Institut de Statistiques de l'Université de Paris

Sorbonne Université

Processus de tarification Non-Vie sur des données chiffrées & anonymisées

Thomas Poinsignon

Responsable entreprise : Antoine Ly
Responsable pédagogique : Claire Boyer

Novembre 2018

Mise à jour : Mai 2019

Mise à jour

Le mémoire ici rédigé correspond à une version légèrement enrichi de celui présenté, et validé à date, lors de la soutenance pour le titre d'actuaire certifié de l'Institut des actuaires en Janvier 2019. Il prend ainsi en compte l'évolution de nos travaux sur le sujet afin d'y incorporer nos dernières analyses.
A ce titre, la liste exhaustive des modifications réalisées au sein de cette version sont :

- La distinction des approches employées pour l'analyse des modèles d'anonymisation et l'ajout de la sous-partie *4.3.2.2 - Approche métier selon l'influence de modalités*

- De succinctes mentions et explications d'une méthodologie alternative d'anonymisation propre aux modèles de tarification non-linéaires au sein de la partie *4.3.3 - Limites de notre procédure de tarification anonymisée*

Ainsi les lecteurs désirant accéder aux travaux originaux sont invités à retrouver le mémoire initial sur le site de l'Institut des actuaires [1].

1. https://www.institutdesactuaires.com/se-documenter/memoire-d-actuariat-38 ?id=29f4e22189bd89633a48154d00bef19c
http://www.ressources-actuarielles.net/C12574E200674F5B/0/A5D0031E07E08426C12584620021C48E

Remerciements

Je tiens à remercier toute l'équipe de l'actuariat non-vie du cabinet Milliman pour m'avoir chaleureusement accueilli et accompagné tout au long de mon stage, et plus particulièrement Antoine Ly pour l'encadrement de ce mémoire et sa grande disponibilité pour échanger sur notre problématique ainsi que Rémi Bellina, notamment pour ses conseils avisés et son expertise dans les modèles de tarification.

Je remercie également Laurent Devineau qui a initié et m'a proposé le sujet atypique de ce mémoire.

Enfin je remercie Claire Boyer, ma tutrice pédagogique à l'ISUP qui m'a suivi tout au long de mon stage.

Summary

The recent increase in the amount of data generated, stored and analyzed by insurers to establish their pricing and underwriting policies has led to the emergence of new needs. Both from a regulatory point of view, with the recent implementation in the European framework of the General Data Protection Regulation (*GDPR*), and with a view to offering new services on the market (*cyber risk*).
The work carried out in this paper is thus devoted to the development and analysis of actuarial methods within the *default security* framework - a principle presented and imposed on companies using personal data by the GDPR. The objective is therefore to extend the elementary mathematical concepts and models used when developing non-life insurance pricing models (Simple Linear Regression and Generalized Linear Models) to their use on secure data in accordance with regulatory requirements.
We will then start by defining the framework of our study in order to specify the regulatory and theoretical contexts within which our problem stands, then we will first focus on the development of an encryption procedure to perform a simple linear regression on encrypted data without ever having to decrypt them during the process. In other words, being able to calculate a linear regression on a pre-cyphered database - in this paper thanks to the *Efficient Integer Vector Homomorphic Encryption* and *Fan & Vercauteren* schemes - without having knowledge of the decryption keys so only the owner has the possibility to decrypt the obtained results.
In a second step, we will focus on an alternative methodology to data encryption : anonymization of the insured portfolio by aggregating policies using non-supervised learning methods (OPTICS, K-Means, etc.). We then obtain for each cluster a new anonymous individual representative of his group. Our idea is then to carry out the pricing of an automobile civil liability insurance based on data thus secured. To analyze the performance of this process, we will compare these results with those obtained from this same pricing model but calculated on non-anonymized data.

Glossary : *General Data Protection Regulation - GDPR, cyber risk, anonymization, peudonymization, homomorphic encryption scheme, simple linear regression, generalized linear models, unsupervised machine learning algorithms, cost/frequency tarification model*

Résumé

La récente démultiplication de la quantité de données générées, stockées et analysées par les assureurs afin d'établir leurs politiques tarifaires et de souscriptions, a conduit à l'émergence de nouveaux besoins : tant du point de vue réglementaire, avec dernièrement la mise en oeuvre dans le cadre européen du règlement général sur la protection des données (*RGPD*), que dans la perspective de proposer de nouveaux services sur le marché (*risque cyber*).
Les travaux réalisés dans le cadre de ce mémoire sont ainsi consacrés au développement et à l'analyse de méthodes actuarielles dans un cadre de *sécurité par défaut* - principe présenté et imposé aux entreprises ayant recours à des données personnelles par le RGPD. L'objectif est donc d'étendre les concepts et les modèles mathématiques élémentaires utilisés lors de l'élaboration de modèles de tarification d'assurance non-vie (Régression linéaire simple et modèles linéaires généralisés) à leurs utilisations sur des données sécurisées conformément aux exigences de la réglementation.
Nous commencerons alors par définir le cadre de notre étude afin de préciser les contextes réglementaire et théorique au sein desquels s'inscrit notre problématique. Puis nous nous concentrerons alors dans un premier temps sur l'élaboration d'une procédure de chiffrement et de calculs permettant d'effectuer une régression linéaire simple sur des données cryptées, sans jamais avoir à les déchiffrer au cours du processus. En somme, nous souhaitons être capable de calculer une régression linéaire sur une base de données chiffrées au préalable - dans ce mémoire grâce aux schémas *Efficient Integer Vector Homomorphic Encryption* et de *Fan & Vercauteren* - sans avoir connaissance des clefs de déchiffrement où seul le propriétaire a alors la possibilité de décrypter les résultats obtenus.
Dans un second temps, nous nous intéresserons à une méthodologie alternative au chiffrement des données : l'anonymisation du portefeuille d'assurés par l'agrégation des polices en utilisant des méthodes d'apprentissage non-supervisées (OPTICS, K-Means, etc.). On obtient alors pour chaque partition formée un nouvel individu anonyme et représentatif de son groupe. Notre idée est alors de réaliser la tarification d'une assurance responsabilité civile automobile à partir de données ainsi sécurisées. Pour analyser la performance de ce processus, nous comparerons les résultats obtenus à ceux issus de ce même modèle de tarification mais calculés sur des données non anonymisées.

Lexique : *Règlement général sur la protection des données - RGPD, risque cyber, anonymisation, pseudonymisation, méthode de chiffrement homomorphe, régression linéaire simple, modèle linéaire généralisé, algorithme d'apprentissage non-supervisé, modèle de tarification coût/fréquence*

Abstract

Context of the study

The management of personal data held by an insurer has now become a major challenge for all players in the insurance industry. The recent consideration of the cyber risk associated with the storage and handling of this sensitive data by insurance companies is one of the main examples.
At the same time, the evolution of the European regulatory framework through the implementation of the General Data Protection Regulation (GDPR) ends to considering the issue of processing data entrusted by the insured to his insurer as a policy of mutual trust that can no longer be neglected.
This is how we find within this regulation the good practices that insurers must respect, among other things, with regard to the data they have in their possession. In particular, we find the *principles of data protection from the design stage* and *security by default*, which aim to clarify and formalize the constraints introduced by this new text. In order to achieve this, the concepts of *anonymized* and *pseudonymized* data are also defined.

Outsourcing of calculations via pseudonymization

Pseudonymization consists in making a data partially anonymous. This means that the data processed this way is difficult to re-attribute to a specific individual. To achieve this, several methods can be used, including secret key encryption methods. In this paper we have examined the possibility of performing a simple linear regression on pseudonymized data by encryption, without ever having to decrypt them :

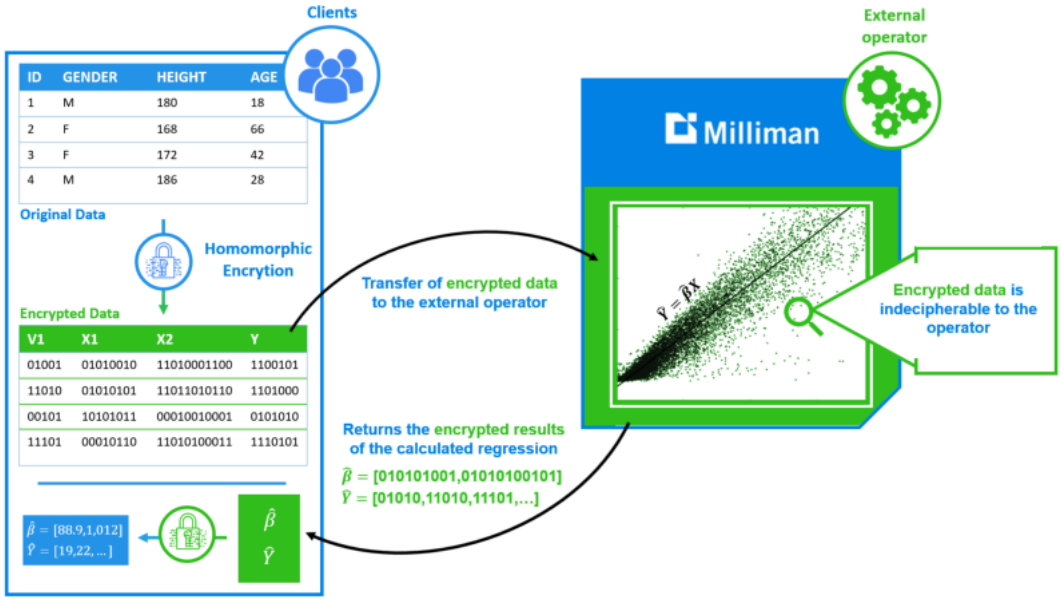


Figure : Procedure for the delegated calculation of a pseudonymized linear regression

We therefore had to use and implement singular encryption schemes to perform certain computational operations directly in the encrypted space. Such methods are called homomorphic. Therefore we chose two schemes : first of all the *Efficient Integer*

Vector Homomorphic Encryption scheme that we implemented in `Python` and whose theoretical aspect is explained in this paper, which has the main advantage of being relatively simple to handle.
By applying this scheme to our moderate sized data, and with some concessions on upstream data processing, we were able to perform our linear regression without decrypting the data during the process.

Table : Memory resources used to perform a linear regression

Elements	**Encrypted space**	**Unencrypted space**
$\hat{\beta}$ and constitutive params.	2,9 kB	480 Bytes
$\hat{X}$	7,8 kB	408 Bytes
$\hat{Y}$	213,2 kB	232 Bytes
Total Cost	223,3 kB	808 Bytes

However, these concessions seem us to be over constraining in a concrete application framework (for example, the secure delegation of calculations from an insurer to an external service provider, *cloud computing*). This is why we decided to reiterate this methodology but using a more robust - and more complex - encryption scheme in `R` : the *Fan & Vercauteren scheme*.
From this model we were able to obtain results equivalent [2] to the previous scheme but without having to make any changes to our data beforehand.
To achieve this, we proceeded differently. With the *Efficient Integer Vector Homomorphic Encryption* scheme we had simply calculated the estimate of the coefficient vector of the regression $\hat{\beta}$ by the formula of ordinary least squares ($\hat{\beta} = (X^T X)^{-1} X^T Y$, with Y the response data vector and X the covariate matrices) in the encrypted space. Here, with the *Fan & Vercauteren* scheme we have chosen instead to obtain an estimate of $\hat{\beta}$ [3] by performing a gradient descent in the encrypted space.
Moreover, this gradient descent in the encrypted space converges well towards the value of $\hat{\beta}$ if the number of iterations is large enough as shown in the diagram below. The problem then encountered concerns the amount of resources used to achieve this. Finally, by using the encryption scheme of *Fan & Vercauteren* and determining an estimate of $\hat{\beta}$, we arrived at the comparative results presented below.
Nevertheless, the significant computation time induced by operations in the encrypted space, as well as the use of "only" pseudonymized data (the GDPR remains restrictive on the use of such data. To get off these constraints it is necessary to anonymize the database) has decided us to consider an alternative approach.

2. The values of the regression coefficients obtained after decryption are identical for both methods but the first method is faster while the second is more secure.

3. This is indeed an estimate of the estimator $\hat{\beta}$.

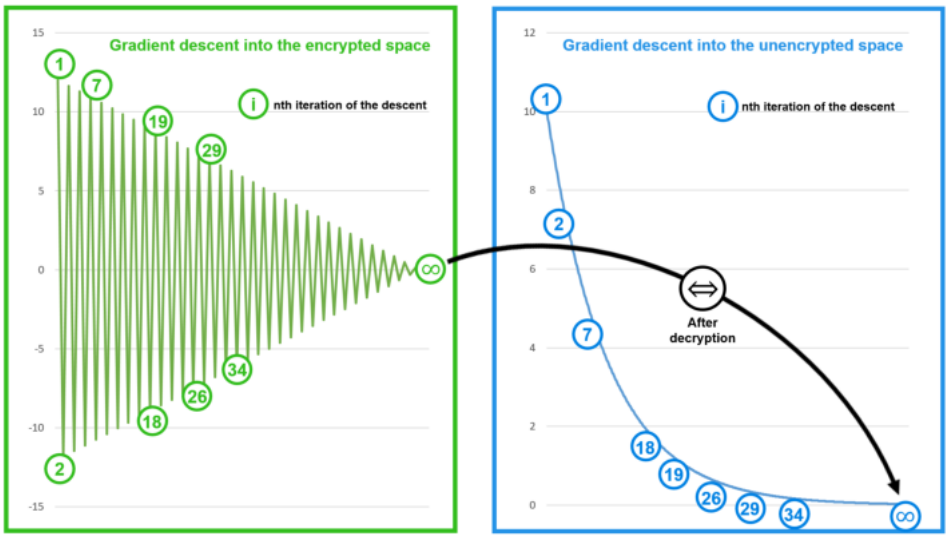


Figure : Diagram establishing the link between the gradient descent in the encrypted space and in the unencrypted space

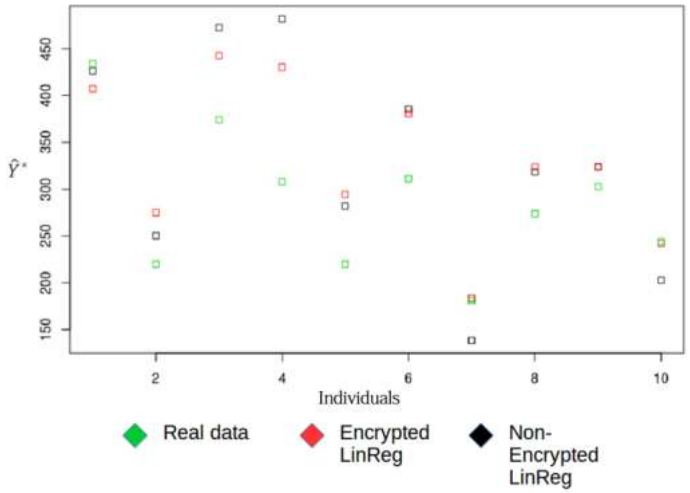


Figure : Comparison of predictions $\hat{Y}$ obtained

Efficient Vector Homomorphic Encryption Method		Gradient descent Method (F&V)	
Advantages	**Disadvantages**	**Advantages**	**Disadvantages**
✓ Easy to implement and understand encryption method	× Security guaranteed by non-optimal encryption method.	✓ Encryption method offering a customizable level of security.	× Very demanding encryption scheme based on complex concepts.
✓ Use of the closed OLS formula to calculate $\hat{\beta}$ in $\mathcal{X}$.	× Evolution of noise during calculations in $\mathcal{X}$ uncontrolled.	✓ The amount of noise introduced during operations can be determined upstream.	× Returns only an estimate of the estimator $\hat{\beta}$ from β.
✓ Necessary material resources and moderate turnaround time.	× Requires transformations/pre-calculations on the data by the customer.	✓ The customer provides directly X and Y encrypted to the calculation operator.	× Very resource-intensive and possibly inappropriate in the current context.
✓ Returns the "real" value of the estimate from β.			

Figure : Summary table of the methods used

Development of an anonymous car pricing system

Thus, we decided, based on *anonymized* data, to establish a cost/frequency model for a motor insurance pricing and to compare the estimated premium amounts with those obtained using the same model but calibrated, in a usual way, on the individual policies in the portfolio (i.e. unanonymized).
A data is said to be anonymized according to the GDPR if it is strictly impossible to re-identify it. It is therefore an irreversible and delicate procedure to be carried out if we want to keep as much information as possible about our data after its anonymisation. Thus the choice of anonymization method is crucial and we finally opted for an innovative approach : rather than delete or group certain variables or modalities as is customary to (horizontal approach), we preferred to aggregate fonts into very small subclasses (vertical approach) using unsupervised learning methods (clustering).
We therefore tested and configured several partitioning methods, including a K-Means, a density estimation algorithm (OPTICS), a hierarchical clustering method (CAH) and an affinity propagation partitioning. Naturally, the criteria for accepting clustering here are different from the usual framework of their uses :

Figure : Diagram of the differences in selection criteria for clustering models

While it is usually desirable to minimize the differences between observations of the same class (in a similar way, to maximize the differences between partitions) by varying the number of clusters to be formed, here the challenge is to obtain the largest possible number of clusters containing at least two insurance policies each (or a maximum of $N/2$ partitions if we have N policies in our portfolio) by minimizing the difference between the observations within each class.
To compare the results obtained, we directly evaluate the estimated sinistrality based on our pricing model on the data anonymized by each of the previous methods compared to the sinistrality estimated by the same model on the individual data. This allows us to categorize the anonymization methods mentioned above according to their characteristics and the quality of the results obtained.Thus, we can observe a difference of only 11% on the average sinistrality estimated on data anonymized by

density estimation compared to the sinistrality computed using the model calibrated on individual policies (benchmark model).

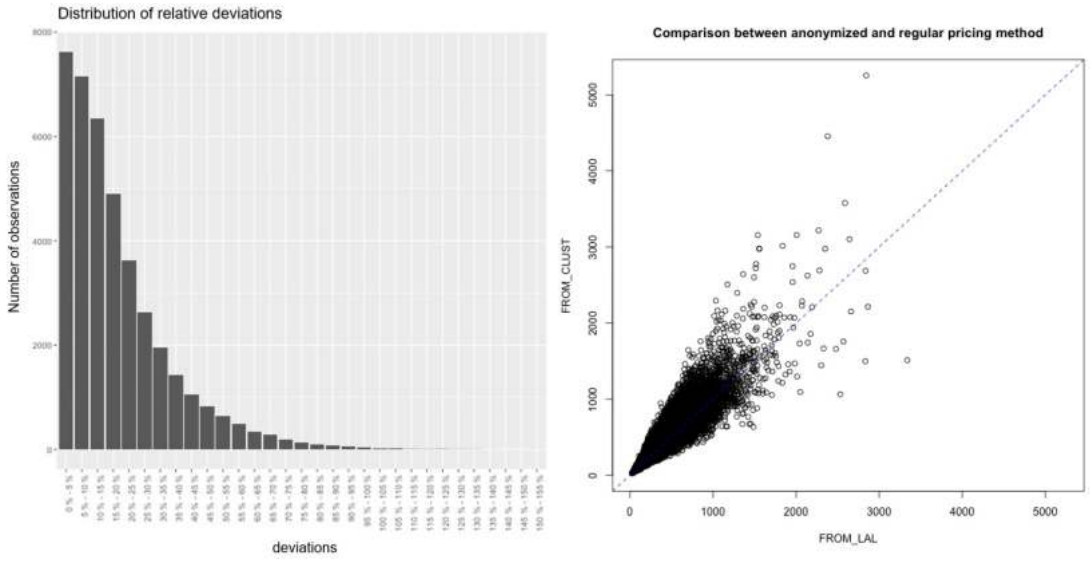


Figure : Distribution of relative deviations & comparison of models

Table : Characteristic quantities (OPTICS(2,3,80))

TOTAL LOSS EXPERIENCE BENCHMARK	+1.266E7
TOTAL LOSS EXPERIENCE ANONYMISED	+1.407E7
QUADRATIC SUM OF DEVIATIONS	+1.389E11
DEVIATION (ANONYMISED - BENCHMARK)	+1.408E6
RELATIVE DEVIATION (FROM BENCHMARK)	+11.12 %

Synthèse

Contexte de l'étude

La gestion des données personnelles détenues par un assureur est désormais devenue un enjeu majeur pour l'ensemble des acteurs du métier. La récente prise en considération du risque cyber lié au stockage et à la manipulation de ces données sensibles par les compagnies d'assurance en est d'ailleurs une des principales démonstrations. En parallèle, l'évolution du cadre réglementaire européen à travers la mise en place du règlement général sur la protection des données (RGPD) finit de considérer la question du traitement des données confiées par l'assuré à son assureur comme une politique de confiance mutuelle qui ne peut dorénavant pas être négligée.
C'est ainsi que l'on trouve au sein de ce règlement les bonnes pratiques que doivent respecter, entre autres, les assureurs vis à vis des données qu'ils ont en leur possession. On retrouve en particulier les principes de *protection des données dès la conception* et de *sécurité par défaut* visant à expliciter et formaliser les contraintes introduites par ce nouveau texte. Pour ce faire, les notions de données *anonymisées* et *pseudonymisées* y sont également définies.

Externalisation de calculs via pseudonymisation

La pseudonymisation consiste à rendre partiellement anonyme une donnée. Il faut ainsi comprendre que la donnée traitée devient difficilement ré-attribuable à un individu spécifique. Pour y parvenir plusieurs méthodes peuvent être employées, et notamment des méthodes de chiffrement à clefs secrètes. Nous avons dans ce mémoire étudié la possibilité de réaliser une régression linéaire simple sur des données pseudonymisées par cryptage, sans jamais avoir à les déchiffrer :

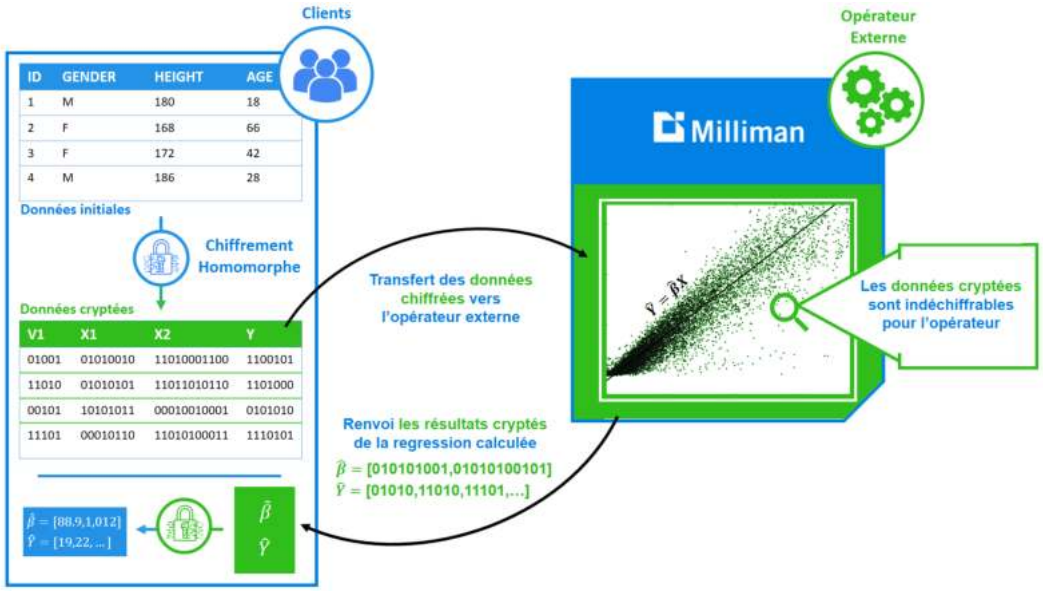


Figure : Procédure du calcul délégué d'une régression linéaire pseudonymisée

Il nous a donc fallu employer et implémenter des schémas de cryptage singuliers permettant de réaliser certaines opérations de calculs directement dans l'espace crypté.

De telles méthodes sont dites *homomorphes*. Notre choix s'est alors porté sur deux schémas : le schéma *Efficient Integer Vector Homomorphic Encryption* et le schéma de *Fan & Vercauteren*. Nous avons d'abord implémenté le schéma *Efficient Integer Vector Homomorphic Encryption* sous `Python` et dont l'aspect théorique est explicité dans ce mémoire, qui présente comme principal avantage d'être relativement simple à appréhender.
En appliquant ce schéma sur des données académiques de taille modérée, et moyennant quelques concessions sur le traitement en amont des données, nous sommes parvenus à réaliser notre régression linéaire sans décrypter les données au cours du processus.

Table : Ressources mémoire employées lors de la régression linéaire

Éléments	**Espace crypté**	**Espace non-crypté**
$\hat{\beta}$ et params. constitutifs	2,9 ko	480 octets
$\hat{X}$	7,8 ko	408 octets
$\hat{Y}$	213,2 ko	232 octets
Coût Tot.	223,3 ko	808 octets

Cependant ces concessions nous ont semblé handicapantes dans un cadre applicatif concret (par exemple la délégation sécurisée de calculs d'un assureur à un prestataire externe, le *cloud-computing*). C'est pourquoi nous avons décidé de ré-itérer cette méthodologie mais en employant un schéma de chiffrement plus robuste - et plus complexe - sous `R` : *le schéma de Fan & Vercauteren*.
A partir de ce modèle nous sommes parvenus à obtenir des résultats équivalents [4] au schéma précédent mais sans avoir à recourir à des modifications sur nos données au préalable.
Pour y parvenir nous avons procédé différemment. Avec le schéma *Efficient Integer Vector Homomorphic Encryption* nous avions simplement calculé l'estimation du vecteur de coefficient de la régression $\hat{\beta}$ par la formule des moindres carrés ordinaires ($\hat{\beta} = (X^T X)^{-1} X^T Y$, avec Y le vecteur des données de réponses et X la matrice des covariables) dans l'espace crypté. Ici, avec le schéma de *Fan & Vercauteren* nous avons choisi au contraire d'obtenir une estimation de $\hat{\beta}$ [5] en réalisant, dans l'espace crypté, une descente de gradient.
Par ailleurs cette descente de gradient dans l'espace crypté converge bien vers la valeur de $\hat{\beta}$ si le nombre d'itérations est suffisamment important comme le schéma ci-dessous le représente. Le problème alors rencontré concerne la quantité de ressources employée pour y parvenir.
Finalement, en utilisant le schéma de chiffrement de *Fan & Vercauteren* et en déterminant une estimation de $\hat{\beta}$, nous sommes parvenus aux résultats comparatifs présentés par la suite.

4. Les valeurs des coefficients de la régression obtenues après déchiffrement sont identiques pour les deux méthodes mais la première méthode s'avère plus rapide alors que la seconde est plus sécurisée.

5. Il s'agit bien d'une estimation de l'estimateur $\hat{\beta}$.

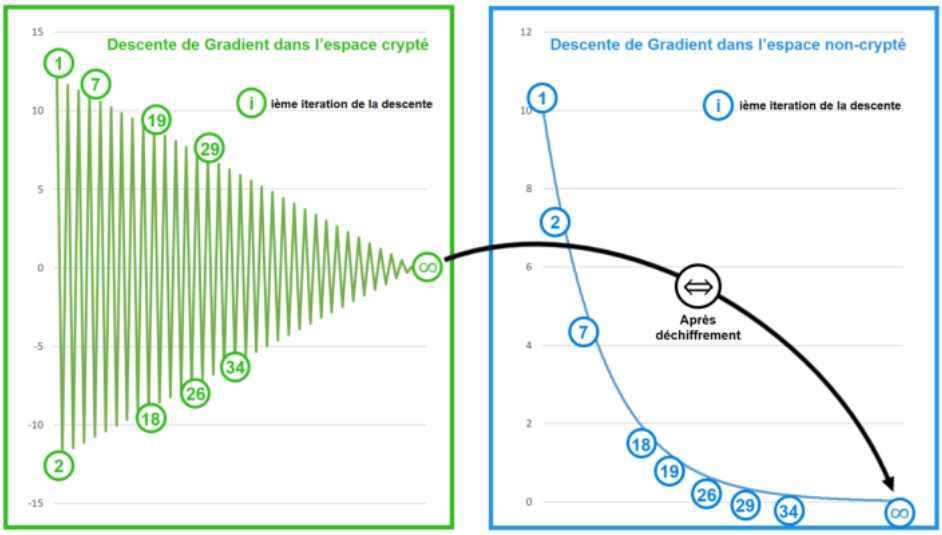


Figure : Schéma établissant le lien entre la descente de gradient dans l'espace crypté et dans l'espace non-crypté

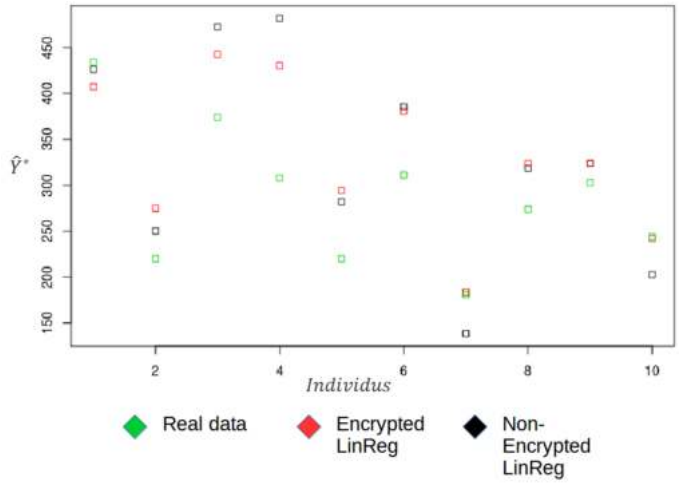


Figure : Comparaison des prédictions $\hat{Y}$ obtenues

Méthode Efficient Vector Homomorphic Encryption		Méthode Descente de Gradient (F&V)	
Avantages	**Inconvénients**	**Avantages**	**Inconvénients**
✓ Méthode de chiffrement simple à implémenter et à comprendre.	× Sécurité garantie par la méthode de chiffrement non-optimale.	✓ Méthode de chiffrement offrant un niveau de sécurité personnalisable.	× Schéma de chiffrement très exigeant et basé sur des notions complexes.
✓ Utilisation de la formule fermée des MCO pour calculer $\hat{\beta}$ dans $\mathcal{X}$.	× Evolution du bruit au cours des calculs dans $\mathcal{X}$ non contrôlée.	✓ La quantité de bruit introduite durant les opérations est déterminable en amont.	× Retourne seulement une estimation de l'estimateur $\hat{\beta}$ de β.
✓ Ressources matérielles nécessaires et temps d'exécution modérés.	× Nécessite des transformations/pré-calculs sur les données par le client.	✓ Le client fournit directement X et Y cryptés à l'opérateur de calculs.	× Très gourmand en ressources et possiblement inadapté au contexte actuel.
✓ Renvoi la valeur «réelle» de l'estimation de β.			

Figure : Tableau synthétique des méthodes employées

Néanmoins le temps de calcul important induit par les opérations dans l'espace crypté, ainsi que l'utilisation de données "simplement" pseudonymisées (le RGPD demeure contraignant sur l'utilisation de telles données, pour lever ces contraintes il est nécessaire d'anonymiser la base de données) nous ont décidés à envisager une approche alternative.

Élaboration d'une tarification automobile anonymisée

Ainsi nous avons décidé, à partir de données *anonymisées*, d'établir un modèle coût/fréquence de tarification en responsabilité civile automobile et de comparer les montants des primes ainsi estimés à ceux obtenus grâce au même modèle mais calibré, de manière habituelle, sur les polices individuelles du portefeuille (donc non-anonymisées).
Une donnée est dite anonymisée selon le RGPD s'il est strictement impossible de la ré-identifier. Il s'agit donc d'une procédure irréversible et délicate à réaliser si l'on souhaite conserver le maximum d'informations de nos données après l'anonymisation réalisée. Ainsi le choix de la méthode d'anonymisation est crucial et nous avons finalement opté pour une approche novatrice : plutôt que de supprimer ou de regrouper certaines variables ou modalités comme il est d'usage de le faire (approche horizontale), nous avons préféré agréger des polices en de très petites sous-classes (approche verticale) grâce à des méthodes d'apprentissage non-supervisé (clustering).
Nous avons donc testé et paramétré plusieurs méthodes de partitionnement, notamment un K-Means, un algorithme d'estimation de densité (OPTICS), une méthode de clustering hiérarchique (CAH) et un partitionnement par propagation d'affinité. Naturellement les critères d'acceptation du partitionnement sont ici différents du cadre habituel de leurs utilisations, comme le montre le schéma suivant.

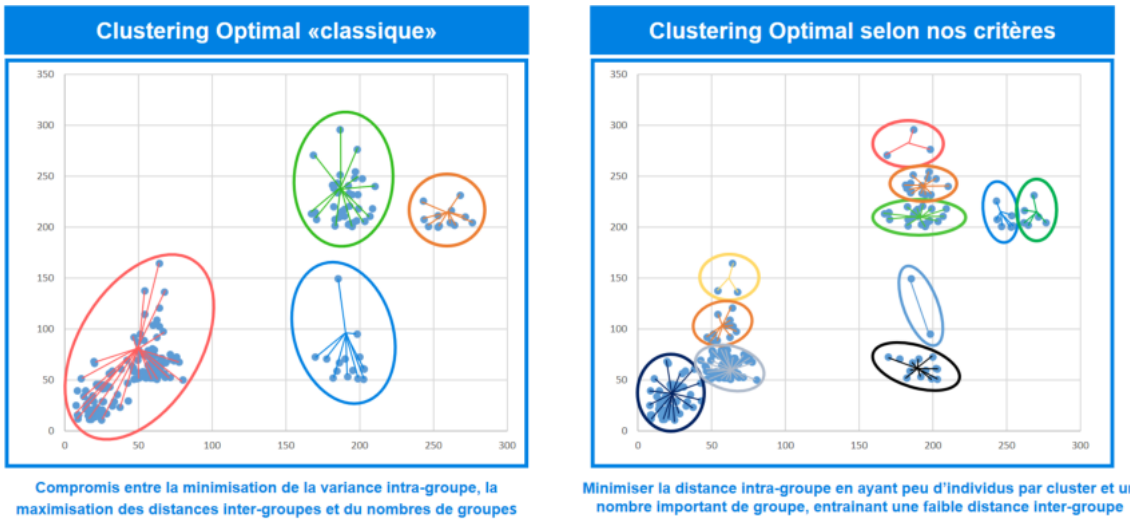


Figure : Schéma des différences de critères de sélection des modèles de clustering

Alors qu'il est usuellement souhaitable de minimiser les écarts entre les observations d'une même classe (de manière analogue, de maximiser les différences entre partitions) en faisant varier le nombre de clusters à former, ici l'enjeu consiste à obtenir un nombre de clusters le plus important possible contenant au moins deux polices

d'assurances chacunes (soit au maximum $N/2$ partitions si l'on compte N polices dans notre portefeuille d'assurés) en minimisant l'écart entre les observations au sein de chaque classe.
Nous commparons alors directement la sinistralité estimée selon notre modèle de tarification sur les données anonymisées par chacune des méthodes précédentes à la sinistralité estimée par ce même modèle sur les données individuelles. Cela nous permet de catégoriser les méthodes d'anonymisation précédemment évoquées selon leurs caractéristiques et la qualité des résultats obtenus. Ainsi on peut observer un écart de seulement 11% entre la sinistralité moyenne estimée sur des données anonymisées par estimation de densité et la sinistralité calculée grâce au modèle calibré sur les polices individuelles (modèle benchmark) :

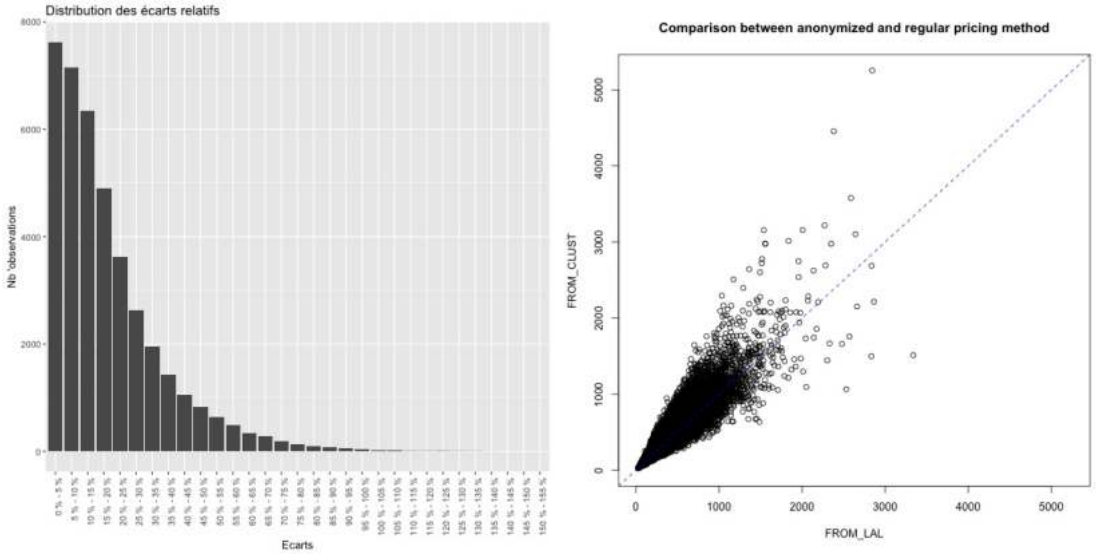


Figure : Distribution des écarts relatifs & comparaison des modèles

Plus d'informations sur l'élaboration de ces figures à l'introduction de la partie (4.3.1)

Table : Grandeurs caractéristiques (OPTICS(2,3,80))

sinistralité totale benchmark	+1.266e7
sinistralité totale anonymisée	+1.407e7
somme quadratique des écarts	+1.389e11
écart abs. (anonymisé - benchmark)	+1.408e6
écart relatif (à benchmark)	+11.12 %

Table des matières

1 Introduction

Le 25 mai 2018 est entré en vigueur au sein de l'union européenne le règlement général sur la protection des données (RGPD) renforçant la protection des données sensibles détenues par les compagnies privées et autres organisations, dont les assureurs exerçant sur le sol européen. Et dans le même temps, nous constatons chaque jour une augmentation considérable, à la fois de la quantité de données à traiter et à stocker pour l'assureur, mais également des ressources nécessaires - et notamment computationnelles - pour mener à bien ses études. L'avènement du traitement des données télématiques dans les procédures de tarification automobile, permettant d'obtenir les informations de conduites des assurés, en est un exemple concret. Si bien que nombre de compagnies ont recours à des prestataires extérieurs pour réaliser leurs calculs sur des architectures informatiques plus performantes.

Seulement, cette démultiplication des données et de leurs échanges entre différents intermédiaires engendrent naturellement de nouveaux risques pour les assureurs et en premier lieu le risque cyber.
Défini par le groupe de travail associé de l'institut des actuaires en 2017[6] comme "*Tout ce qui touche à l'atteinte, la violation ou la perte de données, ainsi qu'à des intrusions de réseau ou à la détérioration d'actifs aussi bien matériels qu'immatériels.*", le risque cyber présente désormais un risque majeur pour les assureurs qu'il leur faut analyser, mesurer et contrôler en particulier dans le nouveau cadre réglementaire européen introduit par le RGPD.
C'est donc dans ce contexte que s'inscrit l'étude menée au sein de ce mémoire. Notre objectif est de proposer des méthodologies à appliquer dans le cadre de modèles de tarification non-vie afin de contrôler en partie le risque cyber (en particulier interception/vols de données) en se conformant aux principes du RGPD (notamment de *protection des données dès la conception*). Nous proposerons ainsi deux procédures distinctes, la première basée sur le concept de pseudonymisation des données dans le cadre de délégation des calculs, puis la seconde sur l'application d'un modèle de tarification automobile à des données anonymisées.
Dans la première partie, nous étudierons donc différents schémas de cryptographie homomorphes permettant de réaliser certains calculs spécifiques dans l'espace crypté (et donc sans avoir à déchiffrer nos données). Nous les implémenterons sous `Python` et `R` et les utiliserons afin de concevoir une régression linéraire simple sur des données pseudonymisées par chiffrement.
Puis dans un second temps, nous nous intéresserons à une méthodologie alternative au chiffrement des données : l'anonymisation par agrégation de polices, en utilisant des méthodes d'apprentissage non-supervisé (OPTICS, K-Means, etc.). Nous réaliserons alors un modèle de tarification coût/fréquence pour une assurance responsabilité civile automobile de référence entraîné sur des polices individuelles, que nous comparerons à un modèle équivalent mais ajusté à partir de données ainsi anonymisées. Nous présenterons les résultats obtenus et analyserons notamment les limites inhérentes à cette procédure.

6. *Emergence du besoin en cyber assurance* - Institut des Acturaires, 2017

Ce mémoire s'articule donc autour de deux parties principales (III) et (IV) présentant précisément pour chacune des procédures concernées, l'aspect théorique et technique de leur implémentation ainsi que les résultats obtenus et leurs analyses pour mesurer l'intérêt de ces méthodes ainsi que leurs limites.
Cependant nous commencerons par définir plus en détails le cadre de notre étude ainsi que les concepts théoriques abordés tout au long du mémoire dans la partie suivante (II).

2 Présentation du cadre de l'étude

Notre étude s'intéresse à l'impact de la réglementation RGPD sur la tarification des produits d'assurance non-vie. Plus particulièrement, dans un souci de respect des données à caractère personnel ainsi que du droit à l'oubli, nous étudions la mise en application de procédures de pseudonymisation et d'anonymisation dans un processus de tarification automobile. Et ce, que les différentes étapes soient réalisées en interne ou par un prestataire extérieur. C'est pourquoi dans ce chapitre nous commencerons par définir les différentes notions élémentaires employées tout au long de ce mémoire, à la fois dans le domaine de la cryptologie et de l'anonymisation de données afin notamment de pouvoir distinguer rigoureusement et scientifiquement les nombreux algorithmes et autres méthodes issus de ces vastes champs de la recherche. Dans un deuxième temps, nous développerons les différentes structures de cryptographie couramment utilisées dans le cadre de la pseudonymisation ainsi que leurs implémentations les plus connues. Puis nous nous pencherons sur les schémas dits *homomorphes*, permettant de réaliser des opérations directement sur les données chiffrées, notamment en définissant, de manière non-exhaustive, les différentes sortes d'algorithmes associés et en explicitant clairement les quelques nuances et subtilités les différenciant.
Enfin en observant les limites inhérentes aux procédures de cryptages nous insisterons ainsi sur les ambitions de ce mémoire dans ce domaine et dans un contexte de tarification en branche non-vie, présenterons les alternatives potentielles : notamment via l'anonymisation des données et la préservation de la confidentialité grâce à une procédure d'aggrégation des données ligne à ligne.

2.1 Définition du cadre et des concepts mathématiques

Cette section a pour objectif de définir précisément et le plus simplement possible les notions couramment rencontrées dans le cadre de calculs et d'applications de méthodes d'apprentissages statistiques sur des données à caractère personnel.

2.1.1 Le réglement général sur la protection des données

Il s'agit du règlement européen n° 2016/679 concernant la protection des données à caractère personnel, le plus souvent dénommé par le sigle RGPD.
Ce texte, voté le 14 Avril 2016, est entré en vigueur le 25 Mai 2018 et prévoit un renforcement ainsi qu'une unification de la protection des données pour les individus dans le cadre européen, notamment en accentuant les droits individuels, grâce à la création d'un droit de portabilité des données personnelles ainsi que des dispositions relatives aux mineurs, mais également une responsabilisation accrue des acteurs traitant les données (les assureurs et les cabinets de conseil, entre autres) et un renforcement du rôle du régulateur à la fois au niveau national, à savoir la Commission Nationale de l'Informatique et des Libertés (CNIL) en France, mais également européen.
Ainsi les principales dispositions de ce règlement impactant notamment le milieu assurentiel et actuariel sont :

- *Un cadre harmonisé*, permettant une meilleure lisibilité législative au sein des pays de l'union européenne où désormais les mêmes règles s'appliquent.

- *Une application en dehors des frontières de l'UE*, notamment pour les entreprises non-européennes traitant des données issues de résidents ou d'organisations européennes.

- *Un consentement " explicite "* de la part des utilisateurs/clients quant à l'utilisation de leurs données personnelles ainsi que davantage de contrôle sur ces dernières.

- *Davantage de responsabilisation des acteurs et des sanctions plus importantes*, visant ainsi à réduire le nombre de formalités à fournir auprès du régulateur afin de faciliter les opérations des entreprises, tout en instaurant des pénalités renforcées en cas de non respect des règles pouvant s'élever jusqu'à 4% du chiffre d'affaires mondial annuel de la compagnie.

- *Un droit à l'effacement et à la portabilité des données personnelles*, permettant à chaque individu de pouvoir demander l'effacement de données à caractère personnel dans les meilleurs délais (une version allégée du droit à l'oubli, ce dernier concernant également les informations antérieures pouvant nuire à un individu) ainsi que la possibilité d'obtenir ses données dans un format structuré et interprétable de manière automatique afin notamment de pouvoir les transmettre à un autre organisme de traitement.

- *La nomination d'un délégué à la protection des données*, obligatoire lorsque les opérations réalisées sur ces données nécessitent un suivi régulier à grande échelle des individus concernés. Il doit alors s'assurer du respect de la réglementation mais également conseiller les différents acteurs des traitements sur son application et ainsi devenir le principal interlocuteur de l'entreprise avec l'autorité de contrôle. Cette fonction est donc naturellement à mettre en perspective avec la fonction de *contrôle de la conformité* comme définie au sein du pilier II (qualitatif) de la directive européenne Solvabilité II dans un cadre assurantiel.

- *Des principes de " protection des données dès la conception " et de " sécurité par défaut "*, qui imposent aux entreprises de prendre en considération les enjeux liés à la protection des données à caractère privé dès la conception de leurs produits et services, et ce durant la totalité de son processus d'élaboration.

Ce dernier point constitue la nouvelle disposition clef introduite par le RGPD, qui couplée à une responsabilisation accrue des différents acteurs, notamment dans le milieu assurantiel, conduit les entreprises du secteur à recourir à des processus spécifiques de protection des données comme *l'anonymisation* (qui peut permettre de se

soustraire à la plupart des dispositions du RGPD) ou à défaut la *pseudonymisation* (proposée au sein même de l'article 25 de ce réglement) [7], en plus par exemple du principe de minimisation des données.
C'est bien dans ce nouveau paradigme que s'inscrit donc les travaux réalisés au sein de ce mémoire, puisqu'il est désormais devenu fondamental pour les différents acteurs du secteur de prendre toutes les précautions nécessaires afin de garantir la confidentialité des données fournies par leurs clients, et notamment au cours de leurs manipulations lors des différentes études statistiques menées, sous peine de sanctions, en particulier financières, plus sévères.
Il s'agit donc d'un nouveau texte réglementaire et dorénavant fondamental pour le milieu actuariel et assurantiel offrant à la fois plus de souplesse pour ces entreprises mais également davantage de responsabilités.

2.1.2 La cryptographie

Étymologiquement, le terme de *cryptographie* trouve ses racines dans deux termes issus du grec ancien "*kruptos*" signifiant caché et "*graphin*", écrire.
Ce terme désigne donc la science consistant à coder, ou à chiffrer des *messages* (non nécessairement des chaînes de caractères, nous verrons par la suite le cas de vecteurs d'entiers dans $\mathbb{Z}^p$) dans le but de les rendre naturellement illisibles et/ou inexploitables par un tiers malveillant ne disposant pas du droit d'accès aux données. Cette notion, assez générale ici, est étroitement associée au principe de pseudonymisation des données que nous définirons dans la partie (2.1.4).
Il s'agit par ailleurs de ne pas confondre la *cryptographie* et la *cryptologie*, cette dernière l'englobant ainsi que la *cryptanalyse* (l'étude des schémas de chiffrement dans le but de les décrypter).
Beaucoup d'autre notions, ainsi qu'un vocabulaire précis sont associés à cette science. Nous développerons davantage ces éléments tout à fait spécifiques et fondamentaux de la cryptographie dans la prochaine section (2.2).

2.1.3 L'anonymisation des données

Également appelée *masquage de la donnée*, il s'agit d'une procédure permettant de rendre très délicate, si ce n'est impossible la ré-identification des individus, et plus généralement des données, à partir des informations obtenues à la suite des traitements statistiques et informatiques usuels.
Ce processus consiste donc à modifier le contenu ainsi que la structure des données afin de respecter les propriétés précédentes.
La Commission Nationale de l'Informatique et des Libertés (CNIL) estime par ailleurs que "*pour qu'une solution d'anonymisation soit efficace, elle doit empêcher toutes les parties d'isoler un individu dans un ensemble de données, de relier entre eux deux enregistrements dans un ensemble de données (ou dans deux ensembles de données séparés) et de déduire des informations de cet ensemble de données.*"

7. RGPD, Art.25 : "*le responsable du traitement met en œuvre [...] des mesures techniques et organisationnelles appropriées, telles que la pseudonymisation, qui sont destinées à mettre en œuvre les principes relatifs à la protection des données*"

La CNIL, par le biais de l'avis du G29 (groupe de travail des différentes CNIL européennes) propose donc trois critères afin de déterminer l'efficacité d'une procédure d'anonymisation :

- **L'individualisation :** Peut-on isoler un individu particulier au sein de la base de données ?
- **La corrélation :** Peut-on associer et relier deux ensembles de données (ou plus) distincts concernant un même individu ?
- **L'inférence :** Est-il possible de déduire des informations personnelles sur un individu précis ?

Enfin la norme internationale ISO/IEC20889 [8] énumère l'ensemble des techniques d'anonymisation et définit cette procédure comme le " *processus par lequel des informations personnellement identifiables (IPI) sont irréversiblement altérées de telle façon que le sujet des IPI ne puisse plus être identifié directement ou indirectement, que ce soit par le responsable du traitement des IPI seul ou en collaboration avec une quelconque autre partie.*"
On note ainsi que cette dernière définition introduit la notion d'irréversibilité de la procédure et rend donc impossible une éventuelle ré-identification, même volontaire, des données.

