = poisson(),
               select = TRUE ,
               cluster = cl)
stopCluster(cl)
freq_bam2
print(Sys.time() - ptn_0)

summary(freq_bam2)

```


Family: poisson

Link function: log

Formula:

NumClaimsYr1 ~ s(AgeYr1, by = classe)

Parametric coefficients:

	Estimate	Std. Error	z value	Pr(> z)
(Intercept)	0.431001	0.007245	59.49	<2e-16 ***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Approximate significance of smooth terms:

	edf	Ref.df	Chi.sq	p-value
s(AgeYr1):classeclasse2	0.0004113	3	0.000	0.50827
s(AgeYr1):classeAiredale Terrier	1.6063961	5	5.231	0.03611 *
s(AgeYr1):classeclasse1	7.9031799	9	774.718	< 2e-16 ***
s(AgeYr1):classeclasse3	0.4060069	9	0.683	0.19455
s(AgeYr1):classeclasse4	0.8735376	8	6.889	0.00498 **
s(AgeYr1):classeBrussels Griffon Brabancon	1.7663835	8	43.152	< 2e-16 ***
s(AgeYr1):classeChihuahua Short Coat	2.8457652	9	59.829	< 2e-16 ***
s(AgeYr1):classeJapanese Chin Spaniel	0.4973572	2	0.987	0.15897
s(AgeYr1):classePoodle Toy	4.9003154	9	56.448	< 2e-16 ***
s(AgeYr1):classePortuguese Water Dog	0.0001543	9	0.000	0.89856
s(AgeYr1):classePyrenean Mountain Dog Great Pyrenees	0.0002810	6	0.000	0.61905
s(AgeYr1):classeSaluki Gazelle Hound	0.6089735	2	1.275	0.13675

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

R-sq.(adj) = 0.0107 Deviance explained = 1.75%

fREML = 53130 Scale est. = 1 n = 12630

```
anova(freq_bam2,freq_bam1,freq_bam)
AIC(freq_bam2,freq_bam1,freq_bam)
lambda.obs=mean(testing_set$NumClaimsYr1)
lambda.predict=mean(predict(freq_bam, data=testing_set, type="response"))
lambda.obs
lambda.predict
lambda.distr=fitdistr(round(predict(freq_bam2, data=testing_set,
type="response")), "Poisson")
```

lambda observé	1.509688
lambda.prédit	1.568963

TABLE A.2.1 – Paramètre de la distribution de la fréquence des réclamations à l'aide GAM

```
set.seed(21)
training = createDataPartition(donnee$NumClaimsYr1, times = 1, p = 0.78, list = FALSE)
dataTrain = donnee[training, ]
dataTest = donnee[-training, ]
```

```
str(dataTain)
```

A.2.3 Étude la Sévérité des réclamations

- GBM

Nous conservons les réclamations qui ont des coûts positifs.

```
dataTain = filter(dataTain,AmtClaimsYr1>0)
dataTest= filter(dataTest,AmtClaimsYr1>0)
set.seed(123)
regLN=gbmt(log(AmtClaimsYr1)~AgeYr1+Breed+annee,
  data=dataTain,
  distribution=gbm_dist("Gaussian"),
  train_params=training_params(num_trees=500,
    shrinkage=0.01,
    interaction_depth=5,
    min_num_obs_in_node=1000,
    bag_fraction=0.5,
    num_train=nrow(dataTain)),
  is_verbose=FALSE,
  keep_gbm_data=TRUE,
  cv_folds = 10,
  par_details=gbmParallel(num_threads=3))

plot(gbmt_performance(regLN),method="cv")
```

Les variables explicatives sont classées dans l'ordre d'importance :
Race, Franchise, Age, Espèces

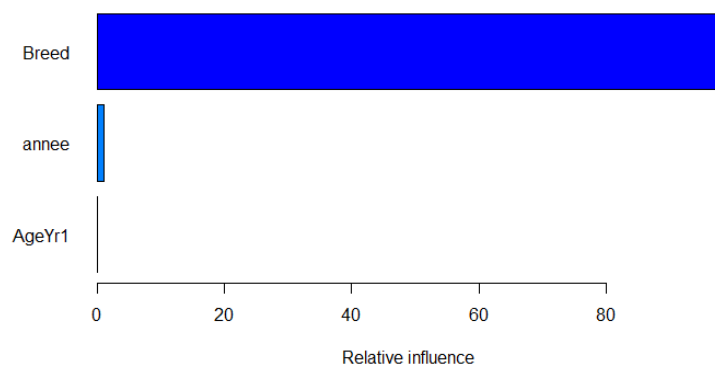


FIGURE A.2.6 – Importance des variables explicatives pour le coût des réclamations

```
best_iter = gbmt_performance(regLN, method="cv")
best_iter
```

```
summary(regLN)
```

La statistique H de Friedman nous montre une forte interaction entre Age et espèce.

```
var_names = c("AgeYr1", "Breed", "annee")
res = matrix(NA, 3, 3)
for (i in 1:2) {
  for (j in (i + 1):3) {
    res[i, j] = interact(gbm_fit_obj = regLN, data = dataTain,
      var_indices = c(i, j), best_iter)
  }
}
diag(res) = 0
row.names(res) = var_names
colnames(res) = row.names(res)

interact_melt <- melt(res, na.rm = TRUE)

ggplot(data = interact_melt, aes(x = Var1, y = Var2, fill = value)) + geom_tile(color = "white") +
  scale_fill_gradient2(low = "white", mid = "gray", high = "blue", name = "Friedman's H-statistic") +
  theme(axis.text.x = element_text(angle = 45, vjust = 1, size = 12, hjust = 1)) +
  coord_fixed()
```

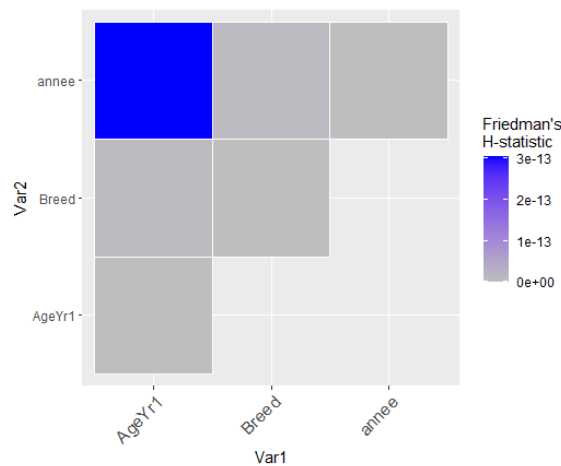


FIGURE A.2.7 – Interaction entre les variables explicatives à l'aide de la statistique de Friedman

- GAM

```
require(parallel)
cl = makeCluster(detectCores() - 1) # Number of cores to use, for parallel computing.
timer = Sys.time()
LN_bam = bam(log(AmtClaimsYr1) ~ s(AgeYr1)+Breed+annee,
             data = dataTain,
             family = gaussian(),
             cluster = cl,
             select= TRUE
            )
stopCluster(cl)
print(Sys.time() - timer)
```

```
par(mfrow = c(1, 2))
plot(LN_bam, trans = exp, scale = 0, shade = TRUE)
```

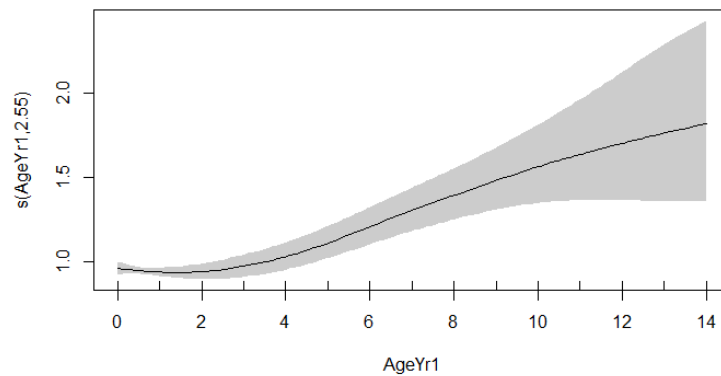


FIGURE A.2.8 – Effet de l'âge des chiens sur le coût des réclamations

```
viz3=visreg(LN_bam,"Breed")
```

```
summary(LN_bam)
```

```
Family: gaussian
Link function: identity
Formula:
log(AmtClaimsYr1) ~ s(AgeYr1, by = classe)
Parametric coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  6.83192    0.01695   403.1  <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
Approximate significance of smooth terms:
```

	edf	Ref.df	F	p-value
s(AgeYr1):classeAiredale Terrier	5.290e-05	3	0.000	0.806
s(AgeYr1):classeclasse1	2.438e+00	9	3.059	1.32e-06 ***
s(AgeYr1):classeclasse3	1.897e-04	9	0.000	0.418
s(AgeYr1):classeBrussels Griffon Brabancon	3.811e-05	3	0.000	0.969
s(AgeYr1):classeChihuahua Short Coat	3.433e-01	9	0.050	0.251
s(AgeYr1):classeJapanese Chin Spaniel	3.042e-04	2	0.000	0.349
s(AgeYr1):classePoodle Toy	2.062e-01	9	0.027	0.273
s(AgeYr1):classePortuguese Water Dog	4.737e-01	7	0.129	0.168
s(AgeYr1):classePyrenean Mountain Dog Great Pyrenees	5.260e-05	5	0.000	0.798
s(AgeYr1):classeSaluki Gazelle Hound	4.194e-05	2	0.000	0.980

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

R-sq.(adj) = 0.00579 Deviance explained = 0.648%

fREML = 8005.1 Scale est. = 1.4347 n = 5001

#Merge dans la base training

```
dataTain$classe=dataTain$Breed%>%
```

```
fct_collapse(
```

```
classe1=c("Airedale Terrierb","Akita","Alaskan Malamute","American Bulldog","American Bull
```

```
classe2=c("Affenpinscher","Alaskan Klee Kai","American Hairless Terrier","Anatolian Shephe
```

```
classe3=c("American English Coonhound","American Eskimo","Argentine Dogo","Australian Catt
```

```
classe4=c("American Foxhound","American Water Spaniel","Barbet","Bedlington Terrier","Blac
```

#Merge dans la base testing

```
dataTest$classe=dataTest$Breed%>%
```

```
fct_collapse(
```

```
classe1=c("Airedale Terrierb","Akita","Alaskan Malamute","American Bulldog","American Bull
```

```
classe2=c("Affenpinscher","Alaskan Klee Kai","American Hairless Terrier","Anatolian Shephe
```

```
classe3=c("American English Coonhound","American Eskimo","Argentine Dogo","Australian Catt
```

```
classe4=c("American Foxhound","American Water Spaniel","Barbet","Bedlington Terrier","Blac
```

```
require(parallel)
```

```
cl = makeCluster(detectCores() - 1) # Number of cores to use, for parallel computing.
```

```
timer = Sys.time()
```

```
LN_bam1 = bam(log(AmtClaimsYr1) ~ s(AgeYr1, by=classe)+annee,
```

```
data = dataTain,
```

```
family = gaussian(),
```

```
cluster = cl,
```

```
select= TRUE
```

```
)
```

```
stopCluster(cl)
```

```
LN_bam1
```

```
print(Sys.time() - timer)
```

```
summary(LN_bam1)
```

```
AIC(LN_bam, LN_bam1)
```

```
obs.mean=mean(dataTest$AmtClaimsYr1)
obs.sd=sd(dataTest$AmtClaimsYr1)
mean.predict=mean(predict(LN_bam1, dataTest, type="response"),na.rm=TRUE)
sd.predict=sd(predict(LN_bam1, dataTest, type="response"),na.rm=TRUE)

obs.mean=mean(dataTest$AmtClaimsYr1)
obs.sd=sd(dataTest$AmtClaimsYr1)

predict.mean=exp(mean.predict+0.5*sd.predict^2)
predict.sd=sqrt(exp(2*mean.predict+sd.predict^2)*(exp(sd.predict^2)-1))
predict.mean
predict.sd

obs.mean
obs.sd
```

mean observée	1753.70700
sd observée	2377.48400
mean prédit	924.59600
sd prédit	74.99795

TABLE A.2.2 – Paramètre de la distribution du coût des réclamations à l'aide GAM

A.3 CONDITIONAL MEAN RISK SHARING CODE

```
requiredpackages=c("actuar", "distr", "distrEx", "conflicted", "esquisse",
  "ggplot2", "latex2exp", "reshape2")
for (i in requiredpackages) {
  if(!require(i, character.only = T))install.packages(i)
  require(i, character.only = T)
}

conflict_prefer("r", "distr")
conflict_prefer("d", "distr")
conflict_prefer("p", "distr")
conflict_prefer("q", "distr")
```

A.3.1 Choix des paramètres des distributions de fréquences et de sévérités

Les valeurs des paramètres σ ; α ; μ ; β et θ sont obtenues de l'article de Pai et al (oct. 2006). « Compound Poisson Model with Covariates : A Case Study on Pet Insurance, p227, p229»

```

#Choix des paramètres
sigma=1.03375
alpha=2.67253
mu=4.90284
q=0.8#quote part

#Résultat obtenu de l'étude menée par de Jeffrey S Pai et al. (2013)
beta_N=c(1.23070,0.00026 ,0.02957,0.04708,0.11882,-0.00058,-0.24512,
         -0.16021,0.42197)#vecteur de paramètres de beta

theta_C=c(-0.10522 ,0.00227,0.12641,0.08862,0.14922,0.00355,0.02991,
          0.02971,-0.90829)#Vecteur de paramètres theta

```

La fréquence moyenne de réclamations est donnée : $\lambda = 12 * \exp(-\alpha + \beta^T \times Z^i)$

Le coût moyen de réclamations est donnée par la relation : $E[\tilde{C}_{ik}] = q * \exp(\mu + \theta^T \times Z^i + \frac{\sigma^2}{2})$

L'écart type est donné par : $\sigma_{\tilde{C}_{ik}} = q * \sqrt{(e^{\sigma^2} - 1) * \exp(2 * (\mu + \theta^T \times Z^i) + \sigma^2)}$

```

#Caractéristique d'un animal
param=function(covar,a,quote){
  val1 = a/12
  #Sévérité et Fréquence
  val2=12*exp(-alpha+(t(beta_N)%*%covar))
  val3=quote*exp(mu+(t(theta_C)%*%covar)+(sigma^2)/2)
  val4= quote*sqrt((exp(sigma^2)-1)*exp(2*(mu+(t(theta_C)%*%covar))+(sigma^2)))
  val5=val2*val3
  return(list(Age=val1, lambda=val2, E_C_ik=val3, Ecart_type=val4, cout=val5))}

```

Considérons un pool composé de trois catégories de participants (propriétaires de chiens) ayant des caractéristiques identiques.

Groupe 1 Les caractéristiques des chiens sont les suivants : Étant à leur premier antécédent médical; vivant en région; âgé de 12 mois de classe 2 (St.Bernards); ayant un poids de 45kg; avec un plan de couverture B.

Groupe 2 Les caractéristiques sont les suivantes : de race Bulldog (Classe 1), étant à leurs premiers antécédents médical, âgé de 60 mois et vivant en région, de masse 60 kg, ayant une couverture santé complète, y compris des soins dentaires. ayant le plan de couverture C

Groupe 3 Les caractéristiques des chiens sont les suivants : De race Bulldogs, n'étant pas au premier antécédent médical, âgé de 120 mois et ne vivant pas en région, de masse 20 kg, ayant une couverture basique complète.

