Institut de Statistique, Biostatistique et Actuariat

DIAGRAMMES LIFT ET EXTENSIONS

Promoteur :
Prof. **Michel DENUIT**
UCLouvain

Mémoire présenté en vue de l'obtention du master en sciences actuarielles (à finalité spécialisé) par : **Sidonie ATSAGMO**

Louvain-la-Neuve

Août 2022

Table des matières

Liste des figures **iv**

Liste des tableaux **vi**

Liste des abréviations **viii**

Résumé **ix**

Remerciements **x**

Introduction **1**

1 Présentation du modèle et généralités **3**

1.1 Modélisation 4

1.1.1 Hypothèses 5

1.1.2 Notation 5

1.1.3 Ordre Convexe 5

1.2 Qualité d'ajustement et stabilité d'un modèle 6

1.3 Stabilité des modèles 8

1.4 Les courbes de performances 9

1.4.1 Courbe de concentration 9

1.4.2 Courbe de Lorenz 10

1.4.3 Estimations 11

1.4.4 Classement des assurés à partir des primes 12

1.4.5 Propriétés des courbes de concentration et de Lorenz 12

1.4.6 Comparaison des performances de deux prédicteurs 14

2 Notions de lift et segmentation d'un modèle **15**

2.1 Modèles Lift et diagramme lift 15

2.1.1 Définitions 15
2.1.2 Lift et diagramme lift 16
2.2 Mesures lift et courbes de concentrations 21
2.2.1 L'aire entre les courbes (ABC) 21
2.2.2 Intégrale des courbes de concentration et de Lorenz 22
2.2.3 Dépendance entre $\pi(X)$ *et* $\mu(X)$ 23
2.2.4 Variations de l'ABC et de l'ICC 23

3 Applications : Assurance de la Responsabilité civile automobile en France 29
3.1 Statistiques descriptives 30
3.1.1 Variables quantitatives 30
3.1.2 Variables qualitatives 31
3.1.3 Corrélation 32
3.1.4 Dépendance entre les variables qualitatives 33
3.2 Modélisation 33
3.2.1 Régressions par le modèle GBM 34
3.2.2 Régressions par le modèle GLM 39
3.2.3 Régressions par le modèle GAM 43
3.2.4 Une généralisation suivant les modèles 47
3.3 Choix du meilleur modèle 49
3.3.1 Choix du modèle selon la qualité d'ajustement 49
3.3.2 Choix du modèle selon le lift 50

Conclusion 52

BIBLIOGRAPHIE 53

Annexes 55

Table des figures

1.1 Exemple de distribution de primes dans un échantillon 7

2.1 Exemple d'un graphique quantile simple 17
2.2 Exemple d'un graphique quantile double 19
2.3 Représentation d'une courbe de Lorenz 20
2.4 Représentation des courbes de concentration et de Lorenz pour $\tau = 0.5$ 24
2.5 Représentation des courbes de concentration et de Lorenz suivant les valeurs de τ 26
2.6 Représentation des courbes de concentration et de Lorenz suivant les différentes distributions 27
2.7 Impact de la distribution de $\mu(X)$ sur la variation de ABC 28

3.1 Box plot des variables quantitatives 30
3.2 Histogrammes des variables quantitatives 31
3.3 Description des variables qualitatives 32
3.4 Tableau des coefficients de corrélation linéaire 32
3.5 Importance des variables explicatives 36
3.6 La statistique H de Friedma 36
3.7 Courbes de concentration 38
3.8 Exemple de tableau pour la construction d'un diagramme quantile simple 38
3.9 Comparaison des primes GBM et pure 39
3.10 Courbes de performance 42
3.11 Comparaison la prime GLM et la prime pure 42
3.12 Courbes de concentration des primes GAM 44
3.13 Exemple de tableau pour la construction des graphiques quantiles doubles 45
3.14 Comparaison de la prime GAM à la prime pure et à la prime GLM 45
3.15 Comparaison des primes GBM à la prime GAM et et à la prime GLM 46
3.16 Courbes de concentration suivant les modèles 48

3.17 Variation de ICC, ILC et ABC en fonction de α 48
3.18 Distribution de primes prédites 49

Liste des tableaux

2.1 Variation de ABC et ICC suivant la loi Log-Normale $\tau = 0.5$ 25
2.2 Variation de ABC et ICC suivant la loi Weibull $\tau = 0.5$ 25
2.3 Variation de ABC et ICC suivant la loi Gamma $\tau = 0.5$ 25
2.4 Loi Log-Normale et Variation de ABC et ICC suivant les valeurs de τ 26
2.5 Loi Weibull et variation de ABC et ICC suivant les valeurs de τ 27
2.6 Loi Gamma et variation de ABC et ICC suivant les valeurs de τ 27
2.7 Loi Gamma et variation de ABC et ICC suivant les valeurs de τ 28
2.8 Loi Gamma et variation de ABC et ICC suivant les valeurs de τ 28

3.1 Statistiques descriptives des variables quantitatives 31
3.2 Les Valeurs du test V de Cramér entre les variables qualitatives 33
3.3 Somme des primes observées et somme des primes prédites (GBM) suivant la fréquence 37
3.4 Somme des primes observées et somme des primes prédites (GBM) suivant le coût moyen. 37
3.5 Somme des primes observées et somme des primes prédites (GLM) suivant la fréquence 41
3.6 Somme des primes observées et somme des primes prédites (GLM) suivant les coûts moyens 41
3.7 Somme des primes observées et somme des primes prédites (GAM) suivant la fréquence. 44
3.8 Somme des primes observées et somme des primes prédites (GAM) suivant les coûts moyens 44
3.9 Somme des primes observées et somme des primes prédites suivant les modèles . 47
3.10 Erreur quadratique sur la base de test 50
3.11 Déviance sur la base de test 50

3.12 ABC, Indice de Gini et lift (TT) suivant les différents modèles 51

Liste des abréviations

ABC : Area Between the Curves

AIC : Akaike Information Criterion

BIC : Bayesian Information Criterion

ICC : Integrated Concentration Curve

ILC : Integrated Lorenz Curve

GAM : Generalized Additive Models

GBM : Gradient Boosting Machine

GLM : Generalized Linear Models

Résumé

Dans le but de tarifer les primes pures d'assurance, les assureurs dans un marché concurrentiel doivent choisir les modèles qui déterminent les primes de manière équitable afin d'éviter la sélection adverse. Les actuaires dans ce sens utilisent généralement les critères basés sur la qualité d'ajustement des différents modèles pour faire un choix parmi plusieurs modèles. Notre travail étudie en plus de la qualité d'ajustement des modèles, leur capacité à segmenter les profils de risques d'assurés, il s'agit des diagrammes lift. Les critères du lift étudiés sont basés sur les courbes de performance, l'indice de Gini, les diagrammes quantiles simples et doubles. Il ressort de cette étude que l'ordre de classement des modèles dépend du critère utilisé.

Mots clés : Tarification, prime pure, diagramme lift, courbes de performance, diagramme quantile simple, diagramme quantile double, GLM, GAM, GBM.

Abstract

In order to price pure insurance premiums, insurers in a competitive market must choose models that determine premiums fairly in order to avoid adverse selection. Actuaries in this sense typically use criteria based on the goodness of fit of different models to make a choice among several models. Our work studies, in addition to the goodness of fit of the models, their ability to segment the risk profiles of the insureds, i.e. the lift diagrams. The lift criteria studied are based on the performance curves, the Gini index, the single and double quantile diagrams. The study shows that the ranking order of the models depends on the criterion used.

Key words : Pricing, pure premium, lift diagram, performance curves, single quantile diagram, double quantile diagram, GLM, GAM, GBM.

Remerciements

J'exprime ma profonde gratitude à tous ceux qui ont participé de près ou de loin au présent travail. Je remercie particulièrement mon promoteur Michel DENUIT pour avoir accepté d'encadrer ce travail. Il a été toujours disponible pour répondre à mes questions tout au long de la rédaction de ce mémoire.

J'adresse mes remerciements aussi à tous mes professeurs et à toutes les personnes qui ont su m'accompagner durant ma scolarité.

Introduction

L'activité d'assurance permet à un assuré ou un preneur d'assurance à transférer un risque à un assureur. Les notions de risque et d'incertitude sont centrales en assurance, en effet, si lors de la demande d'une assurance le risque est déjà survenu ou n'existe pas, aucun contrat ne peut être conclu. Lorsque les conditions sont réunies pour conclure un contrat, l'assuré ou le preneur d'assurance doit payer une prime pour pouvoir transférer son risque. Réciproquement l'assureur qui s'engage à couvrir le risque doit fixer la prime pure en fonction des caractéristiques de l'assuré et du risque, d'où la notion de tarification des primes pures.