Au sein de ce processus, l'une des premières étapes consiste naturellement à faire fît de l'ensemble des variables personnelles et des identifiants des bases de données pouvant facilement décrire et formaliser précisément un individu en regroupant ces informations, comme par exemple :

- Noms, prénoms
- Âge
- Sexe
- Adresses, lieux, ...
- ID (n° de sécurité sociale, n° de série, ...)
- Informations biométriques
- Tout autres éléments distinctifs

Ensuite, il convient le plus souvent de recourir à un algorithme de hachage ou de chiffrement, en adéquation avec les besoins, les objectifs et les exigences à la fois réglementaires mais également spécifiques à chaque entreprise définis par le délégué à la protection des données. Le modèle de cryptographie précédemment défini s'intègre donc parfaitement à cette procédure. Par exemple, le recours à une méthode de *chiffrement avec suppression de la clef*, pour altérer définitivement les attributs de la base de données sans conserver de moyens pour retrouver (facilement) leurs valeurs initiales. Dans ce cadre également, la seule attaque " viable" consisterait à tester des clefs aléatoirement, ce qui s'avèrerait presque sûrement impossible à réaliser compte tenu de la puissance de calcul nécessaire et du nombre de clefs à déterminer (si par exemple, chaque modalité est encodée selon une clef différente).

8. "Techniques d'anonymisation" (Privacy enhancing data de-identification techniques)

L'anonymisation, au sens de la norme ISO/IEC20889, constitue alors une solution efficace pour les entreprises souhaitant s'exempter des contraintes de la RGPD puisque ces données ne permettent plus l'identification d'un individu.
Néanmoins cette méthode peut s'avérer difficile à mettre en place dans un cadre de production en raison des nombreuses contraintes induites par la procédure (en premier lieu de se priver des informations nominatives individuelles) et de ce fait de nombreuses compagnies ont davantage recours à une alternative commme par exemple la *pseudonymisation*.

2.1.4 La pseudonymisation

Le RGPD mentionne cette technique et définit à l'article 4 la *pseudonymisation* comme étant " *le traitement de données à caractère personnel de telle façon que celles-ci ne puissent plus être attribuées à une personne concernée précise sans avoir recours à des informations supplémentaires, pour autant que ces informations supplémentaires soient conservées séparément et soumises à des mesures techniques et organisationnelles afin de garantir que les données à caractère personnel ne sont pas attribuées à une personne physique identifiée ou identifiable.*"
Il s'agit donc d'une manière de préserver la confidentialité des données mais où il demeure possible de ré-identifier un individu lorsque l'on accède à des informations supplémentaires, à ce titre le RGPD considère les données pseudonymisées comme étant identifiables. [9]
Il existe de nombreuses méthodes de pseudonymisation, parmi lesquelles :

- *Un processus cryptographique munie d'une clef secrète*, permettant aux seuls individus possédant la clef de pouvoir décrypter les données et ainsi accéder aux données personnelles conservées dans le base de données. Cependant ici, la question de la confidentialité des données se trouve simplement reportée sur la sécurisation du registre des clefs secrètes.

- *Une fonction de hachage*, qui renvoie une valeur de taille fixe et ayant comme paramètre un attribut particulier ou même une liste d'attributs. Afin de limiter le risque de pouvoir déterminer un intervalle de valeurs prises par ces attributs, il est également possible de recourir à l'ajout d'un terme de bruit à l'attribut (aussi appelé sel) avant le hachage ce qui rend caduque l'utilisation de tables de dictionnaires (ou de rainbow tables) afin d'essayer de déterminer ces valeurs. Seul la technique du brut-force (potentiellement longue et coûteuse) devient envisageable.

La frontière entre *l'anonymisation* et la *pseudonymisation* peut parfois être mince ou peu explicite, c'est pourquoi on considérera ici (conformément à la définition

9. RGPD, Art. 26 " *Les données à caractère personnel qui ont fait l'objet d'une pseudonymisation et qui pourraient être attribuées à une personne physique par le recours à des informations supplémentaires devraient être considérées comme des informations concernant une personne physique identifiable* "

du RGPD) que la *pseudonymisation* constitue l'ensemble des méthodes d'anonymisation partielle qui permettent la ré-identification, voir figure (1) ci-dessous. A contrario une donnée une fois anonymisée doit donc être quasi-certainement impossible à relier à un individu.

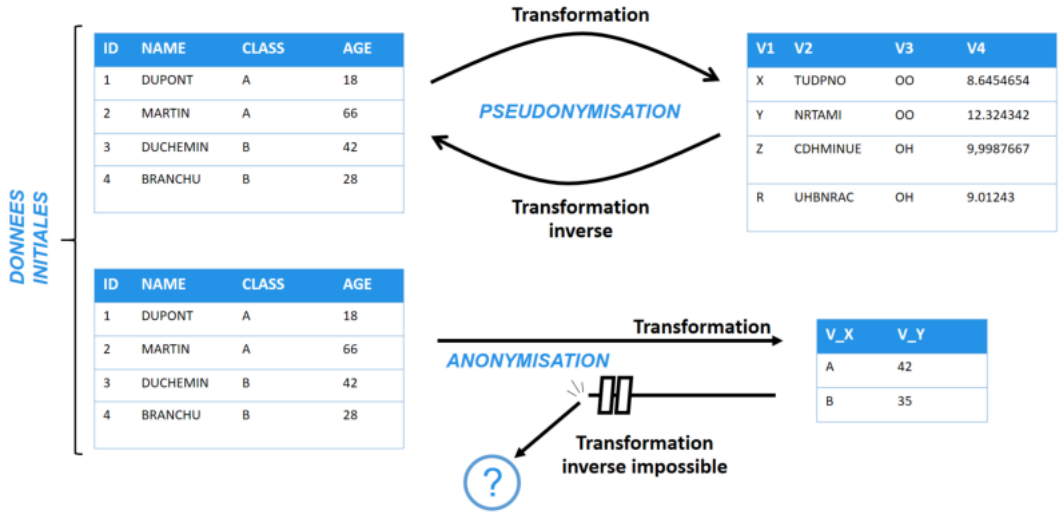


FIGURE 1 : Schéma des procédures de pseudonymisation et d'anonymisation

2.1.5 Le cadre et les objectifs de l'étude

Ce mémoire ayant pour but d'adapter certaines méthodologies actuarielles aux contraintes introduites par la RGPD, nos travaux se sont naturellement articulés autour des deux principales notions définies précédemment :

Dans un premier temps (III) nous nous intéresserons aux méthodes de pseudonymisation des données et plus particulièrement aux procédures de chiffrement offrant des propriétés homomorphiques singulières permettant d'effectuer certaines opérations sur les données cryptées, et ce sans jamais les déchiffrer.
Le contexte général de cette partie se trouve être le calcul délégué (cloud-computing) où un assureur pourrait simplement faire réaliser des calculs lourds/complexes aux serveurs d'un opérateur de service externe. Par exemple dans le cadre d'une tarification automobile, la situation où l'assureur souhaite utiliser une plateforme tierce pour calculer son tarif. Dans le cadre de l'utilisation de données télématiques (volumineuses) cette option est la plus souvent mise en place par la compagnie d'assurance.

Ainsi l'assureur encrypterait ses données x, et alors qu'il serait le seul capable de les déchiffrer, les envoie à l'opérateur externe afin que ce dernier puisse réaliser les calculs souhaités (dans le schéma (2), la fonction f) et renvoyer le résultat chiffré $f(x)$ sans que jamais l'opérateur n'ait eu besoin de déchiffrer les données pour réaliser les opérations ni d'avoir recours à aucune intervention de la part du client.
Puisque nous souhaitons nous placer dans un cadre de tarification non-vie nous nous intéresserons dans cette partie à la possibilité de réaliser une régression linéaire selon

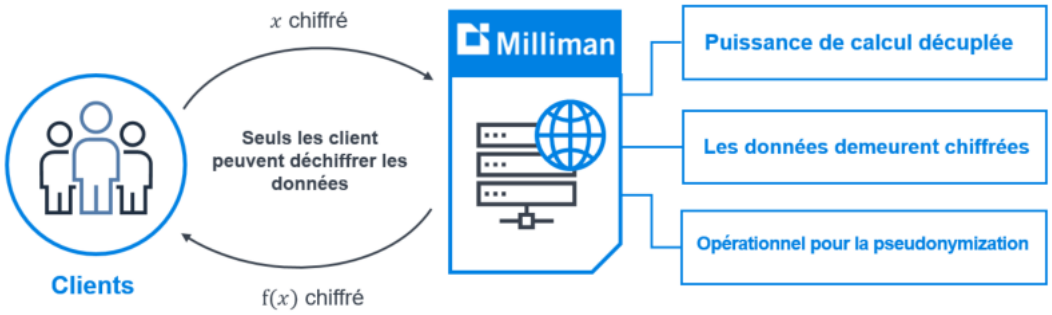


FIGURE 2 : Procédure de calculs délégués via chiffrement homomorphe

la procédure décrite ci-dessus. Pour ce faire nous étudierons, puis implémenterons les différents schémas de chiffrement permettant de réaliser les opérations nécessaires sur les données cryptées, puis nous mettrons en évidence, à la lumière des différents résultats obtenus, les limites inhérentes à ce modèle de calcul sur des données pseudonymisées.

Dans un second temps (IV) nous nous pencherons sur l'implémentation d'une procédure de tarification non-vie (assurance automobile) sur des données anonymisées à l'aide d'une procédure d'agrégation ligne à ligne (*clustering*). L'enjeu ici est d'être capable de montrer l'efficacité et la simplicité de la méthode d'anonymisation en ligne à ligne dans un cadre actuariel concret en proposant une alternative pertinente à la méthode présentée ci-dessus, ainsi qu'aux méthodes de regroupement et de suppressions de variables réalisées parfois en pratique (*k-anonymisation*, etc.).

2.2 Les différents paradigmes de la cryptographie

Comme nous l'avons précédemment évoqué la cryptographie joue un rôle important dans la protection de la confidentialité de données sensibles et plus particulièrement dans le cas de la pseudonymisation.
Il existe ainsi au sein de cette science plusieurs approches issues d'une succession d'évolutions qui ont conduit à des progrès significatifs dans ce domaine. C'est ainsi que l'on peut considérer deux méthodologies de chiffrement distinctes : les méthodes symétriques et asymétriques, au sein desquelles différents algorithmes aux propriétés singulières coexistent.
Nous présentons ici brièvement ces deux méthodes de chiffrement dont nous aurons besoin dans le suite du mémoire, en particulier dans la partie (III) au sujet du traitement d'une régression linéaire à partir de données pseudonymisées par chiffrement.

2.2.1 Description des méthodes de chiffrement

2.2.1.1 Les méthodes symétriques

Il s'agit d'une procédure qui utilise uniquement une *clef secrète* pour chiffrer et déchiffrer les messages, on qualifie d'ailleurs les algorithmes ayant recours à cette technique de *méthodes à clef secrète*. Cette clef peut se matérialiser de différentes façons : un vecteur ou une matrice d'entiers aléatoires par exemple. Cependant, dans ce contexte, la taille de la clef revêt une importance particulière puisque plus celle-ci est grande plus la sécurité du processus est garantie.

En somme, cette technique est comparable à celle d'un coffre contenant un secret et protégé par une serrure unique : quiconque souhaitant ouvrir le coffre (et découvrir/déchiffrer le secret) doit utiliser sa clef, et faire de même pour le refermer (chiffrement du secret). Voir la figure (3).
A titre indicatif certains algorithmes utilisent deux clefs privées distinctes, l'une afin de chiffrer les données et l'autre pour les déchiffrer. Mais ces procédures font tout de même partie intégrante des méthodes de chiffrement symétrique.

L'avantage de ces méthodes réside dans la relative simplicité de leur schéma conceptuel ainsi que dans leur efficacité (rapidité) au regard de la plupart des méthodes asymétriques. Néanmoins cette méthode présente plusieurs inconvénients majeurs :

- La clef secrète doit être transmise à chaque correspondant de façon confidentielle[10], ce qui pose donc la question de la sécurité du transfert et du stockage de ces clefs.

- La prolifération des clefs, dans le cas par exemple où chaque paire de correspondants d'un même réseau utilise une clef secrète différente pour leurs échanges. A titre d'exemple, pour que n individus puissent communiquer en toute confidentialité il est nécessaire d'avoir $n(n-1)/2$ clefs.

10. En théorie seuls les correspondants désirant chiffrer des données ont besoin de la clef. Ceux ne souhaitant que déchiffrer un message peuvent utiliser une méthode d'identification sécurisée.

- Le danger encouru en cas de compromission de la clef. Bien qu'une clef secrète générée au sein des différents algorithmes de chiffrement symétriques ne puisse être théoriquement déterminée de manière efficace (recours nécessaire à la méthode du brut-force), si un agent malveillant venait à être en possession de cette clef, il pourrait déchiffrer l'intégralité des conversations entre les différents protagonistes et même potentiellement se faire passer pour l'un d'entre eux en transmettant lui-même des messages chiffrés.

Les principaux algorithmes basés sur un chiffrement symétrique sont :

- *Le chiffre de Vernam* : méthode offrant une sécurité absolue tant que la clef secrète utilisée est de la même taille que le message à crypter et que cette clef est employée une unique fois.

- *Le DES* : algorithme crée en 1977 par le National Bureau of Standards, même s'il a récemment été découvert des vulnérabilités à certaines nouvelles attaques, il demeure aujourd'hui comme un des moyens de chiffrement les plus sûrs et les plus utilisés pour les données civiles (fédérales, commerciales ou privées).

- *L'AES* : procédure élaborée en 2000 par Rijmen et Daemen basée sur un principe assez similaire au DES qu'elle vise à remplacer. Seulement ici les messages à chiffrer peuvent être découpés en blocs de tailles variables et supérieures à celui du DES (64 bits) : 128, 196 ou 256 bits.

2.2.1.2 Les méthodes asymétriques

Ces algorithmes ont été élaborés dans les années 1970, afin notamment de résoudre les problèmes de prolifération et de transmission des clefs propres aux méthodes symétriques.
L'idée ici est de recourir à une paire de clefs dont on ne peut pas déduire l'une à partir de l'autre. Une clef secrète afin de déchiffrer les messages et l'autre publique qui ne permet que de les chiffrer. Par aileurs on qualifie les algorithmes ayant recours à cette technique de *méthodes à clef publique*.
Ainsi la clef publique de chiffrement peut être divulguée sans crainte puisqu'elle ne peut ni permettre de déchiffrer un message encrypté à l'aide de la clef privée ni fournir quelques informations sur cette dernière.

Pour reprendre notre exemple du coffre, la méthode asymétrique consiste à laisser la possibilité à chacun d'enfermer son secret dans le coffre à l'aide d'un cadenas (la clef publique) alors que seul les individus possédant la clef (la clef secrète) pourront y accéder. Voir figure (3).

Le principal avantage de ces méthodes est qu'il n'est donc pas nécessaire de partager d'informations confidentielles (la clef secrète) avant l'échange des messages cryptés.

Cependant ces algorithmes sont bien plus lents que les chiffrements symétriques, jusqu'à mille fois moins rapide.
On compte parmi les principales implémentations de ces méthodes asymétriques :

- *Le RSA* : inventé en 1977 par Rivest, Shamir et Adleman. Il est basé sur la difficulté de factoriser un très grand nombre en un produit de deux grands facteurs premiers via l'étude de la congruence sur les entiers. La méthode de génération des clefs et de chiffrement introduite par cet algorithme a par la suite été réutilisée dans un assez grand nombre d'autres implémentations.

- *Le DSA* : un algorithme de signature mais pas de chiffrement, il permet par exemple de sécuriser une transaction entre deux individus authentifiables.

La figure (3) ci-contre présente un schéma récapitulatif du fonctionnement de ces deux méthodes de chiffrement.

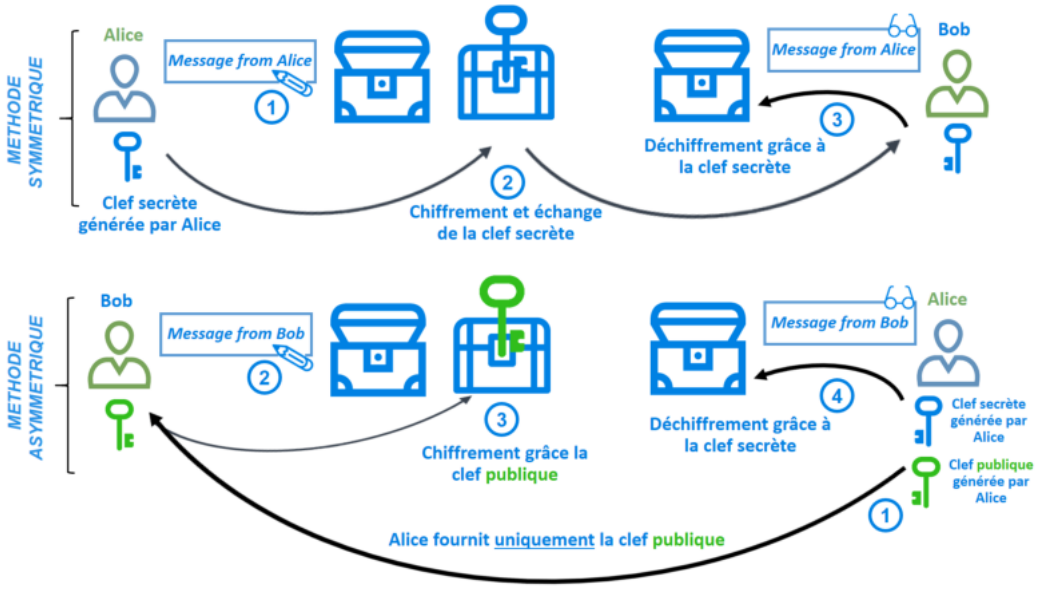


FIGURE 3 : Schéma des chiffrements symétriques et asymétriques

2.2.2 Les propriétés homomorphes de certains chiffrements

Certains algorithmes de chiffrement, qu'ils soient symétriques ou asymétriques, possèdent une propriété singulière : ils peuvent être des homomorphismes additifs et même parfois multiplicatifs. Ainsi une procédure bénéficiant d'une telle propriété permet de réaliser directement certaines opérations sur les éléments chiffrés, et ce dans l'espace crypté, tout en conservant naturellement leurs cohérences et leurs confidentialités.
Ce genre de chiffrement nous sera très utile par la suite lors de la réalisation de la procédure de pseudonymisation dans laquelle nous souhaitons réaliser une régression linéaire sur des données chiffrées dans l'espace crypté.

On rappelle à ce titre qu'un homomorphisme entre deux anneaux A et B se caractérise par une application $\mathcal{E} : A \to B$, qui vérifie :

- $\forall a, b \in A, \mathcal{E}(a +_A b) = \mathcal{E}(a) *_+ \mathcal{E}(b)$
- $\forall a, b \in A, \mathcal{E}(a \times_A b) = \mathcal{E}(a) *_\times \mathcal{E}(b)$
- $\mathcal{E}(1_A) = \mathcal{E}(1_B)$

où $+_A, \times_A$ et 1_A (resp. pour B) représente l'opération additive, multiplicative et le neutre multiplicatif pour A (resp. B).

Exemple : Homomorphisme de groupes simple
On définit plus simplement un homomorphisme de groupes par une application $\mathcal{E} : (\mathcal{H}, .) \to (\mathcal{H}', \diamond)$ tel que :

$$\forall i, j \in \mathcal{H}, \mathcal{E}(i \,.\, j) = \mathcal{E}(i) \diamond \mathcal{E}(j)$$

Ainsi la fonction $\mathbb{R}_+^* \to \mathbb{R}_+$, $x \mapsto \log(x)$, vérifie :

$$\log(x \times x') = \log(x) + \log(x')$$

Il s'agit donc d'un homorphisme de groupes de $(\mathbb{R}_+^*, \times)$ dans $(\mathbb{R}, +)$. On peut même parler d'isomorphisme ici puisque l'application log de $\mathbb{R}_+^*$ dans $\mathbb{R}$ est bijective.

2.2.2.1 Notation et aspects théoriques d'un chiffrement homomorphe

On se place dans cette sous-partie dans un cadre théorique au sein duquel on souhaite modéliser la propriété d'*homomorphisme* additif et multiplicatif d'une méthode de chiffrement quelconque (symétrique ou asymétrique).
En effet, comme notre objectif est de pouvoir réaliser une régression linéaire sur nos données chiffrées dans l'espace crypté, il est évident de considérer la réalisation de ces opérations d'additions et de multiplications dans cet espace comme essentielle au bon fonctionnement de notre procédure.

On suppose alors à titre d'exemple, une donnée d non cryptée que l'on souhaite chiffrer tel qu'on obtienne $\mathcal{E}(d)$ et que l'opération de déchiffrement soit $\mathcal{E}'$ tel que l'on retrouve naturellement :

$$\mathcal{E}'(\mathcal{E}(d)) = d$$

Pour pouvoir exécuter des additions (1) et des multiplications (2), sans décrypter les données il suffit qu'il existe une opération $*_1$ et $*_2$ tel que :

$$\mathcal{E}(d_1 + d_2) = \mathcal{E}(d_1) *_1 \mathcal{E}(d_2) \tag{1a}$$

$$\mathcal{E}(d_1 \times d_2) = \mathcal{E}(d_1) *_2 \mathcal{E}(d_2) \tag{1b}$$

Ainsi, afin de réaliser les calculs $(+,\times)$ sur les données initiales il suffit de réaliser respectivement les opérations $(*_1,*_2)$ sur les données cryptées.
Il est important de noter que les opérations $(*_1,*_2)$ dans l'espace crypté ne sont pas nécessairement identiques à leurs correspondances dans l'espace non-crypté [11], il en est même très rarement le cas.

Si on souhaite par exemple modéliser une régression linéaire monovariée usuelle dans l'espace non crypté $y = a \times x + b$, où x et y sont des vecteurs de réels et a et b des scalaires. Il nous suffit de calculer $\mathcal{E}(y) = (\mathcal{E}(a) *_2 \mathcal{E}(x)) *_1 \mathcal{E}(b)$ dans l'espace crypté, puis de déchiffrer le résultat dans l'espace non crypté tel que :

$$\mathcal{E}'(\mathcal{E}(y)) = \mathcal{E}'\left[(\mathcal{E}(a) *_2 \mathcal{E}(x)) *_1 \mathcal{E}(b)\right]$$

$$\mathcal{E}'(\mathcal{E}(y)) = \mathcal{E}'\left[\mathcal{E}(a) *_2 \mathcal{E}(x)\right] + \mathcal{E}'\left[\mathcal{E}(b)\right]$$

$$\mathcal{E}'(\mathcal{E}(y)) = \mathcal{E}'\left[\mathcal{E}(a)\right] \times \mathcal{E}'\left[\mathcal{E}(x)\right] + b$$

$$y = a \times x + b$$

On dit qu'une méthode de cryptage est homomorphe pour l'addition et la multiplication si elle vérifie les propriétés (1a) et (1b).
Alors de tels chiffrements permettent de réaliser à la fois des additions et des multiplications dans l'espace crypté, c'est pourquoi nous nous intéresserons dans la partie (III) uniquement à ce type de chiffrement. Cependant avant d'expliciter une implémentation en python d'un de ces schémas et de réaliser une régression linéaire sur des données chiffrées à l'aide de celui-ci, nous présenterons, à titre introductif, le tout premier schéma homomorphe réalisé ce qui nous permettra de plus facilement aborder le formalisme ainsi que certaines singularités de ces chiffrements.

2.2.2.2 Cryptage homomorphe additif et multiplicatif

Chiffrage et opérations

La méthode en question proposée par C. Gentry [12], définit un système qui chiffre les bits afin de pouvoir exécuter un nombre quelconque d'opérations logiques de type XOR (ou exclusif, noté ici $\oplus$) et ET (conjonction logique) sur ces bits chiffrés.
On définit ici comme clef secrète (soit un paramètre de $\mathcal{E}$ dans la partie précédente) un long nombre entier impair p. Par ailleurs on peut définir une sorte de "multiple" de p, un nombre sous la forme $qp + e$, avec e un "petit" nombre correspondant au bruit et q un très grand nombre (i.e $q \gg p$). Il est alors impossible de le différencier d'un nombre quelconque, cependant celui qui connaît p peut facilement retrouver q en divisant le nombre précédent par p.

11. i.e, il n'est pas impossible d'avoir : $*_1 \neq +$ et $*_2 \neq \times$

12. Cette procédure de chiffrement a été définie par C. Gentry, M. Van Dijk, S. Halevi et V. Vaikuntanathan en 2009.

On peut donc chiffrer un bit b (i.e $b = 0$ ou $b = 1$) par : $\mathcal{E}(b) = pq + b + 2\epsilon$ où ϵ représente le bruit. On obtient un système de chiffrement homomorphe pour le XOR et le ET, soit pleinement homomorphe, où l'on note $\mathcal{E}'$ la fonction de déchiffrement tel que $\mathcal{E}' := (\mathcal{E})^{-1}$, définie par :

$$\mathcal{E}'(c_i) = (c_i \bmod p) \bmod 2 = b_i$$

Exemple :
En prenant des valeurs numériques simples où $p = 111$, $q_1 = 1000$, $q_2 = 500$ et $\epsilon_1 = 3$, $\epsilon_2 = 1$, on obtient comme chiffrement pour les valeurs $b_1 = 0$ et $b_2 = 1$:

$$c_1 = \mathcal{E}(b_1) = 1000 \times 111 + 2 \times 3 = 111006$$

$$c_2 = \mathcal{E}(b_2) = 500 \times 111 + 1 + 2 \times 1 = 55503$$

En déchiffrant on a bien :

$$\mathcal{E}'(c_1) = (11106 \bmod 111) \bmod 2 = 6 \bmod 2 = 0 = b_1$$

$$\mathcal{E}'(c_1) = (55503 \bmod 111) \bmod 2 = 3 \bmod 2 = 1 = b_2$$

Preuve : Soient b_1 et b_2 deux bits chiffrés respectivement tels que :

$$\mathcal{E}(b_1) = c_1 = q_1 p + b_1 + 2\epsilon_1$$
$$\mathcal{E}(b_2) = c_2 = q_2 p + b_2 + 2\epsilon_2$$

Où : $\epsilon_1 \sim \epsilon_2 \ll p \ll q_1 \sim q_2$

On remarque que le bruit présent dans chacun des chiffrements, $2\epsilon_i$ est toujours pair. En connaissant la valeur de p on peut calculer le reste de la division par p de :

$$c_1 + c_2 = (q_1 + q_2)p + 2(\epsilon_1 + \epsilon_2) + (b_1 + b_2)$$

On obtient alors : $2(\epsilon_1 + \epsilon_2) + (b_1 + b_2)$, or $2(\epsilon_1 + \epsilon_2)$ est pair donc il n'influe pas sur la parité du reste, et ainsi on en déduit que si le reste est impair cela signifie que b_1 est impaire ET b_2 pair (et inversement). On remarque donc que :

$$([\mathcal{E}(b_1) + \mathcal{E}(b_2)] \bmod p) \bmod 2 \equiv b_1 \oplus b_2$$

$$\Leftrightarrow \mathcal{E}'(c_1 + c_2) \equiv \mathcal{E}'(c_1) \oplus \mathcal{E}'(c_2)$$

$$\Leftrightarrow \mathcal{E}\left[\mathcal{E}'(c_1 + c_2)\right] \equiv \mathcal{E}\left[\mathcal{E}'(c_1) \oplus \mathcal{E}'(c_2)\right]$$

$$\Leftrightarrow \mathcal{E}(b_1) + \mathcal{E}(b_2) \equiv \mathcal{E}(b_1 \oplus b_2)$$

b_1	b_2	$b_1 \oplus b_2$
0	0	0
0	1	1
1	0	1
1	1	0

TABLE 1 : Table de vérité du XOR

Puisque l'on peut facilement vérifier à la main que la table de vérité du XOR ci-dessous est toujours vérifiée par la proposition précédente, on peut en déduire la propriété d'homomorphisme additif de cette méthode de chiffrement, où $*_1 = +$.

Exemple : Application numérique
En reprenant les valeurs des paramètres et des données à chiffrer de l'exemple précédent ($b_1 \neq b_2$) on a bien :

$$([c_1 + c_2] \mod 111) \mod 2 = (166509 \mod 111) \mod 2 = 9 \mod 2 = 1 \equiv b_1 \oplus b_2$$

Nous allons désormais nous intéresser à la propriété d'homomorphisme multiplicatif de cette procédure de cryptage, en calculant le reste de la division par p de :

$$c_1 c_2 = (q_1 q_2 p + 2q_1\epsilon_2 + q_1 b_2 + 2q_2\epsilon_1 + q_2 b_1)p + 2(2\epsilon_1\epsilon_2 + \epsilon_1 b_2 + \epsilon_2 b_1) + b_1 b_2$$

On obtient alors $2(2\epsilon_1\epsilon_2 + \epsilon_1 b_2 + \epsilon_2 b_1) + b_1 b_2$, dont la parité ne dépend que de $b_1 b_2$, or $b_1 b_2$ est impair si et seulement si b_1 ET b_2 le sont. On remarque donc que :

$$\begin{aligned}
&([\mathcal{E}(b_1) \times \mathcal{E}(b_2)] \mod p) \mod 2 \equiv b_1 \wedge b_2 \\
\Leftrightarrow\ &\mathcal{E}'(c_1 \times c_2) \equiv \mathcal{E}'(c_1) \wedge \mathcal{E}'(c_2) \\
\Leftrightarrow\ &\mathcal{E}\left[\mathcal{E}'(c_1 \times c_2)\right] \equiv \mathcal{E}\left[\mathcal{E}'(c_1) \wedge \mathcal{E}'(c_2)\right] \\
\Leftrightarrow\ &\mathcal{E}(b_1) \times \mathcal{E}(b_2) \equiv \mathcal{E}(b_1 \wedge b_2)
\end{aligned}$$

Nous pouvons à nouveau nous assurer à la main que la table de vérité du ET ci-dessous est toujours vérifiée par la proposition précédente, on peut en déduire la propriété d'homomorphisme multiplicatif de cette méthode de chiffrement, où $*_2 = \times$.

Exemple : Application numérique
En reprenant les valeurs des paramètres et des données à chiffrer de l'exemple précédent ($b_1 \neq b_2$) on a bien :

$$([c_1 \times c_2] \mod 111) \mod 2 = (6161166018 \mod 111) \mod 2 = 18 \mod 2 = 0 \equiv b_1 \wedge b_2$$

b_1	b_2	$b_1 \wedge b_2$
0	0	0
0	1	0
1	0	0
1	1	1

TABLE 2 : Table de vérité du ET

□

Nettoyage périodique du bruit

Un problème se pose quant à la quantité de bruit introduite par les différentes opérations successives réalisées au sein de l'espace crypté : par exemple pour l'homomorphisme additif (le XOR) on passe d'un niveau de bruit pour b_1 (resp. b_2) de $2\epsilon_1$ (resp. $2\epsilon_2$) à $2(\epsilon_1 + \epsilon_2)$ pour $b_1 \oplus b_2$, soit à peu près le double. Tandis que le bruit issu de l'homomorphisme multiplicatif (le ET) est de l'ordre d'environ $4\epsilon_1\epsilon_2$.
On remarque donc qu'au bout d'un certain nombre d'opérations (en particulier les ET) il est possible d'atteindre un niveau de bruit suffisamment important (de l'ordre de p) tel qu'il deviendrait impossible de pouvoir déchiffrer les données puisque le reste de la division serait altéré, en effet ce dernier augmenterait alors potentiellement de un ou plus, faussant la parité attendue du reste pour retrouver les comportements des opérateurs logiques associés.

Exemple : Bruit trop important
En utilisant cette fois les paramètres $p = 11$, $q_1 = 100$, $q_2 = 50$ et $\epsilon_1 = 3$, $\epsilon_2 = 1$, on obtient comme chiffrement pour les valeurs $b_1 = 0$ et $b_2 = 1$:

$$\begin{array}{l} \mathcal{E}(b_1) = c_1 = 1106 \\ \mathcal{E}(b_2) = c_2 = 553 \end{array} \quad \text{où} \quad \begin{array}{r} \mathcal{E}'(c_1) = (1106 \bmod 11) \bmod 2 = 6 \bmod 2 = 0 \\ \mathcal{E}'(c_2) = (553 \bmod 11) \bmod 2 = 3 \bmod 2 = 1 \end{array}$$

On remarque que l'homomorphisme additif est bien vérifié :

$$([c_1 + c_2] \bmod 11) \bmod 2 = (1659 \bmod 11) \bmod 2 = 9 \bmod 2 = 1 \equiv b_1 \oplus b_2$$

En revanche l'homomorphisme multiplicatif ne donne pas le résultat escompté :

$$([c_1 \times c_2] \bmod 11) \bmod 2 = (611618 \bmod 11) \bmod 2 = 7 \bmod 2 = 1 \not\equiv b_1 \wedge b_2$$

En effet ici on se trouve dans le cas où le bruit issu de la multiplication des termes chiffrés est trop grand comparé à p tel que :

$$\begin{aligned} c_1c_2 &= (q_1q_2p + 2q_1\epsilon_2 + q_1b_2 + 2q_2\epsilon_1 + q_2b_1)p + 2(2\epsilon_1\epsilon_2 + \epsilon_1b_2 + \epsilon_2b_1) + b_1b_2 \\ &= (100 \times 50 \times 11 + 2 \times 100 + 100 \times 1 + 2 \times 50 \times 3)p + 2(2 \times 3 + 3) \\ &= 55600p + \underbrace{18}_{p+7} \\ &= 55601p + 7 \end{aligned}$$

Ce qui conduit inévitablement à fausser la parité de $c_1c_2 \bmod p$.

Nous sommes donc limités à un certain nombre d'opérations réalisable sur les données cryptées, l'idée afin de se séparer de cette contrainte limitative serait donc de nettoyer régulièrement le bruit issu des différentes opérations lorsque celui-ci devient trop important afin de pouvoir continuer les calculs. Pour ce faire on pourrait simplement déchiffrer le résultat partiel (en divisant par p) avant qu'il ne soit trop brouillé puis le chiffrer de nouveau avec un bruit initial. Cependant cela nécessiterait la connaissance de la clef secrète (p) par l'opérateur de calcul afin de déchiffrer le résultat intermédiaire et donc annihilerait finalement toute la procédure de cryptage homomorphe censée pouvoir manipuler des données en conservant justement leurs confidentialités. Il ne s'agit donc pas d'une " solution" viable.

Une astuce conjointe à la méthode proposée par C. Gentry consiste à "simplement" effectuer le déchiffrement mais de manière *homomorphe*, afin de ne jamais avoir à divulguer la clef privée p. Cette phase est alors appelée *bootstrapping*. Pour ce faire, il faut bien comprendre que l'objectif est donc de réaliser une division par p des messages chiffrés et d'examiner la parité du reste au sein de l'espace chiffré mais en utilisant uniquement les opérations disponibles dans l'espace non-crypté. Dans l'exemple précédent il s'agit donc de transcrire cette division euclidienne de l'espace crypté par une succession de XOR et de ET dans l'espace non-crypté.

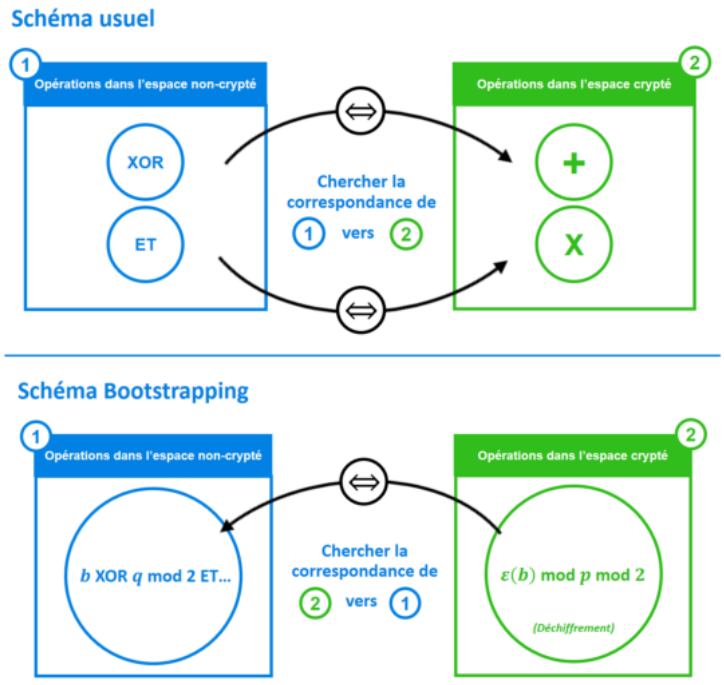


FIGURE 4 : Comparatif du schéma usuel et du nettoyage du bruit par bootstrapping

L'étape de *bootstrapping* constitue donc en quelques sortes une inversion du schéma de réflexion présenté jusqu'à présent où nous essayions de déterminer les calculs à réaliser dans l'espace crypté afin de satisfaire ceux dans l'espace non-crypté, alors que dans ce cadre de réduction du bruit il s'agit plutôt de trouver les opérations à

mener dans l'espace non-crypté afin de pouvoir réaliser l'opération de déchiffrement dans l'espace crypté, comme nous le montre le schéma (4).
Néanmoins pour que cette étape soit un succès il est nécessaire que les opérations induites par le déchiffrement homomorphe soit réalisables sans que le bruit généré ne soit trop important (i.e que le nombre d'opérations dans l'espace non-crypté soit raisonnable). C'est pourquoi les nombres p et q_i se doivent d'être également très grands (avec toujours $q_i \gg p$).

2.2.2.3 Formalisme des chiffrements homomorphes

Afin de définir les différents types de chiffrements homomorphes existants et pour déterminer les propriétés des schémas que nous utiliserons par la suite, il convient de préciser certains éléments et concepts fondamentaux les différenciants et pour ce faire nous nous plaçons dans un cadre tout à fait théorique, dont les définitions et les notations sont issues de *A Guide to Fully Homomorphic Encryption* (F. Armknecht et al, 2015) :

Soit un espace booléen où $\mathcal{P} = \{0, 1\}$ est l'espace des données non-cryptées et une famille F de fonctions de tuples de données non-cryptées vers $\mathcal{P}$. On peut définir une de ces fonctions comme un circuit logique booléen [13] que l'on notera C, tel que $C(m_1, m_2, \ldots, m_n)$ correspond à l'évaluation du tuple de données non-cryptées $(m_1, m_2, \ldots, m_n)$.

Définition (*$\mathcal{C}$-schéma d'évaluation*) Soit $\mathcal{C}$ un ensemble de circuits booléens, un $\mathcal{C}$-schéma d'évaluation pour $\mathcal{C}$ est un tuple d'algorithmes tel que :

- $\text{Gen}(\lambda, \alpha)$ est la fonction génératrice, qui prend en entrée deux paramètres : λ le paramètre de sécurité ($\sim p$ dans le paragraphe précédent) et α un paramètre optionnel. Cet algorithme fournit en sortie trois clefs : sk, la clef secrète utilisée pour décrypter les données chiffrées, pk la clef publique qui servira pour le chiffrement et evk la clef d'évaluation [14].

- $\text{Enc}(pk, m)$ est l'algorithme d'encryptage qui prend en entrée la clef publique et la donnée m à chiffrer. En sortie on obtient c, le message crypté.

- $\text{Eval}(evk, C, c_1, \ldots, c_n)$ est la fonction d'évaluation. Elle prend en entrée un circuit booléen $C \in \mathcal{C}$ ainsi qu'un tuple composé à la fois de données cryptées et de résultats de précédentes évaluations. Cette fonction fournit en sortie le résultat de l'évaluation dans l'espace crypté (le résultat des opérations sur les données dans l'espace crypté selon le circuit C).

13. Il s'agit d'une succession de fonctions logiques booléennes additives (OU) et multiplicatives (ET).

14. Cette clef peut être vu comme une autorisation à réaliser des calculs sur les données dans l'espace crypté. Elle est souvent similaire à la clef publique pk.

- Dec(sk, c) est l'algorithme de décryptage qui prend en entrée la clef secrète ainsi qu'un message chiffré (ou le résultat d'une évaluation) c. En sortie on obtient le message déchiffré m.

De plus on note $\mathcal{X}$ l'espace des données cryptées directement issues de Enc(), $\mathcal{Y}$ l'espace d'évalution [15] et $\mathcal{Z}$ l'union de ces deux espaces, soit $\mathcal{Z} := \mathcal{X} \cup \mathcal{Y}$. De plus $\mathcal{Z}^*$ correspond à l'espace des tuples de tailles arbitraires issus d'éléments de $\mathcal{Z}$.
On définit également les espaces $\mathcal{K}_p, \mathcal{K}_s, \mathcal{K}_e$ pour respectivement les clefs publiques pk, secrètes sk et d'évaluation evk et $\mathcal{A}$ un espace de taille arbitraire. On peut donc définir les domaines de définitions et d'applications des différentes fonctions présentées ci-dessus ainsi :

$$\text{Gen}(\lambda, \alpha) : \mathbb{N} \times \mathcal{A} \to \mathcal{K}_p \times \mathcal{K}_s \times \mathcal{K}_e$$

$$\text{Enc}(pk, m) : \mathcal{K}_p \times \mathcal{P} \to \mathcal{X}$$

$$\text{Dec}(sk, c) : \mathcal{K}_s \times \mathcal{Z} \to \mathcal{P}$$

$$\text{Eval}(evk, C, c_1, \ldots, c_n) : \mathcal{K}_e \times \mathcal{C} \times \mathcal{Z}^* \to \mathcal{Y}$$

On peut finalement formaliser les espaces $\mathcal{X}$ et $\mathcal{Y}$ tels que :

$$\begin{aligned} \mathcal{X} &= \{c \mid \mathbb{P}[\text{Enc}(pk, m) = c] > 0, m \in \mathcal{P}\} \\ \mathcal{Y} &= \{z \mid \mathbb{P}[\text{Eval}(evk, C, c_1, \ldots, c_n) = z] > 0, c_i \in \mathcal{Z} \text{ et } C \in \mathcal{C}\} \end{aligned}$$

Définition (*Déchiffrement efficace*) Un $\mathcal{C}$-schéma d'évaluation est une procédure qui déchiffre efficacement si :

$$\mathbb{P}[\text{Dec}(sk, \text{Enc}(pk, m)) = m] = 1 \qquad \forall m \in \mathcal{P}$$

Où sk et pk sont issus de Gen(λ, α).

Ainsi un tel schéma serait capable de déchiffrer une donnée cryptée sans erreur presque sûrement (probabilité égale à 1).

Définition (*Évaluation efficace*) Un $\mathcal{C}$-schéma d'évaluation est une procédure qui évalue efficacement tout les circuits de $\mathcal{C}$ si $\forall c_i \in \mathcal{X}$ tel que Dec$(sk, c_i) = m_i$ on a :

$$\mathbb{P}[\text{Dec}(sk, \text{Eval}(evk, C, c_1, \ldots, c_n)) = C(m_1, \ldots, m_n)] = 1 - \epsilon(\lambda) \qquad \forall C \in \mathcal{C}$$

Où sk, pk et evk sont issus de Gen(λ, α) et ϵ une fonction négligeable telle que $\epsilon(\lambda) \sim 0$.

15. En général l'espace d'évaluation $\mathcal{Y}$ peut être disjoint de l'espace des données cryptées $\mathcal{X}$.

Cela signifie que le déchiffrement de l'évaluation homomorphe issu d'un circuit compatible[16] de $\mathcal{C}$ conduit au résultat escompté de manière quasi-certaine (probabilité proche de 1).

Définition (*Compacité*) Un $\mathcal{C}$-schéma d'évaluation est une procédure compacte s'il existe un polynôme p tel que pour tout triplet de clefs (sk, pk, evk) issu de $\text{Gen}(\lambda, \alpha)$ et pour tout circuit $C \in \mathcal{C}$ et toutes données cryptées $c_i \in \mathcal{X}$, la taille de l'évaluation issue de $\text{Eval}(evk, C, c_1, \ldots, c_n)$ peut être majorée par $p(\lambda)$ bits, indépendamment de la taille du circuit.

En d'autres termes cela signifie que la taille des données cryptées augmente peu durant les calculs dans l'espace crypté $\mathcal{X}$ et que la taille du résultat de ces calculs ne dépend que du paramètre de sécurité λ.

Les deux définitions d'efficacités mentionnées ci-dessus sont des propriétés nécessaires pour pouvoir considérer un $\mathcal{C}$-schéma d'évaluation comme une véritable procédure de chiffrement. Par ailleurs on dit qu'un tel schéma est efficace s'il respecte à la fois les conditions d'un déchiffrement et d'une évaluation efficace.

2.2.2.4 Les différents chiffrements homomorphes

Nous avons déjà vu dans la sous-partie précédente un exemple de chiffrement qui s'avère être pleinement homomorphe, mais il existe plusieurs types de chiffrements homomorphes possédant des propriétés singulières, le tout est récapitulé dans le schéma (5) :

Définition (*Chiffrement partiellement homomorphe*) Un $\mathcal{C}$-schéma d'évaluation est une procédure de chiffrement partiellement homomorphe s'il s'agit d'un schéma efficace.

Un tel schéma ne nécessite pas de contraintes ni sur sa compacité ni sur les circuits compatibles de $\mathcal{C}$ utilisés pour l'évaluation.
En conséquence la taille des données cryptées évaluées au sein de l'espace $\mathcal{X}$ peut considérablement augmenter et sans limites, par ailleurs l'ensemble des opérateurs au sein de l'espace crypté faisant sens dans l'espace non crypté n'est soumis à aucune contraintes (ni de tailles ni de types d'opérations). Dans la littérature anglo-saxonne on retrouve ce type de chiffrement sous le nom de *Somewhat Homomorphic Encryption* (SHE).

Définition (*Chiffrement homomorphe contrôlé*) Un $\mathcal{C}$-schéma est une procédure de chiffrement homomorphe contrôlée s'il s'agit d'un chiffrement partiellement homomorphe qui prend en entrée de la fonction Gen() le paramètre facultatif α, tel que $\alpha = z$ correspond à la profondeur maximum des circuits pouvant être évalués dans l'espace crypté et qu'il vérifie également la propriété de compacité de sorte que la taille des éléments évalués dans l'espace crypté soit indépendante de z. On parle en anglais de *Levelled Homomorphic Encryption* (LHE).

16. Un circuit compatible de $\mathcal{C}$ constitue l'ensemble des opérations réalisables dans l'espace crypté $\mathcal{X}$ ayant un sens dans l'espace non crypté $\mathcal{P}$.

Mise à part leurs profondeurs z, il n'y a aucune autre contrainte sur $\mathcal{C}$. Si l'on souhaite définir $\mathcal{C}$ comme l'ensemble des circuits binaires de profondeur inférieure ou égale à z [17], alors on parle de chiffrement pleinement homomorphe contrôlé (*Levelled fully homomorphic encryption* - LFHE).

Il est parfois délicat de comprendre la différence entre les chiffrements homomorphes contrôlés et les chiffrements partiellement homomorphes. Or dans le cas de ces derniers la profondeur de leurs circuits logiques associés peut évoluer selon le choix de paramétrage, ainsi la taille des données cryptées évaluées est généralement amenée à augmenter avec la profondeur des circuits. Or pour les chiffrements homomorphes contrôlés, la profondeur maximale est définie par le paramètre α et la taille des données cryptées évaluées n'en dépend pas.

Définition (*Chiffrement pleinement homomorphe*) Un $\mathcal{C}$-schéma est une procédure de chiffrement pleinement homomorphe s'il s'agit d'un chiffrement partiellement homomorphe qui vérifie également la propriété de compacité et où $\mathcal{C}$ représente l'ensemble de tout les circuits booléens [18].

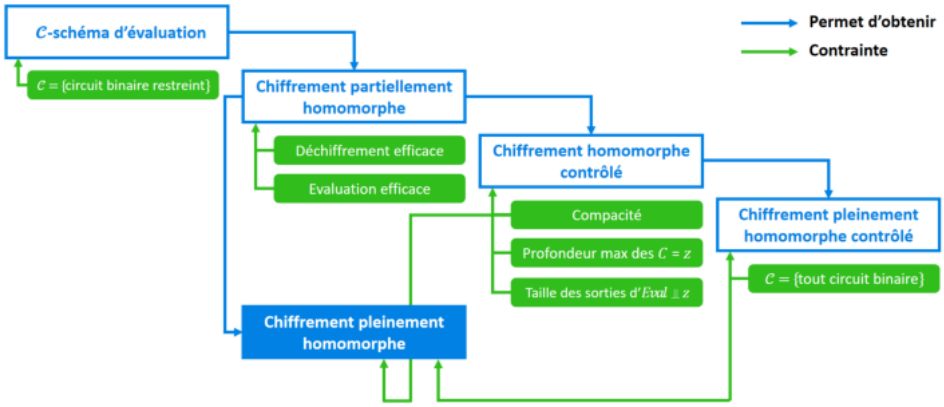


FIGURE 5 : Schéma récapitulatif des différents chiffrements homomorphes

Il existe de nombreuses méthodes de chiffrement partiellement homomorphes, l'annexe (A) en présente deux des plus connues, ainsi que quelques méthodes de chiffrement pleinement homomorphes comme le SEAL (développé par Microsoft). Néanmoins dans la suite de ce mémoire nous nous concentrerons sur une méthode de chiffrement pleinement homomorphe contrôlée (LFHE) définie pour la première fois par A. Yu et al. en 2015 (*Efficient Integer Vector Homomorphic Encryption*) que nous présenterons dans le prochaine partie et dont le principal avantage réside avant tout dans la simplicité de son schéma de cryptage, facilitant d'autant plus son implémentation.

17. Tout les circuits booléens faisant intervenir des additions (OU) et multiplications (ET) dont la profondeur est inférieur ou égale à z.
18. Tout les circuits booléens faisant intervenir des additions (OU) et multiplications (ET).

3 Calculs délégués & données pseudonymisées

Pour rappel, l'enjeu opérationnel de cette partie est de pouvoir fournir un cadre technique permettant à une entité tiers de réaliser certains calculs et notamment de mener des études statistiques sur des données confidentielles pseudonymisées fournies par son client, et ce en toute sécurité.
L'idée pour le client est donc de fournir un ensemble de données (potentiellement pré-calculées) cryptées à l'opérateur, sur lesquelles ce dernier pourra réaliser les calculs escomptés par le client (dans notre cas une régression linéaire simple) sans jamais avoir ni à les décrypter de son côté, ni à solliciter le client pour d'éventuelles étapes intermédiaires nécessitant l'utilisation ou la modification de la clef secrète. Voir figure (6) ci-dessous.