```

grpe=function(quote){

  #Définissons les covariables

```

```

groupe1= as.data.frame(param(c(0, 1, 1, 1, 0, 45, 1, 0, 0), 12,
quote=quote))
groupe2= as.data.frame(param(c(0, 5, 1, 0, 1, 60, 0, 0, 0), 60,
quote=quote))
groupe3= as.data.frame(param(c(1, 10, 0, 0, 0, 20, 0, 0, 1),120,
quote=quote))

return(groupe=rbind(groupe1,groupe2,groupe3))}
groupe=grpe(q)
print(groupe)

```

Représentation de la distribution du montant des réclamations

Le montant des réclamations par animal suit une loi Log-Normale. Les valeurs prises pour le coût de réclamation sont comprises a=0 et b1=qlnorm(0.9999,log (m1), log(sd1))=1.104207e+12. Nous obtenons une valeur très élevée. Pour la bonne supérieure, nous supposons qu'un soin médical peut atteindre un montant de 100 000 (b=100000).

k=100#pas d'itération

```

# Définir les paramètres de la distribution log normale
# Créer un vecteur de valeurs pour tracer la courbe
val = seq(0, 10000000, length=100)

## pour le groupe 1
m1=groupe$E_C_ik[1]
sd1=groupe$Ecart_type[1]

# Calculer de la densité de probabilité pour chaque valeur de x
d_1 = dlnorm(val, meanlog = log(m1), sdlog = log(sd1))

# Calculer de la fonction de répartition pour chaque valeur de x
f_1 = plnorm(val, meanlog = log(m1), sdlog = log(sd1))

## pour le groupe 2
m2=groupe$E_C_ik[2]
sd2=groupe$Ecart_type[2]

# Calculer de la densité de probabilité pour chaque valeur de x
d_2 = dlnorm(val, meanlog = log(m1), sdlog = log(sd2))

# Calculer de la fonction de répartition pour chaque valeur de x pour
f_2 = plnorm(val, meanlog = log(m2), sdlog = log(sd2))

## pour le groupe 3
m3=groupe$E_C_ik[3]
sd3=groupe$Ecart_type[3]

```



```
# Calculer de la densité de probabilité pour chaque valeur de x
d_3 = dlnorm(val, meanlog = log(m3), sdlog = log(sd3))

# Calculer de la fonction de répartition pour chaque valeur de x pour
f_3 = plnorm(val, meanlog = log(m3), sdlog = log(sd3))
```

Représentation

```
# Tracer la courbe de densité de probabilité
plot(val,d_1,col="blue3", xlab = "Coûts", ylab = "Densité de probabilité",
      type = "l",lty=1)
lines(val,d_2,col="brown1", xlab = "Coûts", ylab = "Densité de probabilité",
      type = "l",lty=4)
lines(val,d_3,col="green2", xlab = "Coûts", ylab = "Densité de probabilité",
      type = "l",lty=4)

# Tracer la courbe de la fonction cumulative de probabilité
plot(val,f_1,col="blue3",xlab = "Coûts", ylab = "Fonction de repartition",
      type = "l",lty=1,lwd=1,ylim=c(0.5,1))
lines(val,f_2,col="brown1",xlab = "Coûts", ylab = "Fonction de repartition",
      type = "l",lty=2,lwd=1)
lines(val,f_3,col="green2",xlab = "Coûts", ylab = "Fonction de repartition",
      type = "l",lty=2,lwd=1)
legend("bottomright",legend=c("groupe 1","groupe 2","groupe 3"),
      col = c("blue3","brown2","green2"),lty=1:2,box.lty = 0)
```

Représentation de la fréquence de sinistres pour chaque groupe

```
#Distribution de la sévérité

# Créer un vecteur de valeurs pour tracer la courbe
u=seq(0,10,1)

#Nombre de sinistre pour un chien âgé de 12 mois sans antécédents médicaux
F1 <- dpois(u, groupe$lambda[1])

# Tracer la courbe de densité de probabilité
plot(u, F1, xlab = "Valeurs", ylab = "Densité de probabilité",col="blue3",
      lwd=1,
      type="o",pch=20, lty=1, bg=par("bg"))

#Nombre de sinistre pour un chien âgé de 12 mois sans antécédents médicaux
F2 <- dpois(u, groupe$lambda[2])

# Tracer la courbe de densité de probabilité
lines(u, F2, xlab = "Valeurs", ylab = "Densité de probabilité",col="brown2",
      lwd=1,
```

```

    type="o", lty=2, pch=20, bg=par("bg"))
#Nombre de sinistre pour un chien agé de 12 mois sans antécédents médicaux
F3 <- dpois(u, groupe$lambda[3])

# Tracer la courbe de densité de probabilité
lines(u, F3, xlab = "Nombre de sinistres", ylab = "Densité de probabilité",
      col="green2", lwd=1, type="o", lty=4, pch=20, bg=par("bg"))
legend("topright", inset=0.05, legend=c("groupe 1", "groupe 2", "groupe 3"),
      col = c("blue3", "brown2", "green2"), lty=1:2, box.lty = 0)

```

Pour déterminer la distribution du montant total des réclamations par individus, il est nécessaire de discrétiser la fonction densité des C_{ik} car elle est une variable continue. Pour cela, nous allons utiliser le package ACTUAR de R et nous utiliserons la méthode "Unbiased" donnée par la relation (2.3.6). Les bornes de l'intervalle de discrétisation sont : $a=0$; $b=100\ 000$ et un pas d'itération $k=100$

```

#Considérons un pool constitué 3 groupes d'individus X1; X2 et X3.

#Groupe 1
#Paramètre de sévérité et coûts
a1=12 #chien âgé de 12 mois
m1=groupe$E_C_ik[1]
sd1=groupe$Ecart_type[1]
lambda1=groupe$lambda[1]

#Groupe 2
#Paramètre de sévérité et coûts
a2=60 #chien âgé de 60 mois
m2=groupe$E_C_ik[2]
sd2=groupe$Ecart_type[2]
lambda2=groupe$lambda[2]

#Groupe 3
#Paramètre de sévérité et coûts
a3=120 #chien âgé de 120 mois
m3=groupe$E_C_ik[3]
sd3=groupe$Ecart_type[3]
lambda3=groupe$lambda[3]

```

Fonction discretize, Cette fonction fournit la densité du coût total de sinistre X_i à partir de la fonction de répartition de répartition des C_{ik} calculées aux différents pas de discrétisation

Cette fonction permet de discrétiser la distribution de probabilité des C_{ik} en utilisant la méthode "Unbiased" (Minimisation de l'erreur lors de la discrétisation), entre les bornes a et b avec un pas de discrétisation k . Le paramètre lev spécifie les niveaux (probabilités) correspondantes aux valeurs discrètes correspondantes. Ses niveaux sont calculés en utilisant

la fonction `levlnorm()` qui utilise la distribution Log-Normale spécifique par les paramètres x , $\log(m_1)$, $\log(sd_1)$. Il faut noter que cette fonction ne s'utilise qu'avec la méthode "Unbiased".

Les coûts de sinistres sont des valeurs positives ($C_{ik} \geq 0$), la valeur maximale du montant des réclamations peut être obtenu en utilisant le quantile d'ordre 99.99% de la fonction de répartition des C_{ik} i.e $a = 0$ et $b = 100000$. Dans notre exemple introductif, le montant maximal des réclamations à la charge du pool est 8 500. Nous considérons cette valeur comme borne supérieure de l'intervalle de discrétisation.

```
#Intervalle maximal du coût de sinistre#Valeur maximale du coût de sinistre
a=0
#b1=qlnorm(0.9999,log(m1),log(sd1))
#b1=1.104207e+12 Nous obtenons une valeur très élevée. Pour la borne supérieure,
#nous supposons qu'un soin médical peut atteindre un montant de 100 000
b=100000
k=100#pas d'itération
x = seq(a, b, by=k)
groupe=grpe(q)
#Paramètre de sévérité et coûts
m1=groupe$E_C_ik[1]
sd1=groupe$Ecart_type[1]

#Groupe 2
m2=groupe$E_C_ik[2]
sd2=groupe$Ecart_type[2]

#Groupe 3
a3=120#chien âgé de 120 mois
m3=groupe$E_C_ik[3]
sd3=groupe$Ecart_type[3]

#Discrétisation par la méthode Unbiased du coût de sinistre
f1=discretize(plnorm(x,log(m1),log(sd1)),method = "unbiased",
from = a, to=b,step = k,lev=levlnorm(x,log(m1),log(sd1)))

f2=discretize(plnorm(x,log(m2),log(sd2)),method = "unbiased",
from = a, to=b,step = k,lev=levlnorm(x,log(m2),log(sd2)))
f3=discretize(plnorm(x,log(m3),log(sd3)),method = "unbiased",
from = a, to=b,step = k,lev=levlnorm(x,log(m3),log(sd3)))
```

Groupe 1

```
curve(plnorm(x,log(m1),log(sd1)), from = a, to = b, lwd = 1,
      ylab=TeX(r"${P}[C_{ik} <= 1]${}")
)
op <- par(col = "blue3")
plot(stepfun(x, diffinv(f1)), pch = 20, add = TRUE)
```

Groupe 2

```
curve(plnorm(x,log(m2),log(sd2)), from = a, to = b, lwd = 1,
      ylab=TeX(r"${P}[C_{ik} <= 1]${}")
op <- par(col = "brown2")
plot(stepfun(x, diffinv(f2)), pch = 20, add = TRUE)
```

Groupe 3

```
curve(plnorm(x,log(m3),log(sd3)), from = a, to = b, lwd = 1,
      ylab=TeX(r"${P}[C_{ik} <= 1]${}")
op <- par(col = "green2")
plot(stepfun(x, diffinv(f3)), pch = 20, add = TRUE)
```

A.3.2 Distribution du coût total individuel

```
#Algorithme de Panjer
```

```
#Groupe1
```

```
cdfX1=aggregateDist("recursive", model.freq = "poisson", model.sev=f1,
                    lambda=groupe$lambda[1], p0 = exp(-groupe$lambda[1]),
                    tol=10e-4, x.scale = 1)
```

```
summary(cdfX1)
```

```
#Groupe2
```

```
cdfX2=aggregateDist("recursive", model.freq = "poisson", model.sev=f2,
                    lambda=groupe$lambda[2], p0 = exp(-groupe$lambda[2]),
                    tol=10e-4, x.scale = 1)
```

```
summary(cdfX2)
```

```
#Groupe3
```

```
cdfX3=aggregateDist("recursive", model.freq = "poisson", model.sev=f3,
                    lambda=groupe$lambda[3], p0 = exp(-groupe$lambda[3]),
                    tol=10e-4, x.scale = 1)
```

```
summary(cdfX3)
```

```
summary(cdfX1)
```

```
lim1=quantile(cdfX1,0.9999)
```

```
#borne supérieur de l'intervalle des valeurs possibles
```

```
u1=seq(0,lim1,1)
```

```
summary(cdfX2)
```

```
lim2=quantile(cdfX2,0.9999)
```

```
#borne supérieur de l'intervalle des valeurs possibles
```

```
u2=seq(0,lim2,1)
```

```
summary(cdfX3)
```

```

lim3=quantile(cdfX3,0.9999)
#borne supérieur de l'intervalle des valeurs possibles
u3=seq(0,lim3,1)