La prime pure d'assurance est différente de la prime commerciale. Si la prime pure est l'espérance mathématique des coûts de sinistres, la prime commerciale quant à elle tient compte des frais généraux , des taxes et du chargement de sécurité en plus de l'espérance (GuirGuis et Dutang, 2016). Notre étude est centrée sur la prime pure d'assurance. La littérature montre que connaissant les caractéristiques d'un assuré, l'assureur peut à partir des modèles existants déterminer celui qui prédirait le mieux la prime à payer. Si les résultats du GLM (Generalized Linear Model)sont faciles à interpréter et incitent les assureurs à l'utiliser, on note qu'il existe bien d'autres modèles de tarification, on peut citer entre autres les modèles GAM (Generalized Additive Models) et GBM (Gradient Boosting Machine). Soit Y la variable aléatoire associée au coût total des sinistres par police, la prime pure est l'espérance de Y. Peu importe le modèle utilisé, la prime pure peut être modélisée en utilisant l'approche fréquence-coût moyen.

Cependant, les assureurs devraient mettre en place à partir desquels ils peuvent déterminer les primes de manière juste selon les différents profils de risques. Il devient important de faire un bon choix de modèle pour la tarification. On note de la littérature que le classement des modèles dépend du critère de choix utilisé, l'on peut se baser sur la qualité d'ajustement des modèles (l'erreur quadratique et la déviance par exemple) ou leur capacité à segmenter les profils de risque (notions de lift). L'étude de Meyers et Cumming(2009) est une étude comparative entre la qualité d'ajustement des modèles et leur capacité à segmenter les profils. Tevet(2013) et Medina(2015) ont également fait des études dans ce sens. À partir des courbes de concentration et de Lorenz, Denuit el al. (2019) ont définit une nouvelle mesure, l'aire entre ces deux courbes pour comparer plusieurs modèles de tarification.

L'objectif de cette étude est de classer les modèles de tarification en utilisant les diagrammes lift d'une part et la qualité d'ajustement d'autre part afin de voir si l'ordre de classement des modèles est le même dans les deux cas. Les notions lift étudiées dans ce travail sont basées sur les courbes de performance, les diagrammes quantiles simple et double. Cette étude est divisée

en trois chapitres : le premier présente le modèle et les généralités, le second présente les notions de lift et la segmentation d'un modèle et le troisième est une étude de cas dans lequel on fait une application en assurance de la responsabilité civile automobile en France.

Chapitre 1

Présentation du modèle et généralités

L'idée dans notre travail est de se baser sur une catégorie particulière de courbes pour mieux tarifer les primes réelles en assurance. Il est question de s'appuyer sur un modèle prédictif que les assureurs peuvent utiliser dans le but de rendre la contribution des primes équitables entre les assurés. En effet les assurés n'ont pas toujours les mêmes caractéristiques et chacun devrait contribuer en fonction de ses caractéristiques. Dans le cadre d'un marché concurrentiel d'assurance, un assureur pour éviter de perdre les clients, doit concevoir un modèle de tarification le plus juste possible, car les concurrents visent à attirer les assurés d'une autre compagnie qui paient plus de primes et bénéficient moins de prestations : l'objectif pour nous est de trouver des critères sur lesquels les assureurs doivent se baser pour éviter la sélection adverse. La littérature montre les critères tels que la qualité d'ajustement du modèle et la capacité à segmenter d'un modèle.

Pour effectuer une telle étude, les actuaires se basent le plus souvent sur des modèles de régression dont la réponse Y suit une distribution à dispersion exponentielle. Dans le cadre général, la famille des Y tels que $Y \sim ED(\theta, \frac{\phi}{\nu})$, se caractérise par la forme suivante :

$$f(y|\theta,\phi) = exp\Big(\frac{y\theta - a(\theta)}{\frac{\phi}{\nu}}\Big)c(y,\frac{\phi}{\nu}),\ y \in \mathcal{S}\ avec\ \mathcal{S} \subseteq \mathbb{N}\ ou\ \mathcal{S} \subseteq \mathbb{R}. \tag{1.1}$$

Dans cette caractérisation, θ est le paramètre canonique, $a(.)$ une fonction cumulant convexe, ϕ le paramètre de dispersion et ν le poids. La variance et la moyenne d'une distribution à dispersion exponentielle sont liées à la fonction $a(.)$ de telle manière que $E[Y] = \mu = a'(\theta)\ et\ Var[Y] = \frac{\phi}{\nu}a"(\theta)$. Pour la modélisation de la fréquence de sinistres, on utilise souvent la distribution de Poisson tandis que pour la modélisation du coût moyen, on peut utiliser la distribution de gamma, ces deux distributions obéissent à la famille ED.

Ainsi, si $Y \sim Poi(\lambda)\ avec\ \mathcal{S} = \mathbb{N}$, on a :

$f(y|\theta,\phi) = exp(-\lambda)\frac{\mu^y}{y!} = exp(yln\lambda - \lambda)\frac{1}{y!}$ et par identification avec l'équation (1.1) on a : $\theta = ln\lambda,\ a(\theta) = \lambda = exp(\theta),\ \phi = 1\ et\ c(y,\phi) = \frac{1}{y!}$.

Lorsque $Y \sim Gam(\alpha,\tau)\ avec\ \mathcal{S} = \mathbb{R}^+$, on a :

$$\begin{aligned} f(y|\theta,\phi) &= \frac{\tau^\alpha}{\Gamma(\alpha)} y^{\alpha-1} exp(-\tau y) \\ &= \frac{\alpha^\alpha}{\Gamma(\alpha)} \mu^\alpha y^{\alpha-1} exp(-y\frac{\alpha}{\mu}) \\ &= exp(\alpha(-\frac{y}{\mu} - ln\mu))\frac{\alpha^\alpha}{\Gamma(\alpha)} y^{\alpha-1} \end{aligned}$$

Par identification avec l'équation (1.1) on a :
$\theta = -\frac{1}{\mu},\ a(\theta) = ln\mu = -ln(-\theta),\ \phi = \frac{1}{\alpha}\ et\ c(y,\phi) = \frac{\alpha^\alpha}{\Gamma(\alpha)} y^{\alpha-1}$.

1.1 Modélisation

Suivant les caractéristiques des assurés capturées par un vecteur $X = (X_1, X_2, ..., X_p)$ et une réponse $Y \sim ED(\theta, \frac{\phi}{\nu})$ représentant le coût ou le nombre de sinistres, les régressions sont faites dans le but d'extraire les informations contenues dans X sur Y (Denuit et al., 2019).Le modèle de régression est sous la forme :

$$E[Y] = f(X) \tag{1.2}$$

Les modèles d'arbres de régression (CART) ne répondent pas à cette équation (1.2). Pour ces modèles, $E[Y] = \widehat{f}_w(X)$ avec w un paramètre de complexité représentant le nombre de nœuds dans l'algorithme CART. La fonction $\widehat{f}_w(x)$ étant définie par :

$$\widehat{f}_w(x) = \sum_{j=1}^{w} \overline{Y}_{j,w} \times I\{x \in R_{j,w}\}$$

Où $I\{x \in R_{j,w}\}$ est la fonction indicatrice associée au noeud final $R_{j,w}$ et $\overline{Y}_{j,w}$ est la moyenne empirique dans le groupe j (Paglia et al., 2011).

Il est question pour un actuaire d'utiliser la notion de lift pour segmenter le modèle afin que chaque assuré paye un montant de prime actuariellement juste. La prime pure réelle $\mu(X)$ est inconnue et est évaluée par l'espérance de Y connaissant le contenu de X, c'est-à-dire $\mu(X) = E(Y|X)$. Ceci étant, le fonction $x \mapsto \mu(x) = E(Y|X = x)$ inconnue pour l'actuaire est approchée par la prime actuelle $x \mapsto \pi(x)$ qui a une structure relativement simple par rapport à μ.

Comme on l'a noté plus haut, il est important d'utiliser l'outil lift pour savoir si le modèle défini est valide ou pas, ceci grâce à la performance de prédiction de la prime réelle $\mu(X)$

connaissant X. Pour un portefeuille très large, l'écart $Y - \mu(X)$ s'annule en moyenne. De plus l'écart $\mu(X) - \pi(X)$ doit être le plus petit possible. La modélisation en assurance considère que le score S est formé par des variables explicatives et la prime π est une fonction monotone et croissante de S. Ainsi, $\pi(X) = h(S)$, avec h une fonction croissante.

1.1.1 Hypothèses

On suppose que :

— $\pi(X)$ *et* $\mu(X)$ sont des variables aléatoires continues admettant des fonctions de densité de probabilité, et ceci lorsque X contient au moins une caractéristique continue et π est une fonction continue et strictement croissante du score S.

— $E[\mu(X)] = E[\pi(X)] = E(Y)$.

1.1.2 Notation

Soit F_π la fonction de distribution de $\pi(X)$, et f_π la fonction de densité de probabilité associée à F_π. On a $\forall t \geq 0$:

$$F_\pi(t) = P(\pi(X) \leq t) = \int_0^t f_\pi(s)ds \tag{1.3}$$

La fonction réciproque de F_π notée F_π^{-1} est prise comme quantile associé à F_π ou la Value-at-Risk (VaR). Pour un niveau de probabilité α donné,
$F_\pi^{-1}(\alpha) = inf\{t/F_\pi(t) \geq \alpha\}$ *et* $F_\pi(F_\pi^{-1}(\alpha)) = \alpha$.