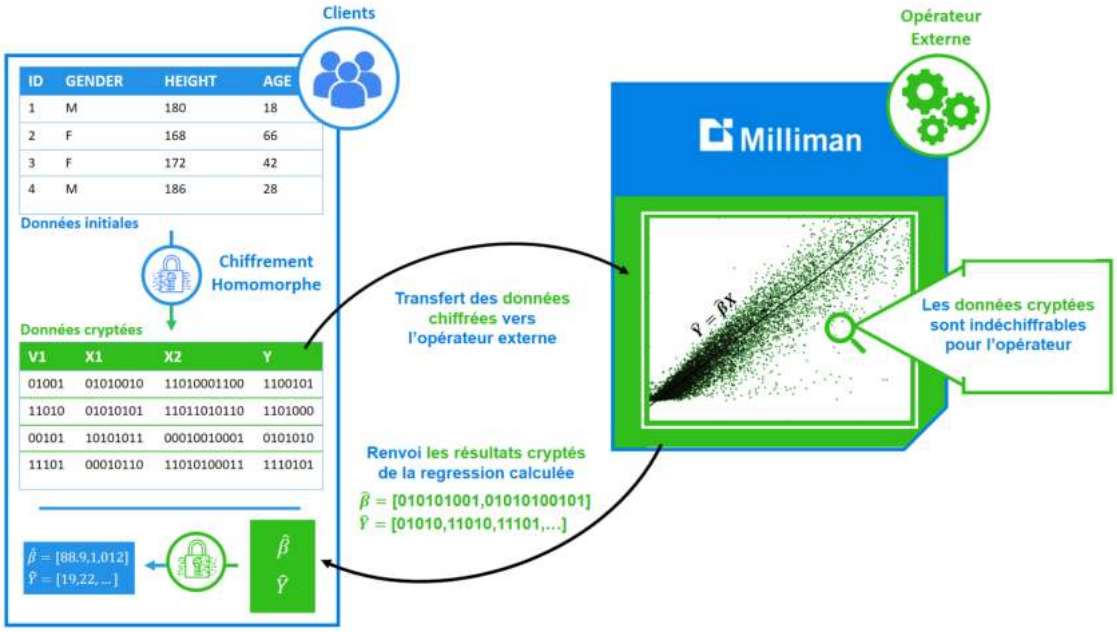


FIGURE 6 : Procédure du calcul délégué d'une régression linéaire pseudonymisée

C'est dans ce cadre que nous avons choisi de commencer par implémenter une procédure de chiffrement pleinement homomorphe (partiellement contrôlée), basée sur le schéma proposé par A. Yu, W. Lok Lai et J. Payor dans *Efficient Integer Vector Homomorphic Encryption*.
Si nous avons souhaité commencer par implémenter cette méthode de chiffrement homomorphe, c'est avant tout parce que la logique sous-jacente à ce schéma est assez simple à comprendre et ne nécessite pas de compétences ou de connaissances annexes particulières. On parvient alors à obtenir des résultats intéressants mais limités justement par la simplicité du schéma en question. C'est pourquoi nous présenterons dans un second temps, toujours dans le but de réaliser un régression linéaire sur des données pseudonymisées, le schéma de chiffrement homomorphe de *Van & Vercauteren*, plus complet et plus complexe qui nous permettra de nous défaire des contraintes rencontrées jusqu'alors.

3.1 Efficient Integer Vector Homomorphic Encryption

Dans cette sous-section nous nous pencherons sur les aspects techniques de ce schéma de chiffrement ainsi que sur ses différentes propriétés mathématiques. Nous présenterons par la suite les résultats obtenus dans la sous-partie suivante.

3.1.1 L'équation générale du chiffrement

Cette méthode de chiffrement s'effectue sur l'ensemble des entiers relatifs $\mathbb{Z}$, et plus particulièrement sur l'ensemble des vecteurs de taille m quelconque de $\mathbb{Z}$. On définit alors dans ce cadre notre vecteur d'entiers $x \in \mathbb{Z}^m$ dans l'espace non crypté $\mathcal{P} \subseteq \mathbb{Z}^m$ que l'on souhaite chiffrer et manipuler, $S \in \mathbb{Z}^{m \times n}$ une matrice d'entiers relatifs à m lignes et n colonnes symbolisant notre clef secrète de cryptage et enfin $c \in \mathbb{Z}^n$ le vecteur d'entier x dans l'espace crypté $\mathcal{X} \subseteq \mathbb{Z}^n$.
On peut ainsi définir le schéma de chiffrement, comme une transformation linéaire de notre vecteur initial x par l'équation générale suivante :

$$Sc = wx + e \tag{3}$$

où $w \in \mathbb{Z}$ un grand entier et $e \in \mathbb{Z}^m$ un vecteur de bruit dont chaque élément est strictement inférieur à $w/2$.
La taille de w est importante, en effet cette variable de pondération permet de retraiter la donnée x en "l'agrandissant" et ainsi de contrôler, au cours du chiffrement, le rapport entre l'information préservée et l'erreur introduite (en traitement du signal, à un ratio entre le signal et le bruit). Ainsi, si w n'est pas assez grand, l'information contenue dans c est beaucoup plus sensible au bruit et lors du déchiffrement nous risquons alors d'obtenir simplement un approximation de x. En revanche une valeur trop élevée de w risquerait simplement de corrompre l'intégralité des données. En d'autres termes le choix de w relève d'un compromis biais-variance.
Nous remarquons également qu'il faut que l'on ait $\max_{ij}\{|S_{ij}|\} \ll w$ afin de garder la quantité de bruit raisonnable lors des calculs dans l'espace crypté $\mathcal{X}$.

3.1.2 La procédure de décryptage

Finalement pour un utilisateur en possession de la clef secrète S, la procédure de décryptage est assez intuitive :

$$\mathrm{Dec}(S, c) = \mathcal{E}'(c) = \left\lceil w^{-1} Sc \right\rfloor = x$$

où $\lceil A \rfloor$ avec $A \in \mathbb{R}^p$ correspond à l'arrondi à l'entier le plus proche de chaque élément de A.

On remarque par ailleurs la disparation du terme d'erreur e, puisque si l'on a bien pour tout élément e_i de e, $0 \leq e_i < w/2$, alors :

$$\begin{aligned}
&\left\lceil w^{-1}(Sc - w/2)\right\rfloor < \mathrm{Dec}(S,c) \leq \left\lceil w^{-1}Sc\right\rfloor \\
\Leftrightarrow &\left\lceil w^{-1}Sc - 1/2\right\rfloor < \mathrm{Dec}(S,c) \leq \left\lceil w^{-1}Sc\right\rfloor \\
\Leftrightarrow &\, x - 1 < \mathrm{Dec}(S,c) \leq x \underset{x \in \mathbb{Z}}{\Longrightarrow} \mathrm{Dec}(S,c) = x \qquad \square
\end{aligned}$$

Cette opération simple pour décrypter les données chiffrées sera donc réalisée dans notre cadre uniquement par le client (seule entité en possession de S) afin de déchiffrer les résultats de la régression linéaire cryptée réalisée par l'opérateur externe.

3.1.3 La méthode de changement de clef

Cette méthode de chiffrement permet d'additionner deux vecteurs, de leur appliquer une transformation linéaire ainsi que de calculer un produit scalaire (pondéré) entre eux. Cependant pour simplifier la réalisation de ces calculs, ainsi que pour faciliter la génération de la clef secrète S, une méthode de changement de clefs a été implémentée.

L'objectif de cette procédure est de permettre au client de changer le couple clef secrète - donnée cryptée par un autre et à partir d'une nouvelle clef secrète définie. Le tout bien entendu en conservant la robustesse du chiffrement de la donnée cryptée initiale.

On définit alors la nouvelle donnée cryptée attendue $c' \in \mathbb{Z}^{n'}$ et la nouvelle clef secrète $S' \in \mathbb{Z}^{m \times n'}$ tel que :

$$S'c' = Sc$$

Le déroulement de cette procédure se décompose en deux étapes, tout d'abord convertir c et S en leurs représentations binaires, nommées respectivement c^* et S^*, puis dans un second temps utiliser ces valeurs afin de procéder au changement de clef.

Étape 1. Conversion en représentation binaire

On commence par définir l tel que $|c| < 2^l$ afin de convertir chaque élément c_i en leur représentation binaire (en base 2), afin de pouvoir écrire :

$$c^* = [b_1, b_2, \ldots, b_n]^T$$

où $b_i = [b_{i(l-1)}, \ldots, b_{i1}, b_{i0}]$ et $b_{ik} \in \{-1, 0, 1\}$.

Exemple : $c = [-1, 5]^T \xrightarrow{donne} c^* = [\underbrace{0, 0, -1}_{A}, \underbrace{1, 0, 1}_{B}]^T$ avec $l = 3$.

En remarquant que l'on a :

A. $-1 = 0 \times 2^2 + 0 \times 2^1 + (-1) \times 2^0$

B. $5 = 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$

Puis on calcule la matrice S^* en convertissant chaque élément de S_{ij} par :

$$B_{ij} = [2^{l-1}S_{ij}, \ldots, 2S_{ij}, S_{ij}]$$

Exemple : $S = \begin{bmatrix} 3 & 1 \\ 4 & 2 \end{bmatrix} \xrightarrow{donne} S^* = \begin{bmatrix} 12 & 6 & 3 & 4 & 2 & 1 \\ 16 & 8 & 4 & 8 & 4 & 2 \end{bmatrix}$ avec $l = 3$

On remarque par ailleurs que la représentation binaire permet d'avoir $\max_{ij}\{|c^*_{ij}|\} \leq 1$ ce qui évite au nouveau terme d'erreur e' de trop augmenter et ainsi garantir l'intégrité de l'encryptage tout en préservant l'égalité $S'c' = Sc$.

Étape 2. Construction de la matrice de passage M

Grâce à la représentation binaire de la clef secrète S^* nous pouvons désormais calculer la nouvelle paire clef secrète - donnée cryptée en utilisant la matrice de passage $M \in \mathbb{Z}^{n' \times nl}$ telle que :

$$S'M = S^* + E$$

où $E \in \mathbb{Z}^{n' \times nl}$ est une matrice de bruit avec $\max_{ij}\{|e_{ij}|\}$ est petit.

Par ailleurs on ne considère que les nouvelles clefs secrètes de la forme $S' = [I, T]$: une concaténation de la matrice identité et d'une matrice $T \in \mathbb{Z}^{m \times (n'-m)}$ afin de pouvoir écrire M tel que

$$M = \begin{bmatrix} S^* - TA + E \\ A \end{bmatrix}$$

où $A \in \mathbb{Z}^{(n'-m) \times nl}$ une matrice aléatoire.

En appelant $c' = Mc^*$, on peut donc écrire :

$$S'c' = S^*c^* + e'$$

où $e' = Ec^*$ le nouveau terme d'erreur tel que $\max_{ij}\{|e'_{ij}|\}$ petit puisque comme nous l'avons déjà indiqué précédemment $\max_{ij}\{|c^*_{ij}|\} \leq 1$.

3.1.4 La procédure d'encryptage et de génération de la clef privée

Il y a deux méthodes distinctes disponibles pour générer les données chiffrées, l'une basée sur la matrice aléatoire S à inverser et la seconde méthode implique le recours à la méthode de changement de clef.

3.1.4.1 Méthode de la clef secrète aléatoire à inverser

Au regard de l'équation générale (3) et en admettant que l'on choisisse de générer notre clef secrète S comme une matrice carrée aléatoire, nous pouvons facilement en déduire la formule d'encryptage suivante :

$$\mathrm{Enc}(S, x) = \mathcal{E}(x) = S^{-1}.(wx + e) = c$$

Cependant une question demeure, comment s'assurer que S, dès lors qu'il s'agit d'une matrice carrée aléatoire, est bien inversible ?

Pour répondre à cette question nous revenons à la définition qu'une matrice est inversible si et seulement si son déterminant est non nul. Puisque le déterminant d'une matrice peut être vu comme un simple polynôme sur les éléments de cette matrice, notre problème peut donc se traduire comme la détermination de la probabilité d'avoir une matrice S (que l'on considère comme un point de $\mathbb{R}^{n\times n}$) vivant dans l'espace des zéros, ou espace d'annulation, d'un polynôme.
Pour répondre à cette problématique nous énonçons le théorème suivant, issu de *The zero set of polynomial* par R. Caron & T. Traynor :

Théorème (*Espace d'annulation d'un polynôme*)
Une fonction polynomiale de $\mathbb{R}^n$ vers $\mathbb{R}$ est soit strictement nulle ou strictement non-nulle presque partout.

Une démonstration de ce théorème est disponible en annexe.

On peut donc en déduire que la mesure de Lebesgue de l'espace d'annulation d'un polynôme vaut zéro, puisque soit un polynôme est toujours nul pour tout élément de $\mathbb{R}^n$, soit il s'annule en un nombre fini de racines.
Alors que l'on pourrait être tenté de conclure que la probabilité définie précédemment est donc nulle, il convient de rappeler que la mesure de Lebesgue n'est pas une mesure de probabilité sur $\mathbb{R}^{n\times n}$ et qu'il faut donc préciser au préalable que les éléments de la matrice S sont distribués en pratique selon certaines lois de probabilités (distribution multivariée, ou sous *Python* grâce à la fonction `numpy.random.rand()` selon une distribution uniforme continue), impliquant l'utilisation d'une mesure de probabilité.
Comme cette mesure est absolument continue selon la mesure de Lebesgue, nous pouvons raisonnablement conclure que la probabilité d'avoir une matrice carrée aléatoire S non inversible vaut zéro.

Nous pouvons donc désormais ne plus nous soucier de l'inversabilité de S. Puisqu'il s'agit d'une matrice carré aléatoire, elle est assurée presque toujours. Cependant nous allons présenter une deuxième méthode de chiffrement plus souple mettant en oeuvre une clef publique ainsi que la procédure de changement de clef évoquée précédemment.

3.1.4.2 Méthode de la clef publique

L'utilisation de la méthode de changement de clef permet d'obtenir une procédure d'encryptage directe en prenant $S = wI$ et $c = x$ comme donnée crytée, on a alors :

$$Sc = wx + e$$
$$\Leftrightarrow (wI)x = wx$$

où e est un terme d'erreur nul.

Désormais nous avons simplement à changer ce couple clef secrète - donnée cryptée par un nouveau grâce à la procédure de changement de clefs. Par ailleurs cette méthode implique la création d'une matrice de passage M pour réaliser la conversion des données de l'espace non-crypté $\mathcal{P}$ (où $x = c$ et $S = wI$) vers l'espace crypté $\mathcal{X}$ avec une clef secrète S' plus complexe et un nouveau $c' \neq x$. Dès lors la matrice M peut être vue comme une clef publique puisqu'elle ne permet que d'encrypter les données (elle ne peut en aucun les déchiffrer) et ainsi être distribuée et partagée sans crainte.

3.1.5 Les opérations supportées sur les données cryptées

Cette procédure de chiffrement permet de réaliser trois types d'opérations sur les données cryptées : l'addition, la multiplication et un produit scalaire pondéré sur les vecteurs d'entiers considérés. Ces trois opérations (et en particulier l'addition et le produit scalaire) seront particulièrement utiles pour la réalisation de notre régression linéaire dans l'espace crypté, puisqu'elles seules permettent de formaliser un modèle linéaire.
Nous verrons par la suite les différentes applications concrètes que nous pourrons réaliser à partir de ces opérateurs.

3.1.5.1 L'addition

Si l'on souhaite obtenir dans l'espace non-crypté $\mathcal{P}$ l'élément $x = x_1 + x_2$, en ayant respectivement c_1 et c_2 comme équivalence dans l'espace crypté $\mathcal{X}$ et partageant la même clef secrète S, alors la procédure d'évaluation est simple et directe :

$$\text{Eval}_+(c_1, c_2) = c_1 + c_2 = c$$

En effet, en repartant de l'équation générale de la méthode de chiffrement (3) on peut écrire de manière triviale :

$$S(c_1 + c_2) = w(x_1 + x_2) + (e_1 + e_2)$$

Dans le cas où chaque x_i serait crypté avec une clef secrète différente, le client aurait simplement à utiliser la procédure de changement de clef précédemment présentée afin d'obtenir une clef unifiée unique pour toutes les données cryptées et ainsi avoir de nouveaux couples clef secrète - donnée cryptée (S',c'_i).

On remarque par ailleurs que les erreurs croissent proportionnellement aux nombres de termes impliqués dans l'addition alors que le sur-coût général de l'opération d'addition en terme de mémoire est naturellement proportionnel à la différence de taille entre les données cryptées et les données non-cryptées. Dans le cas où l'on

choisit une clef secrète S sous forme de concaténation d'une matrice T et de la matrice identité I, les données cryptées obtenues seraient plus longues de seulement 1 bit, ce qui s'avère être un sur-coût négligeable.

3.1.5.2 La transformation linéaire

Nous pouvons une nouvelle fois facilement constater que le calcul d'une transformation linéaire Gx à partir d'une matrice de poids non-cryptée $G \in \mathbb{Z}^{m' \times n}$ dans l'espace crypté $\mathcal{X}$ peut simplement s'écrire sous la forme :

$$\text{Eval}_{\times}(G, c_1) = (GS)c_1 = c$$

En effet, on obtient à partir de (3) :

$$(GS)c_1 = wGx_1 + Ge_1$$

Nous pouvons donc considérer c comme le chiffrement de Gx_1 grâce à la clef secrète GS. Le client doit donc simplement changer de clef en passant de GS à $S' \in \mathbb{Z}^{m' \times (m'+1)}$ en calculant la matrice de passage $M \in \mathbb{Z}^{(m'+1) \times (m'l)}$ et envoyer cette clef publique à l'opérateur de calcul afin que ce dernier puisse déterminer c' tel que :

$$c' = Mc$$

L'utilisation de la méthode de changement de clef permet de réduire la dimension des données cryptées à $m+1$, le sur-coût est également engendré par ce changement de clef, de l'ordre de l fois le coût originel de G.

3.1.5.3 Produit scalaire pondéré

Afin de pouvoir définir l'équivalent du produit scalaire pondéré de $\mathcal{P}$ dans $\mathcal{X}$, il est nécessaire d'énoncer le Lemme suivant :

Lemme Pour tout vecteurs x et y, ainsi qu'une matrice M de dimension compatible on a :

$$x^T My = \text{vec}(M)^T \text{vec}(xy^T)$$

Avec pour tout $A \in \mathbb{R}^{n \times m}$, $\text{vec}(A) := [a_1^T, a_2^T, \ldots, a_n^T]^T$, où a_i est la i^{me} colonne de A ($\sim$ "vecteurisation").

La preuve de ce Lemme réside uniquement dans le développement du terme de droite :

$$\begin{aligned}\mathrm{vec}(M)^T\mathrm{vec}(xy^T) = & \sum_i \sum_j M_{ij}x_i y_j \\ = & \sum_j \left(\sum_i x_i M_i\right)_j .y_j \\ = & \sum_j (x^T M)_j y_j \\ = & x^T M y \end{aligned}$$

□

En prenant x_1 et x_2 deux vecteurs d'entiers de $\mathcal{P}$ chiffrés respectivement en c_1 et c_2 grâce à leurs clefs secrètes respectives S_1 et S_2, on peut calculer le produit scalaire pondéré de ces deux vecteurs, soit $x_1^T H x_2$, où H est une matrice de pondération, tel que :

$$\mathrm{Eval}_{<.>}(w, c_1, c_2) = \left\lceil \mathrm{vec}(c_1 c_2^T).w^{-1} \right\rfloor = c$$

où la clef secrète S de c vaut alors : $\mathrm{vec}(S_1^T H S_2)^T$

On peut ainsi écrire à partir de l'équation générale de la méthode de chiffrement (3) que l'on a :

$$\mathrm{vec}(S_1^T H S_2)^T \left\lceil \mathrm{vec}(c_1 c_2^T).w^{-1} \right\rfloor = w x_1 H x_2 + e$$

où e est un nouveau terme d'erreur.

Le fait de pouvoir réaliser l'équivalent d'un produit scalaire dans l'espace crypté $\mathcal{X}$ nous permet également de calculer des polynômes dans ce même espace en considérant par exemple $x = [x_1, x_2, \ldots, x_n]^T$ et sa valeur chiffrée c par la clef secrète S, que l'on peut étendre tel que $x' = [1, x_1, x_2, \ldots, x_n]^T$ dont la représentation dans $\mathcal{X}$ peut désormais s'écrire $c' = [w, c^T]^T$ où :

$$S' = \begin{bmatrix} 1 & 0 \\ 0 & S \end{bmatrix}$$

Ainsi pour calculer un polynôme de degré deux, il nous suffit de définir une matrice triangulaire $H \in \mathbb{R}^{(n+1)\times(n+1)}$ de coefficients polynomiaux, tel que :

$$
\begin{aligned}
(x')^T H x' = [1, x_1, \ldots, x_n] & \begin{bmatrix} h_{00} & h_{01} & \ldots & h_{0n} \\ 0 & h_{11} & \ldots & h_{1n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & h_{nn} \end{bmatrix} \begin{bmatrix} 1 \\ x_1 \\ \vdots \\ x_n \end{bmatrix} \\
= & h_{00} + x_1 h_{01} + \cdots + x_n h_{0n} \\
& + x_1^2 h_{11} + \cdots + x_n^2 h_{nn} \\
& + x_1 x_2 h_{12} + \cdots + x_1 x_n h_{1n} \\
& + x_2 x_3 h_{23} + \cdots + x_2 x_n h_{2n} \\
& + \ldots \\
& + x_{n-1} x_n h_{n-1n} \\
= & \sum_{j=0}^{n} h_{0j} x_j + \sum_{i,j=1}^{n} h_{ij} x_j^2 + \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} h_{ij} x_i x_j
\end{aligned}
$$

3.1.6 Avantages et contreparties

Le principal avantage que constitue l'utilisation de cette méthode de chiffrement réside avant tout dans la relative simplicité de son schéma. Tant du point de vue de l'écriture de son équation générale que de son implémentation informatique, les notions utilisées et introduites ici sont relativement simple d'accès et ne requiert pas autant de temps que d'autres méthodes de chiffrement plus évoluées (par exemple le chiffrement SEAL de Microsoft [19]). Par ailleurs la plupart des opérations réalisables dans l'espace crypté $\mathcal{X}$ sont relativement triviales (mis à part le produit scalaire pondéré).
En sus, cette méthode de chiffrement est basée sur des vecteurs d'entiers ce qui la rend de suite plus pratique à utiliser dans un contexte réel. En effet les données quantitatives continues auront simplement à être discrétisées au préalable tandis que les données qualitatives seront binarisées.
Enfin le sur-coût engendré par la plupart des opérations peut être contrôlé (pour l'addition et la multiplication en utilisant la méthode de changement de clef) ou même être négligeable dans certains cas (par exemple lors d'additions et de multiplications de petits entiers).

A contrario la complexité du calcul du produit scalaire pondéré dans l'espace crypté ainsi que son sur-coût non négligeable et l'accroissement du bruit difficilement identifiable rend compliqué en pratique une longue succession de ce type d'opérations.
Par ailleurs le recours régulier aux clients et à la méthode de changement de clef au cours des différentes opérations réalisées dans l'espace crypté conduit, au premier abord, à s'éloigner de notre volonté de pouvoir mener les opérations de l'espace crypté sans interventions du client. Néanmoins nous présenterons par la suite grâce à notre implémentation une façon de contourner cette procédure de changement de clef afin de pouvoir tout de même mener efficacement ces calculs dans $\mathcal{X}$.

19. Simple Encrypted Arithmetic Library (SEAL)
(microsoft.com/en-us/research/project/simple-encrypted-arithmetic-library/)

3.2 L'implémentation du schéma et les résultats obtenus

Nous avons implémenté le schéma précédent en *Python 2.7*, et ne requiert que le package `numpy` comme dépendance. Par ailleurs l'objectif de cette sous-partie est avant tout de présenter les résultats obtenus lors de la mise en pratique de la procédure de chiffrement précédemment présentée ainsi que les contraintes sous-jacentes introduites au cours de sa réalisation.
Il n'est pas question ici d'évoquer les détails de son implémentation, qui s'avère être une simple transcription du modèle mathématique en code informatique, mais tout au plus de mentionner les quelques modifications ayant été apportées afin de le rendre conforme à nos besoins.

3.2.1 Chiffrement d'un entier

Pour commencer nous avons naturellement testé notre implémentation avec le chiffrement d'un unique entier naturel, encapsulé dans un tableau de longueur 1 (`np.array`) afin de constater l'efficacité de notre méthode d'encryptage ainsi que le surcout engendré par cette dernière.
Dans ce cas résumé par le tableau suivant, nous avons choisi, à titre indicatif, d'avoir recours à la méthode de chiffrement basée sur l'inversion d'une matrice carrée aléatoire.

TABLE 3 : Synthèse du chiffrement d'un entier x

x	w	n	S	$c := \mathcal{E}(x)$	$\mathcal{E}'(\mathcal{E}(x))$	Mémoire x	Mémoire c	Mémoire S
5	9.10^3	1	$5{,}2.10^7$	$9{,}7.10^5$	5	104 octets	104 octets	120 octets

Outre le fait que nous constatons que notre procédure de chiffrement est capable de déchiffrer x correctement, nous pouvons également voir que globalement le surcoût engendré par cette méthode est nul sous Python [20] puisque l'espace mémoire occupé par l'élément x, donnée non-chiffrée et l'espace occupé par c, la donnée cryptée, sont égaux (104 octets). Le seul surcoût concerne donc la détention de la clef secrète générée à l'issue du chiffrement S, occupant 120 octets supplémentaires en mémoire. Le temps de calcul du chiffrement, compte tenu de l'exemple choisi ici, s'avère absolument négligeable.

3.2.2 Calcul d'un produit vectoriel avec deux vecteurs d'entiers

On définit ici nos deux vecteurs d'entiers à chiffrer $x = [1, 2, 3, 4, 5, 6]$ et $y = [4, 5, 6, 7, 8, 9]$. Nous souhaitons donc calculer, dans l'espace crypté $\mathcal{X}$, le produit yx^t.
On choisit ici d'avoir recours à la procédure d'encryptage basée sur une clef publique. Pour rappel, on obtiendra donc une clef secrète S, matrice concaténée de l'identité

20. Il est important de préciser que cette conclusion valable sous Python peut s'avérer erronée avec d'autres langages ayant une approche plus rigoureuse et moins flexible de la gestion de la mémoire.

I et d'un vecteur colonne T, tel que $S = [I, T] \in \mathbb{Z}^{(n+1)\times n}$, où n représente toujours la longueur des vecteurs x et y.

En prenant $w = 137993$, on obtient comme clef secrète pour x et y :

$$S = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 6 \\ 0 & 1 & 0 & 0 & 0 & 0 & 4 \\ 0 & 0 & 1 & 0 & 0 & 0 & 3 \\ 0 & 0 & 0 & 1 & 0 & 0 & 5 \\ 0 & 0 & 0 & 0 & 1 & 0 & 5 \\ 0 & 0 & 0 & 0 & 0 & 1 & 2 \end{bmatrix}$$

Et comme donnée cryptée c_x pour x (et resp. c_y pour y), le vecteur suivant :

$$c_x^T = [1.36805e^{+05}, 2.75194e^{+05}, 4.13385e^{+05}, 5.50982e^{+05}, 6,88975e^{+05}, 8,27562e^{+05}, 1,98e^{+02}]$$

La structure de données du schéma de chiffrement nous permet alors de retrouver le i^{me} élément d'un de ses vecteurs simplement en utilisant la i^{me} ligne de S comme clef secrète pour la procédure de déchiffrement, i.e pour le quatrième élément de x :

$$\mathrm{Dec}(S[4], c_x) = w^{-1}.[0,0,0,1,0,0,5].c_x = \frac{5.50982e^{+05} + 5.\ 1,98e^{+02}}{137993} = 4$$

Pour réaliser la multiplication vectorielle nous avons simplement à suivre la procédure explicitée précédemment tel que :

$$S' = \mathrm{vec}(S^T S)^T, \text{et } \ c' = \left\lceil \mathrm{vec}(c_x c_y^T).w^{-1} \right\rfloor$$

On utilise alors la méthode *ravel('F')* sur nos tableaux afin de les linéariser.
On remarque par ailleurs que S' dépend de la clef secrète mais pas des données cryptées, et respectivement c' dépend des données cryptées initiales mais pas de leurs clefs secrètes.
On obtient bien $\mathrm{Dec}(S', c') = 154$ où :

TABLE 4 : Surcoût lié au produit matriciel dans $\mathcal{X}$

Mémoire $z = 154$	Mémoire c'	Mémoire S'
104 octets	488 octets	96 octets

On constate ici que le surcoût en mémoire engendré par le calcul du produit vectoriel n'est pas négligeable (une augmentation de près de 370%), même si compte tenu de l'exemple choisi ici l'occupation en mémoire des éléments cryptés peut paraître faible.
Le temps de calcul du chiffrement et de l'opération dans l'espace crypté, compte tenu de l'exemple choisi ici, s'avère absolument négligeable.

3.2.3 Chiffrement d'une matrice d'entiers et produits matriciels

Puisque notre objectif est de pouvoir réaliser une régression linéaire dans l'espace crypté $\mathcal{X}$, nous allons naturellement devoir manipuler des matrices dans cet espace afin par exemple de pouvoir calculer une estimation du coefficient de régression $\hat{\beta}$ grâce à la formule fermée des moindres carrés ordinaires rappelée dans la partie (3.2.4.1).
Or le schéma ne propose pas initialement de solutions directes afin de chiffrer des matrices d'entiers. C'est pourquoi il nous a fallu réfléchir à une procédure spécifique pour les crypter.
Notre idée est simple : puisque nous sommes facilement capable de chiffrer des vecteurs d'entiers, il nous suffit de considérer chaque matrice d'entiers comme une liste (ligne par ligne) de vecteurs d'entiers.
Au final on obtient alors comme structure de données pour les matrices chiffrées des listes de vecteurs c_i et respectivement pour la clef secrète des listes de matrices S_i correspondant au couple clef-vecteur de chaque ligne de la matrice initiale.

Une fois le chiffrement des matrices défini et réalisé il est désormais possible de s'intéresser à l'implémentation du produit matriciel. Pour ce faire nous nous basons simplement sur un produit ligne à ligne des matrices grâce à la procédure de produit vectoriel défini au sein du schéma.
La seule contrainte résultant de cette méthode s'avère être le formatage des matrices. En admettant des dimensions compatibles, afin de pouvoir calculer le produit matriciel usuel XY^T dans l'espace crypté $\mathcal{X}$ il faut en réalité fournir à la fonction du produit matriciel les paramètres X et Y. En effet puisque nous ne pouvons que réaliser des produits vectoriels ligne à ligne il est nécessaire que l'une des deux matrices (ici Y) ait été transposée préalablement à son chiffrement afin que les vecteurs (et donc les lignes de la matrice ainsi obtenue) qui composent c_Y correspondent aux colonnes de la matrice initiale Y^T. Le schéma (7) ci-dessous précise le fonctionnement du produit matriciel dans l'espace crypté.

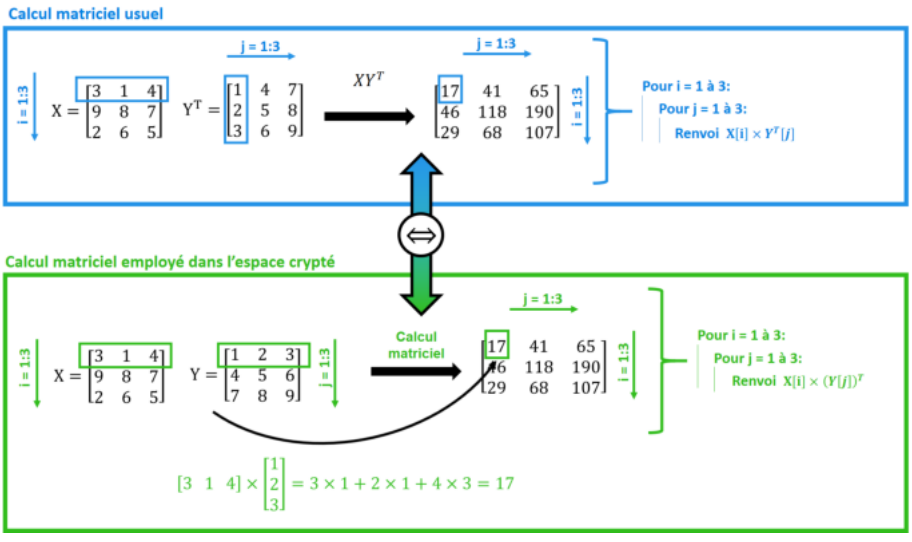


FIGURE 7 : Expression du fonctionnement du produit matriciel dans l'espace crypté

On constate donc qu'avec ce choix de structure pour les matrices d'entiers, une fois cryptées, elles ne peuvent être ni transposées ni inversées dans l'espace crypté. En effet, la structure d'une matrice dans l'espace crypté ne permet que de manipuler les vecteurs-lignes qui la composent et accéder à ses colonnes directement ou indirectement (on ne peut pas non plus isoler la j^{me} composante d'un de ses vecteurs-lignes cryptés) est donc impossible.

3.2.4 Effectuer une régression linéaire sur les données chiffrées

Notre idée initiale est de permettre à un assureur de faire réaliser ses calculs statistiques pour la réalisation d'un modèle de tarification non-vie (ici nous considérons volontairement le cas simple de la régression linéaire) à un prestataire extérieur afin de pouvoir profiter de sa puissance de calcul en garantissant la sécurité et la confidentialité des données du client grâce au maintien du caractère pseudonymisé de celles-ci tout au long du processus. Nous pouvons donc désormais mettre en place cette idée grâce à la structure et à la procédure de calcul matriciel définies précédemment afin de calculer une estimation du coefficient de régression linéaire $\hat{\beta}$ par la méthode des moindres carrés directement sur les données chiffrées.
Cependant, avant de présenter l'implémentation de notre méthode et les résultats que nous avons obtenus, nous commencerons par rappeler les fondements du modèle de régression linéaire classique.

3.2.4.1 Rappel sur le modèle de régression linéaire

Le modèle de régression linéaire standard est défini tel que :

$$Y = X\beta + \epsilon, \;\; \epsilon \sim \mathcal{N}(0_N, \sigma^2 I_N)$$

où Y est le vecteur réponse de taille N, X la matrice des données pour N observations et P covariables de dimension $N \times P$ (ou éventuellement $N \times (P+1)$ si on ajoute une constante au modèle) et ϵ le résidu, vecteur gaussien de taille N.

Sachant que $X^T X$ est inversible sous l'hypothèse de non co-linéarité des colonnes de X, la solution pour déterminer un estimateur de β par la méthode des moindres carrés ordinaires qui soit solution de :

$$\min_{\beta} \|Y - X\beta\|$$

$$\text{est :} \quad \hat{\beta} = (X^T X)^{-1} X^T Y$$

L'idée est donc ici de calculer $\hat{\beta}$ grâce à l'équation précédente en partant directement de données X et Y chiffrées et d'être capable de fournir des estimations Y^* issues du modèle à partir de nouvelles observations X^*, tel que :

$$Y^* = \hat{\beta} X^*$$

3.2.4.2 Implémentation et résultats obtenus

Afin de bien comprendre le fonctionnement de notre méthode pour déterminer $\hat{\beta}$ directement dans l'espace crypté $\mathcal{X}$, il convient de préciser explicitement la logique inhérente à notre procédure. Notre idée consiste à calculer l'estimation du coefficient de régression à partir de la formule fermée mentionnée à la partie précédente (3.2.4.1) dans l'espace crypté grâce aux données X et Y chiffrées fournies par le client. Les calculs matriciels (additions et multiplications) engendrés sont alors naturellement réalisés grâce aux définitions de ces opérations tel que présentées en (3.1.5). Il est fondamental de bien comprendre que nous ne calculons donc pas ici $\hat{\beta}$ en minimisant la fonction de coût quadratique itérativement, ce qui ne serait pas pertinent comme nous le verrons par la suite dans la partie (3.2.5).
Les données employées afin de réaliser notre régression linéaire sur nos données cryptées sont issues d'un contexte académique. Elles ont été choisies ici, bien qu'elles ne soient pas directement reliées au milieu actuariel, car elles présentent tout de même des caractéristiques réalistes en conservant une taille [21] qui permet de facilement - et le plus rapidement possible - tester nos modèles et obtenir des résultats relativement convaincants.
Notre jeu de données provient donc des travaux de D.G. Kleinbaum et L.L Kupper *Applied Regression Analysis and Other Multivariate Methods* (1978), dont l'objectif est de pouvoir modéliser grâce à une régression linéaire le taux de graisse dans le sang (Y) en fonction du poids (en kg) et de l'âge des 25 patients (X).

Pour pouvoir vérifier l'exactitude de la régression linéaire réalisée sur nos données cryptées, nous avons d'abord effectué une régression linéaire sur ces mêmes données mais dans l'espace non crypté afin de connaître les valeurs des composantes de $\hat{\beta}$ ainsi que celles des prédictions X^* :

TABLE 5 : Estimateur de β dans l'espace non crypté

$\hat{\beta}_0$	$\hat{\beta}_1$	$\hat{\beta}_2$
89,27	0,72	4,88

L'enjeu ici, n'est pas de comparer les résultats de la régression linéaire avec les Y observés mais simplement de s'assurer que les résultats obtenus lors de la régression sur nos mêmes données une fois chiffrées soient similaires.

Avant tout nous devons impérativement pré-calculer certains éléments en amont de la procédure de chiffrement. En d'autre termes, le client ne doit pas seulement fournir au prestataire extérieur les données X et Y cryptées, mais également y avoir appliqué certaines transformations.

21. seulement deux covariables en plus d'une éventuelle constante dans le modèle pour 25 observations

TABLE 6 : Prédiction de Y^* dans l'espace non crypté

Y_{rel}	Y^*
395	424,55
434	373,1
220	298,3
...	...
303	324
244	280,9

Comme nous l'avons précédemment évoqué nous ne pouvons ni transposer ni inverser des matrices dans l'espace crypté c'est pourquoi toute ses opérations doivent être réalisées avant le chiffrement, ainsi et conformément également aux exigences de notre procédure de produit matriciel, le client est invité à chiffrer et fournir les deux paramètres suivants [22] $B = (X^T Y)^T$, soit $B = Y^T X$, et $A = (X^T X)^{-1}$ encryptés.
Puisque l'idée est de calculer dans l'espace crypté $\mathcal{X}$ le résultat de l'équation des moindres carrés ordinaires (cf. 3.2.4.1), et compte tenu des contraintes rappelées précédemment on constatera qu'on a bien dans l'espace crypté :

$$\hat{\beta}^T = \text{Dec}[S', \text{Enc}(S, A) *_\times \text{Enc}(S, B)] = \mathcal{E}'[\mathcal{E}(A) *_\times \mathcal{E}(B)]$$

où S' correspond à la transformation de la clef secrète évoquée lors du développement du produit vectoriel dans l'espace crypté et rappelé dans la partie 3.2.2, et $*_\times$ l'opérateur équivalent au produit vectoriel dans $\mathcal{X}$, tel que :

$$\text{Dec}[S', \text{Enc}(S, X) *_\times \text{Enc}(S, Y)] = \mathcal{E}'[\mathcal{E}(X) *_\times \mathcal{E}(Y)] = XY^T$$

Par ailleurs, les transformations successives à appliquer à S pour effectuer les différents produits au cours du processus (une fois pour déterminer $\hat{\beta}$ et une seconde fois pour obtenir les estimations Y^*) ne peuvent naturellement pas être réalisées par l'opérateur de calcul qui n'a jamais accès à la clef S. Seul le client est donc en mesure de procéder à ses transformations sur la clef secrète.
Si l'on peut penser à première vue que cette contrainte contredit certains des principes que nous évoquions lors de la présentation du cadre et des objectifs de l'étude [23] (2.1.5), en réalité ses transformations ne constituent pas des points bloquants pour l'opérateur. En d'autres termes, les calculs réalisés dans l'espace crypté sont totalement indépendants de ces transformations et il n'est donc pas nécessaire que ces changements sur les clefs soient réalisés conjointement avec les calculs de l'opérateur [24] : sachant que le client sait quels calculs ont été effectués sur ses données, ce dernier a

22. En plus de l'équivalent chiffré de X^*s'il souhaite obtenir des prédictions et non seulement les estimations des coefficients de la régression

23. En particulier le fait, pour l'opérateur, de ne pas avoir à nécessiter la moindre intervention du client au cours des calculs.

24. Au regard des formulations de S' et de c' rappelées en (3.2.2) on constate que S' dépend des deux clefs secrètes, mais pas des données cryptées, et respectivement c' dépend des données cryptées initiales mais pas de leurs clefs secrètes.

simplement à réaliser toutes les transformations nécessaires sur les clefs au moment du déchiffrement.

En réalisant les calculs précédemment évoqués dans l'espace crypté et en décryptant les données après avoir effectué les transformations nécessaires sur S, on obtient bien un $\hat{\beta}^T$ égal (à 10^{-5} près) à son calcul dans l'espace non crypté.
Par ailleurs, en conservant la valeur chiffrée de $\hat{\beta}^T$ obtenue, si le client fournie la valeur encryptée de X^*, alors on parvient naturellement à obtenir un Y^* strictement similaire à celui calculé précédemment dans l'espace non crypté, tel que :

$$Y^{*T} = \text{Dec}[S'', \hat{\beta}^T *_\times X^*]$$

où S'' correspond à la transformation des clefs secrètes S et S' conformément à la formulation rappellée en (3.2.2), soit :

$$S'' = \text{vec}(S'^T S)^T$$

Nous présentons dans les tableaux suivants les consommations mémoires ainsi que les durées d'exécution (temps machine exclusivement) des régressions dans $\mathcal{X}$ et dans l'espace non-crypté :

TABLE 7 : Ressources mémoire employées

Éléments	Espace crypté		Espace non-crypté
	c	S	
A	232 octets	216 octets	144 octets
B	488 octets	440 octets	168 octets
$\hat{\beta}$	776 octets	776 octets	168 octets
X^*	7,8 ko	7,8 ko	408 octets
Y^*	213,2 ko	213,2 ko	232 octets

TABLE 8 : Temps de calcul machine nécessaire

Éléments	Chiffrement	Déchiffrement	Opérations
A & B	8ms		
X^*	4ms		
$\hat{\beta}$		$\varnothing$	$\varnothing$
Y^*		4ms	8ms
Client	12ms	4ms	
Opérateur			8ms

On remarque grâce au tableau (7) que, même si la taille des éléments en mémoire dans notre exemple demeure très raisonnable, le passage par le chiffrement induit toutefois une augmentation importante de l'occupation en mémoire des variables, comparativement au calcul réalisé dans l'espace non crypté.
En particulier, on constate que la place occupée en mémoire par la variable réponse estimée Y^* augmente drastiquement lors du calcul dans $\mathcal{X}$ conformément à ce que nous avions évoqués précédemment (3.1.6) concernant l'important sur-coût engendré par les produits vectoriels dans l'espace crypté (et donc plus encore dans un produit matriciel composé d'une succession de produits vectoriels).
Le tableau (8) [25] récapitule les durées nécessaires pour réaliser les chiffrements, déchiffrements et opérations permettant d'obtenir les différents éléments pour réaliser la régression. Ces mesures ont été réalisées sur une machine équipée d'un i7-5600 @2.6Ghz (2 coeurs physiques, 4 coeurs logiques) et 8Go de RAM (+ 8Go de swap).
Une case grise signifie que la méthode n'a pas été appliquée à cet élément (i.e $\hat{\beta}$ est issu d'une opération dans $\mathcal{X}$ et peut être déchiffré par le client - X^* est simplement chiffré en amont de la procédure mais ne résulte pas d'une opération dans $\mathcal{X}$ et son déchiffrement est superflu).
Les durées obtenues peuvent paraître négligeables mais il convient également de les mettre en perspective avec la très petite taille de notre jeu de données. [26]

3.2.5 Contraintes inhérentes au modèle et solutions envisagées

Comme nous avons pu le voir cette méthode présente quelques avantages, notamment au travers de sa simplicité mais également des temps de calcul et d'une consommation mémoire relativement modérée.
Cependant une contrainte majeure demeure : pour que l'opérateur extérieur puisse réaliser la régression linéaire sur les données chiffrées avec cette méthode, il est nécessaire que le client encrypte les données sous une forme pré-calculée avant de les lui transmettre, puisque nous ne pouvons ni inverser ni transposer de matrices dans $\mathcal{X}$. En outre, l'assureur devrait préalablement traiter les données sur ses assurés avant de la transmettre à un service tiers pour le calcul.
Cette contrainte peut donc s'avérer particulièrement limitative pour des bases de données importantes, dans lesquelles finalement le client aurait à prendre à sa charge une partie non négligeable des calculs (et notamment des calculs assez lourds) ce qui annihilerait finalement complètement l'intérêt de vouloir recourir à la puissance de calcul d'un prestataire extérieur.
Afin de contourner cette limitation nous avons donc essayé plusieurs autres alternatives pour déléguer le calcul et la prédiction issue d'une régression linéaire, parmi lesquels :

- *Utiliser une autre méthode de chiffrement*, notamment le chiffrement Paillier que l'on a vu précédemment (2.2.2.5), mais s'agissant d'une procédure de

25. Le signe $\varnothing$ dans le tableau signifie que le temps de calcul est insignifiant ($\sim \mu s$). Par ailleurs les temps de calculs pour réaliser la régression dans l'espace non-crypté sont également insignifiants.

26. Ce choix d'un jeu de données de taille réduite est issu de la comparaison de différentes procédures de régressions cryptées réalisées par la suite et dont certaines s'avèreront bien plus gourmandes.

cryptage partiellement homomorphe (seulement additif entre deux données cryptées) il est impossible de réaliser les opérations nécessaires à une régression linéaire dans $\mathcal{X}$.

- *Réaliser directement la régression linéaire dans $\mathcal{X}$*, c'est à dire minimiser l'équation des moindres carrés ordinaires $\sum_{i=1}^{n} \|Y_i - \beta X_i\|$ dans l'espace crypté afin de déterminer $\hat{\beta}$ sans recourir à la formule fermée précédente. Malheureusement le schéma de chiffrement employé ici [27], comme toutes les autres méthodes de chiffrement homomorphes connues à ce jour, n'autorise pas le recours aux opérateurs de comparaison entre les données cryptées car $\mathcal{X}$ n'est pas un ensemble ordonné. Dès lors la valeur de $\hat{\beta}$ obtenue par cette méthode dans l'espace crypté - garantissant donc la minimisation de notre fonction de coût dans $\mathcal{X}$ - ne correspond presque sûrement pas, une fois décrypté, à la valeur de $\hat{\beta}$ dans l'espace non-crypté minimisant cette même fonction de coût. En d'autre termes le minimum local d'une fonction convexe de l'espace crypté n'est pas nécessairement équivalent à un minimum local dans l'espace non-crypté. Et inversement.

Étant donné qu'aucune de ces tentatives ne s'est avérée fructueuse, nous avons cherché une méthode afin de contourner notre problème initial, à savoir éviter de devoir calculer des inversions et des transposés de matrices - que ce soit par le client ou par l'opérateur externe. Il nous a donc fallu considérer une méthode alternative à la formule fermée employée jusqu'ici. Et finalement nous sommes parvenus grâce aux travaux de P.M Esperança, L.J.M Aslett et C.C Holmes *Encrypted accelerated least squares regression* (2017) à utiliser une descente de gradient afin d'obtenir une estimation de l'estimateur du paramètre de régression $\hat{\beta}$ dans l'espace crypté $\mathcal{X}$ au travers d'un schéma de chiffrement plus sophistiqué. La méthode de chiffrement en question est un cryptage pleinement homomorphe contrôlé basé sur le schéma de *Fan & Vercauteren*.
Dans la prochaine partie nous présenterons brièvement ce nouveau schéma puis les résultats obtenus de notre descente de gradient sur notre jeu de données [28] cryptées.

27. Efficient Integer Vector Homomorphic Encryption scheme
28. Notre jeu de 25 observations et deux variables pour prédire le taux de graisse dans le sang

3.3 Introduction au schéma de Fan & Vercauteren

Afin de pouvoir réaliser une descente de gradient dans l'espace crypté $\mathcal{X}$ pour déterminer une approximation de $\hat{\beta}$, il nous faut pouvoir utiliser une méthode de chiffrement robuste nous permettant notamment de contrôler efficacement l'évolution du bruit au cours des différentes itérations de l'algorithme. C'est pourquoi nous avons décidé d'abandonner dans cette partie le schéma précédent simplement basé sur une sorte de transformation linéaire, en effet étant données les contraintes du chiffrement précédent nous nous sommes demandés s'il n'existait pas une autre méthode permettant de réaliser une régression. C'est ainsi que nous avons fait le choix d'un schéma pleinement homomorphe contrôlé plus complexe.
L'idée de cette partie est de pouvoir rapidement introduire ce nouveau schéma. L'enjeu n'est donc pas d'expliciter entièrement cette procédure comme cela a été fait pour la méthode précédente, cette dernière s'avérant bien plus simple que celle que nous allons présenter maintenant.
A ce titre, plus de précisions et d'informations peuvent être obtenues sur le schéma FV (i.e *Fan & Vercauteren*) au travers de l'article de ses auteurs J. Fan et F. Vercauteren *Somewhat Practical Homomorphic Encryption* (2012).

3.3.1 Idées directrices

Cette méthode de chiffrement est basée sur le principe *d'apprentissage avec erreurs sur des anneaux* (i.e RLWE), qui s'avère être une transposition du cadre classique *d'apprentissage avec erreurs* (i.e LWE) sur lequel repose aujourd'hui la plupart des chiffrements homomorphes dont celui précédemment présenté, restreint ici au cas particulier d'anneaux polynomiaux finis.

3.3.1.1 Apprentissage avec erreurs

Il s'agit d'un problème calculatoire à la base de nombreux cryptosystèmes actuels qui constitue actuellement l'une des principales pistes de recherches dans le domaine de la cryptographie. Il a notamment valu à O. Regev le prix Gödel en 2018 pour l'introduction de cette notion et ses travaux dans ce domaine.
Considérons un vecteur d'entiers s supposé secret. Il est facile de retrouver s grâce à des produits scalaires $\langle a, s\rangle$ et en ayant accès à suffisamment de vecteurs a différents afin de passer d'un système sous-déterminé à un système parfaitement déterminé offrant une unique solution s.
Cependant il n'existe pas à l'heure actuelle d'algorithmes efficaces afin de résoudre ce même problème en ayant cette fois accès uniquement à différentes valeurs de l'ensemble $\{(a, \langle a, s\rangle \ + e)\}$ où e est un terme de bruit aléatoire et distribué selon une distribution spécifique (le plus souvent une distribution gaussienne discrète).
Ainsi une première "définition" du problème d'apprentissage avec erreurs nécessite de *chercher* s. Une autre approche consiste également à essayer de distinguer les éléments de l'ensemble $\{(a, \langle a, s\rangle \ + e)\}$ d'autres échantillons tirés uniformément et aléatoirement de ce même espace [29]. Cette deuxième vision correspond à la forme de

29. i.e en notant $a \in \mathbb{A}$ et $(\langle a, s\rangle \ + e) \in \mathbb{B}$ - l'espace $\mathbb{A} \times \mathbb{B}$

décision, Regev a notamment prouvé que ces deux problèmes étaient bien équivalents sous certaines conditions.