#Représentation des fonctions de répartition
par(mfrow=c(1,3))
plot(cdfX1,do.points=FALSE,ylim=c(0.5,1),
     main = "Groupe 1",
     sub = comment(x))
points(0,cdfX1(0),pch=19,col="blue3")#probability mass function at 0
plot(cdfX2,do.points=FALSE,
     main = "Groupe 2",
     sub = comment(x))
points(0,cdfX2(0),pch=19,col="brown3")#probability mass function at 0
plot(cdfX3,do.points=FALSE,
     main = "Groupe 3",
     sub = comment(x))
points(0,cdfX3(0),pch=19,col="green2")#probability mass function at 0

```

Déterminons et construisons les masses de probabilité du coût total individuel

```

k1=10#pas d'itération
par(mfrow=c(1,3))
valX1=list(ValX1=u1[1:lim1],probX1=diff(cdfX1(u1)))
valX2=list(ValX2=u2[1:lim2],probX2=diff(cdfX2(u2)))
valX3=list(ValX3=u3[1:lim3],probX3=diff(cdfX3(u3)))

plot(valX1$ValX1[seq(0,lim1,k1)],valX1$probX1[seq(0,lim1,k1)],xlab="x",
     ylab=TeX(r"${P[X_{i} = k]}"), type="h")
lines(valX1$ValX1[seq(0,lim1,k1)],valX1$probX1[seq(0,lim1,k1)],xlab="x",
     ylab=TeX(r"${P[X_{i} = k]}"), type="l",col="blue3")
title(main = "Groupe 1")

k2=10#pas d'itération
plot(valX2$ValX2[seq(0,lim2,k2)],valX2$probX2[seq(0,lim2,k2)],xlab="x",
     ylab=TeX(r"${P[X_{i} = k]}"), type="h")
lines(valX2$ValX2[seq(0,lim2,k2)],valX2$probX2[seq(0,lim2,k2)],xlab="x",
     type="l",col="brown3")
title(main = "Groupe 2")

k3=10#pas d'itération
plot(valX3$ValX3[seq(0,lim2,k3)],valX3$probX3[seq(0,lim2,k3)],xlab="x",
     ylab=TeX(r"${P[X_{i} = k]}"), type="h")
lines(valX3$ValX3[seq(0,lim2,k3)],valX3$probX3[seq(0,lim2,k3)],xlab="x",
     type="l",col="green2")
title(main = "Groupe 3")

```

```

#Calcul direct
#Fonction de répartition des Y_i
Gy=function(y,F,d,M){
  I1=ifelse(y < 0,1,0)*0
  I2=ifelse(y==0,1,0)*F(d)
  I3=ifelse(0<y & y < M,1,0)*F(y+d)
  I4=ifelse((y >= M),1,0)*1
  return(I1+I2+I3+I4)
}

#Approximation collective
#déterminer la fonction de répartition des Y_i sachant que Y_i>0
Fy=function(y, F, d, M){
  I1=ifelse(y <= 0,1,0)*0
  I2=ifelse(0<y & y < M,1,0)*(F(y+d) - F(d))/(1-F(d))
  I3=ifelse((y >= M),1,0)*1
  return(I1+I2+I3)
}

```

A.3.3 Calcul de la distribution de la charge S_n

```

#coût total maximal pour chaque par individus
# plafond=data.frame(plafond1=lim1,plafond2=lim2,plafond3=lim3)
#Nombre de chiens de même caractéristiques
#Considérons un groupe de 200 individus
deduc=250
Maxi=8500
nb_ind=800
u=seq(0,1000,10)
#Distribution des réclamations à la charge du groupe
u1=seq(0,lim1,1)
u2=seq(0,lim2,1)
u3=seq(0,lim3,1)
#Distribution de la variable aléatoire représentant la
#réclamation à la charge du pool
#CDF des réclamations des groupes 1, 2 et 3
cdfY1=data.frame(valeurs=u1,cdf=Gy(u1,cdfX1,deduc,Maxi))
cdfY2=data.frame(valeurs=u2,cdf=Gy(u2,cdfX2,deduc,Maxi))
cdfY3=data.frame(valeurs=u3,cdf=Gy(u3,cdfX3,deduc,Maxi))

#PDF des réclamations des groupes 1, 2 et 3
# pdfY1=list(valeurs=u1,pdf=diff(cdfY1$cdf))
# pdfY2=list(valeurs=u2,pdf=diff(cdfY2$cdf))
# pdfY3=list(valeurs=u3,pdf=diff(cdfY3$cdf))

#PDF des réclamations des groupes 1, 2 et 3
# pdfY1=list(valeurs=u1[-1],pdf=diff(cdfY1$cdf))

```

```
pdfY1=list(valeurs=u1,pdf=c(Gy(0,cdfX1,deduc,Maxi),diff(cdfY1$cdf)))
pdfY2=list(valeurs=u2,pdf=c(Gy(0,cdfX2,deduc,Maxi),diff(cdfY2$cdf)))
pdfY3=list(valeurs=u3,pdf=c(Gy(0,cdfX3,deduc,Maxi),diff(cdfY3$cdf)))
```

```
gpe1=DiscreteDistribution(supp = pdfY1$valeurs, prob = pdfY1$pdf)
#Moyenne des coûts de réclamations du groupe1
E(gpe1)
```

```
gpe2=DiscreteDistribution(supp = knots(cdfX2), prob = pdfY2$pdf)
#Moyenne des coûts de réclamations du groupe2
E(gpe2)
```

```
gpe3=DiscreteDistribution(supp = knots(cdfX3), prob = pdfY3$pdf)
#Moyenne des coûts de réclamations du groupe3
E(gpe3)
```

```
#Cette fonction determine la distribution de $S_n$
FS=function(y,Ind_grpe,num,ded,Max,q){
  # cdfX=cdf(quote)
  nb_grp1=0.5*num #groupe 1
  nb_grp2=0.3*num #groupe 2
  nb_grp3=0.2*num #groupe 3

  #déterminer les paramètres de fréquence de chaque groupe
  # et celui de coût total S du pool
  #Initialisation de algorithme de Panjer pour le coût total S
  lambda_grp1=1-(cdfX1(ded))
  lambda_grp2=1-(cdfX2(ded))
  lambda_grp3=1-(cdfX3(ded))
  lambda_S=nb_grp1*lambda_grp1+nb_grp2*lambda_grp2+nb_grp3*lambda_grp3

  if(Ind_grpe==0){
    val_S=(1/lambda_S)*(nb_grp1*lambda_grp1*Fy(y,cdfX1,ded,Max)+
      nb_grp2*lambda_grp2*Fy(y,cdfX2,ded,Max)+
      nb_grp3*lambda_grp3*Fy(y,cdfX3,ded,Max))
    return(list(val=val_S, lambda=lambda_S,n=num))
  }
  #Fonction de répartition de S-Yi
  else if(Ind_grpe==1){
    val_grp1=(1/(lambda_S-lambda_grp1))*((nb_grp1-1)*lambda_grp1*Fy(y,cdfX1,ded,Max))+
      (1/(lambda_S-lambda_grp1))*nb_grp2*lambda_grp2*Fy(y,cdfX2,ded,Max)+
      (1/(lambda_S-lambda_grp1))*nb_grp3*lambda_grp3*Fy(y,cdfX3,ded,Max)
    return(list(val=val_grp1,lambda=lambda_grp1,n=nb_grp1))
  }
  else if(Ind_grpe==2){
    val_grp2=(1/(lambda_S-lambda_grp2))*((nb_grp1)*lambda_grp1*Fy(y,cdfX1,ded,Max))+
      (1/(lambda_S-lambda_grp2))*((nb_grp2-1)*lambda_grp2*Fy(y,cdfX2,ded,Max))+
```

```

        (1/(lambda_S-lambda_grp2))*(nb_grp3)*lambda_grp3*Fy(y, cdfX3, ded, Max)
return(list(val=val_grp2, lambda=lambda_grp2, n=nb_grp2))}
else {
    val_grp3=(1/(lambda_S-lambda_grp3))*((nb_grp1)*lambda_grp1*Fy(y, cdfX1, ded, Max))+
        (1/(lambda_S-lambda_grp3))*(nb_grp2)*lambda_grp2*Fy(y, cdfX2, ded, Max)+
        (1/(lambda_S-lambda_grp3))*(nb_grp3-1)*lambda_grp3*Fy(y, cdfX3, ded, Max)
    return(list(val=val_grp3, lambda=lambda_grp3, n=nb_grp3))}
#checking
sum(diff(FS(0:15000, 0, nb_ind, deduc, Maxi)$val))

```

Le checking nous permet de s'assurer que la somme des probabilités est bien égale à 1

```

timer=Sys.time()
pdfS=FS(0:15000, 0, nb_ind, deduc, Maxi, q)
cdfS=aggregateDist("recursive", model.freq = "poisson",
                    model.sev=diff(pdfS$val),
                    lambda=pdfS$lambda,
                    maxit = 250000)
cdfS=DiscreteDistribution(supp=knots(cdfS), prob=diff(cdfS))

lim=q(cdfS)(0.9999)
u=seq(0, lim, 1)
#Représentation de la fonction de répartition et la masse de probabilité
plot(u, p(cdfS)(u), ylab=TeX(r"${P}[S_{1000}] <= s]$" ), type="l", lty=1,
      main = "Aggregate Claim Amount Distr",
      sub = comment(x))

plot(u, d(cdfS)(u), xlab="s", ylab=TeX(r"${P}[S_{1000}] = s]$" ), type="h")

Sys.time()-timer

```

A.3.4 Calcul de l'espérance conditionnelle

```

#checking
sum(diff(FS(0:15000, 1, nb_ind, deduc, Maxi, q)$val))

```

```

sum(diff(FS(0:15000, 2, nb_ind, deduc, Maxi, q)$val))

```

```

sum(diff(FS(0:15000, 3, nb_ind, deduc, Maxi, q)$val))

```

```

pdfS_Y1=FS(0:15000, 1, nb_ind, deduc, Maxi, q)

```

```

pdfS_Y2=FS(0:15000, 2, nb_ind, deduc, Maxi, q)

```

```

pdfS_Y3=FS(0:15000, 3, nb_ind, deduc, Maxi, q)

```

```

timer=Sys.time()

```

```

#Calcul de la fonction de répartition de S-Yi

```

```

#Fonction de répartition de S-Y1 en utilisant la méthode recursive de

```

```

# AggregateDist

```



```

cdfS_Y1=aggregateDist("recursive", model.freq ="poisson",
                      model.sev=diff(pdfS_Y1$val),
                      lambda=(pdfS$lambda-pdfS_Y1$lambda),maxit =1000000 )
cdfS_Y1=DiscreteDistribution(supp=knots(cdfS_Y1),prob=diff(cdfS_Y1))
#Déterminons la perte totale maximale de S-Y1
lim_S_Y1=q(cdfS_Y1)(0.9999)
u1=seq(0,lim_S_Y1,1)
#Fonction de répartition de S-Y2 en utilisant la méthode recursive de
# AggregateDist

cdfS_Y2=aggregateDist("recursive", model.freq ="poisson",
                      model.sev=diff(pdfS_Y2$val),
                      lambda=(pdfS$lambda-pdfS_Y2$lambda),maxit =1000000)
cdfS_Y2=DiscreteDistribution(supp=knots(cdfS_Y2),prob=diff(cdfS_Y2))
#Déterminons la perte totale maximale de Y2
lim_S_Y2=q(cdfS_Y2)(0.9999)
u2=seq(0,lim_S_Y2,1)
#Fonction de répartition de S-Y3 en utilisant la méthode recursive de
# AggregateDist

cdfS_Y3=aggregateDist("recursive", model.freq ="poisson",
                      model.sev=diff(pdfS_Y3$val),
                      lambda=(pdfS$lambda-pdfS_Y3$lambda),maxit =1000000 )
cdfS_Y3=DiscreteDistribution(supp=knots(cdfS_Y3),prob=diff(cdfS_Y3))
#Déterminons la perte totale maximale de Y3

lim_S_Y3=q(cdfS_Y3)(0.9999)
u3=seq(0,lim_S_Y3,1)
Sys.time()-timer

```

Représentation de la distribution de $S_n - Y_i$

```

#CDF et PDF des S-Yi pour les groupes 1,2,3
# par(mfrow=c(1,2))
##CDF et PDF des S-Y1 du groupe 1
plot(u1,d(cdfS_Y1)(u1),ylab=TeX(r"${F}_{S-Y1}(s)"}),type="l")

plot(u1,p(cdfS_Y1)(u1),xlab="x",
     ylab=TeX(r"${P}[S_{1} = s]"}), type="l")

# par(mfrow=c(1,2))
##CDF et PDF des S-Y2 du groupe 2
plot(u2,d(cdfS_Y2)(u2),ylab=TeX(r"${F}_{S-Y1}(s)"}),type="l")

plot(u2,p(cdfS_Y2)(u2),xlab="x",
     ylab=TeX(r"${P}[S_{1} = s]"}), type="l")

```

```
# par(mfrow=c(1,2))
##CDF et PDF des S-Y3 du groupe 3
plot(u3,d(cdfS_Y3)(u3),ylab=TeX(r"${F}_{S-Y1}(s)$"),type="l")

plot(u3,p(cdfS_Y3)(u3),xlab="x",
      ylab=TeX(r"${P}[S_{1} = s]$"), type="l")
```

Calcul direct de S

Le calcul de manière exacte se fait en utilisant la Binomiale, et utilisant le package Discrete-Distribution {distr} permettant d'effectuer les calculs de manière exacte.