1.1.3 Ordre Convexe

La quantité d'information sur la prime réelle dépend de la dispersion de $\pi(X)$. On n'a aucune information sur le risque relatif de polices différentes lorsque $\pi(X)$ est constant. Ainsi une variabilité élevée de $\pi(X)$ donne plus d'information sur le risque. Pour deux vecteurs d'information X_1 *et* X_2, la dispersion de $\pi(X_1)$ *et* $\pi(X_2)$ renseigne la variable qui fournit plus d'informations. Selon Frees et al.(2019), l'information joue un rôle très important dans la tarification des primes, plus elle est disponible plus les assureurs s'éloignent des modèles de tarification traditionnels. En effet, lorsque les nouvelles informations sont disponibles, les assureurs sont capables de se baser sur des caractéristiques supplémentaires pour mettre des nouveaux modèles en place.

Il devient important de faire la comparaison de la variabilité sous-jacente dans un problème de tarification. Denuit et al. (2019) note qu'au-delà des indicateurs simples comme les écarts-

types, l'utilisation de l'ordre convexe comme outil probabiliste est efficace quant à l'évaluation de la dispersion des variables aléatoires.

Soient Z et T deux variables aléatoires, on dit que T est plus petite que Z dans l'ordre convexe et on note $T \leq_{CX} Z$ si pour toute fonction convexe g, on a :
$\mathbb{E}[g(T)] \leq \mathbb{E}[g(Z)]$, lorsque les espérances existent.

Une caractérisation importante de l'ordre convexe se fait par construction sur le même espace de probabilité en utilisant les espérances conditionnelles. Ainsi, les variables aléatoires Z et T satisfont $T \leq_{CX} Z$ si, et seulement s'il existe deux variables aléatoires $\tilde{Z}$ et $\tilde{T}$, définies sur le même espace de probabilité, telles que $\tilde{T}$ est de même distribution que T, $\tilde{Z}$ est de même distribution Z, et $\{\tilde{T}, \tilde{Z}\}$ est une martingale, c'est-à-dire $\mathbb{E}[\tilde{Z}|\tilde{T}] = \tilde{T}$ est vraie presque sûrement. Cela montre directement que l'augmentation du nombre de caractéristiques est bénéfique car $X_1 \subseteq X_2 \Longrightarrow \mu(X_1) \leq_{CX} \mu(X_2)$. On peut alors avoir des primes plus dispersées μ en passant de X_1 à l'information plus riche X_2.

Selon Tevet(2015), après avoir construit et analysé un modèle prédictif, il ne suffit pas de résumer les résultats et mettre fin à l'étude, une étape à prendre toujours en compte dans la modélisation est la validation du modèle dont les composantes sont généralement : La qualité d'ajustement, le Lift et la stabilité des résultats du modèle. La validation du modèle est ainsi une étape qui permet de juger si le modèle prédictif défini est performant ou non. La section suivante est la qualité d'ajustement et la stabilité d'un modèle, pour ce qui est du lift, il sera présenté au chapitre suivant.

1.2 Qualité d'ajustement et stabilité d'un modèle

Cette composante de la validation d'un modèle prédictif permet d'évaluer l'adéquation du modèle aux données. La littérature montre qu'il existe plusieurs manières de la mesurer, si nous désignons par y *et* μ les valeurs observées de la variable cible et prédites respectivement, des travaux de Tevet(2016) et Maydeu-Olivares and Garcıa-Forero (2010) on note quelques outils de mesure de la qualité d'ajustement d'un modèle :

- **L'erreur quadratique** qui pour chaque enregistrement est le carré de la différence entre la variable cible réelle et la variable cible prédite, $EQ_i = (y_i - \mu_i)^2$.
- **L'erreur absolue** qui pour chaque enregistrement est la valeur absolue de la différence entre la variable cible réelle et la variable cible prédite $EA_i = |y_i - \mu_i|$.

  Si ces deux premiers outils sont faciles et intuitifs, ils ne sont pas très pertinents pour les données d'assurance qui sont généralement non normales et peuvent par conséquent

conduire à un mauvais choix de modèle. On note avec Frees et al.(2014) que la distribution des primes a une tendance asymétrique à droite à queue épaisse, cette forme pouvant être due au fait que les caractéristiques diffèrent d'un assuré à un autre et aussi parce que les contrats sont de plusieurs types suivant par exemple les couvertures, les déductibles, les plafonds d'intervention qui y sont liés. La figure suivante représente une distribution des primes suivant les classes d'assurés, l'axe des Y représente les montants des primes et l'axe des abaisses les classes d'assurés. Le graphique suivant utilise les données brésiliennes issues de la base CASdataset, il s'agit d'un ensemble de données sur l'assurance des véhicules au Brésil (brvehins1).

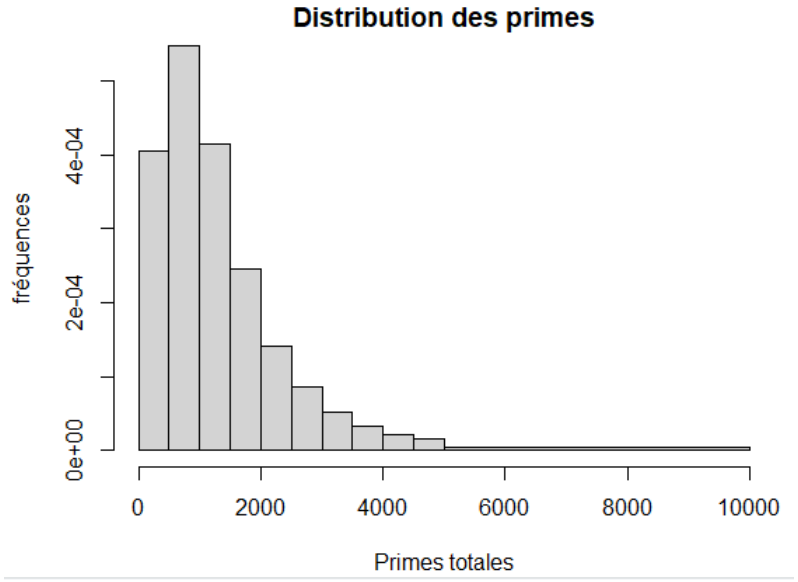

FIGURE 1.1 – Exemple de distribution de primes dans un échantillon

— **La vraisemblance** : elle mesure la probabilité prédite par le modèle pour que ce qui s'est réellement produit puisse encore se produire, les paramètres sont estimés grâce aux GLM maximisant la vraisemblance. Pour deux modèles donnés, le plus adapté est celui ayant la plus grande vraisemblance. La limite de la vraisemblance est que l'ajout d'une variable l'améliore toujours. L'Akaike Information Criterion(AIC) et le Bayesian Information Criterion(BIC) sont des mesures pénalisées. Lorsqu'une variable améliore l'AIC ou le BIC, il est maintenu dans le modèle. Soit k le nombre de paramètres à estimer dans le modèle, n le nombre d'observations de la base de données et $\widehat{L}(\theta)$ la valeur maximale de la fonction de vraisemblance avec θ les valeurs des paramètres maximisant la fonction de vraisemblance : $AIC = 2(k - ln(\widehat{L}))$ *et* $BIC = k \times ln(n) - 2ln(\widehat{L})$.

— **La déviance** : elle dépend à la fois du logvraisemblance des modèles saturé et ajusté, elle est minimisée par les GLM. Si $\widehat{L}$ *et* L^* sont respectivement les logvraisemblances

des modèles saturé et du modèle ajusté, on a : $Deviance = 2(\widehat{L} - L^*)$. Si on juge la qualité d'un modèle ayant les données non normales grâce à l'erreur quadratique, on peut choisir un modèle incorrect, car l'erreur quadratique n'est rien d'autre que la déviance d'un modèle de régression ayant une distribution normale : tel est un avantage de la déviance par rapport à l'erreur quadratique. Lorsque Y suit une distribution de gamma, $Deviance = 2\sum_{i=1}^{n} w_i(\frac{y_i - \mu_i}{\mu_i} - log(\frac{y_i}{\mu_i}))$ avec w_i le poids et μ_i la prédiction de y_i, on retrouve cette définition des Charpentier (2013) avec les poids constants.

Les distributions asymétriques pouvant être adaptées aux données d'assurance, la déviance est mieux dans un cadre d'assurance que l'erreur quadratique. Elle s'applique aussi bien aux GLMs qu'aux procédures de biais minimum. La déviance est aussi calculée à partir du moment où l'on est dans l'hypothèse d'existence des données issues d'une distribution à dispersion exponentielle.

— **Les résidus** : Pour une observation i donnée, le résidu $\varepsilon_i = y_i - \mu_i$. L'une des hypothèses d'un modèle de régression linéaire simple est que ε_i soit supposé distribuer normalement. Si les ε_i sont trop éloignés de la distribution normale, on fait face à un modèle d'ajustement faible. Ainsi, pour une étude qui utilise les données d'assurance, les résidus ne sont pas généralement utiles à cause de leur forme asymétrique.