3.3.1.2 Apprentissage avec erreurs sur des anneaux

Similairement au problème précédent, nous pouvons ici distinguer les deux formulations du problème, en commençant par la vision *recherche* :
On définit $a_i(x)$ un ensemble de polynômes aléatoires mais connus de $\mathbb{K}[x]$ dont les coefficients sont issus de l'anneau d'entiers fini $\mathcal{F}$, $e_i(x)$ un ensemble de *petits*[30] polynômes aléatoires et inconnus de $\mathbb{K}[x]$ et $s(x)$ un unique *petit*[23] polynôme inconnu de $\mathbb{K}[x]$. En posant $b_i(x) = (a_i(x) \cdot s(x)) + e_i(x)$, la vision *recherche* propose de tenter de déterminer le polynôme inconnu $s(x)$ à partir de l'ensemble des couples $(a_i(x), b_i(x))$.
La forme *décisionnaire* tente, à partir des paires $(a_i(x), b_i(x))$ de déterminer si $b_i(x)$ est construit selon la forme précédemment définie ou s'il s'agit d'un polynôme aléatoire de $\mathbb{K}[x]$ dont les coefficients sont également issus de l'anneau d'entiers fini $\mathcal{F}$.

3.3.2 Représentation des données cryptées et opérations réalisables

Dans le schéma FV, les données cryptées sont donc représentées sous la forme d'un polynôme de degré fini et dont les coefficients sont issus d'un anneau d'entiers fini $\mathcal{F}$. Ainsi m est par exemple représenté dans $\mathcal{X}$ par le polynôme $\hat{m}(x) = \sum a_i x^i$, où les coefficients a_i correspondent à la décomposition binaire de m, tel que $\hat{m}(2) = m$.

Il s'agit par ailleurs d'un schéma de chiffrement pleinement homomorphe contrôlé[31] possédant donc les propriétés additive et multiplicative dans $\mathcal{X}$. Ces deux opérations étant respectivement des additions et des multiplications polynomiales dans ce cadre. En sus, cette méthode incorpore une procédure de relinéarisation des données requérant une clef spécifique. Cette dernière peut être transmise sans crainte par l'assureur à l'opérateur extérieur (par exemple un service de *cloud-computing*) puisqu'elle permet uniquement de réduire la taille des éléments chiffrés, notamment à l'issue d'une multiplication entre deux éléments, passant d'une donnée cryptée (un polynôme) de degré deux à un élément de degré unitaire.

3.3.3 Implémentation et paramétrage de la procédure

Dans une recherche d'efficacité nous avons décidé d'utiliser une implémentation existante de ce schéma réalisé par L. Aslett au sein du package R `HomomorphicEncryption`[32].
L'utilisation de cette librairie nous permet de travailler directement et facilement dans un environnement multi-thread où chaque calcul (chiffrement, déchiffrement,

30. Au sens d'une norme sur le polynôme, le plus souvent la norme infinie correspondant au coefficient maximal du polynôme, i.e $\|p(x)\|_\infty = b$ où b est le plus grand coefficient de p.

31. Dans la littérature il est en réalité décrit comme un schéma pleinement homomorphe - possédant une procédure de nettoyage du bruit par bootstrap - mais l'implémentation que nous utilisons est cantonnée au cadre contrôlé.

32. Ce package n'est pas présent sur le CRAN et nécessite d'être compilé manuellement

opérations) dans $\mathcal{X}$ est automatiquement parallélisé. Par ailleurs ce package propose une implémentation des différentes opérations réalisables dans l'espace crypté directement à partir de scalaires, de vecteurs mais également de matrices d'entiers (addition, multiplication, produit matriciel). A noter que les opérateurs usuels sont surchargés [33] grâce à cette librairie : si l'on souhaite par exemple additionner deux éléments de $\mathcal{X}$, il suffit d'utiliser le symbole usuel + sans avoir à réécrire la totalité des opérations équivalentes à l'addition dans l'espace crypté.
Néanmoins, dans ce mémoire nous définirons par la suite les opérations d'addition, de soustraction, de multiplication et de produit matriciel dans l'espace crypté $\mathcal{X}$ par les symboles respectifs $\boxplus, \boxminus, \boxtimes$ et $\boxdot$. Cependant, et dans un souci de lisibilité, les éléments cryptés ne bénéficient pas d'une symbolique particulière [34].

Enfin ce package facilite également le paramétrage de la méthode de chiffrement (qui peut s'avérer complexe étant donné le nombre de paramètres différents et interdépendants qu'elle comporte) en proposant de déterminer les valeurs des paramètres les plus adéquates selon la définition d'un paramètre de sécurité et/ou de la profondeur multiplicative minimale nécessaire :

- *Le niveau de sécurité* (λ), exprimé en bits, tel qu'un niveau de sécurité de n-bits correspond à un crypto-système nécessitant à un attaquant de réaliser au moins 2^n opérations avant de réussir à le casser.

- *La profondeur multiplicative* (L), correspond au nombre de multiplications successives à évaluer sur les données chiffrées. Par exemple : $c_1 \boxtimes c_2 \Rightarrow L = 1$ et donc $c_1 \boxtimes c_2 \boxtimes \cdots \boxtimes c_n \Rightarrow L = n - 1$, mais $c_1 \boxtimes (c_2 \boxplus c_3) \Rightarrow L = 1$!

33. En informatique, signifie qu'une procédure/fonction à été ré-écrite pour satisfaire de nouveaux besoins et qu'elle peut ainsi remplacer l'implémentation existante si nécessaire.

34. Ici X peut à la fois faire référence au X non-crypté et au X crypté. Pour les différencier il suffit de prêter attention à la qualité des symboles des opérateurs. Par exemple : $X \boxdot Y$ signifie clairement qu'il s'agit d'un produit matriciel dans l'espace crypté et que donc X et Y sont cryptés.

3.4 Descente de gradient dans $\mathcal{X}$ et résultats obtenus

Nous présentons ici la méthode de descente de gradient employée dans l'espace crypté afin de pouvoir estimer la valeur de $\hat{\beta}$ sans utiliser la formule fermée précédemment énoncée, toujours dans le cadre d'une régression linéaire simple.

3.4.1 Estimer $\hat{\beta}$ à l'aide d'une descente de gradient

On se place dans ce paragraphe dans l'espace non crypté, avec $X \in \mathbb{R}^{n \times p}$ et $Y \in \mathbb{R}^p$:

Nous commençons par définir notre fonction objectif (que l'on cherche à minimiser), à partir de l'équation des moindres carrés ordinaires on pose $S(\beta) = \|y - X\beta\|_2^2$ et on définit le gradient de notre fonction objectif par :

$$\nabla S(\beta) = \frac{\partial S(\beta)}{\partial \beta} = -2X^T(Y - X\beta)$$

On obtient alors l'équation itérative de la descente de gradient suivante :

$$\begin{aligned} \beta^{[k]} &= \beta^{[k-1]} - \delta \nabla S(\beta^{[k-1]}) \\ &= \beta^{[k-1]} + \delta X^T(Y - X\beta^{[k-1]}) \end{aligned}$$

où k représente la k^{me} itération de l'algorithme de descente et $\delta \in \mathbb{R}$ son pas.

Par ailleurs un **lemme** nous permet de garantir la convergence de l'estimateur obtenu par la méthode de descente de gradient vers le coefficient de régression linéaire des moindres carrés, avec $\beta^{[0]} = 0_n$, tel que :

$$\lim_{k \to +\infty} \beta^{[k]} = (X^T X)^{-1} X^T Y = \hat{\beta}_{MC}, \ \ avec\, \delta \in [0, 2/\mathcal{S}(X^T X)]$$

où $\mathcal{S}(X^T X)$ représente le rayon spectral [35] de $X^T X$.

De plus Ryaben'kii et Tsynkov ont pu déterminer le choix optimal du pas de descente δ^*, soit $\delta^* = 2/(\lambda_{min} + \lambda_{max})$ permettant d'obtenir un rayon spectral de $X^T X$ optimal tel que $S^* = (\lambda_{max} - \lambda_{min})/(\lambda_{max} + \lambda_{min})$, avec λ_{max} (resp. λ_{min}) la valeur propre maximale (resp. minimale) de $X^T X$.

3.4.2 Réaliser une descente de gradient dans l'espace crypté

Afin de pouvoir réaliser notre descente de gradient dans $\mathcal{X}$ il est nécessaire de l'adapter aux contraintes engendrées par notre nouveau schéma de chiffrement.
On rappelle alors que cette méthode de chiffrement n'est valable que pour des éléments dans $\mathbb{Z}$ (le package R s'adaptant aux différentes structures de données : scalaires, vecteurs ou matrices), il est par ailleurs, à ce titre, impossible de réaliser des divisions entre deux éléments de $\mathcal{X}$ [36].

35. $\mathcal{S}(X) := \max_i |\lambda_i|$, avec λ_i la i^{me} valeur propre de X.
36. Ni une multiplication comprenant des valeurs initiales $\in]0, 1[$.

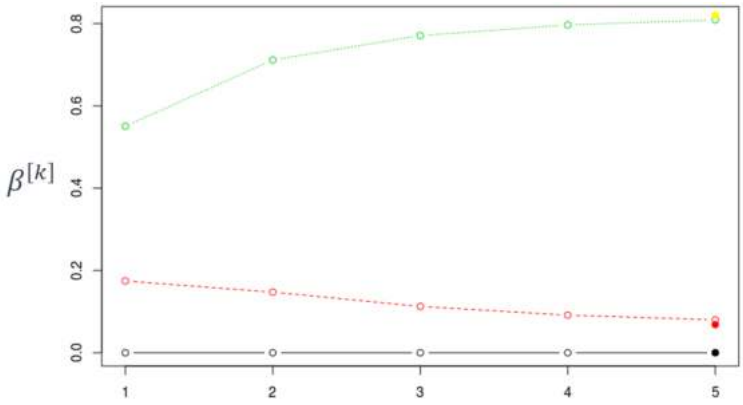


FIGURE 8 : Evolution de $\beta^{[k]}$ en fonction du nombre d'itérations sur notre jeu de données non crypté - chaque couleur correspond à une composante de $\beta^{[k]}$ différente, les points entièrement colorés sont les valeurs des composantes de $\hat{\beta}_{MC}$.

Il nous est donc nécessaire pour manipuler les réels d'utiliser un paramètre de précision que l'on notera $\gamma \in \mathbb{N}$, tel que $\forall x \in \mathbb{R}$, on a $\hat{x} = \lfloor 10^{\gamma} x \rceil \in \mathbb{Z}$.
Finalement, afin de calculer notre descente de gradient dans l'espace crypté nous devons d'abord procéder à une transformation de certains paramètres, et en particulier le pas δ, en effet puisqu'il nous est impossible de réaliser des divisions dans $\mathcal{X}$, nous posons désormais $\delta \equiv 1/\nu$ où $\nu \in \mathbb{N}$, tel que l'équation itérative de la descente de gradient dans l'espace crypté soit [37] :

$$\begin{aligned}\tilde{\beta}^{[k]} &\equiv 10^{\gamma}\tilde{\nu}\tilde{\beta}^{[k-1]} \boxplus \tilde{X}^T(10^{k\gamma}\tilde{\nu}^{k-1}\tilde{Y} \boxminus \tilde{X}\tilde{\beta}^{[k-1]}) \\ &= 10^{(2k+1)\gamma}\nu^k\beta^{[k]}\end{aligned}$$

où toutes les variables munies d'un tilde sont transformées tel que, par exemple : $\tilde{X} = 10^{\gamma}X$, hormis $\beta^{[k]}$ dont le facteur de mise à l'échelle dépend du nombre d'itérations (cf. équation ci-dessus). La totalité du raisonnement et des étapes de calculs pour obtenir l'équation ci-dessus est disponible en annexe (C).
En remarquant que les facteurs d'échelle sont indépendants des données on peut regrouper certains d'entre eux (notamment $10^{k\gamma}\tilde{\nu}^{k-1}$) afin de les chiffrer directement en un unique bloc et ainsi notamment réduire la profondeur multiplicative nécessaire à $2K$, où K représente le nombre limite d'itérations pour la descente de gradient.

Une fois la descente de gradient terminée (calculée dans $\mathcal{X}$, lorsqu'elle atteint l'itération limite K) il suffit pour retrouver les valeurs des coefficients de $\beta^{[K]}$ de calculer pour le client $\mathrm{Dec}(sk, \tilde{\beta}^{[K]})/(10^{(2K+1)\gamma}\nu^K)$ [38], où sk représente la clef secrète du chiffrement.

Nous avons testé cette méthodologie sur notre jeu de données [39] en définissant notre

37. Les opérations $\boxtimes$ et $\boxdot$ n'ont pas été explicitées pour plus de lisibilité.
38. Dans le cas de la prédiction, calculer $\mathrm{Dec}(sk, \tilde{X}\tilde{\beta}^{[K]})/(10^{(2K+2)\gamma}\nu^K)$
39. Notre donnée de 25 observations et deux variables pour prédire le taux de graisse dans le sang

paramètre de précision $\gamma = 1$ (soit de ne garder qu'une décimale pour les nombres réels).
Nous avons également ajouté dans notre méthodologie une étape de *pseudo-bootstrapping* à la sixième itération ($k = 6$), qui déchiffre $\widetilde{\beta^{[k]}}$ afin de nettoyer la donnée (et surtout réduire sa taille en mémoire) avant de la ré-encrypter pour continuer les calculs. Cette étape, totalement irréaliste en pratique, nous permet ici de poursuivre la descente de gradient jusqu'à la dixième itération sur notre machine de test alors que cette procédure, comme nous le soulignerons par la suite, est très exigeante en termes de ressources matérielles.

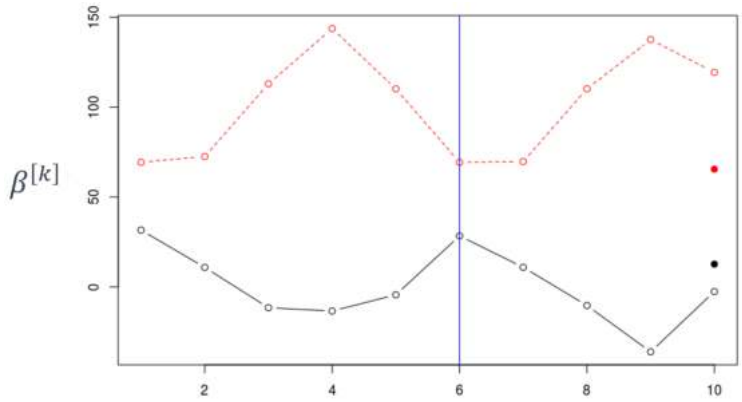


FIGURE 9 : Évolution de $\beta^{[k]}$ en fonction du nombre d'itérations sur notre jeu de données cryptées, après déchiffrement - chaque couleur correspond à une composante de $\beta^{[k]}$ différente, les points colorés sont les valeurs des composantes de $\hat{\beta}_{MC}$.

La procédure de descente de gradient a été réalisée sur le même jeu de données que pour la procédure de régression linéaire cryptée précédente, néanmoins ici les données ont été centrées en amont ce qui explique des valeurs de $\beta^{[k]}$ et de $\hat{\beta}_{MC}$ différentes de celles calculées précédemment. Par ailleurs le ν choisi ici (où $X^T X$ est une matrice carrée 2×2) est optimal et correspond à $(2/\mathrm{Tr}(X^T X))^{-1}$.

On remarque alors grâce à la figure (9) que les estimations des coefficients $\beta^{[k]}$ semblent alterner dans l'espace non-crypté, ce qui souligne la nature oscillatoire de cette méthode d'estimation.
Il est alors naturellement difficile de déterminer la valeur de K stoppant le processus en espérant obtenir l'estimation la plus précise possible de $\hat{\beta}$. En effet, même si l'on est tenté de vouloir prendre K le plus grand possible, on se rend compte ici que si malgré tout notre K n'est pas suffisamment important (ici $K = 10$ "seulement") , une valeur plus petite de K (i.e $K = 2$ ici) peut fournir un résultat plus convaincant. Il est par ailleurs impossible de se fier à un critère d'arrêt du type $\|\beta^{[k]} - \beta^{[k-1]}\| \leqslant \xi$ dans l'espace crypté puisque même si celui-ci peut être muni d'une métrique, l'ensemble $\mathcal{X}$ n'est pas ordonné.
Au regard de la figure (10) on peut d'ailleurs constater que les résultats des régressions linéaires cryptées (où $K = 5$) et usuelles sont dans l'ensemble assez similaires ce qui tend à nous laisser penser que l'erreur engendrée par le choix de K s'avère

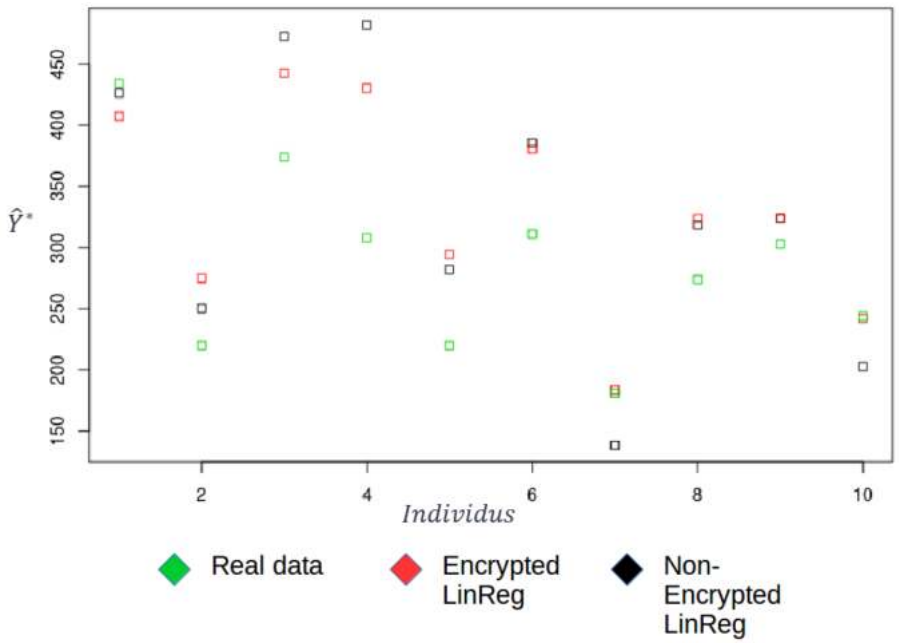


FIGURE 10 : Comparaison des prédictions $\hat{Y}^*$ obtenues grâce aux différentes méthodes de régression linéaire sur notre jeu de données (Encrypted LinReg avec $K = 5$)

raisonnable pour le calcul des prédictions. Et ce d'autant plus que le modèle de régression linéaire est généralement assez imprécis, si bien qu'ici par exemple la méthode cryptée se révèle en moyenne plus précise que la méthode non-cryptée, en comparant aux données réelles (en vert).

Enfin à l'issue du *pseudo-bootstrapping* on constate sur la figure (9) que les sauts suivants des estimateurs des coefficients paraissent à la fois avoir moins d'amplitude et être plus nets ce qui laisse effectivement présager que si l'on était en mesure de réaliser suffisamment d'itérations (mais combien ?) nous serions capables d'obtenir une approximation proche de $\hat{\beta}_{MC}$.
C'est d'ailleurs cette idée d'un nombre d'itérations suffisant qui permet d'expliquer pourquoi la descente de gradient dans l'espace crypté parvient à déterminer une approximation de $\hat{\beta}$. En effet lorsque la descente de gradient est calculée dans l'espace cryptée, on sait qu'à chacune de ses itérations l'approximation tend à se rapprocher de la valeur du point critique dans l'espace non-crypté (si le pas est bien choisi et que la fonction de coût est convexe), tandis ce que dans $\mathcal{X}$ les valeurs vont alterner : seul un nombre suffisamment important d'itérations permet de garantir que le résultat finalement obtenu dans l'espace crypté correspond bien à une approximation de $\hat{\beta}_{MC}$. Une représentation schématique de ce comportement est modélisé au travers de la figure (11).
Cependant cette méthode demande beaucoup de ressources afin de pouvoir réaliser ces opérations, comme nous allons l'évoquer par la suite.

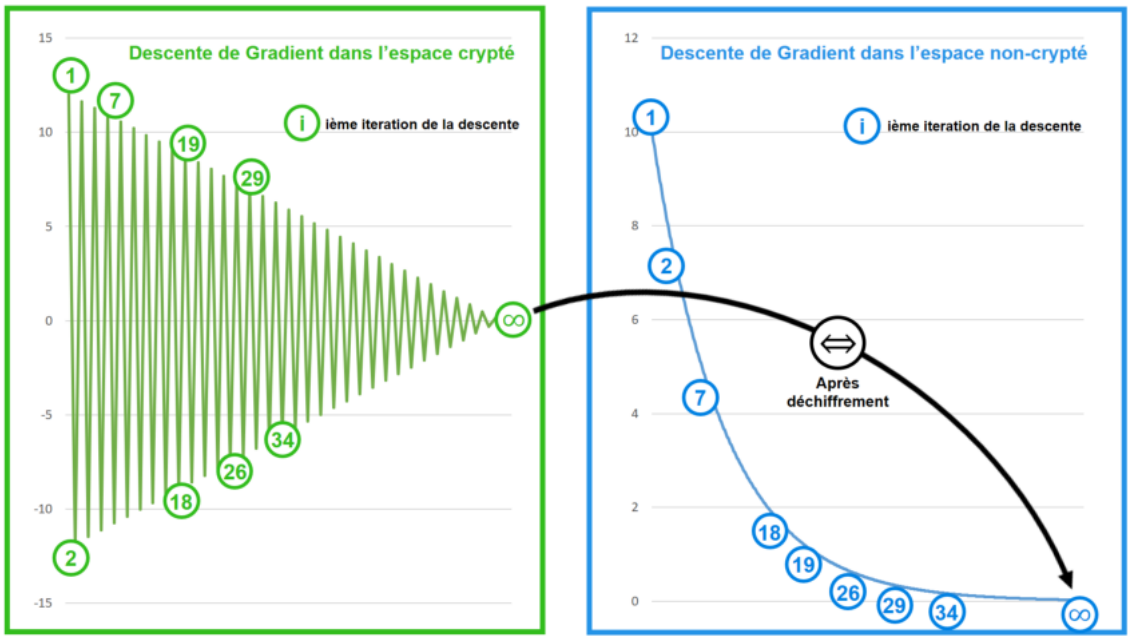


FIGURE 11 : Schéma établissant le lien entre la descente de gradient dans l'espace crypté et dans l'espace non-crypté

3.4.3 Avantages & inconvénients de la méthode de descente de gradient

Parmi les points positifs de cette méthode on retrouve le fait que la procédure de chiffrement FV employée s'avère efficace et permet notamment de pouvoir facilement gérer l'augmentation du bruit généré au cours des calculs grâce à la relinéarisation des données.
Par ailleurs les prédictions obtenues avec le modèle s'avèrent assez précises tout en évitant au client de devoir pré-calculer certains éléments pour réaliser les opérations comme cela était le cas avec la méthode précédemment présentée en section (3.1). En effet ce dernier fournit ici simplement X et Y à l'opérateur externe afin de calculer les coefficients de la régression linéaire et les prédictions $\hat{Y}^*$.

Cependant la descente de gradient réalisée ici ne fournit qu'une estimation de l'estimateur du coefficient de régression linéaire β [40]. De plus le critère d'arrêt de l'algorithme est délicat à déterminer puisqu'il est forcément basé sur le nombre maximal d'itérations K qui est contraint par ailleurs par la capacité physique de calcul de l'opérateur. Alors que le nombre d'itérations à réaliser afin de s'approcher de la valeur de $\hat{\beta}_{MC}$ dépend avant tout de la corrélation entre les variables de X, une forte corrélation conduit nécessairement à des propriétés spectrales de X^TX moins favorables (la matrice est moins bien conditionnée, c'est à dire qu'elle peut être numériquement difficile à inverser par exemple) or le pas de notre descente de gradient dépend de cette propriété ce qui donc rend la convergence d'autant plus lente.
Toutefois le principal défaut de cette méthode de calcul réside dans la configuration matérielle nécessaire pour réaliser convenablement les opérations. A titre indicatif,

40. En quelques sortes, il s'agit "d'une estimation d'une estimation".

pour notre jeu de données composé de seulement deux variables et 25 observations, il nous a fallu près de 50 minutes de calcul pour atteindre la sixième itération sur notre machine de test [41] (contre 8 minutes si l'on s'arrête à la deuxième itération). Nous atteignions alors une saturation de la mémoire disponible (16 Go) et l'utilisation CPU plafonnait à 100% pour chacun des coeurs logiques.
Le coût et la complexité des calculs de notre procédure s'expliquent notamment par le fait que le choix du nombre d'itérations à réaliser K influe sur le paramétrage du chiffrement au travers de la valeur de la profondeur multiplicative maximale L [42]. Or L influe également indirectement sur le paramètre de sécurité λ ce qui conduit inévitablement lorsque l'on souhaite calculer un nombre plus important d'itérations à devoir le faire sur des données à la fois plus à même de supporter davantage de multiplications successives mais également plus sécurisées et donc naturellement plus "lourdes". Finalement réaliser la descente de gradient en augmentant le nombre d'itérations conduit à réaliser plus d'opérations alors que chacune d'entre elles s'avère plus difficile à calculer à mesure que K augmente [43].
C'est pourquoi nous avons décidé d'ajouter à notre procédure de test une étape de *réinitialisation de la mémoire* qui nous permet d'augmenter artificiellement le nombre d'itérations que l'on peut calculer en conservant une occupation mémoire raisonnable (sans cette étape il nous est impossible de dépasser la 6^{me} itération sur notre machine de test sans crash en raison d'un dépassement mémoire). A titre indicatif au sein de l'article de P.M Esperança, L.J.M Aslett et C.C Holmes *Encrypted accelerated least squares regression* (2017), en réalisant cette procédure de calcul sur un jeu de données de 25 variables et 100 observations, il aura fallu près 80 minutes pour atteindre la quatrième itération ($K = 4$) sur une machine autrement plus performante (48 coeurs et 32Go de RAM).
Dès lors, on comprend pourquoi nous avons décidé de nous intéresser ici exclusivement à un jeu de données académique de petite taille, tant le recours à des données réelles semble rédhibitoire au niveau de la puissance de calcul requise.

41. i7-5600 @2.6Ghz (2 coeurs physiques, 4 coeurs logiques) et 8Go de RAM (+ 8Go de swap)
42. Pour rappel, on a $L = 2K$.
43. Attention on parle bien de K le nombre maximal d'itérations et pas de k représentant la k^{me} itération.

3.5 Synthèse des méthodes employées

Au cours de cette partie nous avons étudié deux méthodes qui peuvent permettre à un assureur de déléguer des calculs (ici nous nous sommes attachés en particulier à l'ajustement d'une régression linéaire et au calcul des prédictions) à un opérateur externe en toute sécurité grâce à des schémas de chiffrement homomorphes dans le cadre de données pseudonymisées.

La première méthode nous a permis de comprendre facilement le fonctionnement concret d'une méthode de chiffrement homomorphe (Efficient Integer Vector Homomorphic Encryption), notamment en l'implémentant sous `Python`, et d'obtenir rapidement à la fois $\hat{\beta}$ et $\hat{Y}^*$, moyennant le calcul au préalable de la part du client de certains éléments.
Tandis que la seconde méthode, basée sur le schéma FV permet de déterminer une approximation de $\hat{\beta}$ directement à partir des données X et Y fournies par le client grâce à une descente de gradient dans l'espace crypté, mais au détriment de la rapidité d'exécution. Une synthèse des propriétés des deux méthodes employées est réalisée au sein de la table (12) ci-dessous.

Méthode Efficient Vector Homomorphic Encryption	
Avantages	**Inconvénients**
✓ Méthode de chiffrement simple à implémenter et à comprendre.	✗ Sécurité garantie par la méthode de chiffrement non-optimale.
✓ Utilisation de la formule fermée des MCO pour calculer $\hat{\beta}$ dans $\mathcal{X}$.	✗ Evolution du bruit au cours des calculs dans $\mathcal{X}$ non contrôlée.
✓ Ressources matérielles nécessaires et temps d'exécution modérés.	✗ Nécessite des transformations/pré-calculs sur les données par le client.
✓ Renvoi la valeur «réelle» de l'estimation de β.	

Méthode Descente de Gradient (F&V)	
Avantages	**Inconvénients**
✓ Méthode de chiffrement offrant un niveau de sécurité personnalisable.	✗ Schéma de chiffrement très exigeant et basé sur des notions complexes.
✓ La quantité de bruit introduite durant les opérations est déterminable en amont.	✗ Retourne seulement une estimation de l'estimateur $\hat{\beta}$ de β.
✓ Le client fournit directement X et Y cryptés à l'opérateur de calculs.	✗ Très gourmand en ressources et possiblement inadapté au contexte actuel.

FIGURE 12 : Tableau synthétique des méthodes employées

Finalement, après avoir réalisé ses deux procédures et les avoir rendues fonctionnelles, nous sommes forcés de constater qu'aucune d'elles ne paraît, à l'heure actuelle et à la lumière des schémas de chiffrement homomorphes existants ainsi qu'à la puissance de calcul disponible aujourd'hui, disposée à pouvoir être utilisée en production. Et ce notamment en raison des contraintes inhérentes à chacune d'elles exposées ci-dessus et développées précédemment au cours de cette partie. De plus notre étude était restreinte à un modèle de régression linéaire simple. Les modèles de tarification en pratique étant plus complexes, il faudrait alors les remplacer par des modèles tarifaires plus simples si ces méthodes venaient à être appliquées. Il est cependant fort à parier que les choses seront amener à changer rapidement dans

ce domaine tant il représente une branche particulièrement dynamique de la recherche, commune à la fois à la statistique, la cryptographie ainsi qu'aux sciences informatiques.

Après avoir exploré la possibilité de réaliser des calculs, et plus précisément une régression linéaire, sur des données pseudonymisées, nous présenterons dans la section suivante l'élaboration d'une méthodologie de tarification automobile sur des données anonymisées en aggrégeant lignes à lignes les individus de la base grâce à des méthodes de machine learning non-supervisées et de clusterings.

4 Procédure anonymisée de tarification non-vie

Alors que le RGPD est entré en vigueur le 25 mai 2018, les entreprises d'assurance doivent se soumettre à ces nouvelles contraintes réglementaires. Si la plupart d'entre elles choisissent de pseudonymiser leurs données conformément aux préconisations du règlement européen, nous avons pu constater dans la partie précédente qu'il n'était pour autant ni simple ni efficace (pour le moment) de réaliser certaines procédures statistiques élémentaires directement sur ces données cryptées. Il devient alors le plus souvent nécessaire pour ces entreprises de s'extraire temporairement de ce cadre afin de pouvoir procéder à l'analyse de leurs jeux de données. De plus, même s'il existe d'autres techniques de pseudonymisation que la cryptographie proposée précédemment, ce choix ne dispense pas de l'application des règles du RGPD, à l'inverse de procédures d'anonymisation.

C'est pourquoi nous nous intéressons dans cette partie à l'élaboration d'une technique d'anonymisation [44] d'une base de données de sinistres automobiles dans le but de pouvoir réaliser un modèle de tarification directement sur ces données anonymisées.
Dans la précédente partie nous avons présenté une méthode basée sur un cryptage homomorphe afin de protéger des données sensibles lors de leurs échanges à des tiers et ainsi permettre de transmettre ces informations à des entités externes sans craindre que ces dernières puissent utiliser ces données, ou qu'elles soient interceptées lors de l'échange par des tiers indésirables et qu'elles soient exploitées à des fins illégales.
Ici l'objectif est de protéger les données en interne, c'est à dire d'éviter que des informations sensibles contenues au sein de la base ne puissent être divulguées de manière intentionnelle ou non. On rappelle ici que le caractère anonymisé d'une base de données s'apprécie au regard de l'impossibilité de ré-identifier un individu spécifique (relativement à des moyens jugés raisonnables employés pour y parvenir) au sein du jeu de données ainsi obtenu. Une approche plus approfondie de ce concept ainsi que sa définition selon le RGPD sont présentées au sein de la partie (2.1.3) de ce mémoire.

Avant de présenter la méthode d'anonymisation que nous proposons dans notre cadre de tarification automobile, il convient de faire brièvement un tour des méthodes d'anonymisation les plus répandues actuellement afin de voir leurs avantages et inconvénients.
En premier lieu on retrouve la technique mentionnée dans la partie (2.1.3), consistant à crypter la base de données à l'aide d'une procédure à clef secrète puis de supprimer cette clef afin de détruire toute possibilité de ré-identification (voir schéma (1)), hormis par la *force brute*, en testant toute les clefs possibles une à la fois (ce qui ne constitue pas véritablement une méthode raisonnable en terme de temps de calcul

44. Données dont il est impossible de ré-identifier des informations individuelles.

si le cryptage fournit un niveau de sécurité décent [45]). Cependant cette méthode ne permet plus de manipuler les données une fois anonymisées puisque ces dernières sont devenues totalement inexploitables (perte d'interprétation, relation d'ordre, quantification, etc.). Dès lors il s'agit d'une méthode d'anonymisation définitive.
Une autre méthode d'anonymisation répandue consiste à ajouter du bruit aux données. Par exemple transformer l'âge des individus d'une base afin de le présenter avec une précision de +/-10 ans ou encore ajouter à des données numériques une variable aléatoire gaussienne centrée si l'étude porte particulièrement sur les valeurs moyennes alors l'ajout de termes d'erreurs n'a pas d'impact sur le résultat final. Néanmoins avec cette technique les individus restent potentiellement identifiables, leurs données en revanche deviennent moins fiables. Il devient alors discutable au regard du RGPD de considérer cette méthode comme une procédure d'anonymisation [46].

Il existe également des principes d'anonymisation, dits par *généralisation* dont nous nous inspirerons en partie pour l'élaboration de notre méthode, notamment les principes de *k-anonymisation* et *I-diversité*.
La technique de la *k-anonymisation* doit permettre d'empêcher d'identifier un individu au sein d'une base de données. A ce titre elle correspond donc pleinement à la définition de l'anonymisation telle que mentionnée au sein du RGPD. Avant de commencer, on considère la base exempte de clef primaire (succession de variables qui permet de déterminer chaque individu de la base de manière unique, le plus souvent le couple nom-prénom ou un ID par exemple) après suppression des variables nécessaires. Par la suite cette méthode regroupe les enregistrements possédant une clef d'identification pouvant servir de clef primaire au sein de groupes de k enregistrements dont la nouvelle clef d'identification du groupe est plus générale et commune à tous ses éléments (*quasi-identifiant*). Le schéma ci-dessous (13) synthétise la procédure de *k-anonymisation*, en prenant $k = 3$ sur une base triviale, où l'on considère la variable `Damage` comme une donnée sensible. L'intérêt de cette méthode réside dans sa capacité à empêcher de pouvoir identifier un individu d'une base en la recoupant avec d'autres bases de données (*record linkage*) comme cela peut parfois être possible avec certaines méthodes de pseudonymisation. De plus cette procédure continue de fournir des résultats exacts à l'issue de l'analyse des données et quelques soient les opérations réalisées (à l'inverse de l'ajout de bruit), cependant il est désormais devenu impossible de dissocier les k enregistrements d'un même groupe.
Néanmoins cette méthode connaît plusieurs vulnérabilités, notamment aux attaques par homogénéité (a) ou par connaissance antérieure (b) :

45. Comme nous l'avons vu précédemment dans la partie (3.3.3), ce niveau exprimé en bits (le plus souvent la taille de la clef) correspond au nombre d'opérations à réaliser avant de réussir à casser le cryptage en brut-force (cela ne tient pas compte d'éventuelles failles du crypto-système). Par exemple pour le cryptage AES-256 bits, l'un des plus robuste à ce jour, il faudra 2^{256} ($\sim 10^{77}$) opérations pour espérer décoder son chiffrement.

46. Des chercheurs ayant étudié une base anonymisée (suppression d'identifiants) de 100 millions d'évaluations (bruitées, tant sur l'échelle de notation que sur la date de rédaction) de Netflix provenant de 500 000 utilisateurs sur près de 18 000 films sont parvenus à ré-identifier 99% des enregistrements à l'aide de seulement 8 évaluations et des dates grâce à une étude combinatoire.

a. Assez simplement, le schéma (13) représente une base 3-anonyme vulnérable à une attaque par homogénéité. Lorsque les enregistrements d'un même groupe possède la même valeur de la variable sensible (celle dont on souhaite spécifiquement protéger l'accès), alors naturellement les données ne sont plus anonymisées : dans le schéma, on peut facilement en déduire qu'une personne exerçant une profession libérale, âgée de 39 ans a nécessairement subit un sinistre très important.

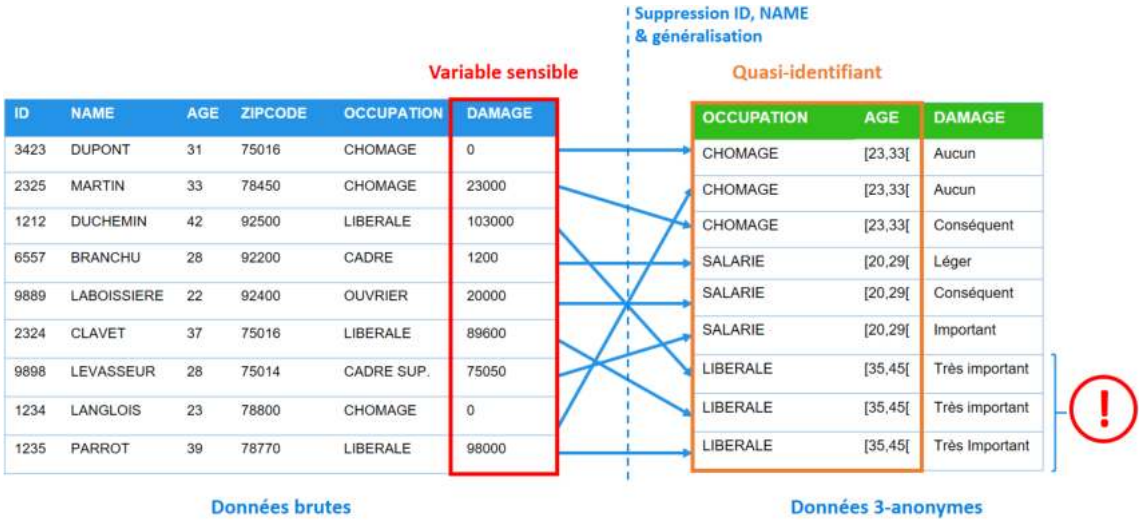


FIGURE 13 : Schéma & limites de la k-anonymisation

b. Si un attaquant connaît suffisament d'informations sur un individu, alors il peut inférer de la base k-anonymisée des informations sensibles. Par exemple à partir du schéma (13), si M. DUMONT sait que son voisin (M. MARTIN) est au chômage alors qu'il vient de fêter son 33^{eme} anniversaire et qu'il a récemment eu un accident, alors il peut en déduire qu'il s'agissait d'un sinistre conséquent.

De plus cette procédure implique un problème technique de taille : En effet le problème posé par la détermination des généralisations à effectuer sur les variables afin d'obtenir les quasi-identifiants optimaux est NP-complexe et requiert donc l'emploi d'heuristiques [47] afin de pouvoir être résolu efficacement. Afin de limiter les risque de fuites d'informations de type (a), le modèle de I-diversité a été développé.

La procédure de *I-diversité* est basée sur le principe de *k-anonymisation* que l'on a vu précédemment mais en ajoutant une contrainte supplémentaire au sein de chaque groupe. Non seulement chaque classe doit être composée d'au moins k individus partageant le même quasi-identifiant, mais désormais chaque classe doit compter au moins I modalités différentes de la variable sensible. Comme nous pouvons le

47. Bayardo and Agrawal, *Data Privacy through Optimal k-anonymization* (2005)

constater grâce au schéma ci-dessous (14), il est parfois nécessaire de changer le quasi-identifiant de la base pour y parvenir.

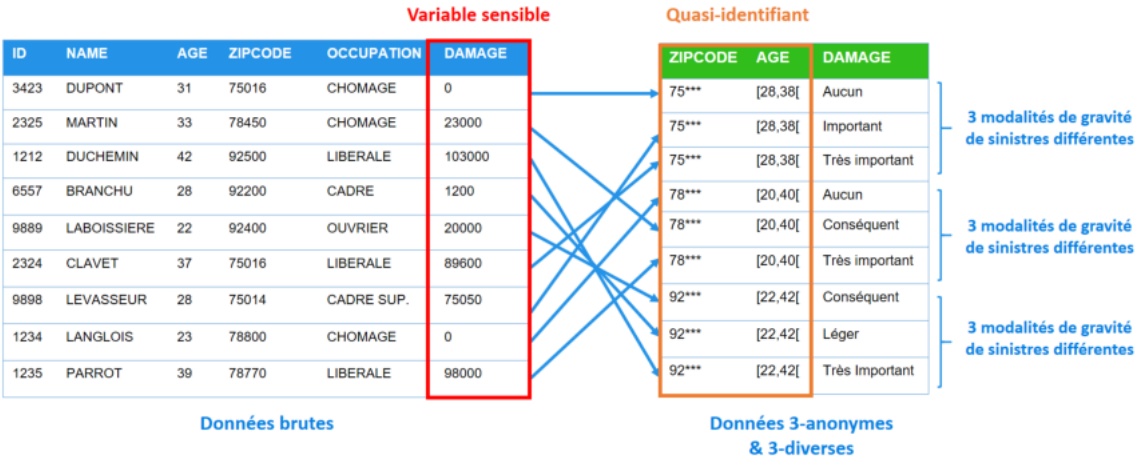


FIGURE 14 : Schéma & limites de la I-diversité

Bien qu'il existe plusieurs manière de réaliser cette méthode [48] (le schéma présente la méthode de I-diversité distincte), cette procédure comporte toujours des limitations très importantes.
En effet si les modalités de la variable sensible ne sont pas convenablement réparties au sein des différentes classes alors il est possible d'inférer certaines informations. Par exemple dans le schéma (14), on peut être certains que toutes les personnes habitant dans le 92 ont subit un sinistre.
Mais la répartition au sein d'une même classe des modalités de la variable sensible a également son importance. En effet en considérant une variable sensible à seulement deux modalités (booléenne) dans le cadre d'une base 100-anonymisée, si une des classes de 100 enregistrements comporte 99 fois la valeur 1 et une fois la valeur 0 pour la variable sensible, alors si un attaquant connaît le quasi-identifiant d'un individu appartenant à cette classe il peut en déduire avec une grande probabilité que la variable sensible pour cet individu vaut 0.
Afin de contourner ce dernier problème la méthode de t-proximité a été développée. Elle permet en particulier de réduire la corrélation entre les variables constituants le quasi-identifiant et la variable sensible, permettant ainsi d'obtenir des partitions homogènes vis à vis de cette variable. Malheureusement le recourt à cette méthode conduit inévitablement à réduire considérablement le domaine d'étude de la base finale, le restreignant le plus souvent à de simple statistiques descriptives. Par exemple, nous ne pourrions pas utiliser cette procédure pour élaborer notre méthode de tarification anonymisée puisque les résultats des méthodes prédictives (GLM, CART, Random Forest,etc.) à partir des covariables du quasi-identifiant (BONUS/MALUS, VEHICULE, etc.) ne permettront pas d'expliquer convenablement les valeurs de la variable sensible (coût/fréquence des sinistres), puisque ces dernières sont désormais le plus possible décorrélées.

48. Notamment la I-diversité par l'entropie ou encore la I-diveristé récursive.

Enfin ces méthodes et en particulier la k-anonymisation s'étend très difficilement à des bases de données de grandes dimensions (en termes de nombre de variables-modalités, parallèlement au concept de fléau de la grande dimension [49]) comme l'a démontré C.C Aggarwal en 2005 [50].
En effet ce que montre cette étude c'est qu'il devient impossible de préserver la confidentialité d'une base 2-anonymisée (soit pour un même quasi-identifiant ne pouvoir distinguer un enregistrement d'au moins un autre) contenant beaucoup de variables avec un nombre important de modalités sans sacrifier une quantité très importante d'information lors de l'étape de généralisation en raison du nombre exponentiel de combinaisons pouvant servir à constituer un quasi-identifiant, même lorsque les variables ont été préalablement formatées au sein d'intervalles, si bien que le résultat final n'en devient plus véritablement exploitable.

C'est en sens que nous avons décidé de travailler sur une alternative aux méthodologies proposées jusqu'ici. Pour rappel, nous nous plaçons ici dans le cadre de données anonymisées [51], à partir desquels nous établiront un modèle de tarification automobile. L'originalité de la méthode proposée réside dans la manière dont est réalisée l'anonymisation de la base de données sous R (3.4). Alors qu'il est courant de recourir à des techniques d'agrégation, voir de suppressions ou de modifications des variables comme nous l'avons vu précédemment, nous avons décidé au contraire de nous concentrer sur une procédure d'anonymisation basée sur une agrégation ligne à ligne des observations de la base grâce à des méthodes d'apprentissages non-supervisées et de clusterings. En effet cette dernière solution présente deux avantages comparativement aux méthodes d'anonymisations précédentes :

- Elle permet de garantir que les données sont anonymisées puisqu'il est devenu impossible de retrouver de l'information provenant d'un individu spécifique une fois l'agrégation réalisée. Ainsi nous sommes finalement exemptés des contraintes introduites par le RGPD.

- Elle constitue de surcroît un bon moyen de compresser la base de données initiale (afin d'accélérer les calculs mais également de réduire l'espace de stockage nécessaire pour la détenir) en perdant le moins d'informations possible pouvant être utile à l'élaboration d'un tarif, là où les méthodes précédemment mentionnées s'avèrent loin d'être optimales.

Pour implémenter cette procédure nous nous sommes servi d'une base de données que nous présenterons, mais d'abord nous commencerons par définir précisément la méthodologie que nous avons employée afin de tester et comparer les résultats de notre procédure anonymisée de tarification. Par la suite nous détaillerons les différentes méthodes d'agrégation ligne à ligne que nous avons réalisées avant d'interpréter les résultats obtenus et de synthétiser les étapes de la méthode proposée.

49. Voir note (56).
50. C.C Aggarwal *On k-Anonymity and the Curse of Dimensionality*, 2005.
51. Données dont il est impossible de ré-identifier des informations individuelles.

4.1 Méthodologie & modèle benchmark ligne à ligne

Pour commencer nous expliciterons précisément le cadre que nous avons défini pour comparer le modèle ligne à ligne classique et le modèle anonymisé ainsi que les critères qui ont été employés.
Puis la deuxième sous-partie se concentrera sur l'élaboration du modèle de tarification ligne à ligne en soulignant l'approche de modélisation choisie ainsi que les algorithmes mis en oeuvre au sein de la procédure.

4.1.1 Méthodologie globale

Afin de réaliser notre procédure de tarification anonymisée nous avons eu naturellement besoin de réaliser en premier lieu un modèle ligne à ligne classique qui nous servira de base de comparaison au modèle ajusté sur les polices agrégées.
Ce premier modèle ne sert par ailleurs pas uniquement à la comparaison, il est également employé comme base du modèle anonymisé.

En effet, notre idée consiste à établir le modèle anonymisé à partir du même modèle ligne à ligne. L'enjeu de la comparaison de ces deux modèles réside alors dans leurs différents ajustements.
Notre premier modèle ligne à ligne est ajusté à partir d'un échantillon d'apprentissage (60% du jeu de donnée initial) puis évalué selon un échantillon de test (40% du jeu de donnée initial) [52]. Tandis que notre modèle anonymisé reprend exactement les mêmes étapes du précédent modèle (mêmes transformations de variables, mêmes algorithmes, etc.) mais ajusté à partir de la version agrégée par polices du même échantillon d'apprentissage que pour le modèle ligne à ligne. En d'autres termes le même modèle est entraîné non plus sur des données individuelles mais sur des données anonymisées dont chaque ligne représente en quelques sorte un " contrat moyen" au sein d'un même groupe/cluster. Au sein de cet échantillon il est donc devenu impossible de retrouver de l'information individuelle (en partant du principe que chaque groupe/cluster est composé au minimum de deux polices distinctes). Une fois le modèle anonymisé ajusté il est également évalué sur le même échantillon de test que le précédent modèle.

La comparaison des deux modèles s'effectue alors principalement selon deux points de vue : d'une part globalement en observant la prime pure prédite en moyenne sur l'échantillon de test (ici on a simplement calculé la somme des primes pures de l'échantillon de test pour chacune des deux méthodes), puis individuellement en étudiant les écarts de prédictions pour chaque observation de test entre les deux modèles.
On comprend alors que l'enjeu principal pour réaliser notre modèle anonymisé réside dans le choix de la méthode de partitionnement des données afin que celui-ci permette d'obtenir des groupes d'individus dont leurs caractéristiques (les "X") soient les plus homogènes possibles sans pour autant avoir de certitude sur l'homogénéité de leurs sinistralités respectives.

52. Il s'agit d'un découpage du jeu de données assez conservateur : on retrouve également d'autres propositions comme par exemple 80% d'apprentissage - 20% de test en pratique.

Puisque notre objectif n'est pas de présenter le meilleur modèle de tarification possible sur notre jeu de données mais simplement de proposer une méthode de tarification alternative à partir de données anonymisées, nous ne nous intéresserons pas outre mesure ni à la qualité du modèle initial ni aux écarts entre la sinistralité prédite à l'issue de notre méthode anonymisée et la sinistralité observée. L'objet de notre étude portant sur l'écart potentiel de précision entre une tarification classique et une tarification anonymisée, nous considérons ici le modèle classique comme référence, indépendamment de la qualité de ses prédictions comparées à la sinistralité observée.