```
#Calcul des probabilités  $Y_i > 0$  des réclamations des groupes 1, 2 et 3
bino.cdfY1=data.frame(valeurs=u1,cdf=Fy(u1,cdfX1,deduc,Maxi))
bino.cdfY2=data.frame(valeurs=u2,cdf=Fy(u2,cdfX2,deduc,Maxi))
bino.cdfY3=data.frame(valeurs=u3,cdf=Fy(u3,cdfX3,deduc,Maxi))
#Calcul du coût total des groupes 1,2 et 3 en utilisation la méthode recursive
#avec la binomiale
```

Groupe 1

```
#checking
sum(diff(bino.cdfY1$cdf))

bino.cdfSY1=aggregateDist("recursive", model.freq ="binomial",
                          model.sev=diff(bino.cdfY1$cdf),
                          prob=pdfS_Y1$lambda,
                          size= pdfS_Y1$n,
                          maxit =1000000)
summary(bino.cdfSY1)
```

Groupe2

```
#checking
sum(diff(bino.cdfY2$cdf))

bino.cdfSY2=aggregateDist("recursive", model.freq ="binomial",
                          model.sev=diff(bino.cdfY2$cdf),
                          prob=pdfS_Y2$lambda,
                          size= pdfS_Y2$n,
                          maxit =1000000)
summary(bino.cdfSY2)
```

Groupe 3

```
bino.cdfSY3=aggregateDist("recursive", model.freq ="binomial",
                          model.sev=diff(bino.cdfY3$cdf),
                          prob=pdfS_Y3$lambda,
                          size= pdfS_Y3$n,
                          maxit =1000000)
```

```
summary(bino.cdfSY3)
```

Construction des objets de classe "DiscreteDistribution"

```
time=Sys.time()
max1=quantile(bino.cdfSY1,1)
u1=seq(0,max1,1)
bino.pdfS1=list(valeurs=u1,pdf=diff(bino.cdfSY1(u1)))
D1=DiscreteDistribution(supp=u1[1:max1], prob = bino.pdfS1$pdf)

max2=quantile(bino.cdfSY2,1)
u2=seq(0,max2,1)
bino.pdfS2=list(valeurs=u2,pdf=diff(bino.cdfSY2(u2)))
D2=DiscreteDistribution(supp =u2[1:max2], prob = bino.pdfS2$pdf)
max3=quantile(bino.cdfSY3,1)
u3=seq(0,max3,1)
bino.pdfS3=list(valeurs=u3,pdf=diff(bino.cdfSY3(u3)))
D3=DiscreteDistribution(supp=u3[1:max3], prob = bino.pdfS3$pdf)
S=D1+D2+D3

Sys.time()-time
```

```
plf<-distr::q(S)(0.9999)

u<-seq(0,plf,1)
Représentation
plot(u,d(S)(u),type="l",xlab="x",ylab="d(x)", main="Masse de probabilité",
     col="darkred")

lines(u,d(cdfS)(u),lty=2,col="black")
legend("topleft",inset=0.05,cex = 0.6,
legend =c("Calcul Exact","Approximation Collective"),
     col=c("darkred","black"),lty=c(1,4),box.lty = 0)
```

Calcul de l'espérance conditionnelle

Cette fonction nous permet de calculer les contributions

$$h_{in}(S_n) = E[Y_i | S_n = s]$$

Cette fonction prend en entrée la réalisation s de S_n , la franchise $deduc = d$, les fonctions de répartition des différents sous groupes ($Groupe_1$; $Groupe_2$; $Groupe_3$)

```
# Calcul de l'espérance conditionnelle de Y1; Y2 et Y3 sachant que S=s
# s: réalisation de S
# FY1: Fonction de répartition de S-Y_1
# FY2: Fonction de répartition de S-Y_2
# FY3: Fonction de répartition de S-Y_3
# Retourne E[Y1|S=s] E[Y2|S=s] et E[Y3|S=s]
```

```

contri<-function(s,deduc,Maxi,Fs,FY1,FY2,FY3){
  if(s==0){
    partY1=0
    partY2=0
    partY3=0
  }
  return(list(c1=0,c2=0,c3=0))
} else{
  ## Fs: Fonction de repartition de S
  probs=p(Fs)(s)-p(Fs)(s-1) #Pr[S=s]
  ## Groupe1
  # Pr[Yi=k] k allant de 1 à s
  prob1=Gy(1:s,cdfX1,deduc,Maxi)-Gy(1:s-1,cdfX1,deduc,Maxi)

  prob2=p(FY1)(s-1:s)-p(FY1)(s-1:s-1) # Pr[S-Yi=s-k]
  partY1=sum((1:s)*prob1*prob2/probs) # E[Yi|S=s]

  # Groupe2
  prob3=Gy(1:s,cdfX2,deduc,Maxi)-Gy(1:s-1,cdfX2,deduc,Maxi)
  prob4=p(FY2)(s-1:s)-p(FY2)(s-1:s-1)
  partY2=sum((1:s)*prob3*prob4/probs)

  # Groupe 3
  prob5=Gy(1:s,cdfX3,deduc,Maxi)-Gy(1:s-1,cdfX3,deduc,Maxi)
  prob6=p(FY3)(s-1:s)-p(FY3)(s-1:s-1)
  partY3=sum((1:s)*prob5*prob6/probs)
}
return(list(c1=partY1,c2=partY2,c3=partY3))
}

timer=Sys.time()
s=seq(0,q(cdfS)(0.9999),10000)
result=data.frame(s=s,CMRS1=rep(NA,length(s)),CMRS2=rep(NA,length(s)),
                  CMRS3=rep(NA,length(s)),Total=rep(NA,length(s)))
for (i in 1:length(s)) {
  CE= contri(result$s[i],deduc,Maxi,cdfS,cdfS_Y1,cdfS_Y2,cdfS_Y3)
  result$CMRS1[i]= CE$c1
  result$CMRS2[i]= CE$c2
  result$CMRS3[i]= CE$c3
  result$Total[i]= sum(pdfS_Y1$n*CE$c1,pdfS_Y2$n*CE$c2,pdfS_Y3$n*CE$c3)
}
Sys.time()-timer

```

Représentation des différentes contributions des groupes

```

plot(s,result$CMRS3,col="green3",type="l",lty=2,ylab="Contribution par groupe"
,xlab = "Valeurs de S" )

```

```

lines(s,result$CMRS1,type="l",lty=1,col="blue3")
lines(s,result$CMRS2,col="brown4",lty=4)
# lines(u[1:plf],diff(cdfS(u)),lty=2,col="black")
legend("topleft",inset=0.05,legend=c("groupe 1","groupe 2","groupe 3"),
      col = c("blue3","brown4","green3"),lty=1:2,box.lty = 0)

```

```

timer=Sys.time()
s=seq(0,q(cdfS)(0.9999),1000)

result=data.frame(s=s,CMRS1=rep(NA,length(s)),CMRS2=rep(NA,length(s)),
                  CMRS3=rep(NA,length(s)),Total=rep(NA,length(s)))
for (i in 1:length(s)) {
  CE= contri(result$s[i],deduc,Maxi,cdfS,cdfS_Y1,cdfS_Y2,cdfS_Y3)
  result$CMRS1[i]= CE$c1
  result$CMRS2[i]= CE$c2
  result$CMRS3[i]= CE$c3
  result$Total[i]= sum(pdfS_Y1$n*CE$c1,pdfS_Y2$n*CE$c2,pdfS_Y3$n*CE$c3)
}
Sys.time()-timer

```

```

plot(s,result$CMRS3,col="green3",type="l",lty=2,ylab="Contribution par groupe",
     ,xlab = "Valeurs de S",ylim=c(0,130) )
lines(s,result$CMRS1,type="l",lty=1,col="blue3")
lines(s,result$CMRS2,col="brown4",lty=4)
lines(u,d(cdfS)(u)*1000000,type="l",col="black")
legend("topleft",inset=0.05,legend=c("groupe 1","groupe 2","groupe 3"),
      col = c("blue3","brown4","green3"),lty=1:2,box.lty = 0)

```

```

plot(s,s,type="l",col="blue",xlim=c(0,40000))
lines(s,result$Total,type = "l",lty=1,col="red")

```

A.4 Étude de la sensibilité des contributions : Taille ; Franchise ; Plafond

A.4.1 Sensibilité à la taille du pool

Point de départ, Franchise=250, Plafond de réclamation=8500, Taille du pool=800.

Ensuite, faisons varier la taille du pool et analysons l'effet de cette variation sur les contributions des membres.

```

timer=Sys.time()
n=c(800,1600,2000,2500,3000,3500)
cdf.S_n=list()
cdf.S_Y1=list()
cdf.S_Y2=list()
cdf.S_Y3=list()

```

```

quant.S_n=rep(NA,length(n))
for (i in 1:length(n)){
  U=seq(0,15000,1)

##S_n
  fS_n= FS(0:15000,0,n[i],deduc,Maxi,q)
FS_n=aggregateDist("recursive", model.freq="poisson",
                   model.sev=diff(fS_n$val),
                   lambda=fS_n$lambda,
                   maxit = 250000)
FS_n=DiscreteDistribution(supp = knots(FS_n),prob = diff(FS_n))

##groupe1: S-Y1
  fs_y1= FS(0:15000,1,n[i],deduc,Maxi,q)
Fs_y1=aggregateDist("recursive", model.freq="poisson",
                   model.sev=diff( fs_y1$val),
                   lambda=(fS_n$lambda-fs_y1$lambda),maxit =1000000 )
Fs_y1=DiscreteDistribution(supp = knots(Fs_y1),prob = diff(Fs_y1))

##groupe2: S-Y2
  fs_y2= FS(0:15000,2,n[i],deduc,Maxi,q)
Fs_y2=aggregateDist("recursive", model.freq="poisson",
                   model.sev=diff(fs_y2$val),
                   lambda=(fS_n$lambda-fs_y2$lambda),maxit =1000000)
Fs_y2=DiscreteDistribution(supp = knots(Fs_y2),prob = diff(Fs_y2))

##groupe3: S-Y3
  fs_y3= FS(0:15000,3,n[i],deduc,Maxi,q)
Fs_y3=aggregateDist("recursive", model.freq="poisson",
                   model.sev=diff(fs_y3$val),
                   lambda=(fS_n$lambda-fs_y3$lambda),maxit =1000000 )
Fs_y3=DiscreteDistribution(supp = knots(Fs_y3),prob = diff(Fs_y3))

  quant.S_n[i]=min(q(FS_n)(0.9999),q(Fs_y1)(0.9999),
                  q(Fs_y1)(0.9999),q(Fs_y1)(0.9999))

  cdf.S_n[[length(cdf.S_n)+1]]=FS_n
  cdf.S_Y1[[length(cdf.S_Y1)+1]]=Fs_y1
  cdf.S_Y2[[length(cdf.S_Y2)+1]]=Fs_y2
  cdf.S_Y3[[length(cdf.S_Y3)+1]]=Fs_y3
  rm(fS_n,FS_n,Fs_y1,fs_y1,Fs_y2,fs_y2,Fs_y3,fs_y3)
}
print(Sys.time()-timer)

```

```

names(cdf.S_n)=paste("N",as.character(n),sep="")
names(cdf.S_Y1)=paste("N",as.character(n),sep="")

```

```
names(cdf.S_Y2)=paste("N",as.character(n),sep="")
names(cdf.S_Y3)=paste("N",as.character(n),sep="")
```

Contribution en fonction du nombre de participants

```
n=c(800,1600,2000,2500,3000,3500)
cmrs.grpe1=list()
cmrs.grpe2=list()
cmrs.grpe3=list()

timer=Sys.time()
  for (i in 1:length(n)) {
S=seq(0,quant.S_n[i],1000)
S.quant=S/quant.S_n[i]
esp.grpe1=data.frame(S=S,S.quant=S.quant,exp=rep(NA,length(S)))
esp.grpe2=data.frame(S=S,S.quant=S.quant,exp=rep(NA,length(S)))
esp.grpe3=data.frame(S=S,S.quant=S.quant,exp=rep(NA,length(S)))

for (j in 1:length(S)) {

  #Conditional mean risk sharing
  A=contri(s=S[j],deduc,Maxi,Fs=cdf.S_n[[i]],FY1=cdf.S_Y1[[i]],
FY2=cdf.S_Y2[[i]],FY3=cdf.S_Y3[[i]])
  esp.grpe1$exp[j]=A$c1
  esp.grpe2$exp[j]=A$c2
  esp.grpe3$exp[j]=A$c3
}
cmrs.grpe1[[length(cmrs.grpe1)+1]]=esp.grpe1
cmrs.grpe2[[length(cmrs.grpe2)+1]]=esp.grpe2
cmrs.grpe3[[length(cmrs.grpe3)+1]]=esp.grpe3
rm(S,A,esp.grpe1,esp.grpe2,esp.grpe3)  }

print(Sys.time()-timer)
```

```
names(cmrs.grpe1)=paste("N",as.character(n),sep="")
names(cmrs.grpe2)=paste("N",as.character(n),sep="")
names(cmrs.grpe3)=paste("N",as.character(n),sep="")
```

Loi de probabilité des contributions

```
timer=Sys.time()
  n=c(800,1600,2000,2500,3000,3500)
  l=1500
  loi.N=list()
  for (i in 1:length(n)) {
    S=r(cdf.S_n[[i]])(l)
    esp=data.frame(S=S,exp.grpe1=rep(NA,l),exp.grpe2=rep(NA,l),
```