1.3 Stabilité des modèles

Tevet(2016) présente les mesures de stabilité interne comme un processus permettant d'étudier la robustesse de résultats d'un modèle donné. Il distingue à ce sujet la validation hors-échantillon dont le but est d'évaluer l'ajustement du modèle en utilisant les données non considérées lors de la construction du modèle. Cette validation est basée sur deux approches, soit dans un premier temps répartir le jeu de données en deux sous-jeux, l'un pour le traitement et l'autre pour le test ; soit faire une validation croisée en utilisant de manière répétée un sous-jeu pour la construction et l'autre pour le test.La phase de construction du modèle utilise la base de données de traitement et la base de données test est utilisée dans la phase de validation du modèle.

L'évaluation de la stabilité dans le temps nécessite généralement la stabilité des résultats du modèle dans le temps. Pour une évaluation de la stabilité temporelle, on peut exécuter le modèle sur les années individuelles et rechercher la variabilité. Le modèle est plus stable lorsque l'on ne note pas un changement significatif des paramètres estimés suivant les sous-ensembles de données. La déviance quant à elle, est obtenue à partir des résultats de l'ensemble ayant

servi la validation du modèle.

Dans le cadre des GLMs, une notion importante pour l'étude de la stabilité d'un modèle est la distance de Cook, c'est un diagnostic qui nous renseigne l'effet de chaque enregistrement sur les résultats du modèle. Plus elle est grande plus on a l'impact impliquant qu'on peut avoir un changement significatif des résultats du modèle lorsque l'enregistrement est éliminé.

Dans la tarification des primes d'assurance, plusieurs auteurs ont utilisé les courbes de concentration et de Lorenz, ces courbes permettent en fait d'étudier les répartitions des primes dans une population considérée. Il est plus juste que chaque assuré paye les primes en fonction de ses caractéristiques et des sinistres pour lesquels il souscrit une assurance. Il est alors important de définir ces deux notions dans le cadre d'assurance afin de voir dans quelle mesure elles sont utiles pour des assureurs.

1.4 Les courbes de performances

1.4.1 Courbe de concentration

Définition 1.4.1. *D'après Denuit et al. (2019), la courbe de concentration de la prime pure* $\mu(X)$ *par rapport à la prime actuelle* $\pi(X)$ *suivant l'information disponible dans le vecteur* X *et pour un niveau de probabilité* α *donné est définie par :*

$$\alpha \mapsto CC[\mu(X), \pi(X); \alpha] = \frac{E[\mu(X) I[\pi(X) \leq F_{\pi}^{-1}(\alpha)]]}{E[\mu(X)]}$$

.

Elle est donc le pourcentage des primes totales perçues par l'assureur du portefeuille ayant les $\alpha\%$ plus petites primes $\pi(X)$.

Lorsque la courbe de concentration dépend de la prime $\pi(X)$ et qu'on est dans un marché concurrentiel, les assurés ayant un profil à faible risque sont attirés par les concurrents, les polices associées à ces profils peuvent quitter le portefeuille, ce qui réduit le profit de l'assureur. Pour ne pas faire face au problème de sélection adverse, l'assureur doit être prudent en évitant la sur-facturation de ce groupe d'assurés. On voit alors l'importance de l'usage de la courbe de concentration, car elle permet de mieux évaluer la prime π convenable à un tel groupe d'assurés. Très peu de travaux ont utilisé la courbe de concentration pour l'évaluation des primes. Pour certains auteurs, son utilisation ne permet pas d'évaluer de manière efficace la performance de la prime π, car il peut arriver dans une étude que ses valeurs réelles n'aient pas

d'importance, c'est le cas l'étude de Gourieroux et Jasiak (2007)[1]. Mais en tarification, il est nécessaire de tenir compte à la fois des montants des primes $\pi(X)$ et du revenu réel des primes basé sur $\mu(X)$. Pour cette raison, beaucoup d'auteurs ont pensé à la courbe de Lorenz, qui permet d'étudier de manière efficace la distribution des primes au sein d'un groupe d'assurés afin d'avoir une idée sur les inégalités parmi eux.

1.4.2 Courbe de Lorenz

Une courbe de Lorenz est définie dans le contexte économique comme une courbe grâce à laquelle l'équité dans la distribution des revenus est étudiée pour une population donnée. La littérature montre que plusieurs études dans le domaine de l'assurance ont utilisé la courbe de Lorenz aussi bien en assurance-maladie qu'en assurance automobile. Jusot et al. (2016) ont utilisé cette courbe pour étudier la variabilité des inégalités de niveau de vie. Ces auteurs comparent d'une part la courbe de consommation de soins et la courbe, et d'autre part la courbe de financement des soins en France. Ainsi, en assurance-maladie, la courbe de Lorenz peut être définie comme une courbe de concentration permettant de distribuer la consommation des soins et le financement des soins au sein d'une population. Par ailleurs, la littérature montre que la courbe de Lorenz peut être utilisée pour la distribution des primes techniques basées sur les coûts ou sur les demandes. Les actuaires utilisent à la fois les mesures d'assurance et les mesures purement statistiques afin de faire une étude comparative entre plusieurs modèles de tarification.

Ainsi, la décision de la mise en place d'un modèle en assurance nécessite des critères plus économiques. Denuit et al. (2019) définissent la courbe de Lorenz comme une courbe de performance permettant de distribuer les primes techniques. On note de leur travail qu'une fois les assurés classés en sous-groupes selon les montants des primes versées, les courbes de concentration et de Lorenz permettent d'estimer la proportion des primes correspondant $\alpha\%$ des primes prédites les plus faibles, et par suite cette proportion est comparée à la perte agrégée dans l'ensemble du portefeuille. Ils se basent sur la courbe de concentration pour définir la courbe de Lorenz. En supposant que la prime réelle $\mu(X)$ est connue i.e $\mu(X) = \pi(X)$ dans l'expression définissant la courbe de concentration, on obtient la courbe de Lorenz.

Définition 1.4.2. *La courbe de Lorenz liée à la prime* $\pi(X)$ *est définie par :*
$\alpha \mapsto LC[\pi(X), \alpha] = CC[\pi(X), \pi(X); \alpha] = \frac{E[\pi(X) I[\pi(X) \leq F_{\pi}^{-1}(\alpha)]]}{E[\pi(X)]}$.

1. Cité par Denuit et al. (2019)

On note de la relation ci-dessus que la courbe de Lorenz est un cas particulier des courbes de concentration.

Lorsque $\pi(X)$ croît suivant l'ordre convexe, on assiste à un déplacement de la courbe de Lorenz vers le bas. Lorsque $\mu(X) = \pi(X)$ la courbe de concentration coïncide avec la courbe de Lorenz de $\mu(X)$ $\forall\alpha$, dans ce cas, il y a équilibre dans le sous-portefeuille vérifiant $\pi(X) \leq F_{\pi}^{-1}(\alpha)$. Plus l'écart entre les deux courbes est grand, moins le prédicteur $\pi(X)$ approxime la prime réelle $\mu(X)$. Même si la majorité d'auteurs utilise seulement la courbe de Lorenz pour l'évaluation de l'efficacité d'un modèle prédictive, il est à noter qu'il reste nécessaire d'utiliser les courbes vu qu'en réalité la prime réelle est différente de la prime actuelle.

1.4.3 Estimations

La courbe de concentration dépend de la prime inconnue $\mu(X)$, donc évaluer la courbe de concentration revient à estimer pour tout niveau de probabilité α, ceci peut se faire en utilisant la réponse Y pour estimation.

$$\begin{aligned} CC[\mu(X), \pi(X); \alpha] &= CC[Y, \pi(X); \alpha] \\ &= \frac{E[\mu(X) I[\pi(X) \leq F_{\pi}^{-1}(\alpha)]]}{E[\mu(X)]} \\ &= \frac{E[E[Y|X] I[\pi(X) \leq F_{\pi}^{-1}(\alpha)]]}{E[E[Y|X]]} \\ &= \frac{E[E[Y I[\pi(X) \leq F_{\pi}^{-1}(\alpha)]|X]]}{E(Y)} \\ &= \frac{E[Y I[\pi(X) \leq F_{\pi}^{-1}(\alpha)]}{E(Y)} \end{aligned}$$

La part des pertes totales attribuée au sous-portefeuille contenant $\alpha\%$ des polices ayant les plus faibles prédictions est obtenue à partir de la courbe de concentration. La relation précédente est d'une grande importance dans la tarification parce qu'en évaluant l'efficacité du prédicteur considéré, la prime pure inconnue $\mu(X)$ est remplacée par la variable expliquée du modèle de régression. Ceci étant, la relation d'estimation peut s'écrire :

$$\begin{aligned} \widehat{CC}[\mu(X), \pi(X); \alpha] &= \widehat{CC}[Y, \pi(X); \alpha] \\ &= \frac{\sum_{i|\widehat{\pi}(X_i) \leq \widehat{F}_{\pi}^{-1}(\alpha)} Y_i}{n\overline{Y}} \\ &= \frac{\sum_{i|\widehat{\pi}(X_i) \leq \widehat{F}_{\pi}^{-1}(\alpha)} Y_i}{\sum_{i=1} Y_i} \end{aligned}$$

où $\widehat{\pi}$ est l'estimation du predicteur π et $\widehat{F}_{\pi}$ la fonction de distribution empirique obtenue

de $\widehat{\pi}$. $\widehat{F}_{\pi}(t) = \frac{1}{n}\sum_{i=1}^{n} I[\widehat{\pi}(X_i) \leq t]$. Denuit et al.(2021) estiment les courbes de concentration grâce à cette relation.