4.1.2 Présentation du jeu de données

La base de données qui nous servira à établir notre modèle de tarification correspond à un historique de 100 000 polices d'assurances automobiles sur une période de deux ans. Ces données sont issue du *Pricing Game* qui s'est déroulé lors de la journée *100% Actuaires* en 2015, elles sont toujours disponibles à cette adresse [53].
Nous avons au préalable réduit notre jeu de données à exactement 100 000 lignes et renommé certaines variables dont voici la description :

- `POL_NUM` correspond au numéro de la police ($\sim$ ID).
- `CAL_YEAR` spécifie l'année de souscription du contrat.
- `EXPOSITION` correspond à l'exposition en jours de la police ($\in [\![91, 365]\!]$).
- `DRIVER_GENDER` spécifie le genre du conducteur/assuré (*Booléen*).
- `DRIVER_OCC` défini la statut professionnel de l'assuré ($\subseteq$ {'Employed', 'Unemployed', 'Self-employed', 'Housewife', 'Retired'}).
- `DRIVER_AGE` correspond à l'âge du conducteur/assuré (en années).
- `VEH_TYPE` défini le type de véhicule conduit (6 modalités allant de A à F).
- `VEH_CATEGORY` spécifie la catégorie du véhicule ($\subseteq$ {'Small', 'Medium', 'Large'}).
- `VEH_GROUP` correspond au groupe du véhicule ($\in [\![1, 20]\!]$)
- `VEH_VALUE` correspond à la valeur de véhicule ($\in [\![1000, 49995]\!]$).
- `POL_DURATION` défini l'ancienneté du contrat ($\in [\![0, 15]\!]$).
- `BONUS_MALUS` spécifie le montant de BONUS/MALUS du conducteur ($\in [\![-50, 150]\!]$).
- `DAMAGE_GUAR` correspond à l'indicateur d'une garantie dommage (*Booléen*).
- `STATE` défini la région de résidence du conducteur (10 modalités)
- `COUNTY` spécifie la sous-région de résidence du conducteur (471 modalités)
- `COUNTY_DENSITY` correspond à la densité de population de la sous-région de résidence du conducteur ($\in$[14.38, 297.39])
- `NUM_LIAB_DAM` correspond aux nombres de sinistres RC matériels ($\in [\![0, 7]\!]$).
- `NUM_LIAB_BOD` correspond aux nombres de sinistres RC corporels ($\in [\![0, 3]\!]$).

53. `http://freakonometrics.free.fr/training.csv`

- `INC_LIAB_DAM` correspond au coût total des sinistres RC matériels ($\in [0, 12878.4]$).
- `INC_LIAB_BOD` correspond au coût total des sinistres RC corporels ($\in [0, 69068]$).

Nous comptons donc 20 variables au sein de notre jeu de données qui ne nécessite pas de retraitements particuliers puisqu'il ne présente ni données manquantes ni entrées aberrantes. Les principales informations sur ces données sont contenues dans le tableau ci-dessous (15).

name	simple_type	nb_modalities	max	mean	min	stddev
POL_NUM	numeric	100000	200285805	200200338	200114871	62166.635234436595
CAL_YEAR	numeric	2	2016	2015.5	2015	0.5000025000138331
EXPOSITION	numeric	275	365	327.58025	91	73.5704059115063
DRIVER_GENDER	string	2	Male	-	Female	-
DRIVER_OCC	string	5	Unemployed	-	Employed	-
DRIVER_AGE	numeric	58	75	41.12506	18	14.298972851315035
VEH_TYPE	string	6	F	-	A	-
VEH_CATEGORY	string	3	Small	-	Large	-
VEH_GROUP	numeric	20	20	10.69284	1	4.6873737262058155
VEH_VALUE	numeric	9395	49995	16454.6194	1000	10507.019806457773
POL_DURATION	numeric	16	15	5.47093	0	4.591135162065123
BONUS_MALUS	numeric	21	150	-6.93	-50	48.62794203713991
DAMAGE_GUAR	numeric	2	1	0.5122	0	0.4998536371144925
COUNTY	string	471	U9	-	L1	-
STATE	string	10	U	-	L	-
COUNTY_DENSITY	numeric	471	297.3851697	117.15688049021784	14.37714238	79.49898804465151
NUM_LIAB_DAM	numeric	8	7	0.14724	0	0.4366947325339446
NUM_LIAB_BOD	numeric	4	3	0.04678	0	0.2195270223028032
INC_LIAB_DAM	numeric	12257	12878.36991	106.15729505072366	0	444.9932471326946
INC_LIAB_BOD	numeric	4505	69068.02629	222.7432719670955	0	1859.5625777467174

FIGURE 15 : Récapitulatif des variables du jeu de données

Afin de continuer à réaliser notre étude descriptive du jeu de données nous avons également calculé les distributions empiriques des principales variables que nous pourrons utiliser par la suite pour réaliser notre tarification ainsi que pour procéder à l'agrégation des polices. Ces résultats sont synthétisés au sein de graphiques et conjointement avec d'autres statistiques descriptives pour chacunes des variables en annexe (An. I).

On notera ici que la variable `POL_NUM` n'est pas une variable explicative et qu'à ce titre elle ne sera utilisée que pour réaliser les jointures nécessaires à l'enrichissement de la base de données. Cette variable sera par la suite naturellement supprimée avant l'agrégation des polices. La variable `CAL_YEAR` est quant à elle directement supprimée.

Nous n'utiliserons pas non plus dans nos modèles la variable `COUNTY` qui présente un nombre très important de modalités risquant de ralentir l'ajustement des modèles et qui s'avère être redondante avec la variable `STATE`, tandis ce que l'information supplémentaire qu'elle fournie en terme de précision de localisation est finalement assez anecdotique voir contre-productif en menant à une forme de sur-apprentissage des modèles. Toutefois nous conserverons cette variable à la suite de l'agrégation ligne à ligne.
Finalement nous réaliserons nos modèles à partir de 12 variables explicatives (en omettant donc les variables `POL_NUM` et `CAL_YEAR` et `COUNTY` ainsi que les variables `NUM_***` et `INC_***` qui sont les variables à expliquer et la variable `EXPOSITION` qui nous servira uniquement à pondérer nos observations).

Désormais nous allons présenter le modèle de tarification ligne à ligne que nous avons réalisé et qui sera par la suite employé comme benchmark pour le modèle anonymisé ainsi que la méthodologie globale que nous avons employé pour réaliser et comparer ces modèles et notamment les modifications que nous avons apportées à la base de données.

4.1.3 Élaboration du modèle ligne à ligne

4.1.3.1 L'approche Coût/Fréquence

Le modèle de tarification ligne à ligne que nous proposons est basé sur une approche fréquence/sévérité. C'est à dire que nous souhaitons dans un premier temps modéliser séparément les fréquences de sinistres (à la fois matériels et corporels) et leurs coûts individuels moyens. En posant $Y_1 =$ Fréq. matérielles, $Y_2 =$ Fréq. corporelles, $Y_3 =$ Coût Moy. matériels et $Y_4 =$ Coût Moy. corporels, on souhaite pouvoir prédire le montant de la prime pure annuelle à payer pour un assuré comme étant l'espérance de sa charge sinistre, soit :

$$\text{Prime Pure}_{\text{pred}} = \mathbb{E}[Y_1.Y_3 + Y_2.Y_4]$$

Afin de tenir compte de l'exposition de chaque police au sein de notre portefeuille, il nous faut notamment "normaliser" les fréquences corporelles et matérielles par la durée de prise en charge de l'assuré par l'assureur, soit la variable `EXPOSITION` dans notre base de données.
Pour ce faire nous allons donc enrichir notre jeu de données de nouvelles variables, parmi lesquelles :

- $\texttt{WEIGHT} = \frac{\texttt{EXPOSITION}}{365}$, utilisée pour normaliser les variables en fonction de l'exposition de la police dans le portefeuille.
- $\texttt{MEAN\_INC\_BOD} = \begin{cases} \frac{\texttt{INC\_LIAB\_BOD}}{\texttt{NUM\_LIAB\_BOD}} \text{ Si } \texttt{NUM\_LIAB\_BOD} > 0 \\ 0 \text{ Sinon} \end{cases}$, coût moyen des sinistres corporels.

- $\texttt{MEAN\_INC\_DAM} = \begin{cases} \frac{\texttt{INC\_LIAB\_DAM}}{\texttt{NUM\_LIAB\_DAM}} \text{ Si } \texttt{NUM\_LIAB\_DAM} > 0 \\ \\ 0 \text{ Sinon} \end{cases}$, coût moyen des sinistres matériels.

On considérera désormais $Y_1 = \texttt{NUM\_LIAB\_DAM}$, $Y_2 = \texttt{NUM\_LIAB\_BOD}$, $Y_3 = \texttt{MEAN\_INC\_DAM}$ et $Y_4 = \texttt{MEAN\_INC\_BOD}$.

4.1.3.2 Modèle GLM

Nous nous sommes dans un premier temps consacrés à la réalisation d'un modèle ligne à ligne initial à partir de modèles linéaires généralisés (GLM) afin de réaliser les prédictions de nos variables de coûts et de fréquence de sinistre. Notre choix s'est porté sur ces modèles statistiques car ils représentent une évolution par rapport à la regression linéaire simple précédemment étudiée et demeure la procédure "standard" en tarification non-vie malgré l'émergence des modèles d'apprentissage statistiques (CART, Gradient boosting, etc.). Nous rappelons désormais les définitions et propriétés élémentaires de ces modèles.

Avant tout, il convient de rappeler la définition d'une famille de lois exponentielle. Un modèle statistique $(\Omega, \mathcal{F}, (\mathbb{P}_{\theta,\phi})_{\theta\in\Theta,\phi>0})$ est dit exponentiel si la mesure de probabilité $\mathbb{P}_{\theta,\phi}$ admet une densité $f_{\theta,\phi}$ selon une mesure dominante μ (le plus souvent la mesure de Lebesgue) tel que :

$$f_{\theta,\phi}(y) = c_\phi(y) \exp\left(\frac{y\theta - a(\theta)}{\phi}\right)$$

θ est appelé le paramètre canonique et ϕ la paramètre de dispersion (souvent considéré comme un paramètre de nuisance). On note également que $a(\theta)$ est $\mathcal{C}^2$ et convexe et que $c_\phi(y)$ ne dépend pas de θ.

Dans le cadre de modèles linéaires généralisés, en posant la variable à expliquer $Y \in \mathbb{R}$ et le vecteur de covariables $X \in \mathbb{R}^d$, réaliser une régression de Y à partir de X suppose deux hypothèses :

1. $Y|X = x \sim \mathbb{P}_{\theta(x),\phi}$ où $\mathbb{P}_{\theta(x),\phi}$ appartient à une famille exponentielle.
2. $g(\mathbb{E}[Y|X]) = \beta^T X$ avec g une fonction bijective.

On appelle g la fonction de lien associée à la distribution de $Y|X$, et alors que nous avons comme propriété des familles exponentielles que $\mathbb{E}[Y|X] = a'(\theta)$, on peut réecrire le point 2. tel que $g(a'(\theta(X))) = \beta^T X$. On en déduit alors une fonction de lien canonique pour chaque loi issue d'une famille exponentielle en posant $g(t) = a'^{-1}(t)$, on obtient alors $\theta(X) = \beta^T X$. Les principales distributions issues de familles exponentielles ainsi que leurs fonctions de liens canoniques sont répertoriés dans le tableau (9).

TABLE 9 : Principales distributions exponentielles et fonctions canoniques associées

Distribution	$g(\mu)$	Lien canonique
Gaussienne	μ	Indentité
Poisson	$\ln(\mu)$	Log
Gamma	μ^{-1}	Inverse
Binomial	$\log(\mu/(1-\mu))$	Logit

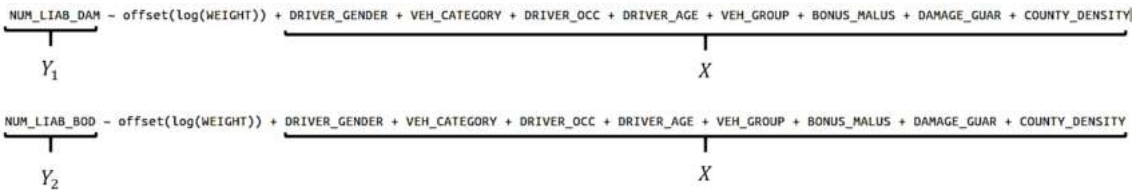
Pour modéliser les fréquences des sinistres corporels et matériels au sein de notre portefeuille nous avons choisis d'utiliser une distribution binomiale négative avec la fonction log comme lien. Alors que nous avions commencé par utiliser un GLM poisson qui s'avère également être une distribution tout à fait recommandée pour modéliser des variables de comptages ou de ratios, nous nous sommes rendu compte que les résultats des prédictions (MSE) étaient légèrement meilleurs en employant finalement cette autre distribution. A noter que le choix d'une distibution binomiale négative permet notamment de modéliser des effets de surdispersion des données (lorsque l'on a $\mathbb{E}[Y|X] < \text{Var}(Y|X)$).
Comme nous l'avions précédemment évoqué, nous avons naturellement besoin de normaliser ces fréquences (soit le nombre de sinistres comptabilisés) par rapport à la durée sur laquelle nous avons pu les mesurer (l'exposition au sein du portefeuille). C'est pourquoi nous avions alors crée la variable `WEIGHT`. L'idée étant donc de simplement diviser nos variables Y_1 et Y_2 à prédire (soit respectivement `NUM_LIAB_DAM` et `NUM_LIAB_BOD`) par la variable `WEIGHT`. Ainsi on procède à une sorte d'interpolation proportionelle pour les assurés sinistrés dont l'exposition au sein du portefeuille n'est pas égale à 365 (soit `WEIGHT` $\neq 1.0$) [54].
Seulement nous ne pouvons pas directement prédire à l'aide d'un GLM binomial négatif les variables transformées Y_1/`WEIGHT` et Y_2/`WEIGHT`. En effet le support de cette loi requiert des données entières $\in \mathbb{N}$. Or ces nouvelles variables évoluent dans $\mathbb{R}$. L'idée consiste alors simplement à contourner cette limitation en introduisant un `Offset`.

On rappelle que nous utilisons une distribution binomiale négative avec la fonction de lien $g = \ln()$. En posant X la totalité des variables explicatives du modèle (nous les expliciterons juste après), on peut donc écrire :

$$g\left(\frac{\mathbb{E}[Y|X]}{\texttt{WEIGHT}}\right) = \beta^T X \;\Leftrightarrow\; \ln(\mathbb{E}[Y|X]) = \underbrace{\ln(\texttt{WEIGHT})}_{\texttt{Offset}} + \beta^T X$$

Soit la modélisation des fréquences de sinistres corporels et matériels suivante :
A titre de remarque, les variables quantitatives continues `DRIVER_AGE` et `COUNTY_DENSITY` notamment, n'ont pas été discrétisées en buckets comme il est pourtant d'usage en tarification, afin de faciliter l'élaboration du modèle ligne à ligne dont on rappelle

54. i.e un individu qui n'a été assuré au sein de la compagnie que la moitié de l'année (`EXPOSITION`=183, soit `WEIGHT` $\sim$`0.5`) et qui compte deux sinistres au cours de cette période verra sa sinistralité normalisée sur l'année s'élever à 4.



que sa précision et la qualité de ses prédictions sont moins le sujet d'étude que la comparaison de celles-ci avec le modèle anonymisé par agrégation de polices.

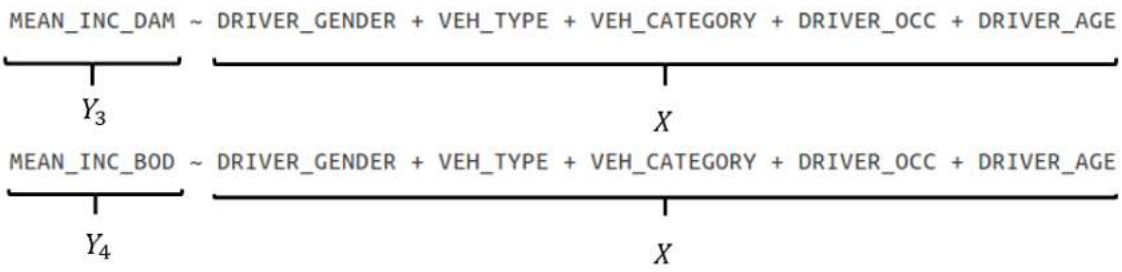
Pour la modélisation des coûts moyens des sinistres nous nous baserons sur les variables `MEAN_INC_BOD` et `MEAN_INC_DAM` précédemment crées, que nous écrêterons afin "de redistribuer" les montants extrêmes. Pour ce faire nous avons donc affecté à tous les sinistres corporels supérieurs à 6861,94€ (valeur du quantile 99%, soit les polices dont les sinistres corporels sont parmi les 1% les plus élevés) ce montant maximal. Cette méthode d'écrêtage est employée ici pour sa simplicité et n'est pas forcément d'usage dans le milieu actuariel. Par la suite on calcul un facteur de correction tel que :

$$\text{Correction} = \frac{\sum_{i=1}^{100000} \texttt{INC\_IAB\_BOD}_i - \sum_{i=1}^{100000} \texttt{MEAN\_INC\_BOD}_i.\texttt{NUM\_LIAB\_BOD}_i}{\sum_{i=1}^{100000} \texttt{WEIGHT}_i}$$

Que l'on appliquera en amont de la prédiction tel que :

$$(\hat{Y_4}.\hat{Y_2})_{\text{finale}} = \hat{Y_4}.\hat{Y_2} + \text{Correction}$$

On fait de même pour les coûts moyens matériels, avec un seuil de 2278,65€ (quantile à 99,5%).
Afin de prédire les variables Y_3 et Y_4, on utilise un GLM avec une distribution inverse-gaussienne et une fonction de lien *inverse*. En ne considérant que les variables précédentes filtrées pour ne conserver que les polices d'assurances sinistrées (dont les montants sont strictement positifs) on obtient la modélisation des coûts moyens des sinistres corporels et matériels suivant :



Finalement afin d'obtenir la charge sinistre totale prédite pour chaque police au sein de notre modèle il nous suffit alors de calculer :

$$\text{Charge sinistre}_{\text{Pred}} = (\hat{Y_1}.\hat{Y_3})_{\text{finale}} + (\hat{Y_4}.\hat{Y_2})_{\text{finale}}$$

Pour correctement paramétrer et évaluer notre modèle nous avons scindé notre jeu de données en deux échantillons, l'un servant à ajuster nos procédure (60% des données pour le jeu d'apprentissage) et l'autre afin de mesurer sa qualité (40% des données pour le jeu de validation).
Nous conservons les échantillons ainsi obtenus afin de pouvoir les réutiliser dans la suite de la procédure d'anonymisation, afin de s'assurer que lorsque l'on comparera les résultats issus des modèles benchmark et anonymisé, nous comparerons des procédures paramétrées et testées sur des données équivalentes.
Des résultats illustrant la qualité du modèle sont répertoriés à titre indicatif dans le tableau (10).

TABLE 10 : Résultats de la méthode GLM

$\sum$ tot. prédictions	12662065.20€
$\sum$ tot. observées	12580726.73€
Ecart (predictions - observées)	-81338.48€
Ecart relatif	0.66%
RMSD [55]	1863,19€

Maintenant que nous avons un modèle de référence calibré à partir des polices d'assurance individuelles nous allons pouvoir nous consacrer à l'anonymisation de ces données grâce à différentes méthodes de clustering afin de constituer une base agrégée d'individus réunis au sein de groupes homogènes au regard des variables utilisés lors des différents GLM exécutés lors de la procédure de tarification.

Dans la prochaine partie nous nous concentrerons donc sur le choix de ces algorithmes de clustering ainsi que sur les méthodes d'agrégations à appliquer aux polices régroupées au sein d'un même cluster pour obtenir un individu abstrait et représentatif du groupe, afin finalement d'avoir un modèle de tarification anonymisé le plus efficace et précis possible au regard de notre modèle benchmark présenté ici.

55. Root-mean-square deviation, i.e : RMSD $= \sqrt{\frac{\sum_{i=1}^{N}(\hat{y}_i - y_i)^2}{N}}$

4.2 Anonymisation par agrégation des polices d'assurances

Dans cette deuxième partie au sujet de l'élaboration d'une procédure de tarification en responsabilité civile automobile matérielle et corporelle anonymisée, nous allons commencer par présenter différents algorithmes d'apprentissage non-supervisés qui nous permettront de réaliser simplement et efficacement l'agrégation des contrats d'assurances de notre portefeuille en de petits groupes homogènes comme le schématise la figure (16) ci-dessous. Nous évoquerons alors les contraintes inhérentes à chaque méthode ainsi que nos contraintes opérationnelles compte tenu du contexte d'application.

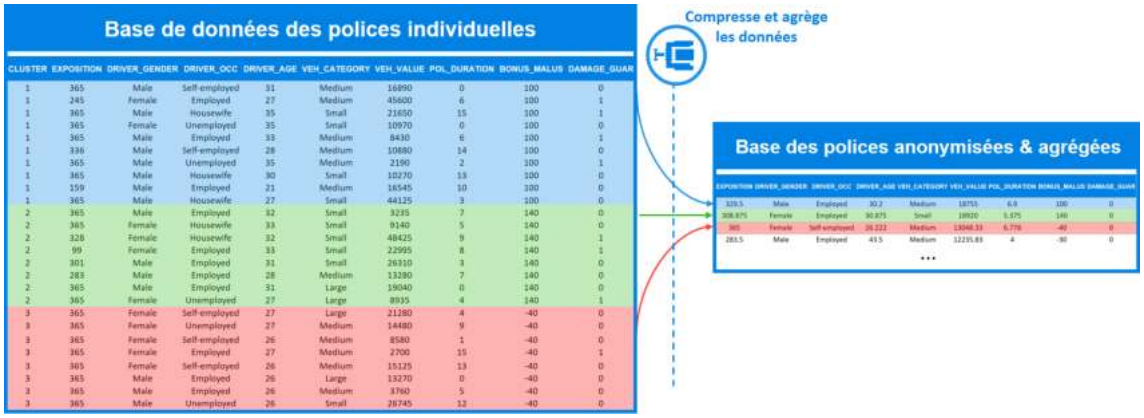


FIGURE 16 : Schéma de l'agrégation de polices d'assurances dans un contexte d'anonymisation

Puis nous présenterons et analyserons les résultats des clusters obtenus sur nos données au travers des algorithmes précédemment évoqués et nous définirons également la méthode que nous emploierons pour parvenir à élaborer un individu caractéristique pour chacun des groupes formés tout en garantissant la confidentialité de chaque police au sein de son cluster.

4.2.1 Algorithmes non-supervisés de regroupement de données

Afin de constituer les regroupements des polices de notre base nous avons eu recourt à des algorithmes d'apprentissage non-supervisés de clustering.
On distingue en effet deux familles de méthodes d'apprentissage machine :

- **Les procédures supervisées** qui requièrent une labellisation des données d'apprentissage et d'évaluer éventuellement la paramétrisation retenue du modèle sur des données de validation avant de procéder à l'évaluation des données de test à classifier ou à régresser.

  Il existe de très nombreux algorithmes d'apprentissage supervisés, mais les méthodes les plus répandues aujourd'hui demeurent les réseaux neuronaux (et leurs dérivés : CNN, Perceptron, etc.) ainsi que les méthodes basées sur la construction d'arbres de décisions (CART, RandomForest, etc.).

L'utilisation précédemment des modèles linéaires généralisés lors de l'élaboration de notre procédure de tarification de référence entre naturellement dans ce cadre puisque nous avons entraîné plusieurs modèles à partir de variables explicatives X en fournissant en même temps le résultat attendus (montants moyens puis fréquence des sinistres) Y afin de fournir une estimation $\hat{Y}$ de ces valeurs pour de nouvelles données dont nous n'avons renseigné que les variables X.

- **Les procédures non-supervisées** regroupe un ensemble de méthodes qui permettent de classer des observations au sein de groupes les plus homogènes possibles en leurs seins mais les plus différenciables entre eux, à partir seulement des variables explicatives X. A la différence des algorithmes supervisés qui nécessitent de connaître les classes à priori, ces méthodes reposent sur la capacité à déterminer des motifs et des liens logiques non-présupposés entres les individus qui constituent la base de données.

  Parmi les nombreuses applications de ce type de méthodes on retrouve en premier lieu le partitionnement ou le regroupement hiérarchique des données (la plupart des méthodes que nous avons employés dans cette étude sont issues de ce domaine, elles seront détaillées par la suite), mais également l'estimation de la distribution des données (qui peut alors donner lieu à un regroupement à posteriori) et enfin la réduction de dimension d'un jeu de données, en diminuant le nombre de variables lorsque par exemple nous avons plus de variables/modalités que d'individus dans la base, nous trouvant alors dans le cadre du fléau de la dimension [56].

Afin de regrouper nos polices d'assurances au sein de groupes homogènes et dont la présence d'un membre dans une classe demeure inconnue à l'issue de l'agrégation, nous nous sommes penchés sur quatres méthodes de clustering, à savoir *le clustering par propagation d'affinité*, *l'algorithme du K-means*, *le clustering ascendant hiérarchique* et enfin un *clustering par estimation de densité*. Nous allons désormais définir et préciser le principe et la mise en oeuvre de ces méthodes dans le cadre de notre étude.
Nous traiterons ici séparément les algorithmes nécessitant de paramétrer directement le nombre de classes désirées à l'issue de la procédure (pouvant naturellement induire un biais important par rapport à un partionnement optimal si le nombre choisi en est très éloigné) et ceux capables de déterminer le nombre optimal de clusters au regard du choix des critères de convergence et des métriques employées.

56. Le fléau de la dimension est un concept inventé par R. Bellman en 1961 afin de définir certains phénomènes spécifiques à l'analyse de base de données de grandes dimensions, où par exemple le nombre de variables et de modalités dépassent largement le nombre d'individus. Dans ces cas, l'espace dans lequel évoluent les individus devient tellement volumineux au regard du nombre très important de dimensions différentes qu'il comporte que les observations paraissent finalement très éparses au sein de cet environnement. Dès lors les méthodes basées sur le calcul de métriques (soit un grand nombre d'algorithmes d'apprentissages supervisés comme les k-plus proches voisins et non-supervisés comme le K-means par exemple), dans ce cadre, deviennent caduques.

4.2.1.1 Algorithmes de clustering dont le nombre de classes est un paramètre du modèle

Avant d'évoquer plus en détails le principe des algorithmes du *K-means* et du *clustering ascendant hiérarchique* (CAH) que nous avons employés dans cette étude, il convient de rappeler précisément nos besoins et nos attentes de ces méthodes.

Contrairement à l'usage "classique" de ces procédures, ici nous ne recherchons pas l'optimalité du partitionnement des données[57], mais nos critères de sélection du modèle sont davantage portés sur les points suivants, schématisés par la figure (17) :

- La minimisation de l'hétérogénéité au sein des classes (symbolisée par la variance intra-classe) au risque d'avoir une variance inter-classe faible (signifiant pourtant dans le cadre "classique" que certains groupes assez similaires devraient fusionner afin d'être plus optimal).
- Avoir suffisamment de clusters à l'issue de la procédure afin de pouvoir réajuster à partir de ces données le modèle de tarification de référence avec suffisamment de données (à l'issue de la procédure de clustering, chaque classe sera caractérisée par une unique pseudo-observation , comme l'agrégation des polices du groupe).
- Conserver des clusters pertinents et cohérents avec un nombre d'individus suffisant afin de garantir l'anonymat des données (concrètement avoir des groupes d'au moins deux polices à chaque fois, sinon l'agrégation en une pseudo-observation est caduque puisque le résultat obtenu correspond directement à une police existante).

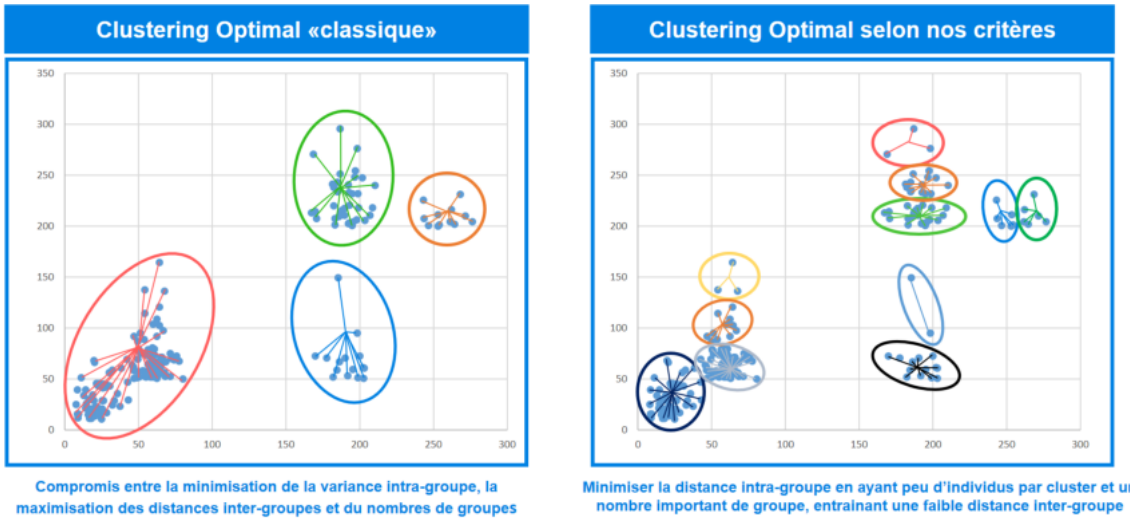


FIGURE 17 : Schéma des différences de critères de sélection des modèles de clustering

57. Au sens où nous n'espérons pas nécessairement obtenir un partitionnement où le nombre de groupes est optimal et une hétérogénéité maximale entre les classes.

Nous avons donc commencé par tester la méthode du *K-means* sur les polices de notre portefeuille, un algorithme classique de partitionnement qui permet d'obtenir des résultats grâce à une procédure simple à assimiler.

- **K-means**

Le principe de cette méthode de partitionnement réside dans la minimisation de la somme des carrés des distances d'un point à la moyenne des points de sa classe.
Soit un ensemble de n individus $(x_1, x_2, \ldots, x_n)$ que l'on souhaite partitionner en K classes de C, tel que $C = \{C_1, C_2, \ldots, C_K\}$ où $K \leqslant n$, on cherche à minimiser la distance des points d'une classe k au barycentre du cluster μ_k (soit la variance intra-cluster) :

$$\underset{C}{\operatorname{argmin}} \sum_{k=1}^{K} \sum_{i=1}^{n_k} (x_i - \mu_k)^2 \Leftrightarrow \underset{C}{\operatorname{argmin}} \sum_{k=1}^{K} \frac{1}{n_k} \operatorname{Var}(C_k)$$

Où $\mu_k := \overline{X_k} = \frac{1}{n_k} \sum_{i=1}^{n_k} x_i$ avec n_k le nombre d'observations dans le cluster k.

Par ailleurs grâce au théorème de König-Huygens on peut établir la relation suivante :

$$\sum \text{DIST. CARRES TOT.} = \sum \text{DIST. CARRES INTRA-CLUSTERS} + \sum \text{DIST. CARRES INTER-CLUSTERS}$$

Cette égalité, que nous allons démontrer, nous permet alors d'expliquer plus précisément pourquoi il est nécessaire de fixer à priori le nombre de classes K à obtenir.

Preuve :
Nous commençons par définir l'indicatrice suivante :

$$\mathbb{1}_{x_i \in C_k} = \begin{cases} 1 \text{ Si } x_i \in C_k \\ 0 \text{ Sinon} \end{cases}$$

En posant μ, la moyenne de l'ensemble des observations (barycentre total) et en considérant donc que l'on a pour k fixé $\sum_{i=1}^{n} \mathbb{1}_{x_i \in C_k} = n_k$, on peut alors décomposer la somme des distances au carrés totale tel que :

$$
\begin{aligned}
\underbrace{\sum_{i=1}^{n}(x_i-\mu)^2}_{\text{Dist. totale}} &= \sum_{i=1}^{n}\sum_{k=1}^{K}\mathbb{1}_{x_i\in C_k}(x_i-\mu)^2 \\
&= \sum_{i=1}^{n}\sum_{k=1}^{K}\mathbb{1}_{x_i\in C_k}(x_i-\mu_k+\mu_k-\mu)^2 \\
&= \sum_{i=1}^{n}\sum_{k=1}^{K}\mathbb{1}_{x_i\in C_k}\left[(x_i-\mu_k)^2+2(x_i-\mu_k)(\mu_k-\mu)+(\mu_k-\mu)^2\right] \\
&= \sum_{i=1}^{n}\sum_{k=1}^{K}\mathbb{1}_{x_i\in C_k}(x_i-\mu_k)^2+2\sum_{i=1}^{n}\sum_{k=1}^{K}\mathbb{1}_{x_i\in C_k}(x_i-\mu_k)(\mu_k-\mu) \\
&\qquad +\sum_{i=1}^{n}\sum_{k=1}^{K}\mathbb{1}_{x_i\in C_k}(\mu_k-\mu)^2 \\
&= \sum_{i=1}^{n}\sum_{k=1}^{K}\mathbb{1}_{x_i\in C_k}(x_i-\mu_k)^2+2\sum_{k=1}^{K}(\mu_k-\mu)\left[\sum_{i=1}^{n}\mathbb{1}_{x_i\in C_k}x_i-\sum_{i=1}^{n}\mathbb{1}_{x_i\in C_k}\mu_k)\right] \\
&\qquad +\sum_{k=1}^{K}n_k(\mu_k-\mu)^2 \\
&= \sum_{i=1}^{n}\sum_{k=1}^{K}\mathbb{1}_{x_i\in C_k}(x_i-\mu_k)^2+2\sum_{k=1}^{K}(\mu_k-\mu)(n_k\mu_k-n_k\mu_k) \\
&\qquad +\sum_{k=1}^{K}n_k(\mu_k-\mu)^2 \\
&= \underbrace{\sum_{i=1}^{n}\sum_{k=1}^{K}\mathbb{1}_{x_i\in C_k}(x_i-\mu_k)^2}_{\text{Dist. intra-clusters}}+\underbrace{\sum_{k=1}^{K}n_k(\mu_k-\mu)^2}_{\text{Dist. inter-clusters}}
\end{aligned}
$$

□

En remarquant que la somme des distances au carrés (variance totale) est constante quelque soit la valeur de K, le principe de l'algorithme du K-means qui consiste à chercher à minimiser la somme des variances intra-clusters est équivalent à vouloir maximiser la variance inter-clusters (la distance entre les classes) [58].
Or si nous ne spécifions pas la valeur de K au préalable et que nous laissions l'algorithme déterminer le nombre de classes optimale nous obtiendrions un résultat trivial où simplement $K = n$, c'est à dire que chaque observation appartiendrait à son propre cluster, constitué uniquement de cette dernière. Dans ce cas la variance intra-clusters est nulle (et donc minimale) et la variance inter-clusters maximisée et égale à la variance totale.
Cependant chacun est conscient qu'un tel résultat ne peut être assimilé à un quelconque partitionnement puisqu'en somme aucun individus n'a été regroupé au sein

58. En effet on a : $\underset{intra}{\operatorname{argmin}}\, intra = Cst - inter \Leftrightarrow \underset{inter}{\operatorname{argmax}}\, inter = Cst - intra$

d'une classe. C'est pour éviter ce cas de figure que l'algorithme du K-means requiert la fixation du nombre de classes K en amont de la procédure de partitionnement.

Pour résoudre ce problème complexe l'utilisation d'une heuristique est nécessaire, l'algorithme de Lloyd est alors couramment employé bien qu'il ne garantisse pas l'optimalité et qu'il s'agisse d'un algorithme glouton (en anglais *greedy*) dont l'objectif consiste à converger rapidement en calculant successivement des extremums locaux jusqu'à espérer s'approcher de l'extremum global. Il n'est cependant pas impossible d'aboutir et de converger seulement vers un optimum local.

Algorithme de Lloyd :

1. Sélectionner aléatoirement K points parmi les n points totaux pour devenir les "barycentres" initiaux des K clusters $(\mu_1^{(0)}, \mu_2^{(0)}, \ldots, \mu_K^{(0)})$
2. Tant qu'il n'y a pas convergence, à l'itération t :
   - 2.1 Classer chaque individu au sein du cluster dont le barycentre est le plus proche :
   $$C_k^{(t)} = \left\{x_i, \|x_i - \mu_k^{(t)}\| \leqslant \|x_i - \mu_{k^*}^{(t)}\| \forall k^* = 1, \ldots, K\right\}$$
   - 2.2 Recalculer le nouveau barycentre de chaque cluster :
   $$\mu_k^{(t+1)} = \frac{1}{n_k^{(t)}} \sum_{x_i \in C_k^{(t)}} x_i$$

On notera à l'étape 2 du précédent algorithme que le critère de convergence peut être défini soit comme un nombre d'itérations maximal, soit comme une valeur de tolérance sur l'évolution des barycentres calculés tel que $\forall k$, $\|\mu_k^{(t)} - \mu_k^{(t+1)}\| \leqslant \epsilon$, avec ϵ petit.

Par ailleurs à l'étape 2.1, la méthode employée afin d'assigner une observation à une classe selon sa proximité avec le barycentre de cette dernière conduit à la formation d'une tesselation de Voronoï [59], c'est à dire la création de clusters strictement convexes. Il est donc impossible d'obtenir à l'issue d'un K-means, des clusters ayant une forme étoilée ou encore d'anneaux par exemple. Le schéma du déroulement de l'algorithme de Lloyd présenté ci-dessous (19) permet notamment de visualiser ce partitionnement singulier.

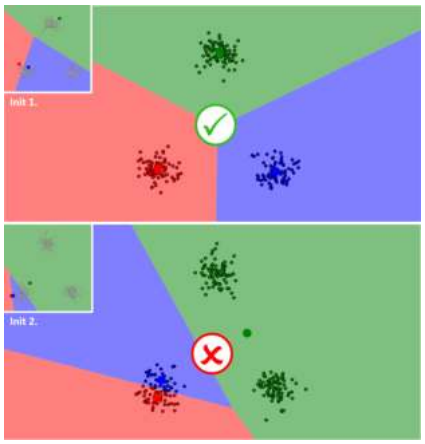


FIGURE 18

L'algorithme du K-means est par ailleurs très dépendant de l'initialisation des barycentres à l'étape 1, c'est pourquoi il est conseillé de recourir à plusieurs initialisations différentes afin

59. Cette structuration de l'espace se retrouve dans bon nombre de méthode d'apprentissage machine, en particulier avec la méthode supervisée des k-plus proches voisins.

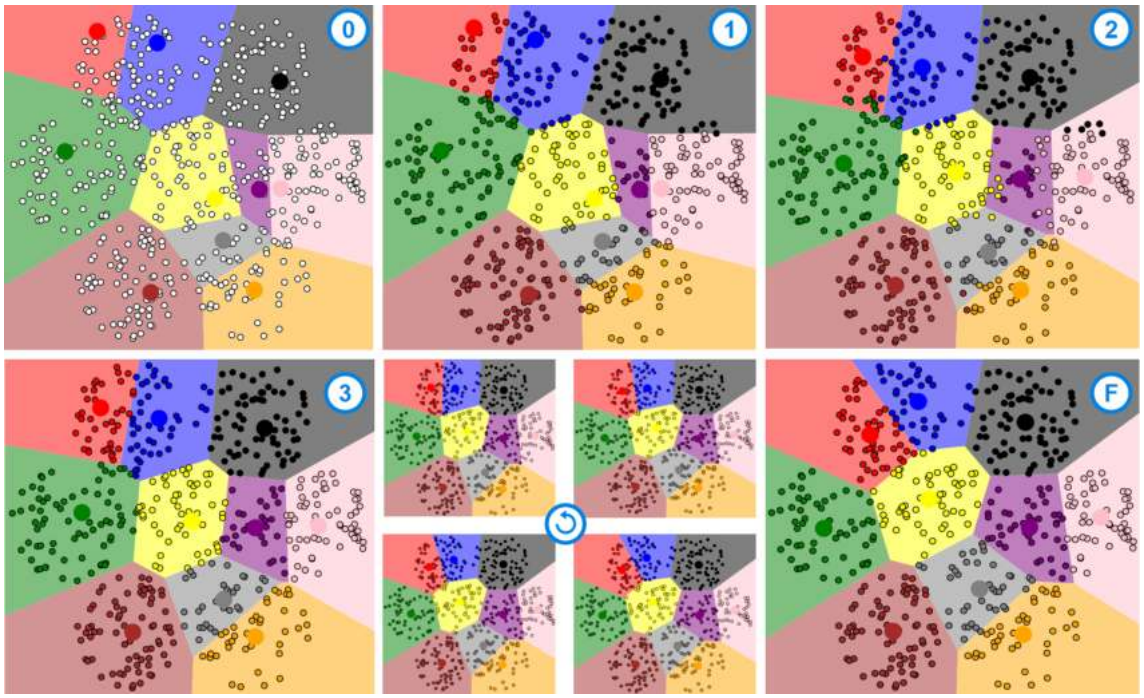


FIGURE 19 : Schéma du déroulement de l'algorithme de Lloyd et d'une partition de Voronoï

d'obtenir un résultat plus homogène. La figure (18) permet de se rendre compte de l'impact d'une "mauvaise" initialisation des barycentres initiaux sur le clustering de données pourtant très différenciables.

Afin de partitionner notre jeu de données d'apprentissage (pour rappel seul le jeu d'apprentissage servant à paramétrer le modèle évoqué en (4.1.3.2) doit être anonymisé par agrégation de polices, les prédictions du modèle obtenus sont, elles, toujours réalisées en ligne à ligne), représentant 60% du jeu de données initial (soit 60 000 polices) nous avons utilisés uniquement les covariables les plus discriminantes et éviter les variables comptant de trop nombreuses modalités (`COUNTY, STATE`, etc.) pour éviter d'introduire du bruit inutilement. C'est pourquoi notre clustering est donc basé seulement sur les variables suivantes :

- `DRIVER_AGE`
- `BONUS_MALUS`
- `COUNTY_DENSITY`
- `DRIVER_GENDER`
- `DRIVER_OCC`
- `VEH_CATEGORY`
- `DAMAGE_GUAR`

Sous R nous avons simplement utilisé la méthode `kmeans` intégrée au langage et ne nécessitant pas d'autres packages supplémentaires afin de partitionner nos données. Nous avons cependant pris soin de paramétrer la fonction afin de prendre en compte nos remarques précédentes, ainsi le paramètre `nstart` permettant de définir le nombre d'initialisations différentes des barycentres initiaux a été définie à 20,

tandis que le nombre d'itérations maximum `iter.max` fut rehaussé à 100 afin de s'assurer que l'algorithme soit en mesure de converger.
Conformément aux critères de sélection du modèle de partitionnement que nous avons établis en début de cette section (4.2.1.1), nous souhaitons obtenir un nombre de clusters suffisamment important car cette valeur correspondra également au nombre de pseudo-observations caractéristiques de chaque clusters que nous utiliserons afin de paramétrer notre modèle de tarification. Il nous faut donc obtenir un nombre important de clusters tout en s'assurant que chaque cluster comporte au moins deux observations, sans quoi la pseudo-observation de la classe serait un individu réel et donc non-anonymisé. L'avantage de l'algorithme du K-means dans notre cas est qu'il nous permet de définir nous même le nombre de classes finales escomptées à l'issue du partitionnement, ainsi nous avons défini 4 partitionnements différents en variant le nombre de clusters à former selon le tableau suivant (11).

TABLE 11 : Les différents partitionnements réalisés à l'aide du K-means

Partitionnement	Nb. clusters base tot.	Nb. clusters base app. (60%)
KM_6000	10 000	6 000
KM_4500	7 500	4 500
KM_3000	5 000	3 000
KM_1200	2 000	1 200

Au regard de nos critères de sélection du modèle de partitionnement, nous nous intéressons principalement à la somme des carrés intra-clusters puisqu'il s'agit du principal critère afin de déterminer l'homogénéité des clusters obtenus.
Cependant après avoir démontré que l'algorithme du K-means consistait à déterminer des partitions dans la donnée afin de minimiser cette somme et qu'une solution triviale consistait alors à simplement choisir le nombre de partitions K, le plus grand possible soit égale au nombre d'observations. On en déduit logiquement, et les diagrammes suivants (20) nous le confirme, que plus nous choisirons un nombre de clusters élevés plus leurs homogénéités (pour rappel, uniquement vis à vis des covariables utilisées pour le clustering) sera assurées. Parmi nos partitionnements on constate donc facilement que celui formant 6000 classes au sein de la base d'apprentissage présente une homogénéité plus importante. Nous avons néanmoins décidé de conserver les autres partitionnements à titre de comparaison.
Enfin nous avons décidé de ne réaliser des partitionnements qu'avec 6000 clusters au maximum sur notre base d'apprentissage alors que nous pourrions imaginer qu'en augmentant encore ce nombre jusqu'à sa limite théorique [60] nous pourrions obtenir de meilleurs résultats, cependant de telles valeurs conduiraient à la formation de trop nombreux clusters ne comptant qu'un seul individu (au sein de l'algorithme du K-means, le nombre d'individus par classe n'est pas contrôlé) ce qui conduirait à leurs suppressions afin de garantir le caractère anonymisé de notre procédure.

60. Notre base d'apprentissage compte 60 000 observations, afin de ne pas conduire de manière certaine à des clusters ne contenant qu'un seul élément il serait théoriquement possible de choisir un partitionnement à 60000/2 soit 30 000 classes.

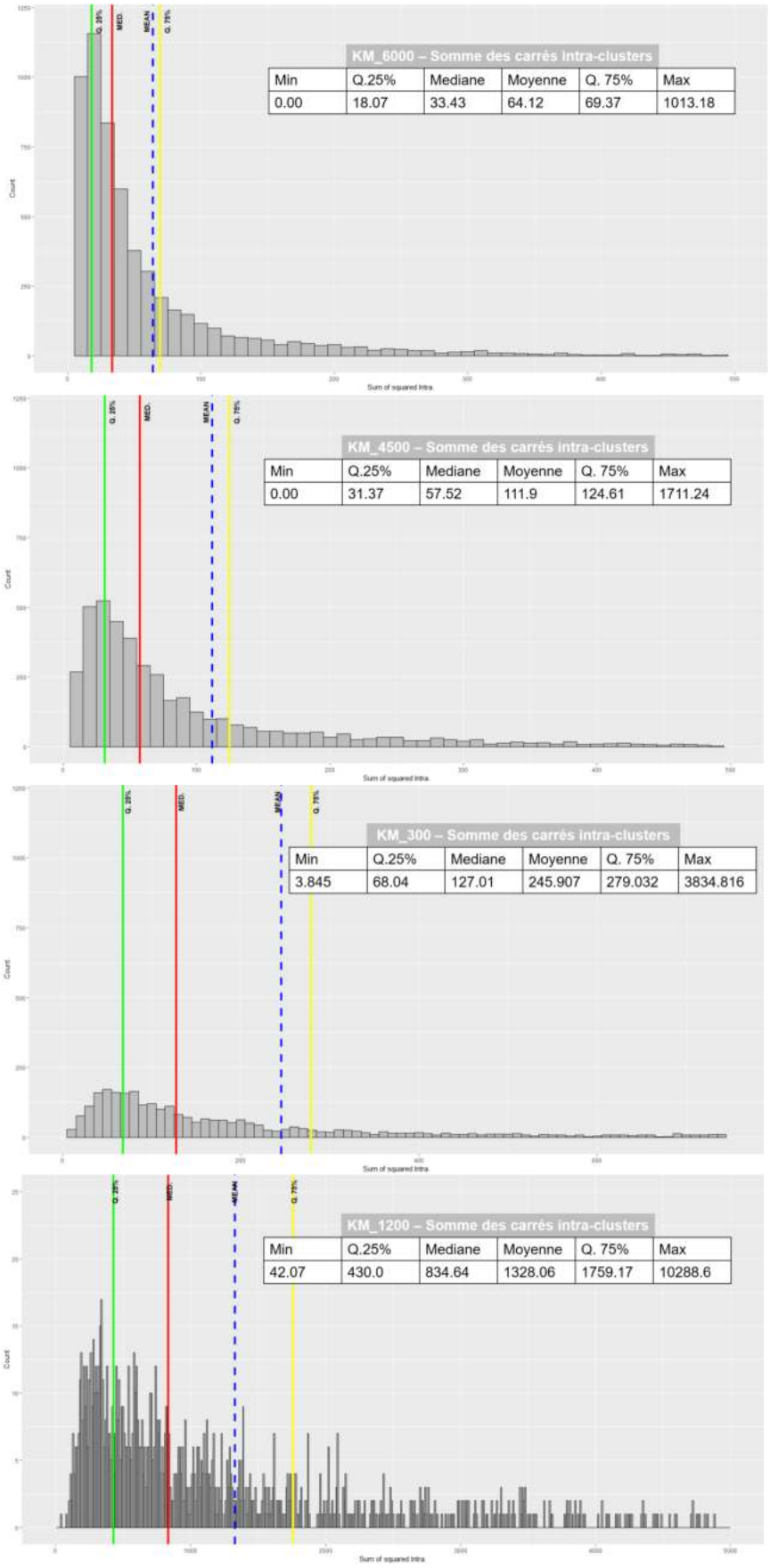


FIGURE 20 : Répartition de la somme des carrés intra-clusters pour les différents partitionnements par l'algorithme du K-means, où $k \in \{6000, 4500, 3000, 1200\}$

- **Clustering ascendant hiérarchique**

Alors que l'algorithme du K-means permet de constituer un partitionnement dont le nombre de classes est défini en amont mais en étant toutefois très dépendant de l'initialisation des barycentres des clusters initiaux. L'approche du clustering hiérarchique est fondée sur une mesure de dissimilarité entre les observations, notée par la suite $d_{x_i x_j}$, qui peut par exemple dans le cas d'un espace euclidien s'avérer être la distance. Les différents individus sont alors regroupés selon cette valeur.
Il existe par ailleurs deux types d'algorithmes de clustering hiérarchique :

- La *méthode ascendante* (ou agglomérative) où au départ toutes les observations sont affiliées à leurs propres clusters (un individu par groupe) puis ces classes sont regroupées en fusionnant les clusters ayant la dissimilarité la plus faible jusqu'à obtenir un unique groupe. Naturellement l'algorithme retourne l'historique et l'état du partitionnement à chaque étape afin de pouvoir obtenir un clustering efficace (on choisit alors de garder par exemple l'état du partitionnement à l'étape i où l'on a k clusters). Nous nous intéresserons en particulier à cette méthode par la suite.