```

        exp.grpe3=rep(NA,1),prob=distr::p(cdf.S_n[[i]])(S))
for (j in 1:length(S)) {
  A=contri(s=S[j],deduc,Maxi,Fs=cdf.S_n[[i]],FY1=cdf.S_Y1[[i]],
          FY2=cdf.S_Y2[[i]],FY3=cdf.S_Y3[[i]])
  esp$exp.grpe1[j]=A$c1
  esp$exp.grpe2[j]=A$c2
  esp$exp.grpe3[j]=A$c3  }
loi.N[[length(loi.N)+1]]=esp
rm(S,esp,A)  }
print(Sys.time()-timer)

names(loi.N)=paste("N",as.character(n),sep = "")

pdf.grpe1=list()
pdf.grpe2=list()
pdf.grpe3=list()
  for (i in 1:length(n)) {
    density.grpe1=density(loi.N[[i]]$exp.grpe1)
    density.grpe2=density(loi.N[[i]]$exp.grpe2)
    density.grpe3=density(loi.N[[i]]$exp.grpe3)
    valeur1=data.frame(val=density.grpe1$x, prob=density.grpe1$y)
    valeur2=data.frame(val=density.grpe2$x, prob=density.grpe2$y)
    valeur3=data.frame(val=density.grpe3$x, prob=density.grpe3$y)
    pdf.grpe1[[length(pdf.grpe1)+1]]=valeur1
    pdf.grpe2[[length(pdf.grpe2)+1]]=valeur2
    pdf.grpe3[[length(pdf.grpe3)+1]]=valeur3
  }
rm(density.grpe1, density.grpe2, density.grpe3, valeur1, valeur2, valeur3 )}
names(pdf.grpe1)=paste("Taille", as.character(n),sep="")
names(pdf.grpe2)=paste("Taille", as.character(n),sep="")
names(pdf.grpe3)=paste("Taille", as.character(n),sep="")

# groupe 1
par(mfrow=c(1,3),font.main=1,cex.main=0.9,cex.sub=0.8)
plot(pdf.grpe1[[6]]$val, pdf.grpe1[[6]]$prob, type="l",lty=3,
     col="#440154",
xlab=list(TeX('$E[Y_i|S_n]$')),ylab="density",main="Groupe 1")
lines(pdf.grpe1[[5]]$val, pdf.grpe1[[5]]$prob, type="l",lty=3,
     col="yellow3",
xlab=list(TeX('$E[Y_i|S_n]$')),ylab="density",main="Groupe 1")
lines(pdf.grpe1[[1]]$val, pdf.grpe1[[1]]$prob, type="l",lty=1,
     col="blue3",
xlab=list(TeX('$E[Y_i|S_n]$')),ylab="density",main="Groupe 1")
lines(pdf.grpe1[[2]]$val, pdf.grpe1[[2]]$prob, type="l",lty=2,
     col="green3",
xlab=list(TeX('$E[Y_i|S_n]$')),ylab="density",main="Groupe 1")
lines(pdf.grpe1[[3]]$val, pdf.grpe1[[3]]$prob, type="l",lty=3,

```



```

    col="red3",
xlab=list(TeX('$E[Y_i|S_n]$')),ylab="density",main="Groupe 1")
lines(pdf.grpe1[[4]]$val, pdf.grpe1[[4]]$prob, type="l",lty=3,
      col="#EB8744",
xlab=list(TeX('$E[Y_i|S_n]$')),ylab="density",main="Groupe 1")
legend("topright",cex=0.6,legend = c("n=800","n=1600","n=2000","n=2500",
"n=3000","n=3500"),col=c("blue3","green3","red3","#EB8744","yellow3","#440154"),
lty=c(1,2,3))

# groupe 2
#par(mfrow=c(1,3),font.main=1,cex.main=0.9,cex.sub=0.8)
plot(pdf.grpe2[[6]]$val, pdf.grpe2[[6]]$prob, type="l",lty=3,
     col="#440154", xlab=list(TeX('$E[Y_i|S_n]$')),ylab="density",main="Groupe 2")
lines(pdf.grpe2[[5]]$val, pdf.grpe2[[5]]$prob, type="l",lty=3,
      col="yellow3", xlab=list(TeX('$E[Y_i|S_n]$')),ylab="density",main="Groupe 2")
lines(pdf.grpe2[[1]]$val, pdf.grpe2[[1]]$prob, type="l",lty=1,
      col="blue3", xlab=list(TeX('$E[Y_i|S_n]$')),ylab="density",main="Groupe 2")
lines(pdf.grpe2[[2]]$val, pdf.grpe2[[2]]$prob, type="l",lty=2,
      col="green3", xlab=list(TeX('$E[Y_i|S_n]$')),ylab="density",main="Groupe 2")
lines(pdf.grpe2[[3]]$val, pdf.grpe2[[3]]$prob, type="l",lty=3,
      col="red3", xlab=list(TeX('$E[Y_i|S_n]$')),ylab="density",main="Groupe 2")
lines(pdf.grpe2[[4]]$val, pdf.grpe2[[4]]$prob, type="l",lty=3,
      col="#EB8744", xlab=list(TeX('$E[Y_i|S_n]$')),ylab="density",main="Groupe 2")
legend("topright",cex=0.6,legend = c("n=800","n=1600","n=2000",
"n=2500","n=3000","n=3500"), col=c("blue3","green3","red3","#EB8744","yellow3","#440154"),
lty=c(1,2,3))

# groupe 3
#par(mfrow=c(1,3),font.main=1,cex.main=0.9,cex.sub=0.8)
plot(pdf.grpe3[[6]]$val, pdf.grpe3[[6]]$prob, type="l",lty=3, col="#440154",xlab=list(TeX('$E[Y_i|S_n]$'))
lines (pdf.grpe3[[5]]$val, pdf.grpe3[[5]]$prob, type="l",lty=3,
      col="yellow3", xlab=list(TeX('$E[Y_i|S_n]$')),ylab="density",main="Groupe 3")
lines(pdf.grpe3[[1]]$val, pdf.grpe3[[1]]$prob, type="l",lty=1,
      col="blue3", xlab=list(TeX('$E[Y_i|S_n]$')),ylab="density",main="Groupe 3")
lines(pdf.grpe3[[2]]$val, pdf.grpe3[[2]]$prob, type="l",lty=2,
      col="green3", xlab=list(TeX('$E[Y_i|S_n]$')),ylab="density",main="Groupe 3")
lines(pdf.grpe3[[3]]$val, pdf.grpe3[[3]]$prob, type="l",lty=3,
      col="red3", xlab=list(TeX('$E[Y_i|S_n]$')),ylab="density",main="Groupe 3")
lines(pdf.grpe3[[4]]$val, pdf.grpe3[[4]]$prob, type="l",lty=3,
      col="#EB8744", xlab=list(TeX('$E[Y_i|S_n]$')),ylab="density",main="Groupe 3")
legend("topright",cex=0.6,legend = c("n=800","n=1600","n=2000",
"n=2500","n=3000","n=3500"),
      col=c("blue3","green3","red3","#EB8744","yellow3","#440154"),lty=c(1,2,3))

```

Boxplot

```

Box.plot.grpe1=list()
Box.plot.grpe2=list()

```

```

Box.plot.grpe3=list()

for (i in 1:length(loi.N)) {
  Box.plot.grpe1[[length(Box.plot.grpe1)+1]]=loi.N[[i]]$exp.grpe1
  Box.plot.grpe2[[length(Box.plot.grpe2)+1]]=loi.N[[i]]$exp.grpe2
  Box.plot.grpe3[[length(Box.plot.grpe3)+1]]=loi.N[[i]]$exp.grpe3}

names(Box.plot.grpe1)=as.character(n)
names(Box.plot.grpe2)=as.character(n)
names(Box.plot.grpe3)=as.character(n)

Box.plotgrpe1.long1=melt(do.call(rbind,Box.plot.grpe1))
Box.plotgrpe2.long2=melt(do.call(rbind,Box.plot.grpe2))
Box.plotgrpe3.long3=melt(do.call(rbind,Box.plot.grpe3))
par(mfrow=c(1,3),font.main=1,cex.main = 0.9, cex.sub = 0.8)
boxplot(value~Var1,Box.plotgrpe1.long1,xlab ="n",ylab = TeX('$E[Y_i|S_n=s]$'),
main="Groupe 1",outline=FALSE)
  abline(h=E(gpe1),lwd = 1.5,lty=4,col="red")
boxplot(value~Var1,Box.plotgrpe2.long2,xlab ="n",ylab = TeX('$E[Y_i|S_n=s]$'),
main="Groupe 2",outline=FALSE)
  abline(h=E(gpe2),lwd = 1.5,lty=4,col="red")
boxplot(value~Var1,Box.plotgrpe3.long3,xlab ="n",ylab = TeX('$E[Y_i|S_n=s]$'),
main="Groupe 3",outline=FALSE)
  abline(h=E(gpe3),lwd = 1.5,lty=4,col="red")

```

A.4.2 Sensibilité au plafond de réclamation

```

timer=Sys.time()
n=2000
deduc=250
#Maximum de relamation applicable à tous les participants
Maxi=c(8500,10000,15000,20000,25000)
timer=Sys.time()
cdf.M.Sn=list()
cdf.M.SY1=list()
cdf.M.SY2=list()
cdf.M.SY3=list()
quant.M.S_n=rep(NA,length(Maxi))
for (i in 1:length(Maxi)){
  U=seq(0,15000,1)
##S_n
  fS_n= FS(0:15000,0,n,deduc,Maxi[i],q)
FS_n=aggregateDist("recursive", model.freq ="poisson",
                   model.sev=diff(fS_n$val),
                   lambda=fS_n$lambda,

```

```

                                maxit = 250000)
FS_n=DiscreteDistribution(supp = knots(FS_n),prob = diff(FS_n))
##groupe1: S-Y1
  fs_y1= FS(0:15000,1,n,deduc,Maxi[i],q)
Fs_y1=aggregateDist("recursive", model.freq = "poisson",
                    model.sev=diff( fs_y1$val),
                    lambda=(fS_n$lambda-fs_y1$lambda),maxit =1000000 )
Fs_y1=DiscreteDistribution(supp = knots(Fs_y1),prob = diff(Fs_y1))
##groupe2: S-Y2
  fs_y2= FS(0:15000,2,n,deduc,Maxi[i],q)
Fs_y2=aggregateDist("recursive", model.freq = "poisson",
                    model.sev=diff(fs_y2$val),
                    lambda=(fS_n$lambda-fs_y2$lambda),maxit =1000000)
Fs_y2=DiscreteDistribution(supp = knots(Fs_y2),prob = diff(Fs_y2))
##groupe3: S-Y3
  fs_y3= FS(0:15000,3,n,deduc,Maxi[i],q)
Fs_y3=aggregateDist("recursive", model.freq = "poisson",
                    model.sev=diff(fs_y3$val),
                    lambda=(fS_n$lambda-fs_y3$lambda),maxit =1000000 )
Fs_y3=DiscreteDistribution(supp = knots(Fs_y3),prob = diff(Fs_y3))
  quant.M.S_n[i]=min(q(FS_n)(0.9999),q(Fs_y1)(0.9999),
                    q(Fs_y1)(0.9999),q(Fs_y1)(0.9999))
  cdf.M.Sn[[length(cdf.M.Sn)+1]]=FS_n
  cdf.M.SY1[[length(cdf.M.SY1)+1]]=Fs_y1
  cdf.M.SY2[[length(cdf.M.SY2)+1]]=Fs_y2
  cdf.M.SY3[[length(cdf.M.SY3)+1]]=Fs_y3
  rm(fS_n,FS_n,Fs_y1,fs_y1,Fs_y2,fs_y2,Fs_y3,fs_y3)
}
print(Sys.time()-timer)

```

```
## Time difference of 2.615033 mins
```

```

names(cdf.M.Sn)=paste("N",as.character(Maxi) ,sep="")
names(cdf.M.SY1)=paste("N",as.character(Maxi),sep="")
names(cdf.M.SY2)=paste("N",as.character(Maxi),sep="")
names(cdf.M.SY3)=paste("N",as.character(Maxi),sep="")

```

Contribution en fonction du maximum de réclamation

```

Maxi=c(8500,10000,15000,20000,25000)
cmrs.M.grpe1=list()
cmrs.M.grpe2=list()
cmrs.M.grpe3=list()
timer=Sys.time()
  for (i in 1:length(Maxi)) {
S=seq(0,quant.M.S_n[i],1000)
S.quant.M=S/quant.M.S_n[i]

```

```

espM.grpe1=data.frame(S=S,S.quant=S.quant.M,exp=rep(NA,length(S)))
espM.grpe2=data.frame(S=S,S.quant=S.quant.M,exp=rep(NA,length(S)))
espM.grpe3=data.frame(S=S,S.quant=S.quant.M,exp=rep(NA,length(S)))

for (j in 1:length(S)) {

  #Conditional mean risk sharing
  A=contri(s=S[j],deduc,Maxi[i],Fs=cdf.S_n[[i]],FY1=cdf.S_Y1[[i]],
FY2=cdf.S_Y2[[i]],FY3=cdf.S_Y3[[i]])
  espM.grpe1$exp[j]=A$c1
  espM.grpe2$exp[j]=A$c2
  espM.grpe3$exp[j]=A$c3}
cmrs.M.grpe1[[length(cmrs.M.grpe1)+1]]=espM.grpe1
cmrs.M.grpe2[[length(cmrs.M.grpe2)+1]]=espM.grpe2
cmrs.M.grpe3[[length(cmrs.M.grpe3)+1]]=espM.grpe3
rm(S,A,espM.grpe1,espM.grpe2,espM.grpe3) }

print(Sys.time()-timer)

```

```

names(cmrs.M.grpe1)=paste("Maxi",as.character(Maxi),sep="")
names(cmrs.M.grpe2)=paste("Maxi",as.character(Maxi),sep="")
names(cmrs.M.grpe3)=paste("Maxi",as.character(Maxi),sep="")

```

Loi de probabilité des contributions