De la même manière, la courbe de Lorenz doit être estimée par :

$$\widehat{LC}(\pi(X);\alpha) = \frac{\sum_{i|\widehat{\pi}(X_i)\leq \widehat{F}_{\pi}^{-1}(\alpha)} \widehat{\pi}(X_i)}{\sum_{i=1}^{n} \widehat{\pi}(X_i)}$$

Cette dernière représente le pourcentage de la prime totale correspondant à $\alpha\%$ des plus petites primes quand elles sont calculées à partir de π. $\widehat{LC}$ est également la proportion des plus faibles primes par rapport à la perte agrégée au niveau du portefeuille entier si $\sum_{i=1}^{n} \widehat{\pi}(X_i) = \sum_{i=1}^{n} Y_i$. Cette relation est également valable dans un modèle GLM avec une fonction de lien canonique et intercep.

1.4.4 Classement des assurés à partir des primes

A partir de l'équivalence $\pi(X_i) \leq F_{\pi}^{-1}(\alpha) \Leftrightarrow F_{\pi}(\pi(X)) \leq \alpha$, le prédicteur $\pi(X)$ est remplacé par le rang $\Pi = F_{\pi}(\pi(X))$ dans les expressions précédentes avec Π qui obéit à la distribution uniforme unitaire. Lorsque les assurés de la population considérée sont classés par ordre croissant de leurs primes, π est le rang de l'assuré.

$$\begin{aligned} E[Y I[\pi(X) \leq F_{\pi}^{-1}(\alpha)]] &= E[Y I[\Pi \leq \alpha]] \\ &= \int_0^{\infty}\int_0^{1} y I[z \leq \alpha] dF_{Y,\Pi}(y,z) \\ &= \int_0^{\infty}\int_0^{1} y I[z \leq \alpha] dC(F_Y(y), z) \\ &= \int_0^{1} F_Y^{-1}(u) \times C_1'(u,\alpha) du \ en,, \ posant\ y = F_Y^{-1}(y),\ ie\ F_Y(y) = u. \end{aligned}$$

Avec C le copule du couple $(Y, \pi(X))$ *et* C_1' la dérivée partielle de C par rapport à u.

1.4.5 Propriétés des courbes de concentration et de Lorenz

Pour une bonne mise en œuvre des courbes de performances, il est nécessaire de connaître les propriétés particulières qui les rendent utiles en assurance.

1.4.5.1 La monotonie

Par définition, la courbe de concentration est définie à partie d'une fonction évaluée aux différents quantiles de la prime $\pi(X)$, plus le niveau de probabilité croît, plus la prime aug-

mente, alors la courbe de concentration est une fonction strictement croissante avec pour points extrêmes $(0;0)$ *et* $(1;1)$.

1.4.5.2 Ligne d'indépendance et ligne d'égalité

La ligne de 45 degrés représente la ligne d'indépendance pour une courbe de concentration et la courbe d'égalité pour une courbe de Lorenz. En cas d'absence de dépendance entre les primes $\mu(X)$ *et* $\pi(X)$, la courbe de concentration coïncide avec la ligne de 45 degrés, donc elle se réduit à la fonction identité $(CC[\mu(X), \pi(X); \alpha] = \alpha)$. Une forte dépendance entre $\mu(X)$ *et* $\pi(X)$ entraîne une augmentation de l'aire du domaine délimité par la ligne d'indépendance et la courbe de concentration.

Proposition 1.4.1. *Si* $\forall t, E[\mu(X)] \geq E[\mu(X)|\pi(X) \leq t]$, *alors la courbe de concentration se situe en dessous de la ligne d'indépendance et on a :* $CC[\mu(X), \pi(X); \alpha] \leq \alpha, \forall \alpha$.

Preuve 1.4.1. *Supposons que* $\forall t, E[\mu(X)] \geq E[\mu(X)|\pi(X) \leq t]$ (a).

Par définition, on a :

$$\begin{aligned} \alpha \mapsto CC[\mu(X), \pi(X); \alpha] &= \frac{E[\mu(X) I[\pi(X) \leq F_\pi^{-1}(\alpha)]]}{E[\mu(X)]} \\ &= \frac{P[\pi(X) \leq F_\pi^{-1}(\alpha)] E[\pi(X)|\pi(X) \leq F_\pi^{-1}(\alpha)]}{E[\mu(X)]} \end{aligned}$$

et D'après (a), *on a* $\mapsto CC[\mu(X), \pi(X); \alpha] \leq \frac{P[\pi(X) \leq F_\pi^{-1}(\alpha)] E[\mu(X)]}{E[\mu(X)]}$ *et* $P[\pi(X) \leq F_\pi^{-1}(\alpha) = P[F_\pi(\pi(X)) \leq \alpha] = \alpha$. ***D'où le résultat.***

Pour la courbe de Lorenz, la ligne de 45 degrés est appelée ligne d'égalité, sur cette ligne, on a une distribution égalitaire, c'est-à-dire $\alpha\%$ d'assurés paye $\alpha\%$ des primes.
Si $\pi(X) = E[Y]$ alors $LC[\pi(X); \alpha] = \alpha$. Car π est constant dans ce cas, ce qui peut s'expliquer par le fait que le vecteur X n'a aucune caractéristique liée à la réponse, Y et $\pi(X)$ sont donc indépendants.

1.4.5.3 Convexité

Proposition 1.4.2. *La courbe de concentration* $\alpha \mapsto CC[\mu(X), \pi(X); \alpha]$ *est convexe si et seulement si* $\mu(X)$ *dépend positivement de* $\pi(X)$ *par régression, c'est-à-dire que la fonction* $t \mapsto E[\mu(X)|\pi(X) = t]$ *est strictement croissante.*

1.4.6 Comparaison des performances de deux prédicteurs

1.4.6.1 Courbes de concentration et de Lorenz

Plusieurs méthodes sont utilisées dans la littérature pour comparer les performances de deux prédicteurs d'une prime pure $\mu(X)$, la différence entre ces prédicteurs pouvant être liée soit à leur forme fonctionnelle, soit aux caractéristiques de l'information qu'ils contiennent. Le plus performant est celui qui est plus efficace à identifier les différents profils de risques et qui est le plus corrélé avec la prime pure $\mu(X)$. Dans le but de comparer les performances de deux prédicteurs, Denuit et al. (2019) utilisent la variance l'ordre convexe entre primes actuelles qui étend la comparaison basée sur les variances. Ainsi pour deux prédicteurs π_1 *et* π_2 et pour deux vecteurs aléatoires X_1 *et* X_2, $\pi_2(X_2) \preceq_{cx} \pi_1(X_1) \Rightarrow Var[\pi_2(X_2)] \leq Var[\pi_1(X_1)]$. De plus, leur travail étend les courbes de performances définies par Gourieroux et Jasiack (2007)[2].

Définition 1.4.3. *La prime* $\pi_1(X_1)$ *est plus discriminatoire que* $\pi_2(X_2)$ *si et seulement si* $\pi_2(X_2) \preceq_{cx} \pi_1(X_1) \Leftrightarrow LC[\pi_1(X_1); \alpha] \leq LC[\pi_2(X_2); \alpha]\ \forall \alpha$ *et*
$CC[\mu(X), \pi_1(X_1); \alpha] \leq CC[\mu(X), \pi_2(X_2); \alpha]\ \forall \alpha$.

Cette définition stipule que π_2 contient la plus grande proportion de la prime totale. En posant $\Pi_1 = F_{\pi_1}(\pi_1(X_1))$ *et* $\Pi_2 = F_{\pi_2}(\pi_2(X_2))$ les scores distribués sur $[0; 1]$ où F_{π_1} *et* F_{π_2} sont des fonctions de distribution de π_1 *et* π_2 respectivement, on a :
$E[Y|\pi_1(X_1) \leq F^{-1}_{\pi_1}(\alpha)] \leq E[Y|\pi_2(X_2) \leq F^{-1}_{\pi_2}(\alpha)] \Leftrightarrow E[Y|\Pi_1 \leq \alpha] \leq E[Y|\Pi_2 \leq \alpha]$.