- La *méthode descendante* (ou par divisions) où au départ toutes les observations sont affiliées à un même cluster, puis ces classes sont divisées en deux groupes ayant la dissimilarité la plus importante jusqu'à obtenir un cluster par observation. Naturellement l'algorithme retourne l'historique et l'état du partitionnement à chaque étape afin de pouvoir obtenir un clustering efficace (on choisit alors de garder par exemple l'état du partitionnement à l'étape i où l'on a k clusters).

Une fois le choix de la méthode employée, il reste à définir quelle mesure de dissimilarité entre les groupes employer [61], il existe de nombreuses méthodes de liens différentes conduisant à des procédures de partitionnement singuliers.

En reprenant les notations précédentes de la partie sur l'algorithme du K-means, on peut par exemple définir la dissimilarité entre deux groupes comme la valeur de dissimilarité la plus faible entre deux observations de chacun des deux groupes :

$$d_{C_k C_l} = \min_{x_i \in C_k, x_j \in C_l} d_{x_i x_j}$$

En d'autres termes il s'agit de mesurer l'écart entre les deux points les plus proches de chacun des deux clusters. On appelle cette méthode **le lien simple**. Utiliser cette méthode durant un partitionnement ascendant hiérarchique revient à regrouper les deux clusters ayant les deux points les plus proches entre eux. Cette stratégie simple peut conduire à un phénomène de *chaînage* où les clusters sont alors élaborés en ajoutant un à un (ou tout du moins par très petits groupes) des individus dans un cluster principal. Cela mène alors inévitablement à obtenir des partitionnements de

61. On les appelle les méthodes de liens.

tailles très différentes dont quelques clusters sont le plus souvent trop éparses [62] et les autres très petits.
Dans notre cas nous n'utiliserons donc pas cette méthode de lien car nous souhaitons pouvoir avoir le maximum de clusters comptant au moins deux observations, or cette méthode conservera le plus longtemps possible au fil des étapes du partitionnement des clusters de tailles très réduites (potentiellement d'une observation) ce qui nous conduira alors à obtenir un nombre très réduit de clusters dont le nombre d'observations est supérieur à un.

Une autre méthode de lien classique consiste cette fois à mesurer la dissimilarité entre les deux points les plus éloignés de deux clusters :

$$d_{C_k C_l} = \max_{x_i \in C_k, x_j \in C_l} d_{x_i x_j}$$

On appelle cette méthode **le lien complet**. Durant le partitionnement cette méthode de lien induit le regroupement de deux clusters si la dissimilarité de leurs deux points les plus éloignés est la plus petite.
Malheureusement ce lien peut engendrer des problèmes de compacité optimale des clusters obtenus. En effet puisque que le regroupement de deux clusters est basé sur la pire dissimilarité possible entre ces deux groupes, il assez naturel de penser que leur fusion formera un cluster homogène en moyenne. Cependant cette homogénéité ne sera pas optimale car certains individus pourront finalement appartenir à une classe alors qu'il sont plus proches de points issus d'un autre cluster. Alors que notre critère de sélection du modèle de partitionnement est principalement basé sur l'homogénéité des clusters obtenus, nous ne retenons pas cette méthode de lien dans notre cadre.

Il existe alors d'autres méthodes de liens qui permettent de se défaire des deux phénomènes évoqués et qui forment une sorte de compromis entre le lien simple et le lien complet.
On retrouve notamment le **lien moyen** pour lequel la dissimilarité de deux clusters est représenté par la moyenne de toutes les dissimilarités par paires des individus des deux clusters :

$$d_{C_k C_l} = \frac{1}{|n_k|} \frac{1}{|n_l|} \sum_{x_i \in C_k} \sum_{x_j \in C_l} d_{x_i x_j}$$

Mais également le **lien centroïdal** qui calcule la dissimilarité entre deux groupes à partir de la dissimilarité de leurs deux barycentres respectifs :

$$d_{C_k C_l} = d_{\mu_i \mu_j}$$

Nous avons alors essayé d'utiliser ces deux liens pour le partitionnement de notre jeu d'apprentissage (toujours selon les mêmes variables que celles employées pour

62. En effet il suffit que deux points de deux clusters distincts soit les plus proches pour que les classes fusionnent indépendamment du niveau de dissimilarité des autres individus des deux groupes.

l'algorithme du K-means), malheureusement nous ne sommes parvenus qu'à obtenir seulement 17 clusters comptant chacun strictement plus d'une observation, ce qui ne s'avère clairement pas suffisant par la suite pour paramétrer notre modèle de tarification [63].

Les méthodes de liens présentées jusqu'ici visent principalement à garantir que les clusters les plus différents soient agrégés le plus tardivement possible dans le processus de partitionnement, soit à chaque étape de faire en sorte que les clusters formés soient, relativement à leur barycentres respectifs [64], biens séparés.
Néanmoins, comme nous l'avons précédemment vu nous nous intéressons principalement à l'homogénéité des clusters que nous obtenons, c'est pourquoi nous allons davantage nous intéresser à la méthode de **lien de Ward**.
En effet cette méthode est basée sur la minimisation de l'augmentation de la variance intra-cluster lors du regroupement de deux clusters :

$$\text{intra-variance} = \sum_{i=1}^{n} \sum_{k=1}^{K} \mathbb{1}_{x_i \in C_k} (x_i - \mu_k)^2$$

Et comme nous l'avons vu avec l'algorithme du K-means, minimiser la variance intra-cluster ou maximiser la variance inter-cluster est équivalent. Il est par ailleurs possible d'implémenter le calcul de la dissimilarité de deux clusters au travers cette méthode de lien grâce à la formule de Lance-Williams :

$$d_{C_k C_l} = \frac{n_k \times n_l}{n_k + n_l} d_{\mu_i \mu_j}$$

Nous avons donc appliqué ce lien à partir de la fonction `hclust` sous R du package `fastcluster` [65] et avons réussi à obtenir 2320 clusters comptant plus d'une observation. L'homogénéité du partitionnement HCLUST_2320 est réprésentée ci-dessous (21).

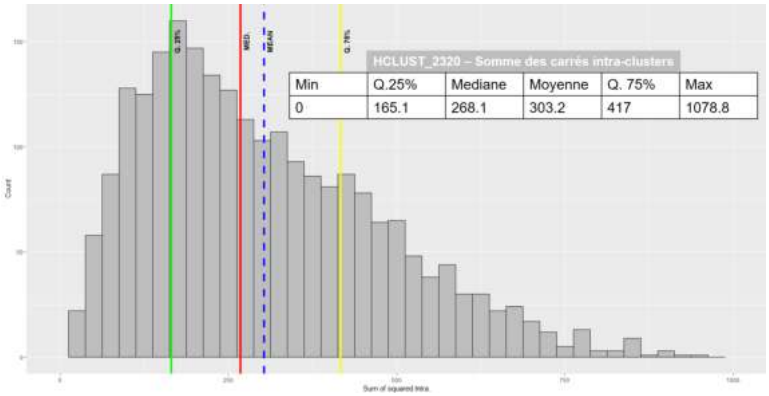


FIGURE 21 : Répartition de la somme des carrés intra-clusters (HCLUST_2320)

63. Obtenir seulement 17 clusters signifierait n'avoir que 17 pseudo-observations anonymisées pour entraîner et définir les paramètres de notre modèle de référence.

64. Pour rappel le lien complet peut faire en sorte que certains points d'une cluster soient plus proches de points issus d'un autre groupe.

65. Ce package permet de réaliser des opérations de clustering plus efficacement, notamment en réduisant la quantité de mémoire nécessaire pour le calcul de matrices de dissimilarité.

4.2.1.2 Algorithmes de clustering dont le nombre de classes est déterminé automatiquement de manière optimale

Alors que nous avions jusqu'ici utilisé des algorithmes de partitionnement dont le nombre de classes à former devait être renseigné par l'utilisateur, soit en amont dans le cadre du K-means, soit à l'issue du clustering pour le partitionnement hiérarchique en définissant l'étape d'arrêt. Nous allons désormais nous intéresser aux méthodes non-supervisées capables de déterminer automatiquement le nombre de classes optimal au sein du jeu de données.
Dans un premier temps nous emploierons et décrirons une méthode de partitionnement basé sur l'estimation de densité (*DBSCAN/OPTICS*) puis dans un second temps un algorithme de clustering reposant sur le principe de partage d'affinité entre les observations (*Propagation d'affinité*).

- **DBSCAN/OPTICS**

Le DBSCAN (density-based spatial clustering of applications with noise) est un algorithme de partitionnement inventé par M. Ester, H-S. Kriegel, J. Sander et X. Xu en 1996 fondé sur la densité de répartition des observations au sein d'une base de données afin de déterminer au mieux des classes le constituant. L'algorithme OPTICS (ordering points to identify the clustering structure), proposé par M. Ankerst, M. Breunig, H-P Kriegel et J. Sander en 1999, en est un dérivé basé sur le même principe de fonctionnement mais permettant de distinguer au sein du jeu de données des clusters de différentes densités. Par la suite nous allons principalement expliciter le fonctionnement de DBSCAN, nous reviendrons alors seulement à la fin sur les changements apportés par OPTICS.

Un des principaux intérêts de ces méthodes, outre le fait de ne pas avoir à spécifier le nombre de classes à former, est d'être en mesure de constituer des clusters non nécessairement convexes (à la différence par exemple du noyau K-means réalisant des partitions de Voronoï), offrant plus de liberté lors du partitionnement mais a contrario également la possibilité d'obtenir des clusters moins équilibrés au regard du nombres d'observations qu'ils comportent comme le montre le comparatif de la figure (22). Or dans notre cas cela peut-être problématique dans la mesure où tous les clusters sont chacun synthétisés en une unique pseudo-observation, toutes équipondérées, indépandamment du nombre de polices présent dans chaque cluster. Ainsi cela pourrait conduire à sur-représenter certains types d'individus (où à l'inverse en sous-représenter d'autres) une fois la base anonymisée.
Il faudra donc rester attentif à ce phénomène en observant et en analysant rigoureusement la répartition des observations au sein des clusters obtenus (en s'assurant également toujours qu'aucun cluster ne comporte qu'une seule observation) avant de porter nos conclusions sur la qualité de la tarification anonymisée réalisée.

L'algorithme DBSCAN nécessite deux paramètres pour réaliser sa méthode de partitionnement : une distance ϵ et un nombre de points minimal MinPoints.

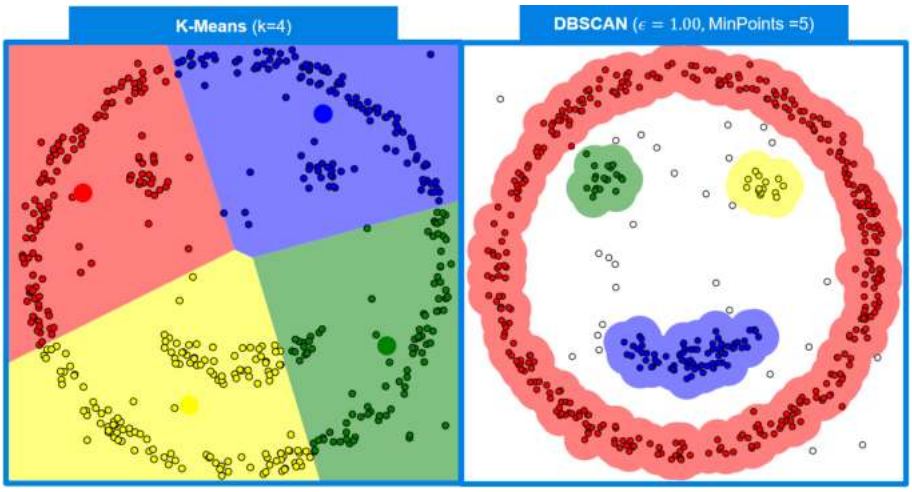


FIGURE 22 : Comparaison des partitionnements à l'aide du K-means et du DBSCAN d'un jeu de données singulier

Le paramètre MinPoints représente le nombre minimum de points présents dans un rayon ϵ autour d'une observation afin de considérer qu'ils appartiennent tous à un même cluster. En d'autres termes ces paramètres permettent de caractériser la densité de ce qui doit être considéré comme un cluster au sein du jeu de données.
Avant d'expliquer plus précisément le fonctionnement et la principe de cet algorithme, il nous faut définir certaines notions :

Définition (*Epsilon-voisinage*) On définit l'*epsilon-voisinage* $N_\epsilon(x)$ d'un point x quelconque du jeu de données C comme étant l'ensemble des observations de C dont la distance par rapport à x est strictement inférieure à ϵ, soit :

$$N_\epsilon(x) = \{u \in C,\ d(x,u) < \epsilon\} \qquad \forall x \in C$$

où d représente une mesure de distance sur C, tel que (C,d) définisse un espace métrique.
D'un point de vue topologique, on peut donc considérer $N_\epsilon(x)$ comme un voisinage de x dans C équivalent à une boule ouverte $\mathcal{B}(x,\epsilon)$ centrée en x de rayon ϵ.

Définition (*Point intérieur*) On considère x comme un point intérieur si son espilon-voisinage contient au moins MinPoints tel que :

$$|N_\epsilon(x)| \geqslant \text{MinPoints}$$

Définition (*Connexion par densité*) On dit que les points x et u sont connectés par densité dans C s'il existe un ensemble $I := \{\nu_1, \ldots, \nu_m\}$ de m points intérieurs tel que :

$$x \in \Omega := \bigcup_{i=1}^{m} N_\epsilon(\nu_i)$$

où $\forall i \in \{2, \ldots, m\}$, $\nu_i \in N_\epsilon(\nu_{i-1})$ et $\nu_1 \in N_\epsilon(u)$.

On peut par ailleurs considérer Ω comme un epsilon-voisinage en série de u, puisqu'il s'agit toujours du point de vue topologique d'un ouvert [66].
En d'autres termes, on dit que x et u sont connectés par densité si l'on peut atteindre x (resp. u) par une suite d'epsilon-voisinages successifs des points intérieurs ainsi formés en partant de u (resp. x) comme le montre le schéma (23).

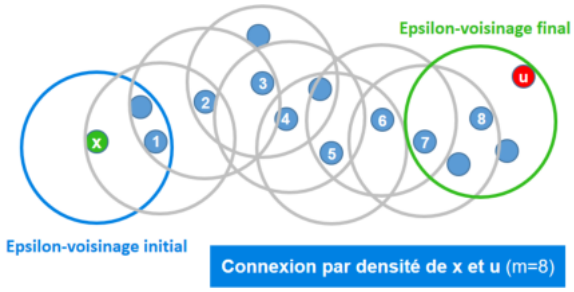


FIGURE 23 : Schéma de connexion par densité de x et u (où MinPoints = 2)

L'algorithme DBSCAN construit justement des clusters de points connectés par densité. Il itère sur les points de la base où il calcule à chaque fois l'epsilon-voisinage du point en question et s'il compte plus de MinPoints les epsilon-voisinages de chacun d'eux. Et ainsi de suite jusqu'au moment où la progression n'est plus possible car l'algorithme n'a pas atteint un point intérieur, alors le cluster n'est plus en expansion. Auquel cas il recommence le processus sur un nouveau point n'appartenant pas encore à un cluster.
Si l'algorithme atteint un point qui n'est pas intérieur, ce dernier est automatiquement écarté et considéré comme du bruit (ou une donnée aberrante). En reprenant l'exemple illustré précédent on remarque bien sur la figure (24) que les points bruités ont été dans l'ensemble correctement écartés (entourés en noir).

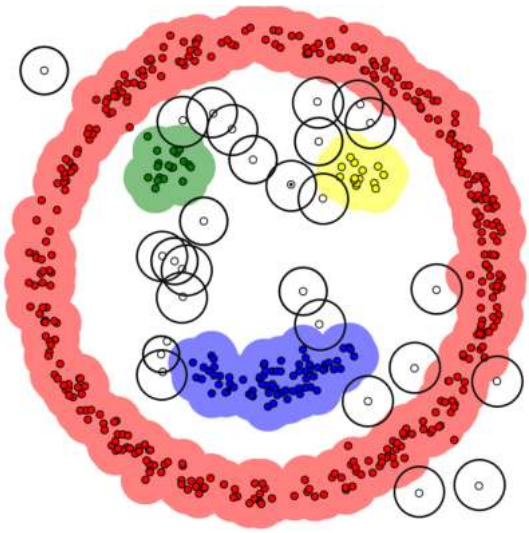
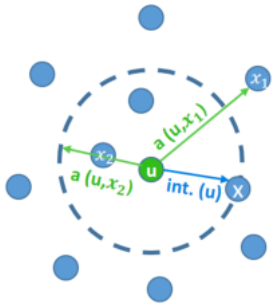


FIGURE 24 : Détection des données aberrantes par l'algorithme DBSCAN

66. La propriété d'ouverture d'un ensemble est stable par réunion.

Bien qu'il s'agisse d'une méthode très efficace dans l'ensemble (qualité du clustering et temps de calcul) quelques problèmes persistent avec cette méthode de partitionnement. Tout d'abord le paramétrage de la méthode s'avère assez complexe car il faut faire en sorte qu'il y ait un nombre suffisant de points intérieurs en modulant les valeurs de ϵ et de MinPoints. Ainsi si ϵ est trop petit et/ou MinPoints trop grand, le nombre de points intérieurs risque d'être limité si bien que le nombre de partitions final sera très important et les clusters très petits (un bon nombre d'entre eux ne compteront qu'une seule observation par exemple).
De plus comme il est possible de l'observer avec d'autres méthodes de clustering ayant recours à des calculs de métriques, DBSCAN est particulièrement sensible au fléau de la dimension, en effet les boules ouvertes calculées lors des epsilon-voisinages en grandes dimensions risquent de ne contenir pratiquement aucun point et alors nous nous retrouvons dans le cas précédemment cité où le nombre de points intérieurs est insuffisant.

Il existe, comme nous l'avons précédemment évoqué, un algorithme alternatif nommé OPTICS basé sur le même principe que DBSCAN mais qui permet notamment de se soustraire du paramétrage de ϵ facilitant ainsi la détection de clusters de densité différentes.
Ainsi l'algorithme définit deux attributs pour chacun des points du jeu de données. Tout d'abord la distance interne (*core-distance*) d'un point u qui s'avère être la distance minimale entre u et le point x le plus proche permettant d'avoir $|N_{\text{dist. interne}}(u)| \geqslant$ MinPoints (en somme que u devienne un point intérieur par rapport à sa distance interne)[67]. Et la distance d'accessibilité (*reachability-distance*) d'un point x_k par rapport à u qui s'avère être le maximum entre la distance interne de u et la distance usuelle issue de la métrique employée entre u et x_k. La figure (25) ci-contre permet de schématiser ces deux valeurs.



Distance interne int.(u) et distances d'accessibilité a(u,x_1), a(u,x_2), pour Min Points =4

FIGURE 25

Il est toutefois possible de renseigner le paramètre ϵ dans l'algorithme OPTICS, dès lors il permet de définir une valeur minimale pour les distances précédentes, tel que :

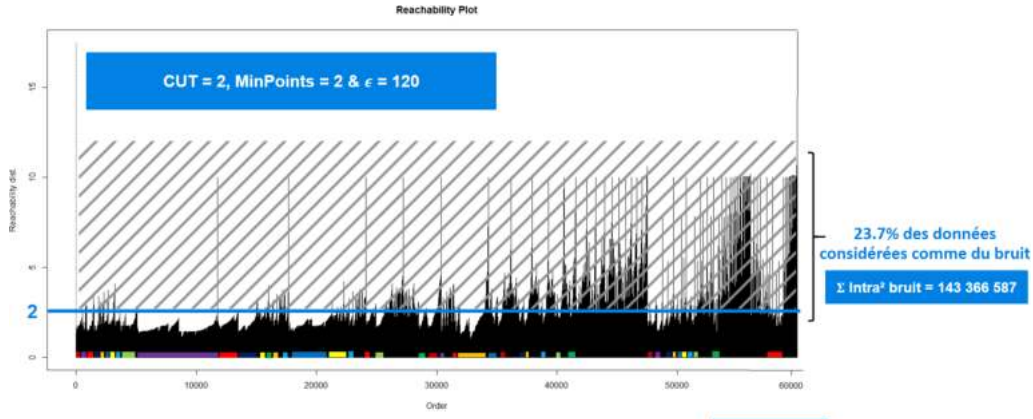
$$\text{interne}(u) = \begin{cases} \textit{Indéfini} & \text{Si } |N_\epsilon(u)| < \text{MinPoints} \\ d(u, \textit{MinPoint}^{\text{ème}}\textit{obs.}) & \text{Sinon} \end{cases}$$

$$\text{accessibilité}(u, x_k) = \begin{cases} \textit{Indéfini} & \text{Si } |N_\epsilon(u)| < \text{MinPoints} \\ max(\text{interne}(u), d(u, x_k)) & \text{Sinon} \end{cases}$$

En définissant le paramètre ϵ ici, nous pouvons donc empêcher la constitution de clusters de densité trop faible (si le paramètre n'est pas renseigné, $\epsilon \to \infty$) et ainsi accélérer le déroulement de l'algorithme.

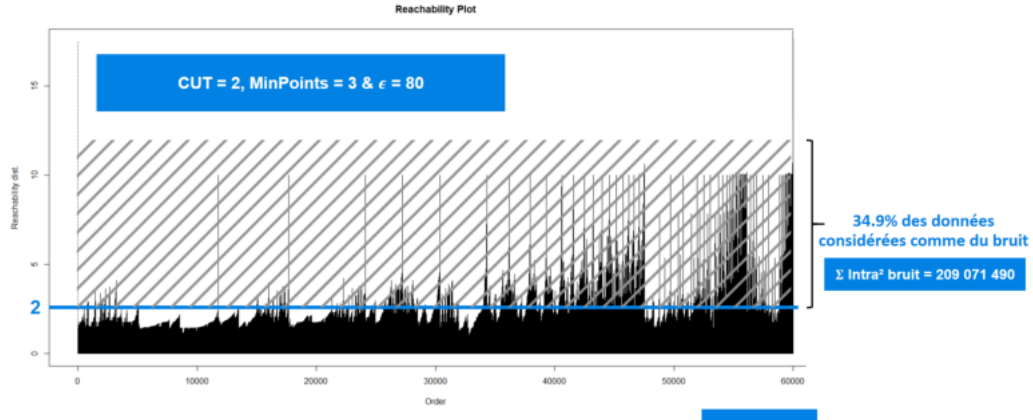
67. Comparativement à DBSCAN ϵ est en quelque sorte variable, en fonction du paramètre MinPoints.

L'algorithme OPTICS calcule alors ces valeurs pour chaque observation du jeu de données puis ordonne les points dans l'ordre croissant selon leur distance d'accessibilité par rapport à leur plus proche voisin. On obtient alors un graphique appelé *reachability-plot* où l'axe x représente les données ainsi ordonnées et en abscisse les distances d'accessibilité associées. Par la suite l'utilisateur défini un seuil au delà duquel les points ayant une distance d'accessibilité supérieure sont considérés comme du bruit tandis ce que les points inférieurs contigus sont regroupés au sein d'un même cluster. Le reachibility-plot ci-dessous (26) illustre ce principe.
Nous avons alors réalisé deux partitionnements avec cet algorithme sous `R` à l'aide du package `dbscan`, d'une part avec MinPoints = 2 et $\epsilon = 120$ puis où MinPoints = 3 et $\epsilon = 80$. Dans les deux cas nous avons choisi un seuil d'écrêtage de 2.



Min	Q.25%	Mediane	Moyenne	Q. 75%	Max
0	0.8	1	577.8	5.2	775912.7

FIGURE 26 : Reachability-plot et résultats d'OPTICS(2,2,120).



Min	Q.25%	Mediane	Moyenne	Q. 75%	Max
0	1.6	4.3	1148.7	14.1	775912.7

FIGURE 27 : Reachability-plot et résultats d'OPTICS(2,3,80).

Afin de définir le seuil d'écrêtage nous nous sommes basés sur les reachability-plot obtenus, l'idée étant d'obtenir le maximum de clusters, et pour déterminer le paramètre MinPoints il fallait qu'il soit le plus petit possible afin de limiter au maximum le nombre de données considérées comme du bruit et d'essayer d'obtenir un nombre de clusters important.

Au final on remarque que la plupart des clusters obtenus sont très homogènes et denses (près de 95% des clusters formés par notre première implémentation a une somme de ses carrés intra clusters inférieure à 57). A priori cela est due au nombre important de petits clusters (moins de cinq observations) que nous sommes parvenus à constituer (77%) parmi les classes crées (5430 pour la première implémentation et 2726 pour la seconde).
Cependant, il semble également assez clair que le fait de vouloir avoir un nombre important de clusters dans notre cas, nous conduit nécessairement à augmenter le nombre d'observations considérées comme du bruit dans l'algorithme. En effet, un des critères déterminant s'avère être le seuil d'écrêtage, or ce dernier en étant très petit entraîne un nombre important de points supérieur à ce niveau à être considérés comme des outliers.
Finalement on constate que les clusters obtenus sont assez différents entre eux, car si une grande majorité des classes constituées sont petites et denses, les autres partitions s'avèrent beaucoup plus volumineuses et bien plus hétérogènes en leur sein [68].
Ces grandes disparités entre les clusters issus de ces partitionnements pourraient donc conduire à la sous-représentation de certaines observations une fois la donnée anonymisée. Il nous faudra donc y être vigilant et conscient lors de l'utilisation de ces données pour le paramétrage de notre modèle de tarification.

- **Propagation d'affinité**

Le partitionnement par propagation d'affinité a été défini pour la première fois par Frey et Dueck en 2007, et tout comme le DBSCAN/OPTICS permet de réaliser un partitionnement sans spécifier au préalable le nombre de classes à constituer. Néanmoins, il n'est pas question ici d'estimation de clusters par densité mais plutôt de la formation de ces clusters par transfert d'information au travers le calcul d'un facteur de similarité entre les observations. L'idée étant alors de regrouper les points autour d'individus représentatifs, unique pour chaque cluster et expliquant le mieux les autres points de la classe.
Un tel facteur s, pour trois observations distinctes x_i, x_j et x_k, est défini tel que $s(x_i, x_j) > s(x_i, x_k)$ si et seulement si l'observation x_i est plus similaire à x_j qu'à x_k [69]. Il est donc nécessaire de calculer une matrice S de taille $n \times n$ regroupant la totalité des similarités des différents individus entre eux avant d'utiliser cet algorithme. On peut donc déjà remarquer facilement une limitation potentielle de cette

68. En particulier le cluster des données bruitées qui représente à lui seuil entre 25 et 35% des données dans nos implémentations et qui regroupent de facto un ensemble très différents d'individus.

69. Bien que nous n'emploierons pas cette mesure, la distance euclidienne négative $-\|x_i - x_j\|^2$ est couramment utilisée.

méthode dans le cas où nous avons beaucoup de données et donc n très grand, la matrice de similarité S prendra d'autant plus de temps à être calculer et à être manipuler par la suite.
De plus les valeurs des diagonales de la matrice S sont particulières puisqu'elle ne représente pas la similarité d'un même individu ($s(x_i, x_i)$) mais la notion de préférence, c'est à dire une forme de pondération influençant la propension d'un individu à être représentatif. Si tous les coefficients diagonaux sont identiques, alors ils permettent d'influer sur le nombre de classes formées à l'issue du partitionnement. Ainsi des valeurs faibles produiront moins de clusters tandis ce que des coefficients élevés conduiront à constituer davantage de classes. Dans notre cas alors que nous souhaitons obtenir un nombre de clusters maximum (mais comptant au moins deux individus en leur seins) nous choisirons d'établir ces coefficients à 1, valeur maximale de la mesure de similarité de *Gower* que nous avons choisi d'utiliser. Dans d'autres cas il est courant d'utiliser la valeur médiane de la similarité calculée sur l'ensemble des paires d'individus du jeu de données.

Nous avons choisis d'utiliser la distance de *Gower* comme mesure de similarité car elle peut facilement être utilisée pour des variables quantitatives et qualitatives. Il s'agit par ailleurs d'une mesure de dissimilarité tel que $s'(x_i, x_j) \in [0, 1]$, où plus $s'(x_i, x_j)$ tend vers 1 plus les deux individus sont différents. Il est alors assez simple de retrouver une mesure de similarité en calculant $s(x_i, x_j) = 1 - s'(x_i, x_j)$.
Par exemple si l'on prend deux observations x_i et x_j comptant chacunes deux variables notés x_{ik}, où $k \in \{1, 2\}$ tel que la première variable soit catégorielle et la seconde numérique, on peut calculer la dissimilarité de la première variable qualitative tel que :

$$s'_{ij1} = \begin{cases} 0 & \text{Si } x_{i1} = x_{j1} \\ 1 & \text{Sinon} \end{cases}$$

Et calculer séparément la dissimilarité des deux individus au regard de la seconde variable quantitative, tel que :

$$s'_{ij2} = \frac{|x_{ik} - x_{jk}|}{\max(x_{.k}) - \min(x_{.k})}$$

Ensuite, afin de calculer la mesure de similartié entre les individus x_i et x_j il suffit de combiner l'ensemble des coefficients de dissimilarité calculés pour les variables sur les deux individus tel que :

$$s(x_i, x_j) = 1 - \frac{\sum_{k=1}^{K} w_{ijk} s'_{ijk}}{\sum_{k=1}^{K} w_{ijk}}$$

où K représente le nombre de variables total et w_{ijk} le poids de la variable k dans la mesure. Un poid pour une variable est nul lorsqu'elle ne doit pas être prise en compte dans l'élaboration de la similarité (données manquante par exemple).

Dans notre cas on considère toujours les mêmes variables que celles explicitées lors du partitionnement par l'algorithme du K-means (soit K=7), étant donnée qu'il

n'y a aucune donnée manquante et que nous ne souhaitons pas accorder davantage d'importance à une variable en particulier l'ensemble des poids vaut 1.
Par la suite l'algorithme de propagation d'affinité représente les points du jeu de données sous la forme d'un graphe complet (où tous les points sont reliés aux autres par une arrête), les individus s'échangent alors deux types de messages sous forme de matrices, voir figure (28) ci-dessous, concernant la détermination des individus représentatifs.

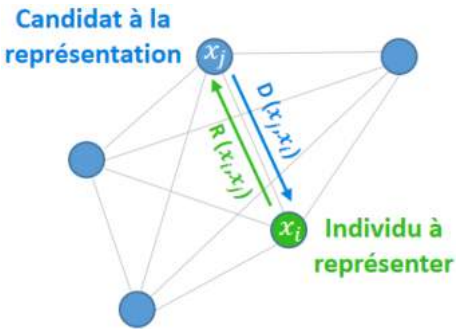


FIGURE 28

La matrice de responsabilité R où $r(x_i, x_j)$ défini la propension du point x_j à être représentatif de x_i comparativement à d'autres points candidats. Cette information est envoyée du point x_i au point x_j.

La matrice de disponibilité D où $d(x_i, x_j)$ défini la pertinence pour le point x_i de choisir x_j comme individu représentatif comparativement aux préférences des autres points à choisir x_j comme représentant. Cette information est envoyée du point x_j au point x_i.

Ces deux matrices sont initialisées à zéros, puis mises à jour itérativement tel que :

1. $r(x_i, x_j) := s(x_i, x_j) - \max_{j \neq j'} \{d(x_i, x_{j'}) + s(x_i, x_{j'})\}$

   2.1 $d(x_i, x_j) := \min\left(0, r(x_j, x_j) + \sum_{x_{i'} \notin \{x_i, x_j\}} \max(0, r(x_{i'}, x_j))\right) \quad \forall x_i \neq x_j$

   2.2 $d(x_j, x_j) := \sum_{x_{i'} \neq x_j} \max(0, r(x_{i'}, x_j))$

Cet algorithme cherche donc pour chaque individu dans la base le point représentatif qui permet de maximiser la somme des disponibilités et des responsabilités. La procédure s'arrête alors lorsque les clusters (ou les individus représentatifs) ne changent plus à l'issue d'un nombre d'itérations prédéfini. Un exemple schématique du déroulement de l'algorithme est présenté ci-dessous (29).

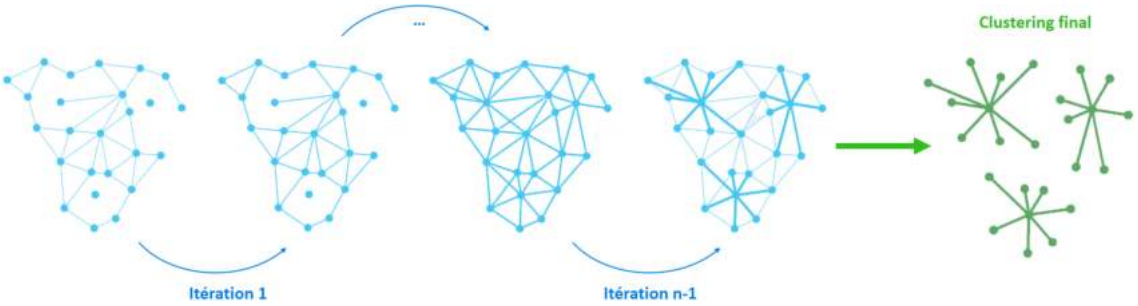


FIGURE 29 : Schéma du déroulement d'un clustering par propagation d'affinité

Nous avons réalisé un tel partitionnement sous `R` à l'aide du package `apcluster` et en découpant notre jeu d'apprentissage en plus petits blocs afin de pouvoir plus facilement calculer les matrices de dissimilarité associées. En effet la quantité de mémoire pour calculer une telle matrice en utilisant le package `daisy` et la mesure de `Gower` précédemment définie, directement sur les 60 000 observations (soit une matrice réeles de taille 60000×60000) excède largement la quantité de RAM disponible sur nos machines.
Nous présentons grâce aux graphiques ci-dessous (30) les résultats des partitionnements obtenus à sur notre jeu de données d'apprentissage pré-découpé en blocs de tailles respectives de 1000, 500, 250 et enfin 100 observations.

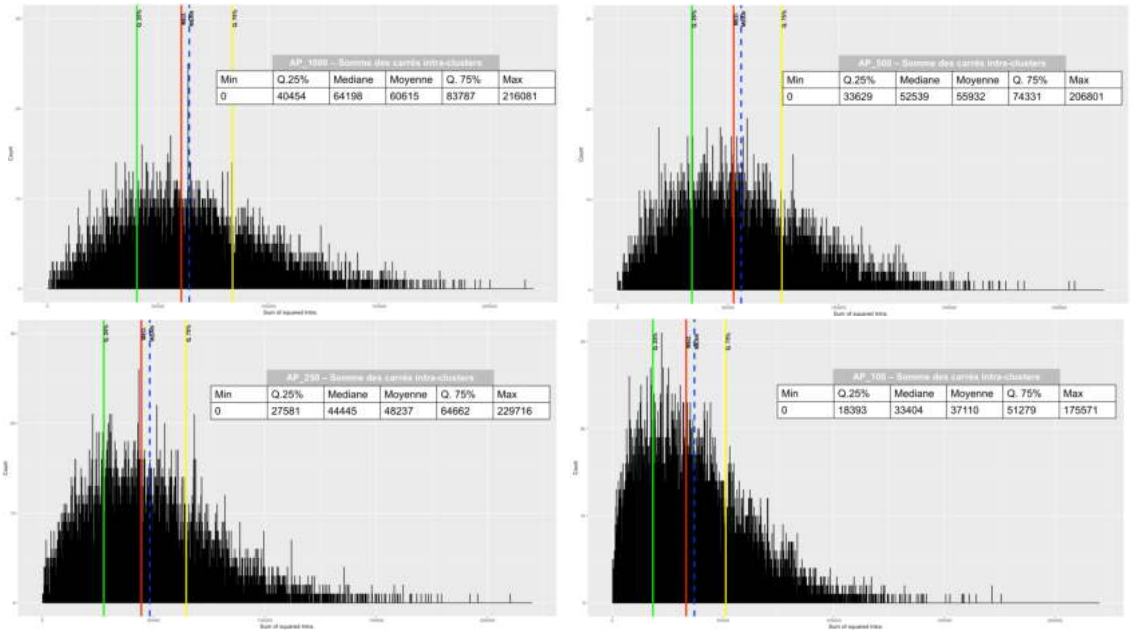


FIGURE 30 : Répartition de la somme des carrés intra-clusters pour les différents partitionnements par l'algorithme de propagation d'affinité

4.2.2 Analyse des partitionnements obtenus

A l'issue des différents partitionnements réalisés nous pouvons nous rendre compte que les méthodes proposées présentent des résultats très variables au regard de la compacité des clusters obtenus.
Si l'algorithme du K-means permet d'obtenir dans l'ensemble de très bons résultats dans ce domaine (en particulier en choisissant de prendre $K = 6000$) avec des clusters ayant en moyenne une distance intra de seulement 64.12, on se rend compte que le choix de k influe très fortement sur la compacité des partitions obtenues puisque par la suite en diminuant cette valeur, la distance moyenne et maximale intra-clusters n'a cessé de se détériorer.

Concernant le clustering hiérarchique les résultats que nous sommes parvenus à obtenir restent bons même s'ils souffrent de la comparaison avec ceux issus du K-means.

Une différence notable réside tout de même dans la répartition de la compacité des clusters obtenus, alors qu'avec le K-means nous avions principalement des partitions assez homogènes au regard de leurs distance internes, ici les différents clusters semblent finalement assez hétérogènes. La répartition des présentée en figure (21) montre bien un étalement des distances intra-clusters du partitionnement.

Lorsque nous avons présenté des algorithmes de partitionnements ne nécéssitant pas de spécifier au préalable le nombre de clusters à créer nous avons commencé par l'algorithme OPTICS, et comme nous l'avions déjà précisé à l'issue de sa première implémentation (26), nous obtenons des clusters très compacts et homogènes (près de 75% des clusters obtenus ont une distance interne inférieure à 5.2) cependant le regroupement des nombreuses observations considérées comme bruitées par l'algorithme engendre la formation de partitions très différentes des autres contenant beaucoup plus d'observations très disparates.
Ces grandes disparités entre les clusters issus de ces partitionnements pourraient donc conduire à la sous-représentation de certaines observations une fois la donnée anonymisée. Il nous faudra donc y être vigilant et conscient lors de l'utilisation de ces données pour le paramétrage de notre modèle de tarification.

Enfin le partitionnement par propagation d'affinité fournit des résultats tout à fait singuliers. En effet ils s'avèrent étonnamment médiocres au regard de nos critères de qualité : nous parvenons au mieux à avoir une distance intra-clusters moyenne de 37110 (soit une valeur près 30 fois plus élevée que la distance maximale des cluters issus des méthodes où k doit être paramétré) en choisissant de partitionner et d'agréger les clusters obtenus à partir de sous-groupes de seulement 100 observations issues du jeu de données d'apprentissage.
Finalement cette méthode de partitionnement engendre des clusters très hétérogènes sur notre jeu de données où il s'avère très difficile de déterminer véritablement une logique de partitionnement, nous avons cependant décidé de conserver ces résultats car il nous semblait intéressant de pouvoir utiliser les données anonymisées issue de cette procédure dans le cadre de notre méthode de tarification et de constater les résultats alors obtenus.

Nous avons résumé les résultats les plus probants de chaque méthode de partitionnement dans le tableau suivant (23).

TABLE 12 : ‘‘Meilleurs” partitionnements pour chaque méthode de clustering

Partitionnement	dist. Min	dist. Moy	dist. Max
KM_6000	0	64.12	1013.18
HCLUSTER_2320	0	303.2	1078.8
OPTICS(2,2,120)	0	577.8	775912.1
AP_100	0	37110	175571

4.3 Tarification anonymisée : analyse des résultats & limites

Alors que nous avons au cours des précédentes étapes préparé notre jeu d'apprentissage de 60 000 polices d'assurance à son anonymisation par agrégation individuelle grâce à de diverses méthodes de clustering, nous allons désormais utiliser les pseudo-observations issues des partitions obtenues pour paramétrer notre modèle de tarification de référence directement sur ces données anonymes.

La méthode d'élaboration des pseudo-individus de chaque cluster est identique à toutes les méthodes d'anonymisation utilisées : Pour chaque partition issue d'une méthode de clustering nous calculons un individu moyen qui servira de peuso-observation dans le cadre de notre anonymisation. Ainsi il s'agit d'une agrégation des variables du jeu de données initial selon leurs valeurs moyennes pour les variables numériques ou leurs valeurs majoritaires pour les variables catégorielles. On obtient ainsi une pseudo-observation représentative par cluster, l'ensemble de ces pseudo-observations constitue alors notre jeu de données d'apprentissage anonymisé.

Nous allons donc pouvoir calculer les primes pures moyennes obtenues à l'issue de l'application du modèle de tarification benchmark basé sur nos données d'apprentissage anonymisée, selon nos données test individuelles (donc non-anonymisées) de 30 000 individus. Nous comparerons alors leurs montants selon la méthode d'anonymisation employée (algorithme de clustering, paramétrage, etc.)

4.3.1 Présentation des primes pures obtenues après anonymisation

Nous présenterons ici les résultats de la procédure de tarification anonymisée que nous avons implémenté en séparant dans chacun des cas selon la méthode d'anonymisation utilisée.
Une fois terminé nous comparerons les primes pures obtenues ainsi que les autres données utiles issues de chacunes des tarification afin de déterminer quelle méthode est la plus pertinente dans notre cadre.

Pour chaque paramétrage des différentes méthodes d'anonymisation employée nous présenterons par la suite deux graphiques, l'un représentant la distribution des écarts relatifs entre la prime pure obtenue à l'aide du modèle anonymisé et celle obtenue à l'issue du modèle de référence (par bucket de 5%). Puis le second, un nuage de points associant en abcisse la prime pure du modèle benchmark (en ligne à ligne - LAL) et en ordonnée la prime du modèle anonymisé (CLUST). Ainsi l'objectif est d'avoir un nuage de point le plus proche possible de la première bissectrice ($y = x$), à l'image d'un QQ-plot classique, ce qui garantirait une certaine cohérence entre les deux types de modèles.
Enfin un tableau synthétique de certaines grandeurs caractéristiques (écart total, sinistralité globale, etc.) est également établit pour chaque implémentation.

4.3.1.1 Anonymisation grâce à l'algorithme du k-Means

Nous commençons par nous intéresser aux résultats issus de l'anonymisation à l'aide de l'algorithme du k-Means, et spécifiquement pour les paramétrisation précédemment étudiées où $K \in (6000, 4500, 3000, 1200)$.

Dans le cas où $K = 6000$, nous obtenons donc naturellement 6000 clusters comptant en moyenne 10 observations chacuns (au minimum 2 et au maximum 27). La sinistralité moyenne observée sur chacunes de ces partitions s'élève alors à 381€.
En paramétrant notre modèle de tarification de référence à partir des données d'apprentissage anonymisées obtenues puis en évaluant les données de test en ligne à ligne à partir de cette nouvelle paramétrisation on obtient les résultats suivants (31) :

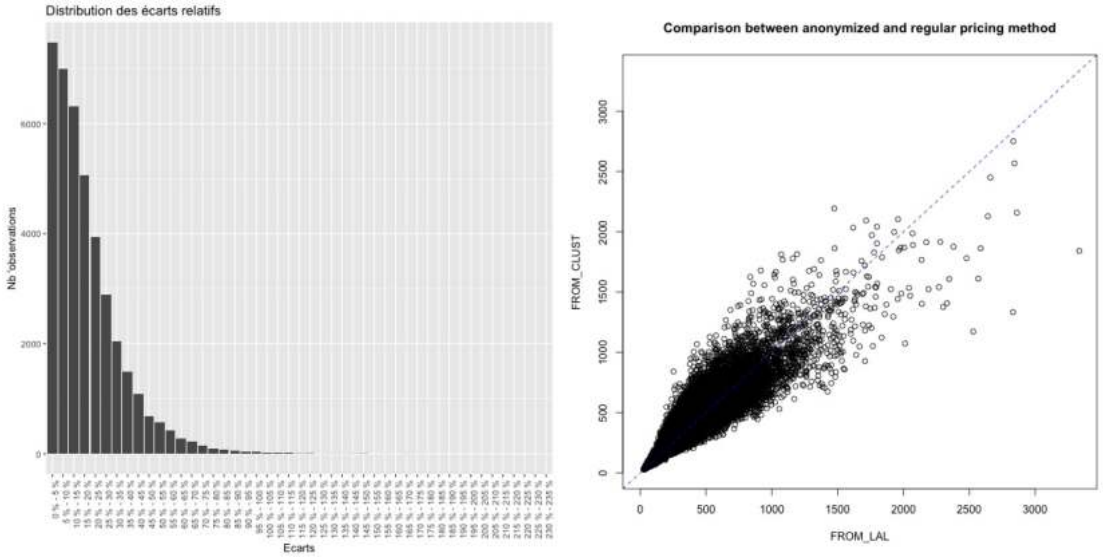


FIGURE 31 : Distribution des écarts relatifs & comparaison des modèles

Plus d'informations sur l'élaboration de ces figures à l'introduction de la partie (4.3.1)

TABLE 13 : Grandeurs caractéristiques (K=6000)

SINISTRALITÉ TOTALE BENCHMARK	+1.266E7
SINISTRALITÉ TOTALE ANONYMISÉE	+1.324E7
SOMME QUADRATIQUE DES ÉCARTS	+1.391E11
ÉCART (ANONYMISÉ - BENCHMARK)	+5.779E5
ÉCART RELATIF (À BENCHMARK)	+4.56 %

Dans le cas où $K = 4500$, nous obtenons donc naturellement 4500 clusters comptant en moyenne 13 observations chacuns (au minimum 2 et au maximum 57). La sinistralité moyenne observée sur chacunes de ces partitions s'élève alors à 373€.