```

timer=Sys.time()
Maxi=c(8500,10000,15000,20000,25000)
l=1500
loi.M=list()
for (i in 1:length(Maxi)) {
  S=r(cdf.M.Sn[[i]])(l)
  esp=data.frame(S=S,expM.grpe1=rep(NA,l),expM.grpe2=rep(NA,l),
                expM.grpe3=rep(NA,l),prob=distr::p(cdf.M.Sn[[i]])(S))
  for (j in 1:length(S)) {
    A=contri(s=S[j],deduc,Maxi[i],Fs=cdf.M.Sn[[i]],FY1=cdf.M.SY1[[i]],
            FY2=cdf.M.SY2[[i]],FY3=cdf.M.SY3[[i]])
    esp$expM.grpe1[j]=A$c1
    esp$expM.grpe2[j]=A$c2
    esp$expM.grpe3[j]=A$c3}

  loi.M[[length(loi.M)+1]]=esp
  rm(S,esp,A)
}
print(Sys.time()-timer)

```

```
## Time difference of 13.01265 mins
```

```

names(loi.M)=paste("Maxi",as.character(Maxi),sep = "")

pdf.grpe1.maxi=list()
pdf.grpe2.maxi=list()
pdf.grpe3.maxi=list()
  for (i in 1:length(Maxi)) {
    density.grpe1.maxi=density(loi.M[[i]]$expM.grpe1)
    density.grpe2.maxi=density(loi.M[[i]]$expM.grpe2)
    density.grpe3.maxi=density(loi.M[[i]]$expM.grpe3)

    valeur1=data.frame(val=density.grpe1.maxi$x, prob=density.grpe1.maxi$y)
    valeur2=data.frame(val=density.grpe2.maxi$x, prob=density.grpe2.maxi$y)
    valeur3=data.frame(val=density.grpe3.maxi$x, prob=density.grpe3.maxi$y)

    pdf.grpe1.maxi[[length(pdf.grpe1.maxi)+1]]=valeur1
    pdf.grpe2.maxi[[length(pdf.grpe2.maxi)+1]]=valeur2
    pdf.grpe3.maxi[[length(pdf.grpe3.maxi)+1]]=valeur3
    rm(density.grpe1.maxi, density.grpe2.maxi, density.grpe3.maxi,
    valeur1, valeur2, valeur3)}

names(pdf.grpe1.maxi)=paste("Plafond", as.character(n),sep="")
names(pdf.grpe2.maxi)=paste("Plafond", as.character(n),sep="")
names(pdf.grpe3.maxi)=paste("Plafond", as.character(n),sep="")

# groupe 1
par(mfrow=c(1,3),font.main=1,cex.main=0.9,cex.sub=0.8)

plot(pdf.grpe1.maxi[[4]]$val, pdf.grpe1.maxi[[4]]$prob, type="l",lty=1,
col="blue3",
xlab=list(TeX('$E[Y_i|S_n]$')),ylab="density",main="Groupe 1")
lines(pdf.grpe1.maxi[[3]]$val, pdf.grpe1.maxi[[3]]$prob, type="l",lty=2,
col="green3",
xlab=list(TeX('$E[Y_i|S_n]$')),ylab="density",main="Groupe 1")
lines(pdf.grpe1.maxi[[5]]$val, pdf.grpe1.maxi[[5]]$prob, type="l",lty=3,
col="yellow3",
xlab=list(TeX('$E[Y_i|S_n]$')),ylab="density",main="Groupe 1")

lines(pdf.grpe1.maxi[[4]]$val, pdf.grpe1.maxi[[4]]$prob, type="l",lty=1,
col="blue3",
xlab=list(TeX('$E[Y_i|S_n]$')),ylab="density",main="Groupe 1")

lines(pdf.grpe1.maxi[[2]]$val, pdf.grpe1.maxi[[2]]$prob, type="l",lty=3,
col="red3",
xlab=list(TeX('$E[Y_i|S_n]$')),ylab="density",main="Groupe 1")

lines(pdf.grpe1.maxi[[1]]$val, pdf.grpe1.maxi[[1]]$prob, type="l",lty=3,
col="#EB8744",

```

```

      xlab=list(TeX('$E[Y_i|S_n]$')),ylab="density",main="Groupe 1")

legend("topright",cex=0.6,legend = c("M=8500",
"M=10000", "M=15000", "M=20000", "M=25000"),
      col=c("blue3", "green3", "red3", "#EB8744", "yellow3"),lty=c(1,2,3))

# groupe 2
  #par(mfrow=c(1,3),font.main=1,cex.main=0.9,cex.sub=0.8)
plot(pdf.grpe2.maxi[[2]]$val, pdf.grpe2.maxi[[2]]$prob, type="l",lty=2,
     col="green3",
     xlab=list(TeX('$E[Y_i|S_n]$')),ylab="density",main="Groupe 2")

lines(pdf.grpe2.maxi[[5]]$val, pdf.grpe2.maxi[[5]]$prob, type="l",lty=3,
      col="yellow3", xlab=list(TeX('$E[Y_i|S_n]$')),ylab="density",main="Groupe 2")

lines(pdf.grpe2.maxi[[1]]$val, pdf.grpe2.maxi[[1]]$prob, type="l",lty=1,
      col="blue3",
      xlab=list(TeX('$E[Y_i|S_n]$')),ylab="density",main="Groupe 2")

lines(pdf.grpe2.maxi[[3]]$val, pdf.grpe2.maxi[[3]]$prob, type="l",lty=3,
      col="red3",
      xlab=list(TeX('$E[Y_i|S_n]$')),ylab="density",main="Groupe 2")

lines(pdf.grpe2.maxi[[4]]$val, pdf.grpe2.maxi[[4]]$prob, type="l",lty=3,
      col="#EB8744",
      xlab=list(TeX('$E[Y_i|S_n]$')),ylab="density",main="Groupe 2")

legend("topright",cex=0.6,legend = c("M=8500",
"M=10000", "M=15000", "M=20000", "M=25000"),
      col=c("blue3", "green3", "red3", "#EB8744", "yellow3"),lty=c(1,2,3))

# groupe 3
  #par(mfrow=c(1,3),font.main=1,cex.main=0.9,cex.sub=0.8)

plot(pdf.grpe3.maxi[[2]]$val, pdf.grpe3.maxi[[2]]$prob, type="l",lty=2,
     col="green3",
     xlab=list(TeX('$E[Y_i|S_n]$')),ylab="density",main="Groupe 3")

lines(pdf.grpe3.maxi[[5]]$val, pdf.grpe3.maxi[[5]]$prob, type="l",lty=3,
      col="yellow3", xlab=list(TeX('$E[Y_i|S_n]$')),ylab="density",main="Groupe 3")

lines(pdf.grpe3.maxi[[1]]$val, pdf.grpe3.maxi[[1]]$prob, type="l",lty=1,
      col="blue3",
      xlab=list(TeX('$E[Y_i|S_n]$')),ylab="density",main="Groupe 3")

lines(pdf.grpe3.maxi[[3]]$val, pdf.grpe3.maxi[[3]]$prob, type="l",lty=3,

```

```

    col="red3",
    xlab=list(TeX('$E[Y_i|S_n]$')),ylab="density",main="Groupe 3")

lines(pdf.grpe3.maxi[[4]]$val, pdf.grpe3.maxi[[4]]$prob, type="l",lty=3,
      col="#EB8744",
      xlab=list(TeX('$E[Y_i|S_n]$')),ylab="density",main="Groupe 3")

legend("topright",cex=0.6,legend = c("M=8500",
"M=10000", "M=15000", "M=20000", "M=25000"),
      col=c("blue3", "green3", "red3", "#EB8744", "yellow3"),lty=c(1,2,3))

maxi.plot.grpe1<-list()
maxi.plot.grpe2<-list()
maxi.plot.grpe3<-list()
for (i in 1:length(Maxi)) {
  maxi.plot.grpe1[[length(maxi.plot.grpe1)+1]]<-loi.M[[i]]$expM.grpe1
  maxi.plot.grpe2[[length(maxi.plot.grpe2)+1]]<-loi.M[[i]]$expM.grpe2
  maxi.plot.grpe3[[length(maxi.plot.grpe3)+1]]<-loi.M[[i]]$expM.grpe3
}
names(maxi.plot.grpe1)<-as.character(Maxi)
names(maxi.plot.grpe2)<-as.character(Maxi)
names(maxi.plot.grpe3)<-as.character(Maxi)

maxi.plotgrpe1.long1<-melt(do.call(rbind,maxi.plot.grpe1))
maxi.plotgrpe2.long2<-melt(do.call(rbind,maxi.plot.grpe2))
maxi.plotgrpe3.long3<-melt(do.call(rbind,maxi.plot.grpe3))

par(mfrow=c(1,3),font.main=1,cex.main = 0.9, cex.sub = 0.8)

boxplot(value~Var1,maxi.plotgrpe1.long1,xlab ="Plafond",ylab = "",main="Groupe 1",
outline=FALSE)
abline(h=E(gpe1),lwd = 1.5,lty=4,col="red")

boxplot(value~Var1,maxi.plotgrpe2.long2,xlab ="Plafond",ylab = "",main="Groupe 2",
outline=FALSE)
abline(h=E(gpe2),lwd = 1.5,lty=4,col="red")

boxplot(value~Var1,maxi.plotgrpe3.long3,xlab ="Plafond",ylab = "",main="Groupe 3",
outline=FALSE)
abline(h=E(gpe3),lwd = 1.5,lty=4,col="red")

```

Franchise globale

```

# tab.deduc=list()
n=2000
Maxi=8500
#franchise annuelle applicable à tous les participants

```

```

deduc=c(50,150,250,300,400)

timer=Sys.time()
cdf.D.Sn=list()
cdf.D.SY1=list()
cdf.D.SY2=list()
cdf.D.SY3=list()
quant.DS_n=rep(NA,length(deduc))

for (i in 1:length(deduc)){
  U=seq(0,15000,1)

  ##S_n
  fs_n= FS(0:15000,0,n,deduc[i],Maxi,q)
  FS_n=aggregateDist("recursive", model.freq = "poisson",
                    model.sev=diff(fs_n$val),
                    lambda=fs_n$lambda,
                    maxit = 250000)
  FS_n=DiscreteDistribution(supp = knots(FS_n),prob = diff(FS_n))

  ##groupe1: S-Y1
  fs_y1= FS(0:15000,1,n,deduc[i],Maxi,q)
  Fs_y1=aggregateDist("recursive", model.freq = "poisson",
                    model.sev=diff( fs_y1$val),
                    lambda=(fs_n$lambda-fs_y1$lambda),maxit =1000000 )
  Fs_y1=DiscreteDistribution(supp = knots(Fs_y1),prob = diff(Fs_y1))

  ##groupe2: S-Y2
  fs_y2= FS(0:15000,2,n,deduc[i],Maxi,q)
  Fs_y2=aggregateDist("recursive", model.freq = "poisson",
                    model.sev=diff(fs_y2$val),
                    lambda=(fs_n$lambda-fs_y2$lambda),maxit =1000000)
  Fs_y2=DiscreteDistribution(supp = knots(Fs_y2),prob = diff(Fs_y2))

  ##groupe3: S-Y3
  fs_y3= FS(0:15000,3,n,deduc[i],Maxi,q)
  Fs_y3=aggregateDist("recursive", model.freq = "poisson",
                    model.sev=diff(fs_y3$val),
                    lambda=(fs_n$lambda-fs_y3$lambda),maxit =1000000 )
  Fs_y3=DiscreteDistribution(supp = knots(Fs_y3),prob = diff(Fs_y3))

  quant.DS_n[i]=min(q(FS_n)(0.9999),q(Fs_y1)(0.9999),
                  q(Fs_y1)(0.9999),q(Fs_y1)(0.9999))

  cdf.D.Sn[[length(cdf.D.Sn)+1]]=FS_n
  cdf.D.SY1[[length(cdf.D.SY1)+1]]=Fs_y1

```



```

cdf.D.SY2[[length(cdf.D.SY2)+1]]=Fs_y2
cdf.D.SY3[[length(cdf.D.SY3)+1]]=Fs_y3
  rm(fs_n,FS_n,Fs_y1,fs_y1,Fs_y2,fs_y2,Fs_y3,fs_y3)
}
print(Sys.time()-timer)

```

```
## Time difference of 2.724257 mins
```

```

names(cdf.D.Sn)=paste("N",as.character(deduc),sep="")
names(cdf.D.SY1)=paste("N",as.character(deduc),sep="")
names(cdf.D.SY2)=paste("N",as.character(deduc),sep="")
names(cdf.D.SY3)=paste("N",as.character(deduc),sep="")

```

Contributions en fonction de la franchise