1.4.6.2 Ordre de concordance

Lorsque deux variables aléatoires ont tendance à varier dans le même sens, elles sont dites concordantes (Denuit et al., 2005). Soient (Z_1, Z_2) *et* (V_1, V_2) deux couples de variables aléatoires ayant les mêmes distributions marginales $(P[Z_k \leq t] = P[V_k \leq t] = F_k(t), k \in \{1, 2\})$.
Si $P[Z_1 \leq t_1, Z_2 \leq t_2] \leq P[V_1 \leq t_1, V_2 \leq t_2]\ \forall t_1,\ t_2$, alors (Z_1, Z_2) est dit plus concordant que (V_1, V_2) et on note $(Z_1, Z_2) \preceq_{conc} (V_1, V_2)$.

Proposition 1.4.3. *Si* $\pi_2(X_2) \preceq_{cx} \pi_1(X_1)$ *et* $(Y, \Pi_2) \preceq_{conc} (Y, \Pi_1)$, *alors, le prédicteur* $\pi_1(X_1)$ *est plus discriminatoire que le prédicteur* $\pi_2(X_2)$ *pour la réponse* Y.

Le plus discriminatoire des prédicteurs π_1 *et* π_2 pour la réponse Y est celui qui domine suivant l'ordre convexe et qui est plus corrélé avec Y au sens de $\preceq_{conc}$.

2. Cité par Denuit et al.(2019)

Chapitre 2

Notions de lift et segmentation d'un modèle

Nous notons de la littérature que la notion de lift joue un rôle important dans la définition des primes suivant les classes des assurés. On note alors qu'il existe plusieurs types de courbes lift, une question importante est de savoir comment elles se construisent et comment choisir l'une d'elles dans un contexte précis. Lecomte (2018), a défini une courbe de lift comme un outil de visualisation de la capacité de segmentation d'un modèle, qui fait une confrontation entre les prédictions du modèle et celles issues d'un modèle aléatoire. Il note plusieurs points importants dans la construction d'un modèle à savoir : comment construire une courbe lift, comment découper les données, les différents types de diagrammes lift, le critère de choix pour un diagramme et comment comparer deux courbes lift. Les travaux existants ressortent plusieurs notions liées au mot "lift", on note entre autres : la valeur de lift, le modèle lift, les courbes et diagrammes lifts.

2.1 Modèles Lift et diagramme lift

2.1.1 Définitions

Le lift selon Tevet(2013) est une mesure de la valeur économique d'un modèle, mais qui ne représente pas le gain d'un assureur dans la mise en œuvre d'un modèle. Bien que certaines mesures lift puissent aider à quantifier le gain, l'idée générale du lift est l'évaluation de la capacité d'un modèle à différencier les pires et les meilleurs risques. En résumé, pour cet auteur, Le"Lift mesure la mesure dans laquelle un modèle aide un assureur à éviter la sélection adverse en facturant à chaque assuré un tarif actuariellement juste". Il est donc question pour chaque

assureur de mettre en œuvre un modèle de tarification équitable de telle manière que les taux de primes proposés aux assurés soient actuariellement juste afin de ne pas faire face à la sélection adverse, car les assurés n'ont pas les mêmes caractéristiques en général.

Après la modélisation, il est nécessaire de savoir si le modèle conçu vaut la peine d'être mise en œuvre, c'est ainsi que Tevet(2013) pense qu'en plus de juger la pertinence du modèle grâce à sa qualité d'ajustement et sa capacité à distinguer les pires risques des meilleurs, il est important de vérifier également sa stabilité afin de dire s'il est valide ou pas. Dans le même sens Medina (2015) note que si certains modèles semblent pertinents, ils peuvent faire face à un certain nombre de faiblesses. On peut citer entre autres : le mauvais ajustement, le sur-ajustement, une distribution mal choisie, l'instabilité des résultats dans un sous-jeu de données ou même dans le temps, une répétition d'enregistrement ayant un grand impact sur les résultats, une faible performance du modèle par rapport à un statu-quo.

Tevet (2013) distingue trois outils de mesure et d'illustration d'un modèle lift : les graphiques quantiles simples, les graphiques lift double et l'indice de Gini. Medina (2015) augmente un quatrième outil qui est le graphique de taux de pertes.

2.1.2 Lift et diagramme lift

Certaines études montrent que le lift peut être calculé à partir du gain (c'est le cas où Y est binaire) tandis que d'autres représentent les digrammes sans faire appel à la notion de gain, tout dépend du contexte. Dans le cas de la vente d'un nouveau produit, l'assureur qui souhaite contacter une grande majorité de ses clients pouvant accepter ce produit peut se baser sur le gain pour contacter un nombre très réduit des clients. Si par contre il dispose des primes pures réelles, il peut concevoir un modèle de prédiction de primes afin de construire son diagramme de lift : c'est le cas de Medina(2016), il construit les courbes de lift sans passer par les gains.Dans ce paragraphe, les données utilisées pour des exemples sont issues de brvehins1.

2.1.2.1 Graphiques quantiles simples

Dans le but de déterminer lequel des modèles de prédiction est mieux, trois critères sont à vérifier lorsqu'on a un graphique quantile simple. Premièrement, on vérifie la correspondance entre les valeurs observées (y_i) et les valeurs prédites $(\pi_i(X))$; ensuite on vérifie si les valeurs augmentent de façon monotone ou avec peu d'inversions et enfin on détermine la capacité du modèle à distinguer les assurés à faibles coûts des assurés à coûts élevés (Tevet, 2016).

En ce qui concerne la construction, elle nécessite comme précédemment les étapes telles

que : le classement des données, leur répartition en quantiles. La première différence réside sur le fait qu'ici, les données sont triées suivant les coûts des pertes prédites $\pi_i(X)$. Pour chaque quantile, on détermine le coût moyen des pertes réelles et le coût moyen des pertes prédites. A chaque niveau de quantile, on a deux points, l'un pour le coût prédit et l'autre pour le coût réel, ceci nous permet alors de tracer les diagrammes lift correspondant aux pertes réelles et aux pertes prédites.

Exemple : La figure suivante présente un graphique quantile simple illustrant la courbe des coûts réels du sinistre et celle des coûts prédits grâce au modèle GLM.

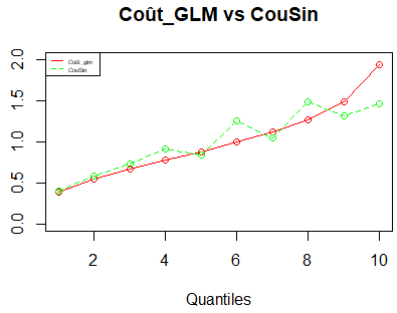

FIGURE 2.1 – Exemple d'un graphique quantile simple

On note de cette figure que les coûts prédits et les coûts réels sont presque égaux dans le premier quantile, ils coïncident également pour $\alpha = 0.5,\ 0.7,\ 0.85$. Par ailleurs, le plus grand écart entre les deux coûts s'observe dans le dernier quantile où les valeurs prédites $\pi_i(X)$ sont supérieures aux valeurs réelles y_i, dans les autres quantiles, les valeurs observées sont supérieures aux valeurs prédites par le GLM. Les valeurs prédites augmentent de façon monotone, alors que les valeurs réelles augmentent avec peu d'inversions.

Tevet(2013) note que les graphiques quantiles simples aident à comparer deux modèles de prédiction différents A et B. Dans ce cas on peut avoir deux graphiques : le premier représentant la courbe des coûts moyens réels et celle des coûts moyens de pertes prédits par le modèle A, et un deuxième graphique sur lequel est tracé la courbe des coûts moyens réels et celle des coûts moyens de pertes prédits par le modèle B. Dans une note d'étude de l'institut CAS du 13 janvier 2019, le lift est définit de la manière suivante :

Soit M=moyenne des primes prédites du dernier quantile et m= moyenne des primes prédites du premier quantile : $lift = M - m$. Après avoir calculé le lift pour chacun des modèles A et B, le meilleur est celui ayant le lift le plus élevé.

L'importance d'une telle étude est de savoir lequel des deux modèles est meilleur pour la conception d'un plan de tarification. Pour ce faire, on doit se baser sur certains critères : la monotonie croissante des coûts de sinistres lorsqu'on quitte d'un quantile à un autre plus élevé,

la précision du modèle qui est de juger l'écart entre les coûts réels et prédits dans chaque quantile, et la capacité de chaque modèle à distinguer les pires risques des meilleurs. Ainsi, entre A et B, le modèle à choisir est celui qui fournit la courbe la plus raide, qui est plus proche de la courbe des coûts de sinistres réels et produit le plus grand écart entre le premier et le dernier quantile.

On note de l'étude de Medina(2016) que les graphiques quantile simples peuvent se construire en faisant le tri soit par les coûts des pertes sous les taux réels, soit par les coûts prédits par le modèle. Un aperçu sur ses graphiques montre que lorsque le tri se fait suivant les coûts prédits par le modèle, les courbes des coûts de sinistres réels et prédits sont plus proches, leur croissance est plus rapide et la courbe des coûts prédits produit une grande différence entre les quantiles d'extrémité. Ainsi, pour mettre en place un meilleur plan de tarification, on choisirait le graphique qui s'obtient en triant les données suivant les coûts prédits par le modèle.