En paramétrant notre modèle de tarification de référence à partir des données d'apprentissage anonymisées obtenues puis en évaluant les données de test en ligne à ligne à partir de cette nouvelle paramétrisation on obtient les résultats suivants (32) :

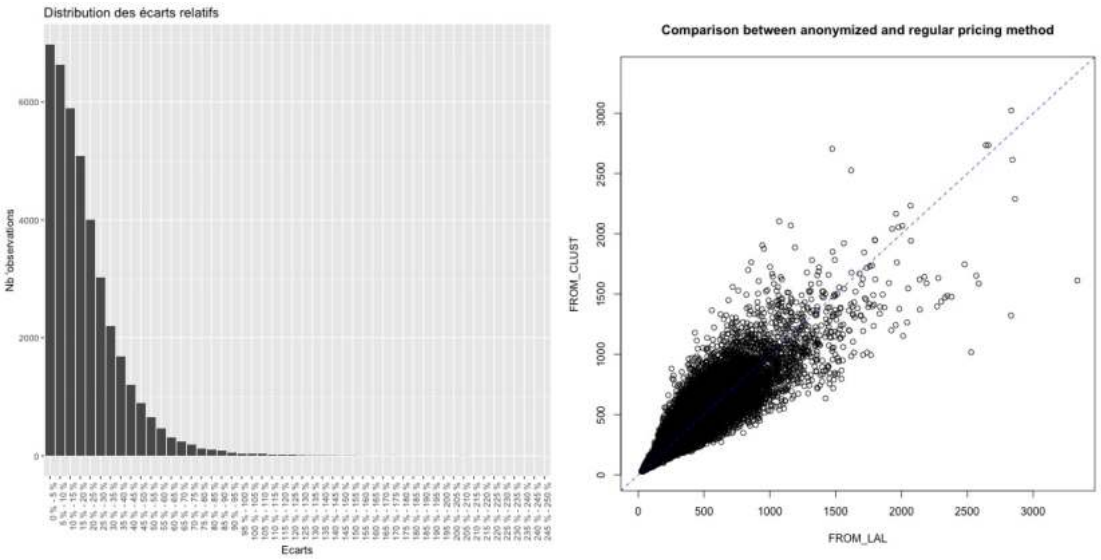


FIGURE 32 : Distribution des écarts relatifs & comparaison des modèles

Plus d'informations sur l'élaboration de ces figures à l'introduction de la partie (4.3.1)

TABLE 14 : Grandeurs caractéristiques (K=4500)

SINISTRALITÉ TOTALE BENCHMARK	+1.266E7
SINISTRALITÉ TOTALE ANONYMISÉE	+1.325E7
SOMME QUADRATIQUE DES ÉCARTS	+1.392E11
ÉCART (ANONYMISÉ - BENCHMARK)	+5.879E5
ÉCART RELATIF (À BENCHMARK)	+4.64 %

Dans le cas où $K = 3000$, nous obtenons donc naturellement 3000 clusters comptant en moyenne 20 observations chacuns (au minimum 8 et au maximum 82). La sinistralité moyenne observée sur chacunes de ces partitions s'élève alors à 375€.
En paramétrant notre modèle de tarification de référence à partir des données d'apprentissage anonymisées obtenues puis en évaluant les données de test en ligne à ligne à partir de cette nouvelle paramétrisation on obtient les résultats suivants (33) :

Plus d'informations sur l'élaboration de ces figures à l'introduction de la partie (4.3.1)

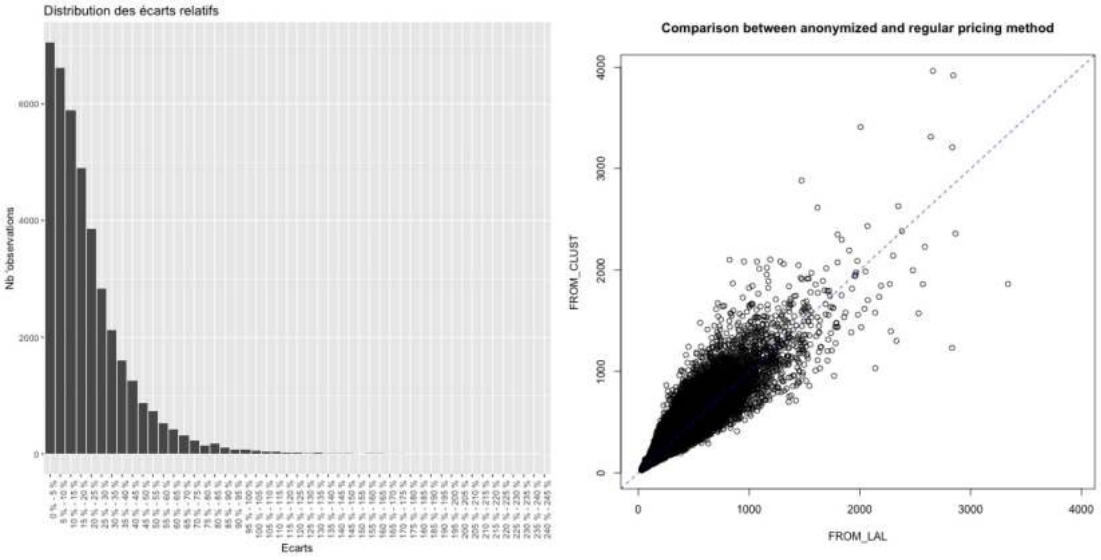


FIGURE 33 : Distribution des écarts relatifs & comparaison des modèles

TABLE 15 : Grandeurs caractéristiques (K=3000)

SINISTRALITÉ TOTALE BENCHMARK	+1.266E7
SINISTRALITÉ TOTALE ANONYMISÉE	+1.388E7
SOMME QUADRATIQUE DES ÉCARTS	+1.390E11
ÉCART (ANONYMISÉ - BENCHMARK)	+1.218E6
ÉCART RELATIF (À BENCHMARK)	+9.62 %

Dans le cas où $K = 1200$, nous obtenons donc naturellement 1200 clusters comptant en moyenne 50 observations chacuns (au minimum 12 et au maximum 151). La sinistralité moyenne observée sur chacunes de ces partitions s'élève alors à 382€.
En paramétrant notre modèle de tarification de référence à partir des données d'apprentissage anonymisées obtenues puis en évaluant les données de test en ligne à ligne à partir de cette nouvelle paramétrisation on obtient les résultats suivants (34) :

Plus d'informations sur l'élaboration de ces figures à l'introduction de la partie (4.3.1)

TABLE 16 : Grandeurs caractéristiques (K=1200)

SINISTRALITÉ TOTALE BENCHMARK	+1.266E7
SINISTRALITÉ TOTALE ANONYMISÉE	+1.592E7
SOMME QUADRATIQUE DES ÉCARTS	+1.399E11
ÉCART (ANONYMISÉ - BENCHMARK)	+3.258E6
ÉCART RELATIF (À BENCHMARK)	+25.73 %

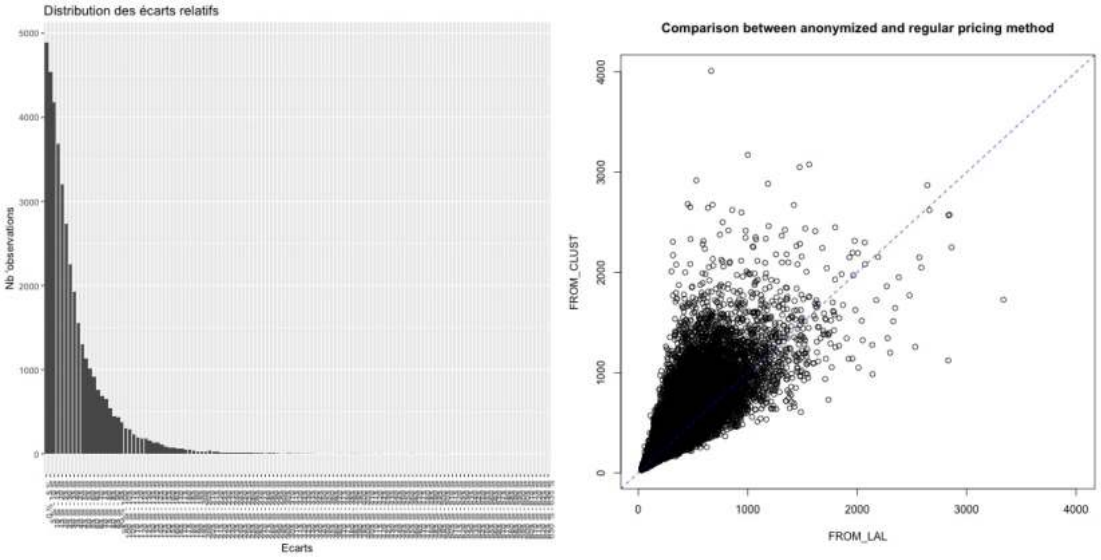


FIGURE 34 : Distribution des écarts relatifs & comparaison des modèles

4.3.1.2 Anonymisation par clustering hiérarchique

Nous n'avons réalisé qu'une implémentation de notre méthode d'anonymisation par clustering hiérarchique à l'aide de la fonction de lien de ward et avons ainsi obtenu près de 2320 partitions.
En moyenne les clusters comptent 26 observations (au minimum 7 et au maximum 152) et la sinistralité moyenne observée sur chacuns d'eux s'élève à près de 443€.
En paramétrant notre modèle de tarification de référence à partir des données d'apprentissage anonymisées obtenues puis en évaluant les données de test en ligne à ligne à partir de cette nouvelle paramétrisation on obtient les résultats suivants (35) :

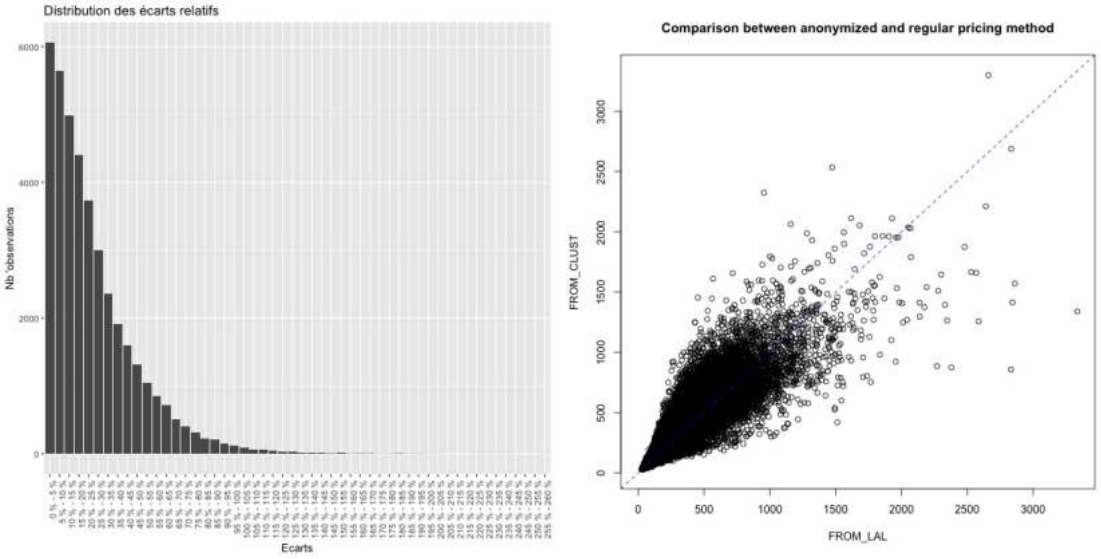


FIGURE 35 : Distribution des écarts relatifs & comparaison des modèles

Plus d'informations sur l'élaboration de ces figures à l'introduction de la partie (4.3.1)

TABLE 17 : Grandeurs caractéristiques (hClust - ward)

SINISTRALITÉ TOTALE BENCHMARK	+1.266E7
SINISTRALITÉ TOTALE ANONYMISÉE	+1.389E7
SOMME QUADRATIQUE DES ÉCARTS	+1.394E11
ÉCART (ANONYMISÉ - BENCHMARK)	+1.228E6
ÉCART RELATIF (À BENCHMARK)	+9.7 %

4.3.1.3 Anonymisation par estimation de densité (OPTICS)

Comme nous l'avons précédemment développé (4.2.1.2), nous avons implémenté deux paramétrages différents de l'algorithme OPTICS, dans les deux cas la valeur du paramètre d'écrêtage est fixé à 2 (CUT=2) et le nombre minimal de points à 2 (et une distance min. de 120) puis à 3 (et une distance min. de 80).
Pour notre première implémentation, nous avons trouvé 5431 clusters comptant en moyenne 11 observations (au minimum 2 et au maximum 15003, on observe ainsi comme nous l'avions précédemment remarqué des partitions très hétérogènes en nombre d'observations) et la sinistralité moyenne observée sur chacuns d'eux s'élève à près de 453€.
En paramétrant notre modèle de tarification de référence à partir des données d'apprentissage anonymisées obtenues puis en évaluant les données de test en ligne à ligne à partir de cette nouvelle paramétrisation on obtient les résultats suivants (36) :

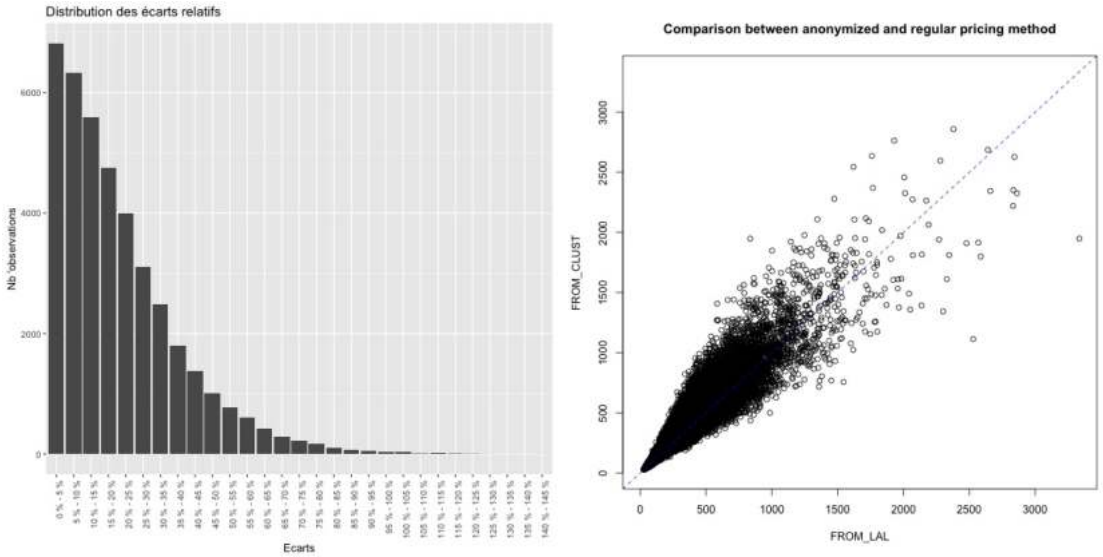


FIGURE 36 : Distribution des écarts relatifs & comparaison des modèles

Plus d'informations sur l'élaboration de ces figures à l'introduction de la partie (4.3.1)

TABLE 18 : Grandeurs caractéristiques (OPTICS(2,2,120))

SINISTRALITÉ TOTALE BENCHMARK	+1.266E7
SINISTRALITÉ TOTALE ANONYMISÉE	+1.445E7
SOMME QUADRATIQUE DES ÉCARTS	+1.391E11
ÉCART (ANONYMISÉ - BENCHMARK)	+1.788E6
ÉCART RELATIF (À BENCHMARK)	+14.12 %

Sur notre seconde implémentation, nous constatons toujours une répartition des individus au sein de nos 2570 clusters très inégale, on trouve en moyenne 23 observations par classe (au minimum 2 et au maximum 208121). La sinistralité moyenne observée sur chaque partition s'élève alors à près de 423€.
En paramétrant notre modèle de tarification de référence à partir des données d'apprentissage anonymisées obtenues puis en évaluant les données de test en ligne à ligne à partir de cette nouvelle paramétrisation on obtient les résultats suivants (37) :

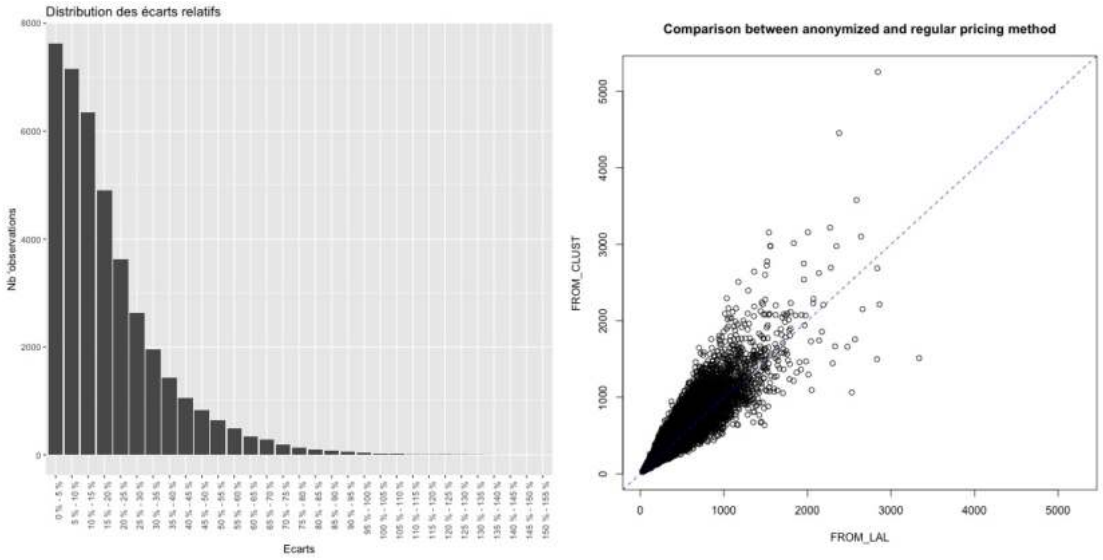


FIGURE 37 : Distribution des écarts relatifs & comparaison des modèles

Plus d'informations sur l'élaboration de ces figures à l'introduction de la partie (4.3.1)

TABLE 19 : Grandeurs caractéristiques (OPTICS(2,3,80))

SINISTRALITÉ TOTALE BENCHMARK	+1.266E7
SINISTRALITÉ TOTALE ANONYMISÉE	+1.407E7
SOMME QUADRATIQUE DES ÉCARTS	+1.389E11
ÉCART (ANONYMISÉ - BENCHMARK)	+1.408E6
ÉCART RELATIF (À BENCHMARK)	+11.12 %

4.3.1.4 Anonymisation par propagation d'affinité

Enfin nous avons réalisé quatre implémentations différentes de notre méthode d'anonymisation par propagation d'affinité en faisant varier la taille des blocs de données sur lesquels l'algorithme s'appliquait itérativement (1000, 500, 250 et 100).

Avec la première implémentation avec des blocs de 1000 individus, nous avons trouvé 6605 clusters comptant en moyenne 9 observations (au minimum 3 et au maximum 21) et la sinistralité moyenne observée sur chacuns d'eux s'élève à près de 378€.
En paramétrant notre modèle de tarification de référence à partir des données d'apprentissage anonymisées obtenues puis en évaluant les données de test en ligne à ligne à partir de cette nouvelle paramétrisation on obtient les résultats suivants (38) :

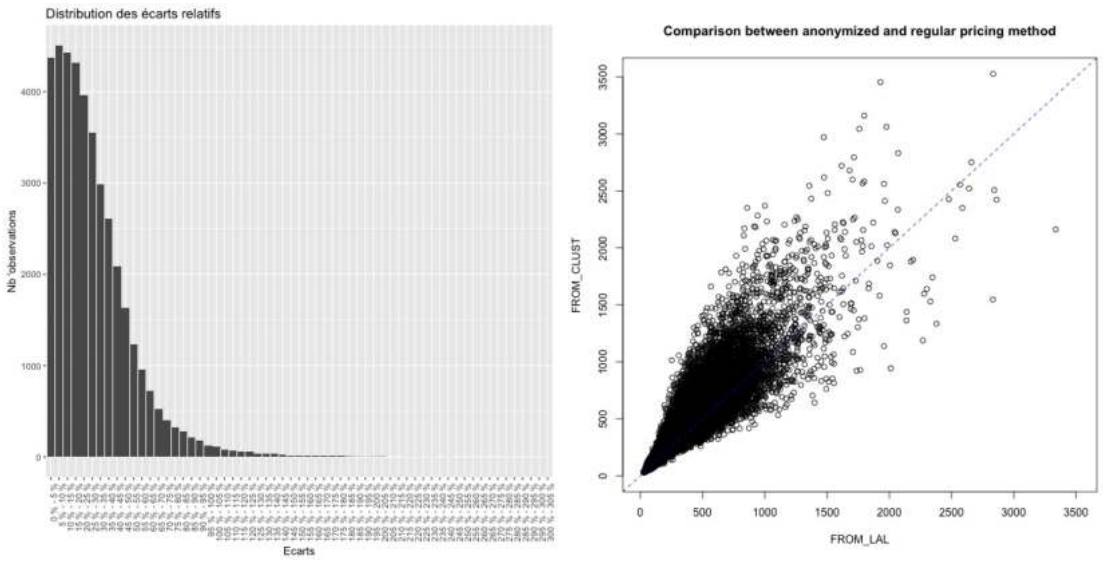


FIGURE 38 : Distribution des écarts relatifs & comparaison des modèles

Plus d'informations sur l'élaboration de ces figures à l'introduction de la partie (4.3.1)

Ensuite nous avons compté 7364 clusters sur notre deuxième implémentation (bloc de 500 individus) comportant en moyenne 8 observations (au minimum 2 et au maximum 20) et la sinistralité moyenne observée sur chacuns d'eux s'élève à près de 383€.

Table 20 : Grandeurs caractéristiques (AP_1000)

sinistralité totale benchmark	+1.266e7
sinistralité totale anonymisée	+1.544e7
somme quadratique des écarts	+1.398e11
écart (anonymisé - benchmark)	+2.778e6
écart relatif (à benchmark)	+21.94 %

En paramétrant notre modèle de tarification de référence à partir des données d'apprentissage anonymisées obtenues puis en évaluant les données de test en ligne à ligne à partir de cette nouvelle paramétrisation on obtient les résultats suivants (39) :

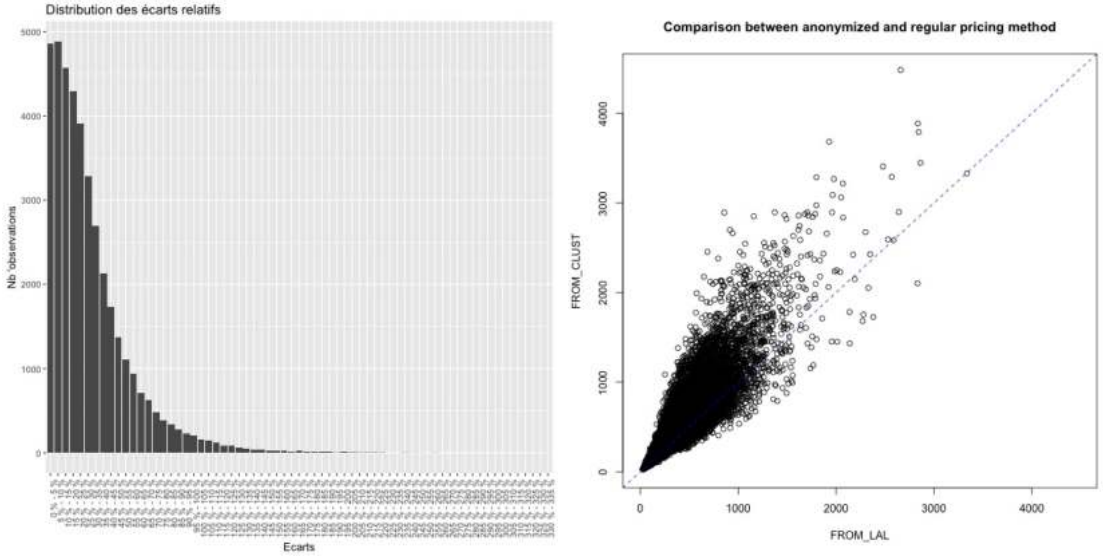


Figure 39 : Distribution des écarts relatifs & comparaison des modèles

Plus d'informations sur l'élaboration de ces figures à l'introduction de la partie (4.3.1)

Table 21 : Grandeurs caractéristiques (AP_500)

sinistralité totale benchmark	+1.266e7
sinistralité totale anonymisée	+1.594e7
somme quadratique des écarts	+1.401e11
écart (anonymisé - benchmark)	+3.278e6
écart relatif (à benchmark)	+25.89 %

Concernant le troisième paramétrage (bloc de 250 individus), nous avons compté 8287 clusters avec en moyenne 7 observations (au minimum 2 et au maximum 13) et la sinistralité moyenne observée sur chacuns d'eux s'élève à près de 375€.

En paramétrant notre modèle de tarification de référence à partir des données d'apprentissage anonymisées obtenues puis en évaluant les données de test en ligne à ligne à partir de cette nouvelle paramétrisation on obtient les résultats suivants (40) :

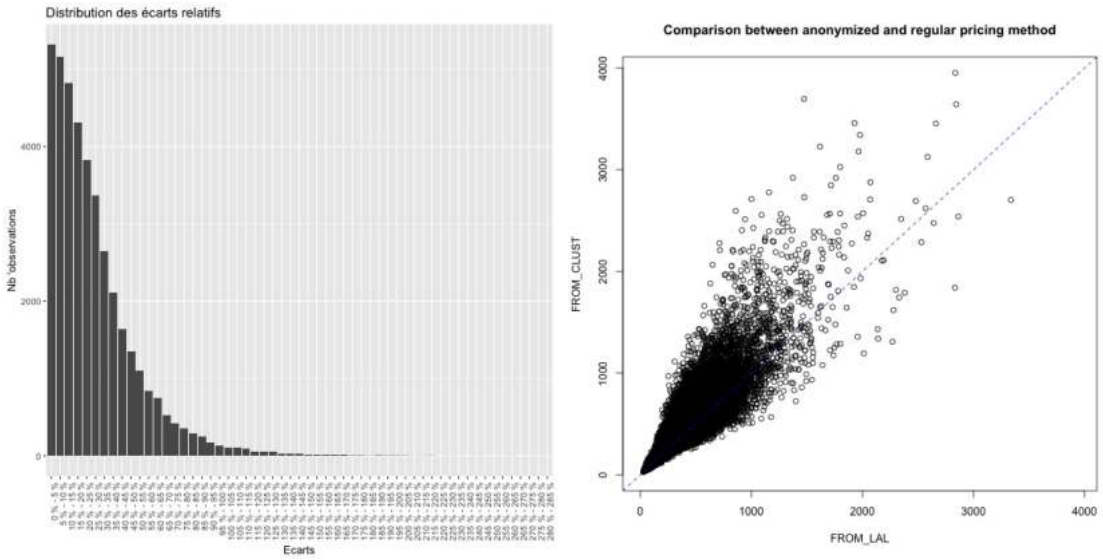


FIGURE 40 : Distribution des écarts relatifs & comparaison des modèles

Plus d'informations sur l'élaboration de ces figures à l'introduction de la partie (4.3.1)

TABLE 22 : Grandeurs caractéristiques (AP_250)

SINISTRALITÉ TOTALE BENCHMARK	+1.266E7
SINISTRALITÉ TOTALE ANONYMISÉE	+1.521E7
SOMME QUADRATIQUE DES ÉCARTS	+1.398E11
ÉCART (ANONYMISÉ - BENCHMARK)	+2.548E6
ÉCART RELATIF (À BENCHMARK)	+20.12 %

Enfin la dernière implémentation (bloc de 100 individus), compte 10044 clusters avec en moyenne 6 observations (au minimum 2 et au maximum 18) et la sinistralité moyenne observée sur chacuns d'eux s'élève toujours à près de 375€.
En paramétrant notre modèle de tarification de référence à partir des données d'apprentissage anonymisées obtenues puis en évaluant les données de test en ligne à ligne à partir de cette nouvelle paramétrisation on obtient les résultats suivants (41) :

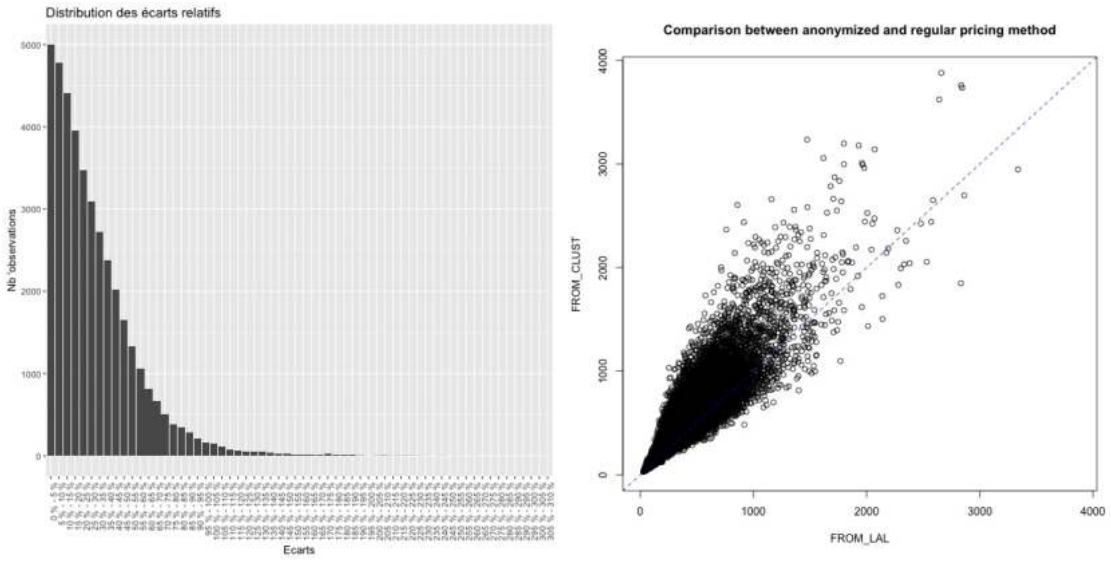


FIGURE 41 : Distribution des écarts relatifs & comparaison des modèles

Plus d'informations sur l'élaboration de ces figures à l'introduction de la partie (4.3.1)

TABLE 23 : Grandeurs caractéristiques (AP_100)

SINISTRALITÉ TOTALE BENCHMARK	+1.266E7
SINISTRALITÉ TOTALE ANONYMISÉE	+1.573E7
SOMME QUADRATIQUE DES ÉCARTS	+1.398E11
ÉCART (ANONYMISÉ - BENCHMARK)	+3.068E6
ÉCART RELATIF (À BENCHMARK)	+24.23 %

Nous allons désormais apporter dans la sous-partie suivante une comparaison des résultats présentés ici ainsi qu'une analyse approfondie du processus de tarification anonymisé ainsi établi.

4.3.2 Analyse & comparaison des modèles d'anonymisation

4.3.2.1 Approche théorique

Tout d'abord, afin d'éviter de tirer une conclusion trop rapide sur les différentes méthodes d'anonymisation employées afin d'obtenir un tarif automobile dans un cadre anonymisé il est important de garder à l'esprit qu'il n'existe pas un unique critère pour définir la qualité d'une telle méthode et qu'il faut dès lors considérer l'ensemble des données quantitatives précédemment présentées afin d'analyser rigoureusement ces résultats.

Algorithme	Paramétrage	Sinistralité tot. Anonymisée	Σ quadratique écarts	Écart (anonymisée − benchmark)	Écart relatif (à benchmark)
K-Means	K=6000	+1.324e7	+1.391e11	+5.779e5	+4.56 %
	K=4500	+1.325e7	+1.392e11	+5.879e5	+4.64 %
	K=3000	+1.388e7	+1.390e11	+1.218e6	+9.62 %
	K=1200	+1.592e7	+1.399e11	+3.258e6	+25.73 %
Hclust	Ward	+1.389e7	+1.394e11	+1.228e6	+9.7 %
OPTICS	(2,2,120)	+1.445e7	+1.391e11	+1.788e6	+14.12 %
	(2,3,80)	+1.407e7	+1.389e11	+1.408e6	+11.12 %
AP	Bloc de 1000	+1.544e7	+1.398e11	+2.778e6	+21.94 %
	Bloc de 500	+1.594e7	+1.401e11	+3.278e6	+25.89 %
	Bloc de 250	+1.521e7	+1.398e11	+2.548e6	+20.12 %
	Bloc de 100	+1.573e7	+1.398e11	+3.068e6	+24.23 %

Ecart quadratique minimal

Écart absolu et relatif minimal

FIGURE 42 : Tableau de synthèse des méthodes d'anonymisation

Nous attacherons ainsi une importance particulière à la distribution des écarts relatifs à l'issue de l'anonymisation ainsi qu'à la forme du nuage de points comparant le modèle de référence aux modèles anonymisés ainsi qu'à l'écart relatif total observé (en % par rapport au modèle benchmark).
Enfin, comme mentionné durant la présentation de la méthodologie globale (4.1.1), notre étude portant avant tout sur l'observation et l'analyse de la différence entre un modèle de référence et son alternative anonymisée, nous rappelons que les montants de sinistralité prédits par les procédures de tarification anonymisées employées ne sont comparées qu'avec ceux issus du modèle de référence.

Pour commencer, même si l'on constate facilement à la vue des résultats présentés précédemment de grandes disparités dans la qualité des modèles d'anonymisation employés, une caractéristique encourageante demeure commune à l'ensemble de ces procédures.
En effet chacune d'entre elles, alors qu'elles retournent des résultats naturellement différents du modèle de référence, s'avère surévaluer le montant de la sinistralité totale et donc en moyenne des primes pures calculées. Un tel comportement est plutôt rassurant dans un cadre prudentiel puisque l'on obtient avec nos modèles anonymisés des montants de sinistres prévisionnels supérieurs en moyenne aux frais modélisés par le modèle ligne à ligne classique, ce qui pour l'assureur est une erreur davantage acceptable (tant que l'écart reste raisonnable) qu'une erreur risquant de sous-estimer la sinistralité du portefeuille. Ce dernier cas de figure pouvant placer l'assureur en position de défaut très rapidement en engendrant des provisions insuffisantes pour faire face aux frais réellement engagés.
Alors que l'erreur supplémentaire engendrée par l'anonymisation de nos données au cours de notre procédure de tarification peut être vue comme une marge de sécurité supplémentaire pour l'assureur, et ce quelque soit la méthode employée, il demeure nécessaire d'évaluer et de comparer ces écarts, à la fois dans leurs totalités en les sommant (pour avoir une vision globale de cette marge sur l'ensemble du portefeuille) mais également individuellement (afin de mesurer l'erreur introduite dans le calcul des primes individuelles). L'idée étant d'avoir la marge la plus petite possible dans chacun des cas.

Nous pouvons alors distinguer quatre catégories (A, B, C et D) de modèles de tarification anonymisés pour notre jeu de données. Les schémas (43) ci-après soulignent les caractéristiques principales de ces différentes classes au regard de la distribution de leurs écarts relatifs ainsi que du nuage de points issue de leur comparaison au modèle benchmark.
La première d'entre elles (A) correspond aux modèles offrant un écart entre le modèle de référence et la sinistralité ainsi évaluée la plus réduite possible individuellement. Le nuage de points comparant les deux types de modèles étant alors assez proche de la première bissectrice tandis que la courbe représentant la distribution des écarts relatifs présente un plateau assez élevée au départ (près de 21 000 points ayant un écart entre les deux types de modèles compris en 0 et 15%) avant de plonger fortement. Cette forme de courbe souligne ainsi le nombre important d'observations ayant un faible écart entre les modèles.
On retrouve parmi cette classe de modèles ceux basés sur un clustering par estimation de densité OPTICS, et en partitculier la seconde implémentation (OPTICS(2,3,80)) qui présente une légère sur-estimation de la sinistralité individuelle tout en étant plus précis les algorithmes de la classe B sur les sinistres importants. Enfin l'écart global demeure contenu (au mieux $\sim$+11%) ce qui fait de cette classe celle présentant le meilleur compromis avec des écarts individuels les plus faibles et un écart global très mesuré.

La catégorie B regroupe les modèles permettant d'obtenir la marge de sécurité la plus réduite globalement.
On y retrouve l'ensemble des modèles basés sur un k-Means (à l'exception du cas où K=1200) et dans une moindre mesure celui basé sur un clustering hiérarchique. En observant leurs nuages de points on constate que le cône formé est centré sur la première bissectrice mais qu'il est égaelement plus large que ceux de la classe A. De plus, on peut aussi percevoir une sous-estimation quasi-systématique des sinitres les plus importants. Concernant la distribution des écarts relatifs entre les modèles, la courbe est très élevée au départ (environ 6 000 points ayant un écart de moins de 5%) mais ne présente pas de plateau comme la classe A avant de drastiquement diminuer. Ainsi, moins d'observations présentent un faible écart individuellement.
L'écart global de ces méthodes est en revanche le plus petit, on obtient par exemple avec la méthode k-Means où K=6000 un écart global de seulement +4.56% par rapport au modèle de référence. Cependant on explique principalement ce bon score par une compensation entre les sur-estimations et les sous-estimations des différents sinistres. En effet en observant la somme quadratique des écarts, nous pouvons constater qu'ils sont dans l'ensemble bien supérieurs à ceux obtenus par les méthodes de la catégorie A.

On retrouve au sein de la classe C l'ensemble des méthodes offrant une marge de sécurité élevée, tant du point de vue individuel que du point de vue global.
Parmi les procédures appartenant à cette catégorie on retrouve les méthodes basées sur le principe de propagation d'affinité, à l'exception de l'implémentation par blocs de 1000 observations (AP_1000) traité à part, ainsi que celle issue du k-Means où

K=1200.
On remarque que le nuage de points comparant ces modèles au benchmark forme un cône plus ou moins large et assez nettement décalé sur la gauche de la première bissectrice, signe que ces méthodes engendrent effectivement une sur-estimation de la sinistralité individuelle, et ce même pour les sinistres les plus importants.
Concernant le k-Means avec K=1200 en particulier, on constate que le cône du nuage de points est très large et que la distribution des écarts relatifs n'est pas satisfaisante (faible pic pour les écarts entre 0 et 15%) et très longue queue à droite, ainsi l'écart global obtenu est lui aussi très élevé (près de +26%). Néanmoins nous n'éliminons pas cette méthode car l'on observe bien sur le nuage de points que même si les écarts individuels sont très élevés, ils demeurent dans une écrasante majorité strictement positif par rapport au modèle de référence.
Ainsi ces méthodes permettent d'obtenir une importante marge de sécurité dans l'estimation de la sinistralité du portefeuille, à la fois globalement mais également du point de vue individuel. De telles procédures peuvent s'avérer utile si l'on considère, par exemple, que le modèle ligne à ligne initial n'est pas assez prudent.

Enfin la dernière catégorie D correspond au modèle éliminé, à savoir celui issu de la propagation d'affinité par blocs de 1000 observations.
Le principal reproche que l'on peut faire à cette méthode réside notamment dans le pic décalé de sa distribution des écarts relatifs, là où toutes les autres méthodes parviennent à avoir leurs pics aux alentours de 0-15%, ici le pic (à un niveau par ailleurs faible) se situe vers 5-25%. Cela souligne ainsi l'incapacité de ce modèle à produire des résultats proches du modèle de référence.
Par ailleurs le nuage de points forme un cône assez large et relativement centré sur la première bissectrice ce qui confirme que les écarts individuels sont importants mais que le modèle ne fournit pas une sur-estimation de la sinistralité individuelle et donc ne peut pas faire partie de la classe C.
Alors que l'écart global est important (∼+22%), il est tout de même amoindri par les compensations des sur-estimations et des sous-estimations des différentes observations, il s'agit donc à ce titre d'un modèle à éviter dans notre cadre.

On présente ci-dessous deux schémas synthétisant les caractéristiques principales de ces quatre catégories concernant la distribution de leurs écarts relatifs ainsi que le nuage de points issue de leur comparaison au modèle benchmark (43).

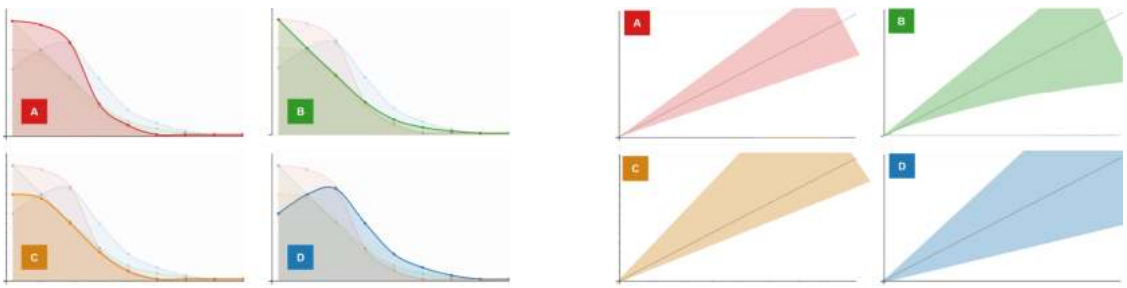


FIGURE 43 : Schéma des distributions des écarts relatifs & des nuages de points en fonction des catégories des modèles anonymisés

Pour conclure sur le choix de la méthode d'anonymisation, il nous semble évident de privilégier les méthodes de la classe A (et en particulier l'implémentation OPTICS(2,3,80)) qui offrent le meilleur compromis avec ce modèle de tarification et notre jeu de données.
Toutefois les modèles des classes B et C peuvent demeurer utiles dans des cas précis (minimisation des écarts individuels, dégager une marge de sécurité plus importante, etc.) D'autant plus que les algorithmes de clustering k-Means issus de la catégorie B sont biens plus simples à paramétrer que les algorithmes d'estimation de densité de la classe A.

4.3.2.2 Approche métier selon l'influence de modalités

Cependant les analyses présentées jusqu'ici ont une portée globale, dans le but avant tout de discerner facilement et rapidement une méthode d'anonymisation efficace. Il serait également intéressant de pouvoir analyser et quantifier l'impact de certaines modalités des variables sur l'écart de sinistralité estimée par le modèle benchmark et le modèle anonymisé. Notamment afin de pouvoir appliquer d'éventuelles corrections ex-post aux estimations du modèle anonymisé.

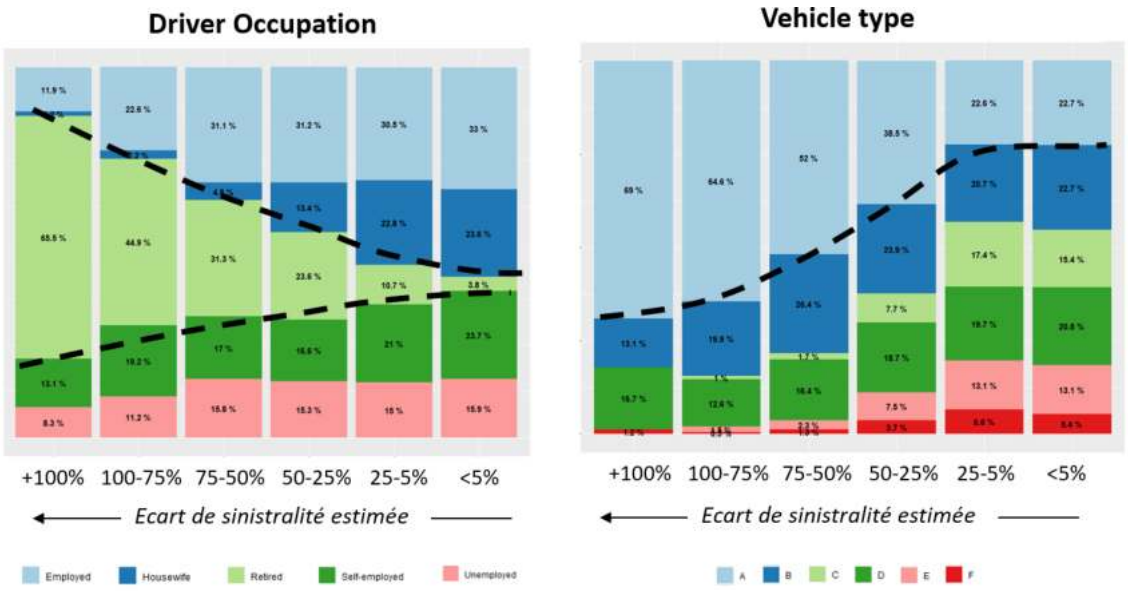


FIGURE 44 : Analyse de la déviation de l'estimation de la sinistralité en fonction des modalités des variables *Driver Occupation* et *Vehicle Type*

Ainsi au regard de la figure (44), traçant les distributions des écarts de sinistralité obtenus entre le modèle anonymisé [70] et le modèle benchmark sur le jeu de données test en fonction des modalités des variables *Driver Occupation* et *Vehicle Type*, nous pouvons facilement déterminés sur quels profils d'assurés appliquer ces corrections, ainsi que leurs niveaux, afin de rapprocher au maximum le modèle anonymisé du comportement du modèle benchmark.

70. Anonymisation OPTICS(2,3,80)

En l'occurrence, on remarque que les assurés en retraites ou ceux conduisant des véhicules de type A ont plus de chances de voir leurs primes sur-évaluées au travers du modèle anonymisé par rapport à celui de référence. On peut donc ajuster globalement à la baisse la sinistralité estimée en sortie du modèle anonymisé pour ces profils, le niveau de la correction pouvant être obtenu itérativement de sorte, à minima, à effacer l'effet ciseau (pour *Driver Occupation*) et la cassure (pour *Vehicle Type*) soulignés par les courbes noires sur la figure (44).

4.3.3 Limites de notre procédure de tarification anonymisée

Il existe plusieurs limitations à la méthode de tarification que nous avons proposé. Tout d'abord les résultats présentés précédemment sont issus d'un type de modèle de tarification basé sur des méthodes linéaires (GLM) afin de modéliser les coûts moyens des sinistres ainsi que leurs fréquences. Cependant d'autres méthodes statistiques peuvent être employées à ces fins, notamment des algorithmes reposant sur la construction d'arbres de décisions tels que le CART ou le random forest, ainsi que le gradient boosting. Nous avons alors essayé d'appliquer notre procédure de tarification anonymisée sur un nouveau modèle de référence dont la sinistralité et la fréquence sont cette fois modélisés à partir de forêts d'arbres de décisions (RF), et également sur un second modèle basé sur des gradient boosting (GD).
Or les résultats obtenus à l'issue de notre tarification anonymisée sur ces deux modèles se sont avérés peu satisfaisants, et ce quelque soit la méthode d'anonymisation employée. En effet les écarts relatifs observés étaient particulièrement importants tandis que le nuage de points comparant les modèles anonymisés à leurs modèles de références semblait totalement décorrélé de la première bissectrice mais particulièrement dense.
Nous en avons ainsi déduit qu'il est impossible de recourir à notre procédure d'anonymisation sur de tels modèles tant les résultats obtenus sont éloignés de ceux du benchmark. Il nous semble que ces mauvais résultats peuvent s'expliquer par la nature même des méthodes de boosting et de bagging alors employés lors de la tarification, qui tendent à diminuer la variance des prédictions au détriment de son biais. De plus lorsque des arbres de décisions sont formés sur des pseudo-observations (représentant l'individu moyen d'un cluster) la nature du sur-apprentissage inhérent à ces méthodes conduit à une forme de "sur-moyennisation" des modèles qui débouche, une fois appliqués aux données de test ligne à ligne, à des résultats très éloignés du modèle de référence.
Il semble donc que la méthode de tarification anonymisée que nous proposons ici ne puisse s'appliquer efficacement qu'à des procédures reposant sur des modèles linéaires, heureusement le recours à ces méthodes sont courantes dans un cadre de tarification non-vie.

Cependant nous continuons de travailler au support des méthodes non-linéaires, basées notamment sur des arbres de décisions. Ainsi nous avons élaborées une méthodologie d'anonymisation alternative pour ces méthodes de tarifications.
Comme le montre le schéma de la figure (45), cette méthodologie alternative reprend le même principe que celle détaillée auparavant dans ce mémoire, si ce n'est qu'ici l'objectif est de préserver davantage de variance dans le jeu de données une fois

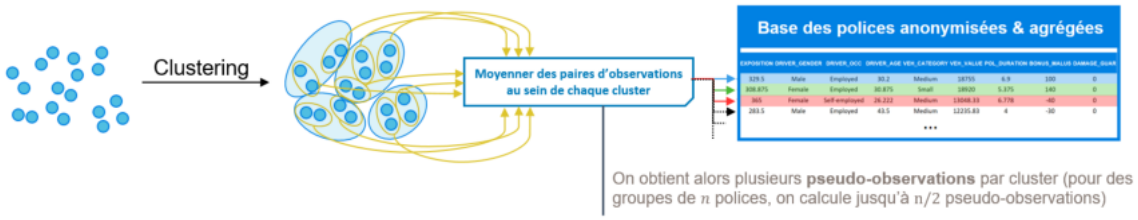


FIGURE 45 : Schéma de la procédure alternative d'anonymisation pour le support de tarifications issues de méthodes non linéaires

anonymisé (qui pour rappel, nous servira de jeu d'apprentissage lors de la phase de tarification), afin de faire face notamment au problème de "sur-moyennisation" évoqué au précédent paragraphe.
Pour ce faire, au lieu de calculer une pseudo-observation par cluster formé, on choisit de former des couples d'observations au sein de chaque cluster et c'est à partir de ces couples[71] que seront calculés les individus moyens formant le jeu de données anonymisé. On obtient alors - à la différence de la méthodologie initiale - plusieurs pseudo-observations par clusters, et leurs nombres est par ailleurs proportionnel à la taille de ce dernier. Ainsi pour un portefeuille initial de N polices scindé en k clusters de n observations, on peut obtenir jusqu'à $N/2(:= kn/2)$ pseudo-observations, la taille de la base de données une fois anonymisée étant donc sensiblement plus importante qu'avec la précédente méthodologie.
Alors qu'avec la méthodologie initiale l'écart de sinistralité estimée entre le modèle de référence et celui anonymisé pour un modèle de tarification basé sur un CART avoisinnait les 100%, en utilisant cette méthodologie nous sommes parvenus à fortement la réduire à hauteur d'environ 30%. Même s'il s'agit d'un écart encore trop important pour pouvoir s'en satisfaire en pratique, il en demeure un progrès notable qui nous permet d'envisager d'autres améliorations par la suite.
A noter par ailleurs que cette méthodologie alternative n'est pas réservée aux modèle de tarifications non-linéaires, cependant nous n'avons pas noter de gains substantiel en l'employant sur des modèles linéaires alors que la base de données anonymisée obtenue est tout de même bien plus volumineuse.

Finalement, même si nous avons pu analyser similairement chacun des modèles de tarification des différentes catégories (A, B, C et D) en faisant varier la taille et la nature des échantillons de test et d'apprentissage ainsi que les fonctions de liens des GLM (poisson, gaussienne, etc.), nous n'avons cependant pas eu l'occasion de tester ces procédures sur d'autres modèles de tarification linéaires à partir de jeu de données différents.
Ainsi les analyses des modèles présentées ici sont liées à notre cadre d'étude et sont donc susceptibles de varier une fois appliqués à d'autres base de données.

71. Si un cluster comprend un nombre d'observations impair, il est entendu que pour conserver l'anonymat des données obtenues in fine, la donnée restante sera regroupée avant le calcul des pseudo-observations à l'un des couples précédemment formés.

5 Conclusion

Nous avons étudié au cours de ce mémoire plusieurs procédures permettant de résoudre certaines difficultés résultant de l'introduction de la nouvelle réglementation européenne sur la protection des données (RGPD) dans un cadre actuariel. Et plus particulièrement lors de la réalisation d'un tarif en assurance non-vie.
Ainsi nous nous sommes plus particulièrement penchés sur la possibilité de proposer des méthodologies de tarification alternatives assurant la sécurisation des données personnelles des assurés et la garantie du maintien de leurs caractères confidentiels en tous points de ce processus.
En différenciant dans notre étude les concepts, mentionnés dans la réglementation, de données pseudonymisées de celles anonymisées nous avons conçu et implémenté deux procédures permettant d'utiliser certains outils mathématiques couramment employés durant les phases de tarification avec des données parfaitement protégées conformément au RGPD.

Le recours aux méthodes de chiffrement homomorphes présentées dans ce mémoire nous a permis de développer une approche sécurisée du *cloud-computing* (calculs délégués). Alors que cette nouvelle manière de traiter les données tend de plus en plus à se généraliser chez certains assureurs en raison de l'augmentation toujours plus importante du volume de données récoltées à analyser il est, selon la réglementation, du ressort de l'assureur de garantir l'intégrité de ses données.
Nous avons ainsi démontré la possibilité de faire réaliser une régression linéaire sur un jeu de données cryptées (pseudonymisées) par un prestataire externe de calculs, sans que ce dernier n'ait à aucun moment ni le moyen d'accéder aux informations contenues dans ces données ni au résultat du calcul exécuté.
Bien entendu la procédure proposée ici s'avèrent assez limitée et requiert pourtant d'importantes ressources de calculs (pour le prestataire). Néanmoins les concepts utilisés dans cette procédure s'avère en constante évolution, tant du point de vue informatique que mathématique, et il ne semble donc pas impossible d'envisager demain l'emploi de ce type de méthodologie pour sécuriser le transfert, le stockage et l'analyse de données sur des serveurs extérieurs dédiés.

Cependant seul le recours à l'anonymisation des données permet de se soustraire à la plupart des contraintes du RGPD. C'est pourquoi nous avons également proposé d'analyser les écarts entre un modèle de tarification automobile classique coût/fréquence en ligne à ligne et ce même modèle entraîné à partir de données anonymisées. L'originalité de la méthode résidait alors dans la manière d'anonymiser le portefeuille d'assurés. En effet, contrairement aux procédures classiques en la matière (k-anonymisation, fonctions bijectives sur les variables, etc.) nous n'avons jamais ni modifié ni supprimé de variables (hormis les variables nominatives). En utilisant plusieurs méthodes de clustering, nous avons pu établir plusieurs catégories de méthodes d'anonymisation pour notre procédure de tarification et sommes parvenus à des résultats très encourageants, tant l'écart entre le modèle de référence et le modèle anonymisé s'est avéré mince dans certains cas.
Naturellement les méthodes d'anonymisation ont été appliquées ici à un modèle de

tarification GLM spécifique et nous avons pu par exemple constater que leur application à des modèles différents (par exemple ceux basés sur des arbres de décisions) engendraient des résultats peu convaincants. Par ailleurs le choix des algorithmes de clustering optimaux peut dépendre du jeu de données et leur utilisation peut nécessiter une paramétrisation particulièrement difficile à réaliser.

Finalement les concepts abordés dans cette étude ainsi que les méthodes proposées poursuivent l'objectif de mise en conformité de certaines procédures actuarielles classiques avec le cadre réglementaire européen du RGPD. Elles trouvent également leur place dans l'émergence de nouveaux besoins, notamment au travers de la gestion du risque cyber qui fait actuellement l'objet d'une attention particulière de la part d'un grand nombre d'assureurs, tant dans l'élaboration de nouveaux produits et services que dans l'évaluation de leurs propres risques (opérationnels, activités déléguées, réputation, etc.).

Annexes

Annexe A

Quelques exemples de chiffrements homomorphes remarquables

Ici nous présenterons brièvement quelques méthodes de chiffrement homomorphes parmi les plus connues.