```

cmrs.D.grpe1=list()
cmrs.D.grpe2=list()
cmrs.D.grpe3=list()

timer=Sys.time()
  for (i in 1:length(deduc)) {
S=seq(0,quant.DS_n[i],1000)
S.quant.D=S/quant.DS_n[i]
esp.grpe1=data.frame(S=S,S.quant.D=S.quant.D,exp=rep(NA,length(S)))
esp.grpe2=data.frame(S=S,S.quant.D=S.quant.D,exp=rep(NA,length(S)))
esp.grpe3=data.frame(S=S,S.quant.D=S.quant.D,exp=rep(NA,length(S)))

for (j in 1:length(S)) {

  #Conditional mean risk sharing
  A=contri(s=S[j],deduc[i],Maxi,Fs=cdf.D.Sn[[i]],FY1=cdf.D.SY1[[i]],
FY2=cdf.D.SY2[[i]],FY3=cdf.D.SY3[[i]])
esp.grpe1$exp[j]=A$c1
esp.grpe2$exp[j]=A$c2
esp.grpe3$exp[j]=A$c3
}
cmrs.D.grpe1[[length(cmrs.D.grpe1)+1]]=esp.grpe1
cmrs.D.grpe2[[length(cmrs.D.grpe2)+1]]=esp.grpe2
cmrs.D.grpe3[[length(cmrs.D.grpe3)+1]]=esp.grpe3
rm(S,A,esp.grpe1,esp.grpe2,esp.grpe3)
}
print(Sys.time()-timer)

```

```
## Time difference of 59.39966 secs
```

```

names(cmrs.D.grpe1)=paste("N",as.character(deduc),sep="")
names(cmrs.D.grpe2)=paste("N",as.character(deduc),sep="")
names(cmrs.D.grpe3)=paste("N",as.character(deduc),sep="")

```

```
par(mfrow=c(1,3),font.main=1,cex.main = 0.9, cex.sub = 0.8)
#groupe 1
plot(cmrs.D.grpe1[[1]]$S,cmrs.D.grpe1[[1]]$exp, type="l",lty=1,
xlim=c(0,max(cmrs.D.grpe1[[5]]$S,0)),ylim=c(0,max(cmrs.D.grpe1[[5]]$exp,0)),
col="blue3",ylab=list(TeX('$E[Y_i|S_n=s]$')),xlab="s",main="Groupe 1")
lines(cmrs.D.grpe1[[2]]$S,cmrs.D.grpe1[[2]]$exp,lty=1,
col="green3")
lines(cmrs.D.grpe1[[3]]$S,cmrs.D.grpe1[[3]]$exp, type="l",lty=1,
col="red3",)
lines(cmrs.D.grpe1[[4]]$S,cmrs.D.grpe1[[4]]$exp, type="l",lty=1,
col="#EB8744")
lines(cmrs.D.grpe1[[5]]$S,cmrs.D.grpe1[[5]]$exp, type="l",lty=13,
col="yellow3")
legend("bottomright",cex=0.6,legend = c("d=50","d=100","d=150","d=200","d=250"),
col=c("blue3","green3","red3","#EB8744","yellow3"),lty=c(1,2,3))

#groupe 2
plot(cmrs.D.grpe2[[1]]$S,cmrs.D.grpe2[[1]]$exp, type="l",lty=1,
xlim=c(0,max(cmrs.D.grpe2[[5]]$S,0)),ylim=c(0,max(cmrs.D.grpe2[[5]]$exp,0)),
col="blue3",ylab=list(TeX('$E[Y_i|S_n=s]$')),xlab="s",main="Groupe 2")
lines(cmrs.D.grpe2[[2]]$S,cmrs.D.grpe2[[2]]$exp,lty=1,
col="green3")
lines(cmrs.D.grpe2[[3]]$S,cmrs.D.grpe2[[3]]$exp, type="l",lty=1,
col="red3")
lines(cmrs.D.grpe2[[4]]$S,cmrs.D.grpe2[[4]]$exp, type="l",lty=1,
col="#EB8744")
lines(cmrs.D.grpe2[[5]]$S,cmrs.D.grpe2[[5]]$exp, type="l",lty=13,
col="yellow3")
legend("bottomright",cex=0.6,legend = c("d=50","d=100","d=150","d=200","d=250"),
col=c("blue3","green3","red3","#EB8744","yellow3"),lty=c(1,2,3))

#groupe 3
plot(cmrs.D.grpe3[[1]]$S,cmrs.D.grpe3[[1]]$exp, type="l",lty=1,
xlim=c(0,max(cmrs.D.grpe3[[5]]$S,0)),ylim=c(0,max(cmrs.D.grpe3[[5]]$exp,0)),
col="blue3",ylab=list(TeX('$E[Y_i|S_n=s]$')),xlab="s",main="Groupe 3")
lines(cmrs.D.grpe3[[2]]$S,cmrs.D.grpe3[[2]]$exp,lty=1,
col="green3")
lines(cmrs.D.grpe3[[3]]$S,cmrs.D.grpe3[[3]]$exp, type="l",lty=1,
col="red3")
lines(cmrs.D.grpe3[[4]]$S,cmrs.D.grpe3[[4]]$exp, type="l",lty=1,
col="#EB8744")
lines(cmrs.D.grpe3[[5]]$S,cmrs.D.grpe3[[5]]$exp, type="l",lty=13,
col="yellow3")
legend("bottomright",cex=0.6,legend = c("d=50","d=100","d=150","d=200","d=250"),
col=c("blue3","green3","red3","#EB8744","yellow3"),lty=c(1,2,3))
```

Densité de probabilité en fonction de la franchise

```

time=Sys.time()
# tab.deduc=list()
n=2000
Maxi=8500
#nombre de valeurs de s à generer
l=150
#franchise annuelle applicable à tous les participants
deduc=c(50,150,250,300,400)
loi.D=list()
for (i in 1:length(deduc)) {
  S=r(cdf.D.Sn[[i]])(1)
  tab=data.frame(S=S,expD.grpe1=rep(NA,1),expD.grpe2=rep(NA,1),
                expD.grpe3=rep(NA,1))
  for (j in 1:l) {
    A=contri(s=S[j],deduc[i],Maxi,Fs=cdf.D.Sn[[i]],
FY1=cdf.D.SY1[[i]],FY2=cdf.D.SY2[[i]],FY3=cdf.D.SY3[[i]])
    tab$expD.grpe1=A$c1
    tab$expD.grpe2=A$c2
    tab$expD.grpe3=A$c3
  }
  loi.D[[length(loi.D)+1]]=tab
  rm(S,tab,A)
}
print(Sys.time()-time)

```

```
## Time difference of 1.579034 mins
```

```
names(loi.D)=paste("Franchise",as.character(deduc),sep="")
```

```

pdf.grpe1.deduc=list()
pdf.grpe2.deduc=list()
pdf.grpe3.deduc=list()
  for (i in 1:length(deduc)) {
    density.grpe1.deduc=density(loi.D[[i]]$expD.grpe1)
    density.grpe2.deduc=density(loi.D[[i]]$expD.grpe2)
    density.grpe3.deduc=density(loi.D[[i]]$expD.grpe3)

    valeur1=data.frame(val=density.grpe1.deduc$x, prob=density.grpe1.deduc$y)
    valeur2=data.frame(val=density.grpe2.deduc$x, prob=density.grpe2.deduc$y)
    valeur3=data.frame(val=density.grpe3.deduc$x, prob=density.grpe3.deduc$y)

    pdf.grpe1.deduc[[length(pdf.grpe1.deduc)+1]]=valeur1
    pdf.grpe2.deduc[[length(pdf.grpe2.deduc)+1]]=valeur2
    pdf.grpe3.deduc[[length(pdf.grpe3.deduc)+1]]=valeur3
    rm(density.grpe1.deduc, density.grpe2.deduc, density.grpe3.deduc,valeur1,
    valeur2,valeur3 )}

```

```
names(pdf.grpe1.deduc)=paste("Franchise", as.character(deduc),sep="")
names(pdf.grpe2.deduc)=paste("Franchise", as.character(deduc),sep="")
names(pdf.grpe3.deduc)=paste("Franchise", as.character(deduc),sep="")

# groupe 1
par(mfrow=c(1,3),font.main=1,cex.main=0.9,cex.sub=0.8)
plot(pdf.grpe1.deduc[[5]]$val, pdf.grpe1.deduc[[5]]$prob, type="l",lty=3,
     col="yellow3",xlim=c(0,max(pdf.grpe1.deduc[[1]]$val)),
     xlab=list(TeX('$E[Y_i|S_n]$')),ylab="density",main="Groupe 1")

lines(pdf.grpe1.deduc[[2]]$val, pdf.grpe1.deduc[[2]]$prob, type="l",lty=2,
      col="green3",
      xlab=list(TeX('$E[Y_i|S_n]$')),ylab="density",main="Groupe 1")

lines(pdf.grpe1.deduc[[3]]$val, pdf.grpe1.deduc[[3]]$prob, type="l",lty=3,
      col="red3",
      xlab=list(TeX('$E[Y_i|S_n]$')),ylab="density",main="Groupe 1")

lines(pdf.grpe1.deduc[[1]]$val, pdf.grpe1.deduc[[1]]$prob, type="l",lty=1,
      col="blue3",
      xlab=list(TeX('$E[Y_i|S_n]$')),ylab="density",main="Groupe 1")

lines(pdf.grpe1.deduc[[4]]$val, pdf.grpe1.deduc[[4]]$prob, type="l",lty=3,
      col="#EB8744",
      xlab=list(TeX('$E[Y_i|S_n]$')),ylab="density",main="Groupe 1")

legend("topright",cex=0.6,legend = c("d=50","d=100","d=150","d=200","d=250"),
      col=c("blue3","green3","red3","#EB8744","yellow3"),lty=c(1,2,3))

# groupe 2
#par(mfrow=c(1,3),font.main=1,cex.main=0.9,cex.sub=0.8)
plot(pdf.grpe2.deduc[[5]]$val, pdf.grpe2.deduc[[5]]$prob, type="l",lty=3,
     col="yellow3",
xlab=list(TeX('$E[Y_i|S_n]$')),ylab="density",main="Groupe 2",
     xlim=c(0,max(pdf.grpe1.deduc[[1]]$val,190)))

lines(pdf.grpe2.deduc[[1]]$val, pdf.grpe2.deduc[[1]]$prob, type="l",lty=1,
      col="blue3",
      xlab=list(TeX('$E[Y_i|S_n]$')),ylab="density",main="Groupe 2")

lines(pdf.grpe2.deduc[[2]]$val, pdf.grpe2.deduc[[2]]$prob, type="l",lty=2,
      col="green3",
      xlab=list(TeX('$E[Y_i|S_n]$')),ylab="density",main="Groupe 2")
```

```

lines(pdf.grpe2.deduc[[3]]$val, pdf.grpe2.deduc[[3]]$prob, type="l",lty=3,
      col="red3",
      xlab=list(TeX('$E[Y_i|S_n]$')),ylab="density",main="Groupe 2")

lines(pdf.grpe2.deduc[[4]]$val, pdf.grpe2.deduc[[4]]$prob, type="l",lty=3,
      col="#EB8744",
      xlab=list(TeX('$E[Y_i|S_n]$')),ylab="density",main="Groupe 2")

legend("topright",cex=0.6,legend = c("d=50","d=100","d=150","d=200","d=250"),
      col=c("blue3","green3","red3","#EB8744","yellow3"),lty=c(1,2,3))

# groupe 3
#par(mfrow=c(1,3),font.main=1,cex.main=0.9,cex.sub=0.8)
plot(pdf.grpe3.deduc[[5]]$val, pdf.grpe3.deduc[[5]]$prob, type="l",lty=3,
     col="yellow3", xlab=list(TeX('$E[Y_i|S_n]$')),
     ylab="density",main="Groupe 3",xlim=c(0,max(pdf.grpe1.deduc[[1]]$val,350)))

lines(pdf.grpe3.deduc[[1]]$val, pdf.grpe3.deduc[[1]]$prob, type="l",lty=1,
      col="blue3",
      xlab=list(TeX('$E[Y_i|S_n]$')),ylab="density",main="Groupe 3")

lines(pdf.grpe3.deduc[[2]]$val, pdf.grpe3.deduc[[2]]$prob, type="l",lty=2,
      col="green3",
      xlab=list(TeX('$E[Y_i|S_n]$')),ylab="density",main="Groupe 3")

lines(pdf.grpe3.deduc[[3]]$val, pdf.grpe3.deduc[[3]]$prob, type="l",lty=3,
      col="red3",
      xlab=list(TeX('$E[Y_i|S_n]$')),ylab="density",main="Groupe 3")

lines(pdf.grpe3.deduc[[4]]$val, pdf.grpe3.deduc[[4]]$prob, type="l",lty=3,
      col="#EB8744",
      xlab=list(TeX('$E[Y_i|S_n]$')),ylab="density",main="Groupe 3")

legend("topright",cex=0.6,legend = c("d=50","d=100","d=150","d=200","d=250"),
      col=c("blue3","green3","red3","#EB8744","yellow3"),lty=c(1,2,3))

```

Boxplot

```

Boxplot.D.grpe1<-list()
Boxplot.D.grpe2<-list()
Boxplot.D.grpe3<-list()
for (i in 1:length(deduc)) {
  Boxplot.D.grpe1[[length(Boxplot.D.grpe1)+1]]<-loi.D[[i]]$expD.grpe1
  Boxplot.D.grpe2[[length(Boxplot.D.grpe2)+1]]<-loi.D[[i]]$expD.grpe2
  Boxplot.D.grpe3[[length(Boxplot.D.grpe3)+1]]<-loi.D[[i]]$expD.grpe3
}

```