2.1.2.2 Les graphiques lift double

Contrairement aux graphiques quantiles simples, le tri ici est basé sur le ratio entre les coûts des sinistres attendus par les deux modèles. Soient $\widehat{\pi}_A$ *et* $\widehat{\pi}_B$ les coûts prédits par les modèles A et B respectivement, on peut définir le ratio de tri par $RA = \frac{\widehat{\pi}_A}{\widehat{\pi}_B}$. La construction de ces graphiques est également basée sur les quantiles. Lorsque deux modèles différents semblent avoir les mêmes performances, les graphiques lift doubles tendent à montrer visiblement un gagnant : c'est celui dont ces valeurs prédites sont plus proches des valeurs réelles ; ainsi, un modèle peut être meilleur dans un quantile et pire dans l'autre. La deuxième différence réside sur le fait que ces types de graphiques ont toujours trois courbes, une pour les coûts réels, une pour les coûts prédits par le modèle A et une pour les coûts prédits par le modèle B. L'avantage qu'un graphique lift double est qu'il montre clairement lequel des modèles est meilleur, ce qui n'est pas toujours le cas pour les diagrammes quantile simple (Tevet,2013).

Exemple : La figure suivante présente le graphique quantile simple illustrant la courbe des coûts réels du sinistre, la courbe des coûts prédits grâce au modèle GLM et celle des coûts prédits par le modèle GAM.

On remarque premièrement de cette figure que la courbe liée au GLM est décroissante, cela est dû au fait que la base de données étant triée suivant le ratio $RA = \frac{\widehat{\pi}_{GAM}}{\widehat{\pi}_{GLM}}$, les faibles valeurs de ce ratio s'obtiennent lorsque le dénominateur est est plus grand et le numérateur plus petit. Ensuite, on constate qu'à l'exception du quatrième quantile où les coûts prédits par le GLM sont meilleurs, ce diagramme favorise les coûts prédits par le GAM dans tous les autres

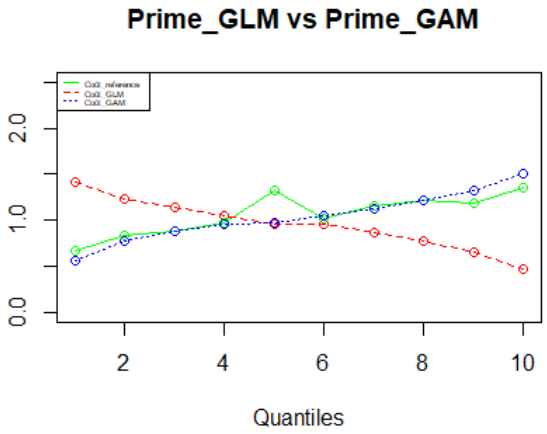

FIGURE 2.2 – Exemple d'un graphique quantile double

quantiles.

Medina(2016) propose une autre méthode de construction d'un diagramme lift double. Dans son étude, les données sont triées suivant le ratio entre les primes prédites et les primes actuelles, ensuite les données triées sont divisées en quantiles d'égales expositions.

2.1.2.3 L'index de Gini

Si la compréhension et l'interprétation des méthodes précédentes sont plus faciles, ce n'est pas le cas pour l'index de Gini. Ce dernier est plus complexe et son avantage est qu'il rapproche le lift à un seul nombre unique. Fortement utilisé en économie dans le cadre d'étude des inégalités dans la distribution des revenus, l'index de Gini se calcule à partir de la courbe de Lorenz. Dans le contexte économique, l'on cherche à obtenir des modèles dans lesquels la courbe de Lorenz est plus proche de la ligne de 45 dégrée (ligne d'égalité parfaite). Mais dans le cadre actuariel, l'on cherche à obtenir un modèle le plus inégalitaire possible, c'est-à-dire obtenir une courbe de Lorenz la plus éloignée possible de la ligne de 45 dégrée.

Si en économie sur les axes des X et des Y sont représentés respectivement les pourcentages cumulés des populations et des pourcentages des salaires, pour une étude actuarielle sur ces axes sont représentés respectivement les pourcentages cumulés d'exposition et les pourcentages cumulés de pertes. Soit $\mathcal{A}$ l'aire du domaine entre la ligne de 45 dégrée et la courbe de Lorenz et $\mathcal{B}$ l'aire du domaine entre la courbe de Lorenz et l'axe des X des abscisses, l'index de Gini I est définit par $I = \frac{\mathcal{A}}{\mathcal{A}+\mathcal{B}}$. Plus $\mathcal{A}$ est grand, plus le résultat est bon.

Plus l'indice de Gini augmente, plus on se rapproche vers des meilleures primes candidates. Bien que cet index ne permet pas d'évaluer la rentabilité d'un plan de notation particulière, il permet de quantifier la capacité d'un tel plan à distinguer les meilleurs risques des pires. Cet

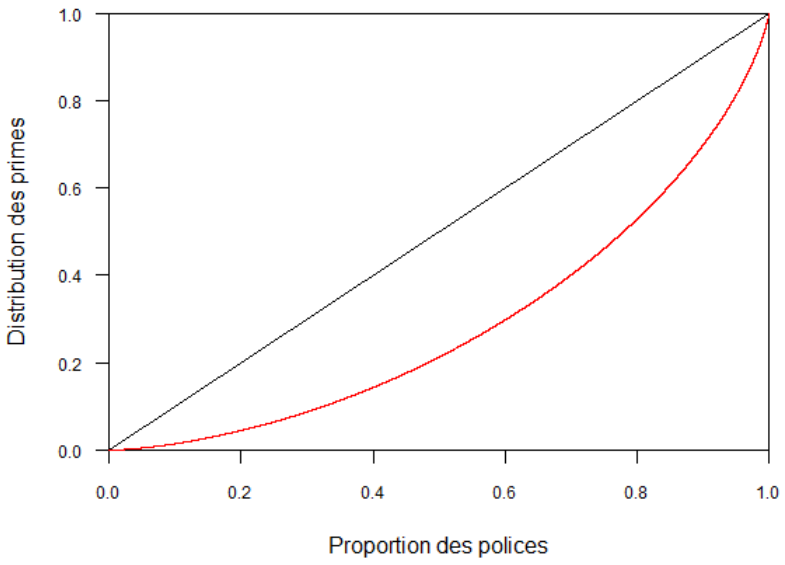

FIGURE 2.3 – Représentation d'une courbe de Lorenz

indice peut aussi être utilisé sur les bases de données. En effet, on note de l'étude de Denuit et al.(2019) que pour une variable aléatoire continue et strictement positive Z, il est possible de mesurer la variabilité sa variabilité de la manière suivante :

$$Gini[Z] = E[|Z_1 - Z_2|] = E[maxZ_1, Z_2] = E[minZ_1, Z_2]$$

avec Z_1 *et* Z_2 indépendants et de même distribution que Z. $Gini[Z]$ est étroitement liée à la variance, qui de manière équivalente, peut être exprimée par :

$$Var[Z] = \frac{1}{2}E[(Z_1 - Z_2)^2]$$

.

$Gini[Z]$ est lié à la différence absolue tandis que $Var(Z)$ est liée à la différence quadratique.

Si Z est continue, alors on peut montrer que $Gini[Z] = 4Cov[Z, F_Z(Z)]$.

Lorsqu'une séquence d'observations est classée par ordre croissant, la différence moyenne de Gini quantifie la relation entre la valeur réelle de Z et sa position dans la séquence. Plus la variabilité de Z est importante, plus sa valeur réelle est grande avec un rang élevé tandis qu'une différence moyenne de Gini plus faible indique que les observations sont plus concentrées autour de leur valeur centrale.

Cette formule met en relation la différence moyenne de Gini avec la courbe de Lorenz. En partant du plus petit écart absolu de Z_1 défini par $E[(t - Z_1)I[Z_1 \leq t]]$, en remplaçant t *par* Z_2 on a :

$$E[(Z_2 - Z_1)I[Z_1 \leq Z_2]] = \frac{1}{2}E[|Z_1 - Z_2|] = \frac{1}{2}Gini[Z] = 2Cov[Z, F_Z(Z)]$$

.

L'aire entre la ligne d'égalité et la courbe de Lorenz de Z est $\frac{1}{E[Z]}Cov[Z, F_Z(Z)]$.

Si la différence moyenne de Gini est élevée, alors la courbe de Lorenz du prédicteur s'éloigne de la ligne d'égalité. Telle est la raison pour laquelle le modèle qui prédit les primes présentant une plus grande différence moyenne de Gini tend à être choisi.

2.2 Mesures lift et courbes de concentrations

L'objectif de prédiction est de trouver une prime actuelle qui soit le plus proche possible de la prime pure, face à cette situation, Denuit et al. (2019) propose deux nouvelles mesures utiles en assurance, permettant de mesurer la capacité d'un modèle à prédire les coûts de sinistres : l'aire entre les courbes de Concentration et de Lorenz, noté ABC(Area Between the Curves) et la courbe de concentration intégrée, noté ICC(Integrated Concentration Curve). Dépendant toutes deux des courbes de concentration et de Lorenz, ces deux courbes mesurent aussi bien la capacité d'un modèle prédictif à différencier les pires et les meilleurs risques. Ce sont deux outils de lift qui s'interprètent intuitivement.