Le chiffrement RSA

Dans le chiffrement RSA, il est nécessaire de suivre une procédure particulière afin de générer les clefs privées et publiques de la méthode :

1. Choisir deux nombres premiers p et q strictement distincts
2. Calculer leur module de chiffrement, soit $N = pq$.
3. Calculer $\phi(N) = (p-1)(q-1)$
4. Déterminer l'exposant de chiffrement, soit un entier e premier avec $\phi(N)$ et strictement inférieur à $\phi(N)$.
5. Calculer l'exposant de déchiffrement d, tel que l'entier d soit l'inverse de $e \pmod{\phi(N)}$, où $d < \phi(N)$.

On définit ainsi d comme étant la clef secrète de déchiffrement (sk) et le couple (N, e) la clef publique pour le chiffrement (pk), tel que :

$$\begin{aligned} \mathrm{Enc}((N,e),x) &:= x^e \pmod{N} \\ \mathrm{Dec}((N,d),y) &:= y^d \pmod{N} \end{aligned}$$

Par soucis de clarté on note dans la suite $\mathcal{E}(x) = \mathrm{Enc}((N,e),x)$, de plus en restreignant l'espace des données non cryptées tel que :

$$0 \leqslant x \leqslant \sqrt{N}$$

On constate la propriété d'homomorphisme multiplicatif suivante :

$$\begin{aligned}\mathcal{E}(x_1) \cdot \mathcal{E}(x_2) &= x_1^e \cdot x_2^e \pmod N \\ &= (x_1 \cdot x_2)^e \pmod N \\ &= \mathcal{E}(x_1 \cdot x_2)\end{aligned}$$

Le système de chiffrement RSA est donc un chiffrement partiellement homomorphe puisque seul l'opérateur multiplicatif présente cette propriété [1].

Le chiffrement Paillier

Afin de simplifier la démarche de génération des clefs, considérons les deux entiers primaires p et q de même tailles, ainsi on peut se replacer dans le cadre de génération de clef RSA précédemment présenté, tel qu'ici on obtienne $g = N + 1$, $\lambda = \phi(N)$ et $\mu = (\phi(N))^{-1} \pmod N$.
On définit alors la clef publique (pk) par le couple (N, g) et la clef privée (sk) comme le couple (λ, μ), on a alors $\forall x \in [\![0, N]\!]$:

$$\begin{aligned}\mathrm{Enc}((N, g), x) &:= g^x \cdot r^N \pmod{N^2} \\ \mathrm{Dec}((\lambda, \mu), y) &:= \frac{(y^\lambda \pmod{N^2} - 1) \cdot \mu}{N} \pmod N\end{aligned}$$

où r est un entier aléatoire, tel que $0 < r < N$

Par soucis de clarté on note dans la suite $\mathcal{E}(x) = \mathrm{Enc}((N, g), x)$, par ailleurs on constate la propriété d'homomorphisme additif suivante :

$$\begin{aligned}\mathcal{E}(x_1) \cdot \mathcal{E}(x_2) &= (g^{x_1} r_1^N) \cdot (g^{x_2} r_2^N) \pmod{N^2} \\ &= g^{x_1 + x_2} (r_1 r_2)^N \pmod{N^2} \\ &= \mathcal{E}(x_1 + x_2)\end{aligned}$$

Nous pouvons également noter une propriété particulière du chiffrement Paillier qui propose notamment un homomorphisme multiplicatif singulier entre un élément crypté et un élément non-crypté tel que :

$$\begin{aligned}(\mathcal{E}(x_1))^{x_2} &= [(g^{x_1} r_1^N)) \pmod{N^2}]^{x_2} \\ &= g^{x_1 \cdot x_2} (r_1)^{N \cdot x_2} \pmod{N^2} \\ &= \mathcal{E}(x_1 \cdot x_2)\end{aligned}$$

Cette propriété signifie concrètement qu'il est possible dans un système de chiffrement de Paillier d'obtenir la multiplication de deux éléments dans l'espace non crypté (x_1 et x_2) à condition d'élever à la puissance x_2 (valeur non-cryptée) le message crypté $\mathcal{E}(x_1)$ de x_1. Dans la pratique cela revient donc à vouloir multiplier un élément crypté par un élément non crypté.

1. Dans ce cas les opérateurs sont identiques dans l'espace crypté et dans l'espace non crypté, soit le couple $(\times, *^1) \Leftrightarrow (\times, \times)$.

Néanmoins le système de chiffrement Paillier s'avère seulement être un chiffrement partiellement homomorphe puisque seul l'opérateur additif présente cette propriété [2].

2. Dans ce cas le couple d'opérateur est tel que $(+, *^1) \Leftrightarrow (+, \times)$.

Annexe B

Mesure de Lebesgue & matrices aléatoires inversibles

1 Rappel sur la mesure de Lebesgue

1.1 Définition

Définition 1 (La mesure extérieure de Lebesgue)**.**

Soit un sous-ensemble $E \subseteq \mathbb{R}$, où la taille d'un intervalle $I = [a, b]$ est donné par $l(I)$, la mesure extérieure de Lebesgue $\lambda^*(E)$ est défini par :

$$\lambda^*(E) = \inf \left\{ \sum_{k=1}^{\infty} l(I_k) \right\}$$

avec : $(I_k)_{k \in \mathbb{N}}$ est une suite d'intervalles ouverts tels que :

$$E \subseteq \bigcup_{k=1}^{\infty} I_k$$

1.2 Propriétés et propositions

Propriété 1 (sur $\mathbb{R}^n$)**.**

1. *Si A est un produit cartésien d'intervalles $I_1 \times I_2 \times \cdots \times I_n$, alors A est Lebesgue-mesurable et $\lambda(A) =\mid I_1 \mid \times \mid I_2 \mid \times \cdots \times \mid I_n \mid$.*
2. *Si A est la réunion disjointe d'une quasi-infinité d'ensembles disjoints Lebesgue-mesurables, alors A est elle-même Lebesgue-mesurable et $\lambda(A)$ est égal à la somme (ou à la série) des mesures des ensembles mesurables en question.*
3. *Si A est Lebesgue-mesurable, alors son complémentaire l'est également.*
4. *$\lambda(A) \geqslant 0$ pour tout ensemble A Lebesgue-mesurable.*
5. *Si A et B sont Lebesgue-mesurables et A est un sous-ensemble de B, alors $\lambda(A) \leqslant \lambda(B)$.*

Propriété 2 (σ-algèbre ou tribu)**.**
Les ensembles Lebesgue-mesurables forment une σ-algèbre qui contient tout les produits des intervalles, et λ est la seule mesure complète et invariante par translation sur cette σ-algèbre tel que $\lambda([0,1] \times [0,1] \times \cdots \times [0,1]) = 1$.

Sur l'ensemble des réels $\mathbb{R}$, la mesure de Lebesgue est σ-finie.

Définition 2 (σ-finite)**.**
Soit (X,Σ, μ) un espace mesuré, la mesure μ est dite σ-finie si X est une union dénombrable d'ensembles mesurables et de mesures finies.

On considère par la suite que la notion de mesure réfère systématiquement à la mesure de Lebesgue.

Propriété 3.
Tous les ensembles dénombrables sont des ensembles négligeables (de mesure nulle). Dès lors, si X est un sous-ensemble mesurable de $\mathbb{R}^n$, $\lambda(X) = 0$.

Remarque : Un unique point a une mesure nulle. Un ensemble dénombrable est l'union dénombrable de ses points, et puisque la mesure est σ-additive, on obtient bien que la mesure de cet ensemble est égal à la somme des mesures (nulles) de ses points.

2 Génération de matrices aléatoires inversibles

2.1 Description

Une matrice carrée A n'est pas inversible si et seulement si son déterminant est nul. Le déterminant d'une matrice étant un polynôme selon ses éléments, déterminer la probabilité qu'une matrice carrée générée aléatoirement ne soit pas inversible revient à évaluer la probabilité qu'une matrice A, que l'on peut considérer comme un point dans $\mathbb{R}^{n^2}$, se trouve dans l'espace d'annulation du polynôme.

2.2 L'espace d'annulation d'un polynôme

Selon les travaux de R. Caron & T. Traynor : *The Zero set of a polynomial*, University of Windsor
Théorème 1 (Espace d'annulation d'un polynôme)**.**
Une fonction polynômiale de $\mathbb{R}^n$ vers $\mathbb{R}$, est soit strictement nulle, ou strictement non-nulle presque partout.

Preuve par récurrence sur n :
Soit λ_n la mesure de Lebesgue sur n dimensions.

Cas $p : \mathbb{R} \to \mathbb{R}$:

Soit p un polynôme de degré m, $(\forall m > 0)$ différent du polynôme nul trivial. Alors p a au plus m racines, donc : $\underline{\lambda_1\{x : p(x) = 0\} = 0}$.

Cas $p : \mathbb{R}^n \to \mathbb{R}$:

Tout d'abord, nous supposons que le résultat précédent est vérifié pour $p : \mathbb{R}^{n-1} \to \mathbb{R}$, tel que $\forall x \in \mathbb{R}^{n-1}, x = (x_1, x_2, \ldots, x_{n-1})$:

$$\lambda_{n-1}\{x : p(x) = 0\} = 0$$

Désormais, on pose $p : \mathbb{R}^n \to \mathbb{R}$ comme :

$$\begin{aligned} p(x) = p(x_1, x_2, \ldots, x_n) = & \sum_{k=(k_1,k_2,\ldots,k_n)} a_k x^k \\ = & \sum_{k_1,k_2,\ldots,k_n} a_{k_1 k_2 \ldots k_n} x_1^{k_1} x_2^{k_2} \ldots x_n^{k_n} \end{aligned}$$

En définissant k comme $k = (i, j)$, avec $i = (k_1)$ et $j = (k_2, k_3, \ldots, k_n)$, nous pouvons alors scinder $\mathbb{R}^n$ tel que $\mathbb{R} \times \mathbb{R}^{n-1}$:

$$p(x) = p(x_1, \bar{x_2}) = \sum_j q_j(x_1) \bar{x_2}^j$$

où $q_j(x_1) = \sum_i a_{ij} x_1^i$.
Puisque p est différent du polynôme nul dans $\mathbb{R}^n$, nous pouvons alors en déduire qu'un $j = (k_2, k_3, \ldots, k_n)$ existe tel que q_j n'est pas strictement nul et donc que $\{x_1, q_j(x_1) = 0\}$ est fini et de mesure nulle.
De plus, $N := \{x_1 : p(x_1, \bar{x_2}), \forall \bar{x_2}\}$ est également fini et donc de mesure nulle.

Pour tout x_1 fixé $\notin N$, $p(x_1, .)$ est strictement non nul presque partout selon l'hypothèse de récurrence.

On a alors :

$$\begin{aligned} \lambda_n(\{x : p(x) = 0\}) = & \int \lambda_{n-1}(\{\bar{x_2} : p(x_1, \bar{x_2}) = 0\}) dx_1 \\ = & \int_N \lambda_{n-1}(\{\bar{x_2} : p(x_1, \bar{x_2}) = 0\}) + \int_{N^c} 0 dx_1 \\ = & 0 \end{aligned}$$

□

Annexe C

Descente de gradient dans l'espace crypté d'un schéma F&V

On remplace le pas de la descente de gradient ordinaire δ en posant $\delta \equiv 1/\nu$ où $\nu \in \mathbb{N}$, avec $\gamma \in \mathbb{N}$ un paramètre de précision tel que : $\forall x \in \mathbb{R}, \hat{x} = \lfloor 10^{\gamma} x \rceil \in \mathbb{Z}$, afin de pouvoir réaliser des calculs avec des réels au sein de l'espace crypté $\mathcal{X}$.
On rappelle que l'on a pour une descente de gradient classique, avec $X \in \mathbb{R}^{n \times p}$ et $Y \in \mathbb{R}^p$:

$$\beta^{[k]} = \beta^{[k-1]} + \delta X^T (Y - X\beta^{[k-1]})$$

Et on pose alors les transformations suivantes :

- $\tilde{X} = 10^{\gamma} X$
- $\tilde{Y} = 10^{\gamma} Y$
- $\tilde{\nu} = 10^{\gamma} \nu$

On peut en déduire en remplaçant par les termes précédemment définis que la formulation de la descente de gradient dans $\mathcal{X}$ peut s'écrire :

$$\tilde{\beta}^{[k]} \equiv 10^{\gamma} \tilde{\nu} \tilde{\beta}^{[k-1]} \boxplus \tilde{X}^T (10^{k\gamma} \tilde{\nu}^{k-1} \tilde{Y} \boxminus \tilde{X} \tilde{\beta}^{[k-1]}) \tag{C.1}$$

Notre objectif est simplement de montrer que l'on peut également écrire que l'estimation de $\hat{\beta}$ à l'itération k d'un descente de gradient dans l'espace crypté peut s'écrire [1] :

$$\tilde{\beta}^{[k]} = 10^{(2k+1)\gamma} \nu^k \beta^{[k]} \tag{C.2}$$

Pour k=1 :
En prenant comme point d'initialisation pour notre descente de gradient $\beta^{[0]} = 0_n$, on a en partant de (C.1) :

$$\begin{aligned} \tilde{\beta}^{[k]} &= 10^{\gamma} \tilde{X}^T \tilde{Y} \\ &= 10^{3\gamma} X^T Y \end{aligned}$$

1. Les opérations $\boxtimes$ et $\boxdot$ n'ont pas été et ne seront pas explicitées dans la suite des calculs pour plus de lisibilité.

Or à partir de (C.2), on obtient :

$$\begin{aligned}\tilde{\beta}^{[k]} &= 10^{(2k+1)\gamma}\nu^k\beta^{[k]} \\ &= 10^{(2k+1)\gamma}\nu^k\nu^{-1}X^TY \\ \tilde{\beta}^{[1]} &= 10^{3\gamma}X^TY\end{aligned}$$

Donc les équations (C.1) et (C.2) sont bien équivalentes pour $k = 1$.

Pour k quelconque :
En supposant (C.2), et en prenant en compte les transformations précédemment définies, cette fois on peut écrire :

$$\begin{aligned}\tilde{\beta}^{[k]} &= 10^{(2k+1)\gamma}\nu^k\beta^{[k]} \\ &= 10^{(2k+1)\gamma}\nu^k\left[\beta^{[k-1]} \boxplus \nu^{-1}X^T(Y \boxminus X\beta^{[k-1]})\right] \\ &= 10^{(2k+1)\gamma}\nu^k\beta^{[k-1]} \boxplus 10^{(2k+1)\gamma}\nu^{k-1}X^T\left(Y \boxminus X\beta^{[k-1]}\right) \\ &= 10^{(2k+1)\gamma}\nu^k\beta^{[k-1]} \boxplus 10^{(2k+1)\gamma}\nu^{k-1}X^TY \boxminus 10^{(2k+1)\gamma}\nu^{k-1}X^TX\beta^{[k-1]} \\ &= 10^{(2k+1)\gamma}\nu^k\beta^{[k-1]} \boxplus 10^{(2k+1)\gamma}\nu^{k-1}X^TY \boxminus 10^{(2k+1)\gamma}\nu^{k-1}X^TX\beta^{[k-1]} \\ &= 10^{(2k+1)\gamma}\nu^k\beta^{[k-1]} \boxplus 10^{2k\gamma}\nu^{k-1}\tilde{X}^TY \boxminus 10^{(2k-1)\gamma}\nu^{k-1}\tilde{X}^T\tilde{X}\beta^{[k-1]} \\ &= 10^{(2k+1)\gamma}\nu^k\beta^{[k-1]} \boxplus 10^{(2k-1)\gamma}\nu^{k-1}\tilde{X}^T\tilde{Y} \boxminus 10^{(2k-1)\gamma}\nu^{k-1}\tilde{X}^T\tilde{X}\beta^{[k-1]} \\ &= 10^{(k+1)\gamma}\tilde{\nu}^k\beta^{[k-1]} \boxplus 10^{k\gamma}\tilde{\nu}^{k-1}\tilde{X}^T\tilde{Y} \boxminus 10^{k\gamma}\nu^{k-1}\tilde{X}^T\tilde{X}\beta^{[k-1]} \\ &= 10^{\gamma}\tilde{\nu}\tilde{\beta}^{[k-1]} \boxplus 10^{k\gamma}\tilde{\nu}^{k-1}\tilde{X}^T\tilde{Y} \boxminus \tilde{X}^T\tilde{X}\tilde{\beta}^{[k-1]} \\ \tilde{\beta}^{[k]} &= 10^{\gamma}\tilde{\nu}\tilde{\beta}^{[k-1]} \boxplus \tilde{X}^T(10^{k\gamma}\tilde{\nu}^{k-1}\tilde{Y} \boxminus \tilde{X}\tilde{\beta}^{[k-1]})\end{aligned}$$

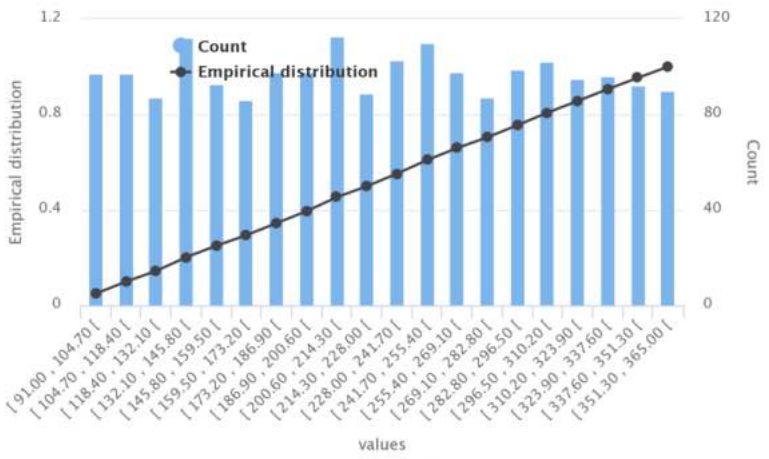
Donc les équations (C.1) et (C.2) sont bien équivalentes pour $\forall k$.

□

Annexe D

Description du jeu de données de tarification RC automobile

EXPOSITION



Quantiles

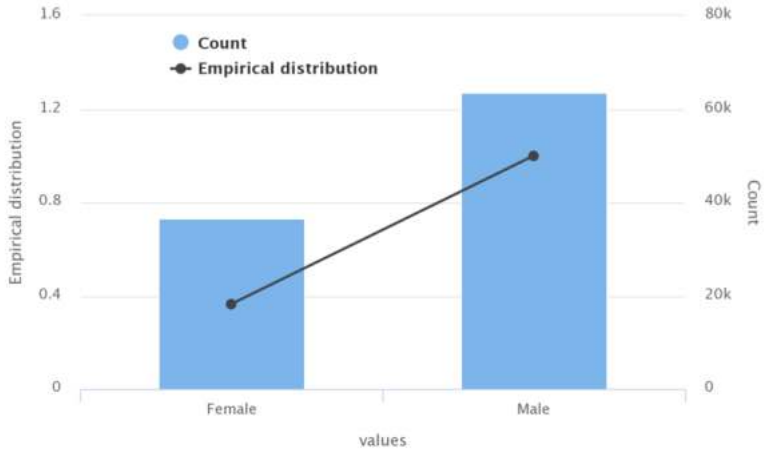
	quantiles	values
0	0.5%	96
1	10.0%	192
2	25.0%	340
3	50.0%	365
4	75.0%	365
5	90.0%	365
6	99.5%	365

Statistics

	statistics	values
0	min	91
1	max	365
2	mean	327.58025
3	stddev	73.57040591150603

Type : numeric
Number of modality : 275
Advise as numeric : true

DRIVER_GENDER



Less frequent

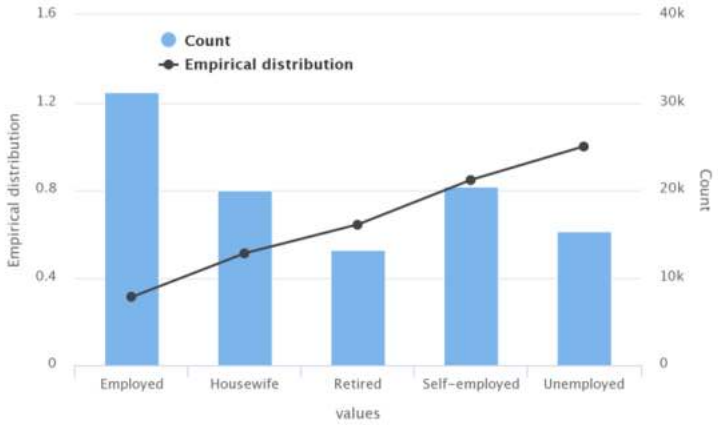
	Values	Count
0	Female	36568

Most frequent

	Values	Count
0	Male	63432

Type : string
Number of modality : 2
Advise as numeric : false

DRIVER_OCC



Less frequent

	Values	Count
0	Retired	13167
1	Unemployed	15315

Most frequent

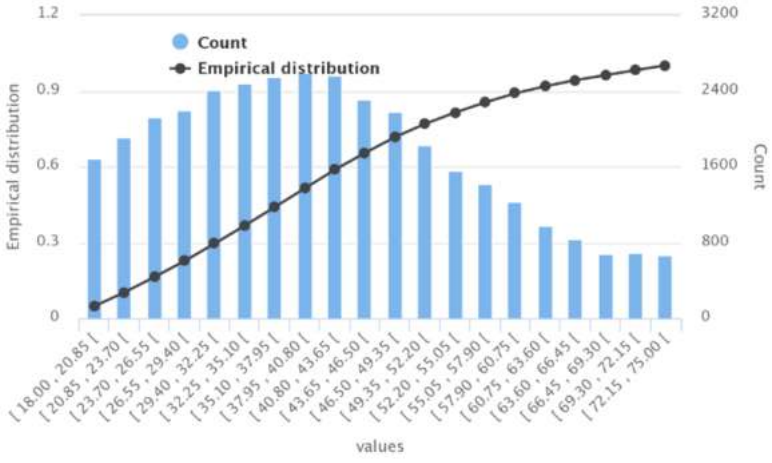
	Values	Count
0	Housewife	20008
1	Self-employed	20371
2	Employed	31139

Type : string

Number of modality : 5

Advise as numeric : false

DRIVER_AGE



Quantiles

	quantiles	values
0	0.5%	18
1	10.0%	23
2	25.0%	30
3	50.0%	40
4	75.0%	51
5	90.0%	62
6	99.5%	75

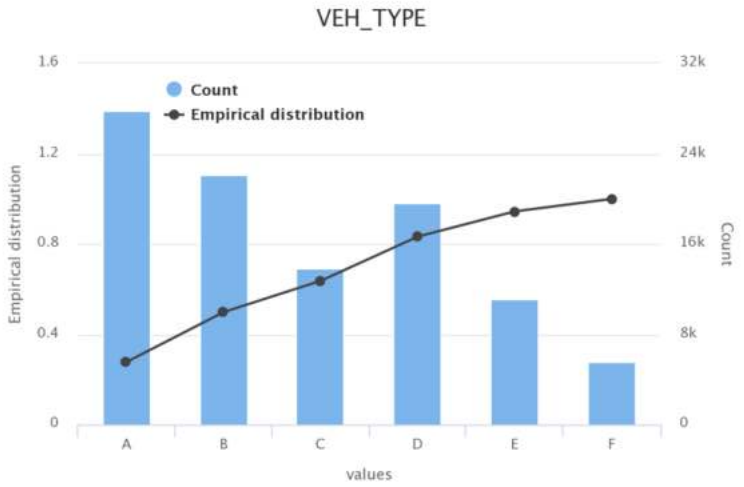
Statistics

	statistics	values
0	min	18
1	max	75
2	mean	41.12506
3	stddev	14.298972851315092

Type : numeric

Number of modality : 58

Advise as numeric : true



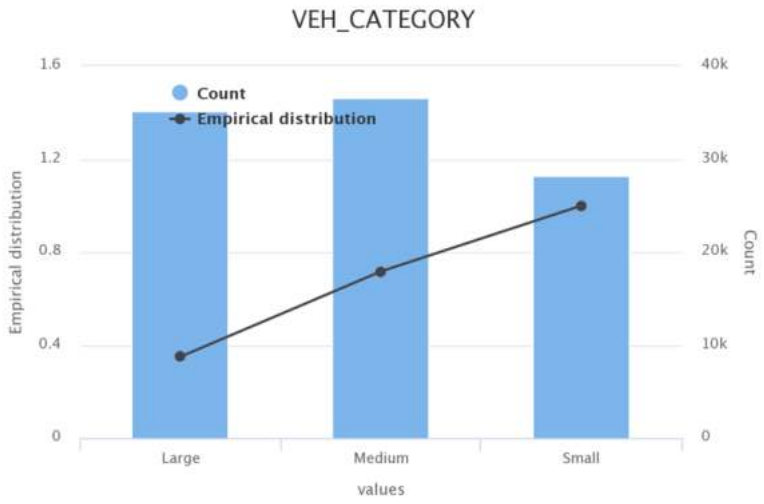
Less frequent

	Values	Count
0	F	5541
1	E	11168
2	C	13858

Most frequent

	Values	Count
0	D	19589
1	B	22088
2	A	27756

Type : string
Number of modality : 6
Advise as numeric : false



Less frequent

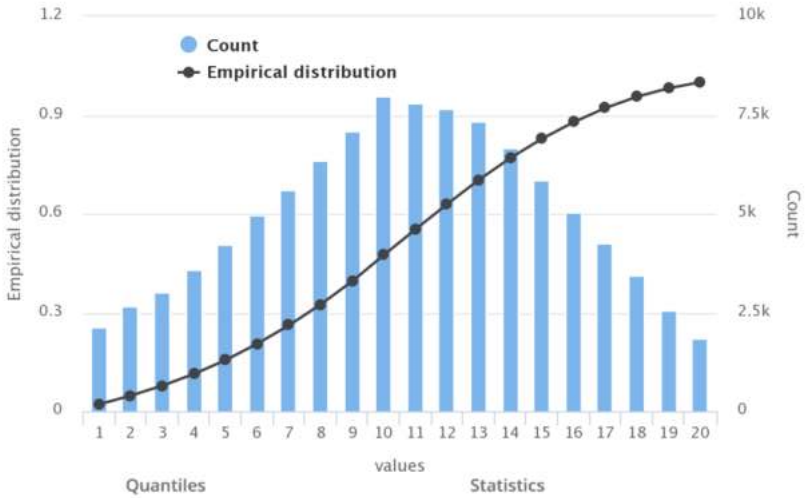
	Values	Count
0	Small	28238

Most frequent

	Values	Count
0	Large	35123
1	Medium	36639

Type : string
Number of modality : 3
Advise as numeric : false

VEH_GROUP



Quantiles

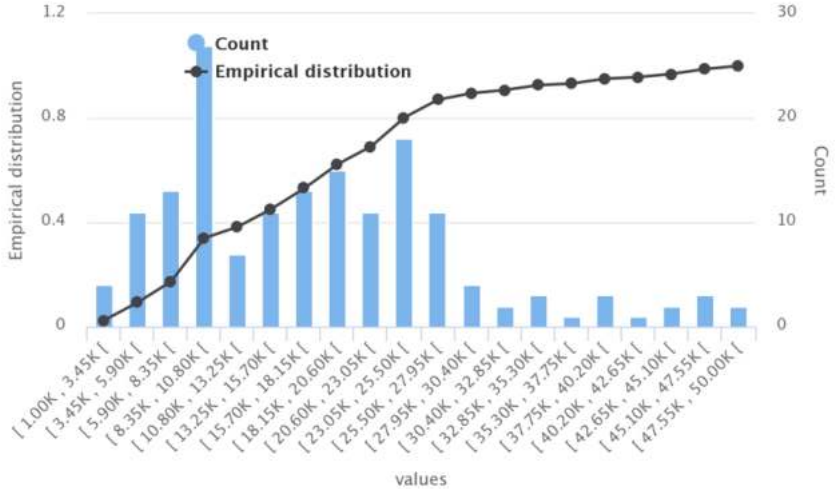
	quantiles	values
0	0.5%	1
1	10.0%	4
2	25.0%	7
3	50.0%	11
4	75.0%	14
5	90.0%	17
6	99.5%	20

Statistics

	statistics	values
0	min	1
1	max	20
2	mean	10.69284
3	stddev	4.687373726205804

Type : numeric
Number of modality : 20
Advise as numeric : true

VEH_VALUE



Quantiles

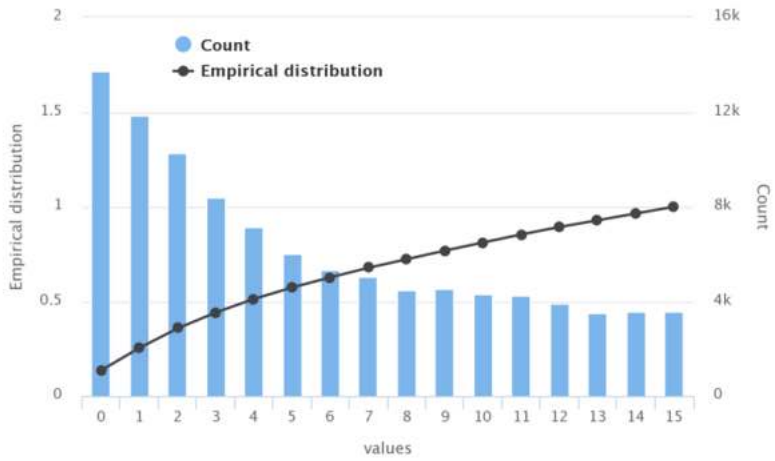
	quantiles	values
0	0.5%	1220
1	10.0%	5130
2	25.0%	8380
3	50.0%	14610
4	75.0%	22575
5	90.0%	29510
6	99.5%	48930

Statistics

	statistics	values
0	min	1000
1	max	49995
2	mean	16454.6194
3	stddev	10507.019806457769

Type : numeric
Number of modality : 939
Advise as numeric : true

POL_DURATION



Quantiles

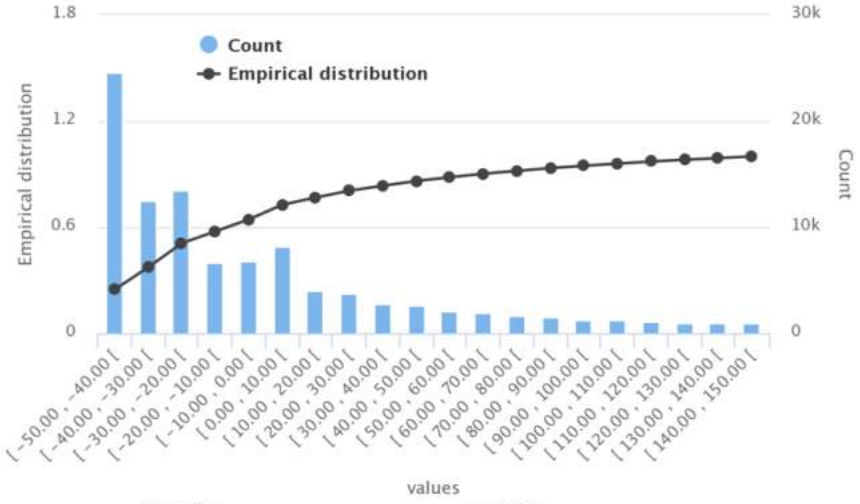
	quantiles	values
0	0.5%	0
1	10.0%	0
2	25.0%	1
3	50.0%	4
4	75.0%	9
5	90.0%	13
6	99.5%	15

Statistics

	statistics	values
0	min	0
1	max	15
2	mean	5.47093
3	stddev	4.591155162065121

Type : numeric
Number of modality : 16
Advise as numeric : true

BONUS_MALUS



Quantiles

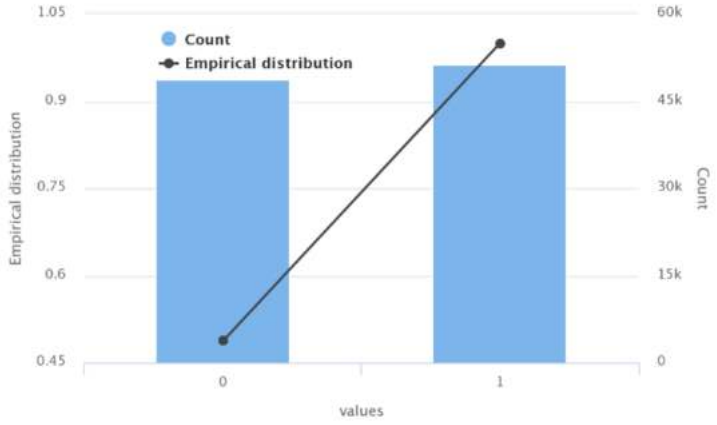
	quantiles	values
0	0.5%	-50
1	10.0%	-50
2	25.0%	-40
3	50.0%	-30
4	75.0%	10
5	90.0%	70
6	99.5%	150

Statistics

	statistics	values
0	min	-50
1	max	150
2	mean	-6.93
3	stddev	48.62794203714018

Type : numeric
Number of modality : 21
Advise as numeric : true

DAMAGE_GUAR



Quantiles

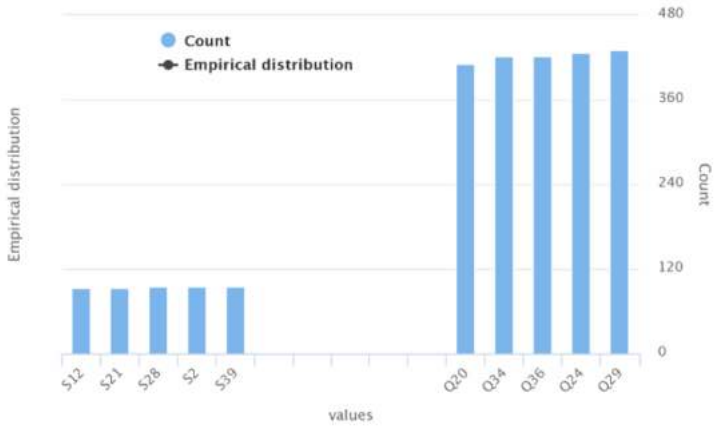
	quantiles	values
0	0.5%	0
1	10.0%	0
2	25.0%	0
3	50.0%	1
4	75.0%	1
5	90.0%	1
6	99.5%	1

Statistics

	statistics	values
0	min	0
1	max	1
2	mean	0.5122
3	stddev	0.49985363711449116

Type : numeric
Number of modality : 2
Advise as numeric : true

COUNTY



Less frequent

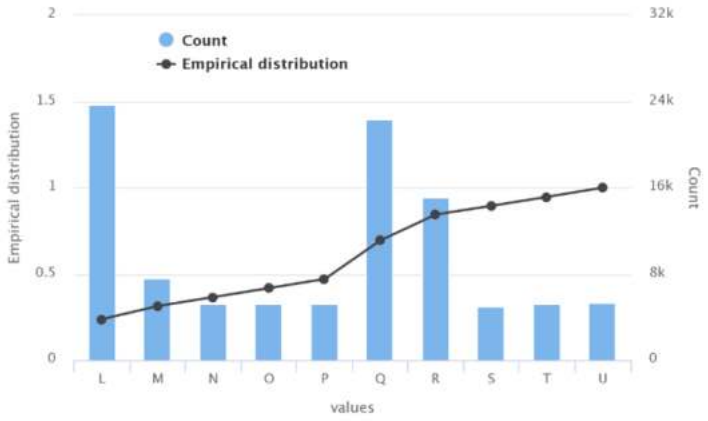
	Values	Count
0	S12	94
1	S21	94
2	S28	95
3	S2	96
4	S39	96

Most frequent

	Values	Count
0	Q20	412
1	Q34	421
2	Q36	422
3	Q24	427
4	Q29	431

Type : string
Number of modality : 471
Advise as numeric : false

STATE



Less frequent

	Values	Count
0	S	4994
1	N	5193
2	T	5195
3	O	5213
4	P	5259

Most frequent

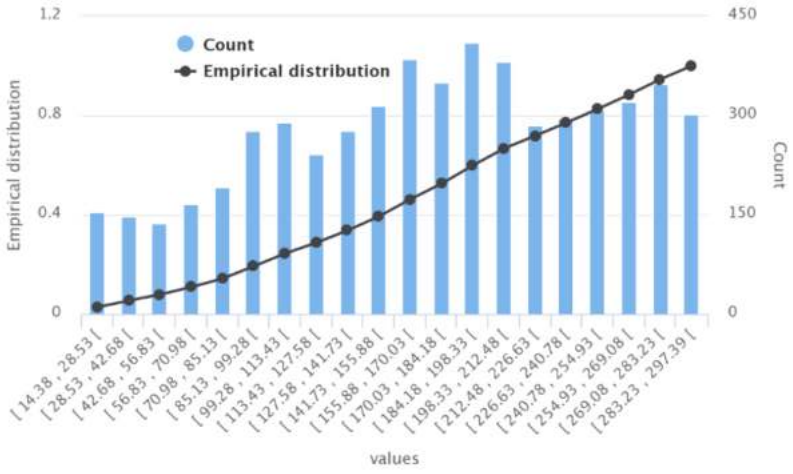
	Values	Count
0	U	5364
1	M	7595
2	R	15076
3	Q	22381
4	L	23730

Type : string

Number of modality : 10

Advise as numeric : false

COUNTY_DENSITY



Quantiles

	quantiles	values
0	0.5%	17.8799582
1	10.0%	30.18943347
2	25.0%	50.6257826
3	50.0%	94.36462339
4	75.0%	174.6445246
5	90.0%	240.0004036
6	99.5%	296.4319078

Statistics

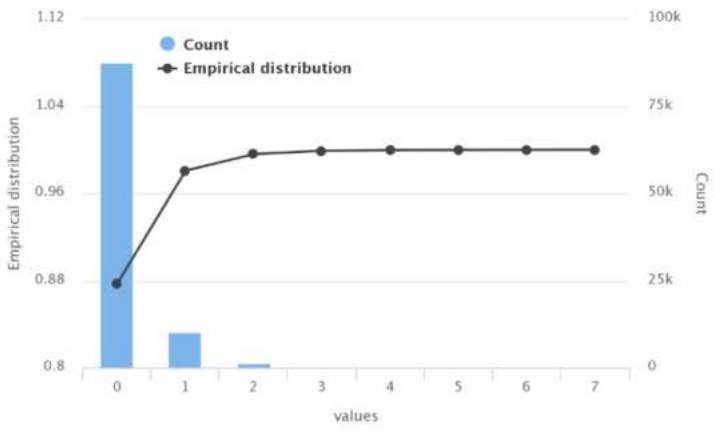
	statistics	values
0	min	14.37714238
1	max	297.3851697
2	mean	117.15688049021138
3	stddev	79.49898804465192

Type : numeric

Number of modality : 471

Advise as numeric : true

NUM_LIAB_DAM



Quantiles

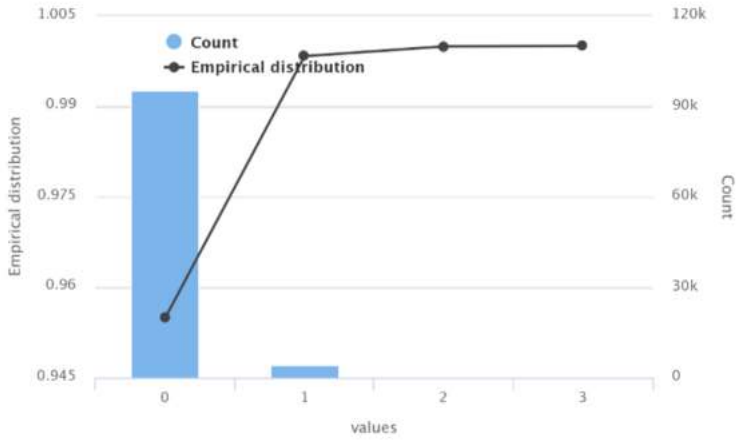
	quantiles	values
0	0.5%	0
1	10.0%	0
2	25.0%	0
3	50.0%	0
4	75.0%	0
5	90.0%	1
6	99.5%	2

Statistics

	statistics	values
0	min	0
1	max	7
2	mean	0.14724
3	stddev	0.4366947325339457

Type : numeric
Number of modality : 8
Advise as numeric : true

NUM_LIAB_BOD



Quantiles

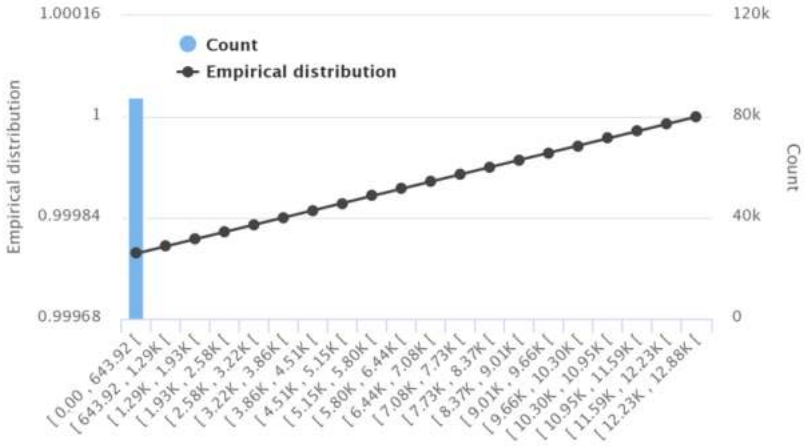
	quantiles	values
0	0.5%	0
1	10.0%	0
2	25.0%	0
3	50.0%	0
4	75.0%	0
5	90.0%	0
6	99.5%	1

Statistics

	statistics	values
0	min	0
1	max	3
2	mean	0.04678
3	stddev	0.21952702230280272

Type : numeric
Number of modality : 4
Advise as numeric : true

INC_LIAB_DAM



Quantiles

	quantiles	values
0	0.5%	0
1	10.0%	0
2	25.0%	0
3	50.0%	0
4	75.0%	0
5	90.0%	160.3307466
6	99.5%	2911.205121

Statistics

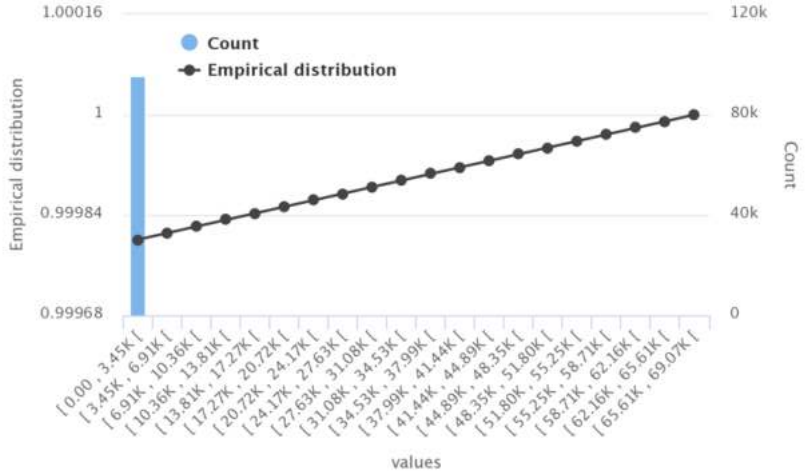
	statistics	values
0	min	0
1	max	12878.36991
2	mean	106.15729505072343
3	stddev	444.9932471326943

Type : numeric

Number of modality : 12257

Advise as numeric : true

INC_LIAB_BOD



Quantiles

	quantiles	values
0	0.5%	0
1	10.0%	0
2	25.0%	0
3	50.0%	0
4	75.0%	0
5	90.0%	0
6	99.5%	12447.40846

Statistics

	statistics	values
0	min	0
1	max	69068.02629
2	mean	222.74327196709487
3	stddev	1859.5625777467199

Type : numeric

Number of modality : 4505

Advise as numeric : true

Bibliographie

[1] L. J. M. Aslett, Pedro M. Esperan ca and Chris C. Holmes, *Encrypted statistical machine learning : new privacy preserving methods.* , 2015.

[2] N. Dowlin, R. Gilad-Bachrach, K. Laine, K. Lauter, M. Naehrig, J. Wernsings, *CryptoNets : Applying Neural Networks to Encrypted Data with High Throughput and Accuracy.* , 2016.

[3] N. Dowlin, R. Gilad-Bachrach, K. Laine, K. Lauter, M. Naehrig, J. Wernsings, *Manual for Using Homomorphic Encryption for Bioinformatics.* , 2015.

[4] S. M. Mahajan and A. K. Reshamwala, *Data Mining Ethics in Privacy Preservation.* , International Journal of Computer Theory and Engineering, Vol. 3, No. 4, August 2011.

[5] G. Jagannathan, K. Pillaipakkamnatt, R.N Wright, *A Practical Differentially Private Random Decision Tree Classifier.* , 2009.

[6] K. Nissim, S. Raskhodnikova, A. Smith, *Smooth Sensitivity and Sampling in Private Data Analysis.* , 2007.

[7] A. Yu, W. L. Lai, J. Payor, *Efficient Integer Vector Homomorphic Encryption.* , 2015.

[8] J. W. Bos, K. Lauter, J. Loftus, M. Naehrig *Improved Security for a ring-based fully Homomorphic Encryption Scheme.* , 2013.

[9] H. Zhou and G. Wornell *Efficient Homomorphic Encryption on Integer Vectors and its Applications.* , 2014.

[10] S. Bogos, J. Gaspoz, S. Vaudenay *Cryptanalysis of a Homomorphic Encryption Scheme.* , 2016.

[11] N. Cai, S. Kou *Econometrics with Privacy Preservation.* , 2017.

[12] F. Armknecht, C. Boyd, C. Carr, K. Gjøsteen, A. Jaschke, C.A.Reuter & M. Strand *A Guide to Fully Homomorphic Encryption.* , 2015.

[13] P.M Esperança, L.J.M Aslett, C.C Holmes, *Encrypted accelerated least squares regression.* , 2017.

[14] J.Fan , F. Vercauteren *Somewhat practical fully homomorphic encryption.* , 2012.

[15] V.S Ryaben'kii , S.V. Tsynkov *A theoretical introduction to numerical analysis.* , 2016.

[16] R. Caron, T. Traynor *The zero set of a polynomial.* , University of Windsor.

[17] X. Yi et al *Homomorphic Encryption and Applications.* , Springer 2014.

[18] M. Van Djik et al *Fully homomorphic encryption over the integers.* , Springer 2010.

[19] J.-P. Delahaye *Déléguer un calcul sans divulguer ses données.* , Pour la science - octobre 2015.

[20] J. Blanc, A. De Georges *Techniques de cryptographies.* , 2004.

[21] E. Teske-Wilson *Homomorphic Cryptosystems.* University of Waterloo, 2011.

[22] S.P Lloyd *Least squares quantization in PCM.* IEEE Transactions on Information Theory, vol. 28, no 2, 1982.

[23] L. Sweeney *k-anonymity : a model for protecting privacy.* International Journal on Uncertainty, Fuzziness and Knowledge-based Systems, 2002.

[24] B. Nguyen *Techniques d'anonymisation.* Statistique et société, Vol. 2, N° 4 Décembre 2014.

[25] J. Eder *Privacy in Biobanks k-Anonymity and l-Diversity, etc.* Université de Klagenfurt, 2012.

[26] R.J. Bayardo, R. Agrawal *Data Privacy through Optimal k-anonymization* Conference : Data Engineering, 2005

[27] C.C. Aggarwal *On k-anonymity and the curse of dimensionality* IBM T. J. Watson Research Center, 2005.

[28] R. Tibshirani *Hierarchical Clustering* Carnegie Mellon University, 2013.

[29] Groupe de travail cyber-risques de l'institut des actuaires *Emergence du besoin en cyber assurance* Institut des actuaires, 2017.

[30] [WEB] Iamtrask, *Building safe A.I : a tutorial for Encrypted Deep Learning.* (https://iamtrask.github.io/2017/03/17/safe-ai/), 2017.

[31] [WEB] CNIL, *Règlement européen sur la protection des données : ce qui change pour les professionnels.* (www.cnil.fr/fr/reglement-europeen-sur-la-protection-des-donnees-ce-qui-change-pour-les-professionnels), 2016.

[32] [WEB] CNIL, *Le G29 publie un avis sur les techniques d'anonymisation.* (www.cnil.fr/fr/le-g29-publie-un-avis-sur-les-techniques-danonymisation-0), 2014.

[33] [WEB] Village Justice, *Données personnelles : anonymisation ou pseudonymisation ?* (www.village-justice.com/articles/donnees-personnelles-anonymisation-pseudonymisation,26194.html), 2017.

[34] [WEB] Global Security Mag, *RGPD : choisir entre l'anonymisation ou la pseudonymisation des données personnelles* (www.globalsecuritymag.fr/RGPD-choisir-entre-l-anonymisation,20171108,74998.html), 2017.

[35] [WEB] Akuiteo, *Pour vos données personnelles, êtes-vous plutôt anonymisation ou pseudonymisation ?* (www.akuiteo.com/blog/rgpd-gdpr-anonymisation-pseudonymisation-donnees-personnelles), 2018.

[36] [WEB] Wikipédia, *Anonymisation.* (https://fr.wikipedia.org/wiki/Anonymisation)

[37] [WEB] Wikipédia, *Apprentissages avec erreurs.* (https://fr.wikipedia.org/wiki/Apprentissage_avec_erreurs)

[38] [WEB] Wikipédia, *Ring Learning with Errors*. (https://en.wikipedia.org/wiki/-Ring_learning_with_errors)

[39] [WEB] Wikipédia, *K-Means clustering*. (https://en.wikipedia.org/wiki/K-means_clustering)

[40] [WEB] Wikipédia, *K-anonymisation*. (https://en.wikipedia.org/wiki/K-anonymity)

[41] [WEB] Wikipédia, *Méthode de Ward*. (https://fr.wikipedia.org/wiki/M%C3%A9thode_de_Ward)

[42] [WEB] OpenClassroom, *Explorer vos données avec des algorithmes non-supervisés*. (https://openclassrooms.com/fr/courses/4379436-explorez-vos-donnees-avec-des-algorithmes-non-supervises)

[43] [WEB] Naftali Harris, *Visualizing K-Means*. (www.naftaliharris.com/blog/visualizing-k-means-clustering/)

[44] [WEB] Wikipédia, *RGPD*. (https://fr.wikipedia.org/wiki/RGPD)

[45] RGPD, *Règlement U.E 2016/679, relatif à la protection des personnes physiques à l'égard du traitement des données à caractère personnel et à la libre circulation de ces données.* , 27 Avril 2016.