```

    }
names(Boxplot.D.grpe1)<-as.character(deduc)
names(Boxplot.D.grpe2)<-as.character(deduc)
names(Boxplot.D.grpe3)<-as.character(deduc)

dedu.plotgrpe1.long1<-melt(do.call(rbind,Boxplot.D.grpe1))
dedu.plotgrpe2.long2<-melt(do.call(rbind,Boxplot.D.grpe2))
dedu.plotgrpe3.long3<-melt(do.call(rbind,Boxplot.D.grpe3))
par(mfrow=c(1,3),font.main=1,cex.main = 0.9, cex.sub = 0.8)

boxplot(value~Var1,dedu.plotgrpe1.long1,xlab ="Franchise",ylab = "",main="Groupe 1")
abline(h=E(gpe1),lwd = 1.5,lty=4,col="red")

boxplot(value~Var1,dedu.plotgrpe2.long2,xlab ="Franchise",ylab = "",main="Groupe 2")
abline(h=E(gpe2),lwd = 1.5,lty=4,col="red")

boxplot(value~Var1,dedu.plotgrpe3.long3,xlab ="Franchise",ylab = "",main="Groupe 3")
abline(h=E(gpe3),lwd = 1.5,lty=4,col="red")

```

A.5 Système avec réassurer

Taille du pool: N=1600
 Franchise: Deduc=250
 Quote Part: Q=0.8
 Remboursement Maximal annuel: Plf=8500

Création des groupes

```

pool=function(Age,prop1=0.5,prop2=0.3,prop3=0.2,N){
  L=length(Age)
  gr1=ceiling(prop1*N)
  gr2=ceiling(prop2*N)
  gr3=N-(gr1+gr2)
  Ni=c(gr1,gr2,gr3)
  Grp=data.frame(Age=Age,Gender=c("Groupe 1","Groupe 2","Groupe 3"),Ni=Ni)
  return(Grp)}
Pool=pool(c(12,60,120),0.5,0.3,0.2,1600)

```

```

##Probabilité cible pour avoir un surplus
beta=0.85
##chargement de sécurité
theta_SL=0.5
##Retentions
w_n=distr::q(cdf.S_n[["N1600"]])(beta)

#Niveau de retention individuelles

```

```

wi=contri(s=w_n,deduc[3],Maxi,Fs=cdf.S_n[["N1600"]],
FY1=cdf.S_Y1[["N1600"]],FY2=cdf.S_Y2[["N1600"]],FY3=cdf.S_Y3[["N1600"]])
Pool$Wi=t(as.data.frame(wi))[1:length(wi)]

```

```

## Valeurs E[Xi|S=s] simulées précédemment
cmrs.level=loi.N[["N800"]][,-5]
timer = Sys.time()
## Vecteur contenant les valeurs de min{S,W}
Charge.P2P = cmrs.level
Charge.P2P[1:nrow( cmrs.level),2:ncol( cmrs.level)] = 0
## Vecteur contenant les valeurs de la prime SL
PI.SL = cmrs.level
PI.SL[1:nrow( cmrs.level),2:ncol( cmrs.level)] = 0
## Vecteur contenant les valeurs du cash-back
BI.n <- cmrs.level
BI.n[1:nrow(cmrs.level),2:ncol(cmrs.level)] = 0
Pool$EYi = rep(NA,nrow(Pool))
Pool

```

Age	Groupe	N_i
12	Groupe 1	800
60	Groupe 2	480
120	Groupe 3	320

TABLE A.5.1 – Composition du pool

```

#Fonction de répartition des différents groupes

```

```

FX = list()
FX[[1]] = gpe1
FX[[2]] = gpe2
FX[[3]] = gpe3

```

```

for (i in 2:ncol(cmrs.level)) {
  #E[Xi]
  Pool$EYi[i-1] <- E(FX[[i-1]])

  #S_P2P=min{h,wi}
  Charge.P2P[,i] = ifelse(
    test = cmrs.level[,i] < Pool$wi[i-1],
    yes = cmrs.level[,i],
    no = Pool$wi[i-1])
  # Calcul de la prime stop-loss
  PI.SL[,i] <- ifelse(
    test = cmrs.level[,i] - Pool$wi[i-1] > 0,
    yes = cmrs.level[,i] - Pool$wi[i-1],
    no = 0)
  # Cash-back

```

```

BI.n[,i] <- ifelse(
  test = Pool$Wi[i-1]-cmrs.level[,i] > 0,
  yes = Pool$Wi[i-1]-cmrs.level[,i],
  no = 0)}
print(Sys.time()-timer)

```

```
## Time difference of 0.05208898 secs
```

```

Pool$prime.SL <- (1+theta_SL)*colMeans(PI.SL[,-1],na.rm = TRUE)
Pool$Prime <- Pool$prime.SL + Pool$Wi
Pool$Ex.Bi <- colMeans(BI.n[,-1],na.rm = TRUE)
Grp.table <- Pool[,-c(4,5)]
Pool

```

A.6 Variabilité des cotisations dans tranche inférieure $[0, \omega_n]$

A.6.1 Comparaison avec une variation du nombre de participant

Étude de la variabilité dans la tranche inférieure en fonction du nombre de participants.

```

timer <- Sys.time()
#nombre de participants.
n=c(800,1600,2000,2500,3000,3500)
## Vecteur des probabilités beta pour octroyer des bonus en fin de période
beta=0.85
## Chargement de sécurité pour la stop-loss
theta_SL=0.5
GRP=list()
down.level.N = list()
up.level.N = list()
SL.prem = data.frame(Grp=seq(1,3,1))
for (i in 1:length(n)) {
  #Création d'un pool avec les 3 classes d'âges
  GRP[[i]]=pool(c(12,60,120),0.5,0.3,0.2,n[i])
  Grp.level = pool(c(12,60,120),0.5,0.3,0.2,n[i])

  # Fonction de répartition de Sn
  Fpart = cdf.S_n[[i]]
  # rétention du groupe
  w_n = distr::q(cdf.S_n[[i]])(beta)
  # rétention individuelles Wi
  wi = contri(s=w_n,deduc[3],Maxi,Fs=cdf.S_n[[i]],FY1=cdf.S_Y1[[i]],
  FY2=cdf.S_Y2[[i]],FY3=cdf.S_Y3[[i]])
  Grp.level$Wi = t(as.data.frame(wi))[1:length(wi)]
  GRP[[i]]$Wi = Grp.level$Wi
  # Récupération des valeurs de E[Xi|S] pour les différentes valeurs de n
  cmrs.level = loi.N[[i]][,-5]
}

```



```

# Vecteur contenant les valeurs de  $\min\{E[X_i/S], W_i\}$ 
S.P2P = cmrs.level
S.P2P[1:nrow(cmrs.level),2:ncol(cmrs.level)] <- 0
#
# Vecteur contenant les valeurs de  $\min\{E[X_i/S], W_i\}$ 
S.UP <- cmrs.level
S.UP[1:nrow(cmrs.level),2:ncol(cmrs.level)] <- 0
#
prem <- data.frame(SL=rep(NA,nrow(Grp.level)))
for (i in 2:ncol(cmrs.level)) {
  for (j in 1:nrow(cmrs.level)) {
    #tranche inférieure:  $S.P2P=\min\{h, w_i\}$ 
    S.P2P[j,i] <- ifelse(
      test = cmrs.level[j,i] < Grp.level$Wi[i-1],
      yes = cmrs.level[j,i],
      no = Grp.level$Wi[i-1])

    # # tranche supérieure  $S.UP=S-\min\{S, w_n\}$ 
    S.UP[j,i] <- ifelse(
      test = cmrs.level[j,i] - Grp.level$Wi[i-1] > 0,
      yes = cmrs.level[j,i] - Grp.level $Wi[i-1],
      no = 0)
  }
  prem[i-1,1] <- (1+theta_SL)*mean(S.UP[,i])
}
SL.prem <- cbind(SL.prem,prem,Grp.level$Wi)
down.level.N[[length(down.level.N)+1]] <- S.P2P
up.level.N[[length(up.level.N)+1]] <- S.UP
rm(Grp.level,w_n,wi,S.P2P,S.UP)
}
colnames(SL.prem)[1] <- "Grp"
names(down.level.N)<-paste("N", as.character(n), sep = "")
names(up.level.N)<-paste("N", as.character(n), sep = "")
print(Sys.time()-timer)

```

```
## Time difference of 3.843795 secs
```

```

Level.plot = list()

for (j in 2:ncol(down.level.N[[3]])) {
  Bx_plt=list()
  for (i in 1:length(down.level.N)) {
    tmp = down.level.N[[i]][[j]]
    tmp = subset(tmp,tmp < GRP[[i]]$Wi[j-1])
    Bx_plt[[length(Bx_plt)+1]]=tmp}
  names(Bx_plt)=as.character(n)
}

```

```

Bx_plt_long=melt(do.call(rbind,Bx_plt))
Level.plot[[length(Level.plot)+1]] = Bx_plt_long}

rm(Bx_plt_long,Bx_plt)

names(Level.plot) = colnames(down.level.N[[1]])[2:ncol(down.level.N[[1]])]

par(mfrow=c(1,3), family = "sans", font.lab = 2, font.axis = 1, cex.lab=0.9,
    cex.main=0.9, font.main=1,font.lab=1,font.axis=1)
for (i in 1:length(Level.plot)) {
  boxplot(value~Var1,Level.plot[[i]],xlab = "",ylab = "",
          main = names(Level.plot)[i],
          outline = FALSE)
  abline(h=E(FX[[i]]),lwd = 1,lty=4,col="red")}

```

Variabilité des cotisations dans la tranche supérieure.

```

#nombre de participants
n=c(800,1600,2000,2500,3000,3500)
Level.Up <- list()
for (j in 2:ncol(up.level.N[[3]])) {
  Bx_plt<-list()
  for (i in 1:length(up.level.N)) {
    tmp <- up.level.N[[i]][[j]]
    tmp <- subset(tmp,tmp > 0)
    Bx_plt[[length(Bx_plt)+1]]<-tmp
    rm(tmp)
  }
  names(Bx_plt)<-as.character(n)
  Bx_plt_long<-melt(do.call(rbind,Bx_plt))
  Level.Up[[length(Level.Up)+1]] <- Bx_plt_long
}
Visualisation du transfert de la tranche superieure des reclamations à un (re)-assureur
rm(Bx_plt_long,Bx_plt)
names(Level.Up) <- colnames(up.level.N[[1]])[2:ncol(up.level.N[[1]])]

par(mfrow=c(1,3), family = "sans", font.lab = 2, font.axis = 1, cex.lab=0.9,
    cex.main=0.9, font.main=1,font.lab=1,font.axis=1)
for (i in 1:length(Level.Up)) {
  boxplot(value~Var1,Level.Up[[i]],xlab = "",ylab = "",
          main = names(Level.Up)[i],
          outline = FALSE)}

n=c(800,1600,2000,2500,3000,3500)
Box.Vs <- list()

```

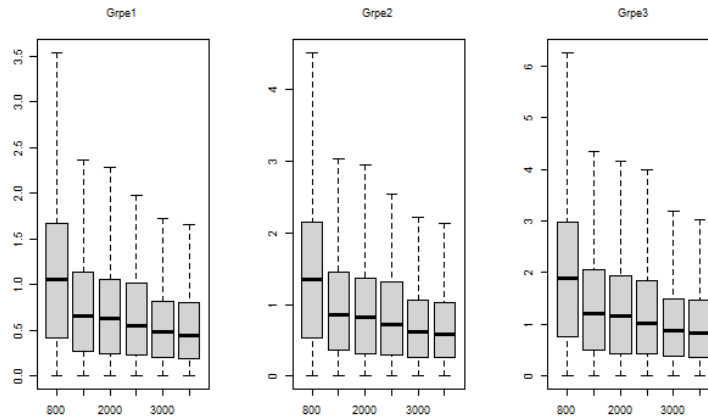


FIGURE A.6.1 – Répartition des contributions dans la tranche supérieure

```

for (j in 2:ncol(down.level.N[[1]])) {
  Bx.vs<-list()
  for (i in 1:length(down.level.N)) {
    tmp <- down.level.N[[i]][[j]]
    tmp <- subset(tmp,tmp < GRP[[i]]$Wi[j-1])
    tmp <- as.data.frame(tmp)
    tmp$Hin <- loi.N[[i]][[j]][1:nrow(tmp)]
    tmp$N <- n[i]
    colnames(tmp) <- c("SP2P","Hin","N")
    Bx.vs[[length(Bx.vs)+1]]<-tmp}
  names(Bx.vs)<-as.character(n)
  Bx.vs_long<-melt(do.call(rbind,Bx.vs),id="N")
  Bx.vs_long$N <- as.factor(Bx.vs_long$N)
  Box.Vs[[length(Box.Vs)+1]] <- Bx.vs_long}
names(Box.Vs) <- colnames(down.level.N[[1]])[2:ncol(down.level.N[[1]])]
par(mfrow=c(1,3), family = "sans", font.lab = 2, font.axis = 1, cex.lab=0.9,
    cex.main=0.9, font.main=1,font.lab=1,font.axis=1)
for (i in 1:length(Level.plot)) {
  boxplot(value~variable+N,Box.Vs[[i]],xlab = "N",ylab = "",
    at = c(1:2, 4:5, 7:8, 10:11, 13:14, 16:17),
    main = names(Level.plot)[i], col = c("chocolate1","cadetblue"),
    names = c("800","1600","2000","2500","3000","","","","","","3500",""),
    outline = FALSE)
  abline(h=E(FX[[i]]),lwd = 1,lty=4,col="red")
legend("topright",box.lty=0,inset = 0.02,
  fill = c("chocolate1","cadetblue"),
  legend = c(TeX('$S^{P2P}_n$'),TeX('$S_n$')), horiz = T)}

```

UNIVERSITÉ CATHOLIQUE DE LOUVAIN
Faculté des sciences

Place des Sciences, 2 bte L6.06.01, 1348 Louvain-la-Neuve, Belgique | www.uclouvain.be/sc