Il est question pour eux d'utiliser les courbes de concentration et de Lorenz $\alpha \mapsto CC[\mu(X), \pi(X); \alpha]$ *et* $\alpha \mapsto LC[\pi(X); \alpha]$ pour évaluer la capacité d'un modèle à prédire $\pi(X)$. En considérant $\mathfrak{P}_{ff}$ le portefeuille contenant les plus faibles valeurs de $\pi(X)$, la courbe de Lorenz est la proportion des primes perçues de $\alpha\%$ de polices de $\mathfrak{P}_{ff}$ et la courbe de concentration est la proportion du coût total des sinistres de $\mathfrak{P}_{ff}$. Ces deux proportions sont comparables puisque le revenu total attendu de π *et* μ coïncide avec la perte totale attendue vérifiant $E[\mu(X)] = E[\pi(X)] = E(Y)$.

2.2.1 L'aire entre les courbes (ABC)

Cette méthode nécessite à la fois l'étude de la courbe de concentration et de la courbe de Lorenz afin d'évaluer la variation de l'écart entre elles. Plus les deux courbes se rapprochent, plus on converge vers une meilleure prime actuelle $\pi(X)$ approchant la prime pure $\mu(X)$, avec la courbe de concentration qui dépend de la réponse Y *et* $\pi(X)$ et la courbe de Lorenz qui dépend uniquement de $\pi(X)$. L'aire du domaine délimité par les droites d'équations $x = 0$, $x = 1$, la courbe de concentration (CC) et la courbe de Lorenz (CL) est cependant une mesure de la

qualité d'ajustement. Elle est notée par $ABC(\pi(X))$ et est définie par :

$$\begin{aligned}
ABC[\pi(X)] &= \int_0^1 (CC[Y,\pi(X);\alpha] - LC[\pi(X);\alpha])d\alpha \\
&= \tfrac{1}{E[\pi(X)]} \int_0^1 (E[Y I[\Pi \leq \alpha]] - E[\pi(X) I[\Pi \leq \alpha]])d\alpha \\
&= \tfrac{1}{E[\pi(X)]} \int_0^1 \int_0^\infty (P[\pi(X) \leq y, \Pi \leq \alpha] - P[Y \leq y, \Pi \leq \alpha])dyd\alpha \\
&= \tfrac{1}{E[\pi(X)]} (Cov[\pi(X), \Pi] - Cov[Y, \Pi]) \\
&= \tfrac{1}{E[\pi(X)]} (Cov[\pi(X), F_\pi(\pi(X))] - Cov[Y, F_\pi(\pi(X))])
\end{aligned}$$

Il ressort de cette relation que, $ABC[\pi(X)]$ dépend à la fois du Gini du prédicteur $\pi(X)$ et de la covariance de la réponse Y et ce prédicteur.

2.2.2 Intégrale des courbes de concentration et de Lorenz

Il est possible de faire face à une situation où deux prédicteurs n'ont pas un ordre constant, un tel cas se produit lorsque les courbes de concentration et de Lorenz se croisent. Ainsi, un prédicteur peut être meilleur pour les risques faibles est pire pour les risques élevés. Ceci étant, on peut faire appel à la notion d'intégrale pour une comparaison plus efficace des courbes. Pour un niveau de probabilité α donné, cette intégrale est définie par :

$$\begin{aligned}
ICC[\mu(X), \pi(X); \alpha] &= \int_0^\alpha CC[\mu(X), \pi(X); \xi]d\xi \\
&= \int_0^\alpha \frac{E[\mu(X) I(\Pi) \leq \xi)]}{E[Y]} \\
&= \tfrac{E[\mu(X)(\alpha - \Pi)_+]}{E[Y]} \\
&= \tfrac{cov[\mu(X), (\alpha - \Pi)_+]}{E[Y]} + E[(\alpha - \Pi)_+]
\end{aligned}$$

Pour $\alpha = 1$,

$$\begin{aligned}
ICC &= ICC[\mu(X), \pi(X); 1] \\
&= \frac{cov[\mu(X), 1 - \Pi]}{E[Y]} + \frac{1}{2} \\
&= \frac{1}{2} - \frac{Cov[\mu(X), \Pi]}{E[Y]}
\end{aligned}$$

Or $\frac{Cov[\mu(X),\Pi]}{E[Y]} = \frac{Cov[\mu(X),F_\pi(\pi(X))]}{E[\pi(X)]}$, si de plus $\mu(X) = \pi(X)$, on a :

$$\begin{aligned}
\frac{Cov[\mu(X), \Pi]}{E[Y]} &= \frac{1}{2} - \frac{Cov[\pi(X), F_\pi(\pi(X))]}{E[\pi(X)]} \\
&= \frac{1}{2} - Gini(\pi(X))
\end{aligned}$$

On constate que si on connait $\mu(X)$, alors l'ICC dépend du Gini de $\pi(X)$.

$E[Y(\alpha-\Pi)_+] = E[E[Y(\alpha-\Pi)_+|X]] = E[\mu(X)(\alpha-\Pi)_+],\ car\ \mu(X) = E(Y|X)$

Grâce à la courbe de concentration intégrale, on peut mesurer la performance de $\pi(X)$ à prédire $\mu(X)$. Comme ICC dépend de $Cov(\mu(X),(\alpha-\Pi)_+)$ pour α fixé alors, plus Π est petit par rapport à α, plus $\pi(X)$ est petite.

Lorsque le ratio $\frac{Cov[(\mu(X)),(\alpha-\pi)_+]}{E[Y]}$ se rapproche de -1, on obtient une meilleure prime candidate.

Pour ce qui est de la courbe intégrale de Lorenz, elle est définie par :

$ILC[\pi(X);\alpha] = \frac{Cov[\pi(X),(\alpha-\Pi)_+]}{E[\pi(X)]} + E[(\alpha-\Pi)_+]$.

Pour $\alpha < \Pi$ on a :

$$\begin{aligned} ILC[\pi(X);\alpha] &= \frac{Cov[\pi(X),(\Pi-\alpha)]}{E[\pi(X)]} + E[\Pi-\alpha] \\ &= \frac{Cov[\pi(X),\Pi]}{E(\pi(X))} + E[\Pi] - \alpha \\ &= \frac{\pi(X), F_\pi(\pi(X))}{E(\pi(X))} + E(F_\pi(\pi(X))) - \alpha \\ &= Gini(\pi(X)) + E(F_\pi(\pi(X))) - \alpha. \end{aligned}$$

On remarque que l'ILC dépend du Gini de $\pi(X)$.

2.2.3 Dépendance entre $\pi(X)$ et $\mu(X)$

Puisque l'on désire approcher la prime pure inconnue $\mu(X)$ avec $\pi(X)$, il est important d'étudier la dépendance entre elles. L'objectif de ce paragraphe est de considérer plusieurs distributions pour l'espérance conditionnelle $\mu(X)$, et de choisir celle qui colle mieux à nos données. Denuit et al. (2019) utilisent à titre d'exemple les copules de Clayton et de Frank pour l'étude de cette dépendance. Ces copules se définissent respectivement par :

$C_\theta(u,v) = (u^{-\theta} + v^{-\theta} - 1)^{-\frac{1}{\theta}},\ \theta > 0$ et

$C_\theta(u,v) = -\frac{1}{\theta} ln(1 + \frac{(exp(-\theta u)-1)(exp(-\theta v)-1)}{exp(-\theta)-1}),\ \theta \neq 0.$

Dans ces deux expressions le paramètre θ sert de mesure de lien de dépendance entre les deux variables. Quelle que soit le copule, on peut utiliser le taux de Kendall pour la corrélation.

2.2.4 Variations de l'ABC et de l'ICC

La variabilité des courbes de performances dépend de la loi de probabilité utilisée pour la modélisation des primes. Partrat et Besson (2005), grâce à plusieurs tests statistiques ont

retenu, les lois Gamma, Log-Normale et Weibull pour la modélisation des coûts de sinistres. Ainsi pour notre étude, nous utiliserons l'une de ces trois lois, mais ce choix ne sera pas aléatoire, la variabilité des courbes de performances aidera à choisir. On considère dans cette section $\mu(X)$ *et* $\pi(X)$ suivent soit la distribution de Gamma, soit celle de Log-Normale ou encore celle de Weibull suivant les variations de la variance et un taux de Kendall constant. L'idée finale étant de faire une comparaison de l'aire entre les courbes de concentration et de Lorenz afin de choisir la distribution qui donne le meilleur candidat.

Cette section avec les copules et les Gamma, Weibull et LogNormale n'est qu'un exemple illustratif.

2.2.4.1 Impact de la variance avec le copule de Clayton pour $\tau = 0.5$

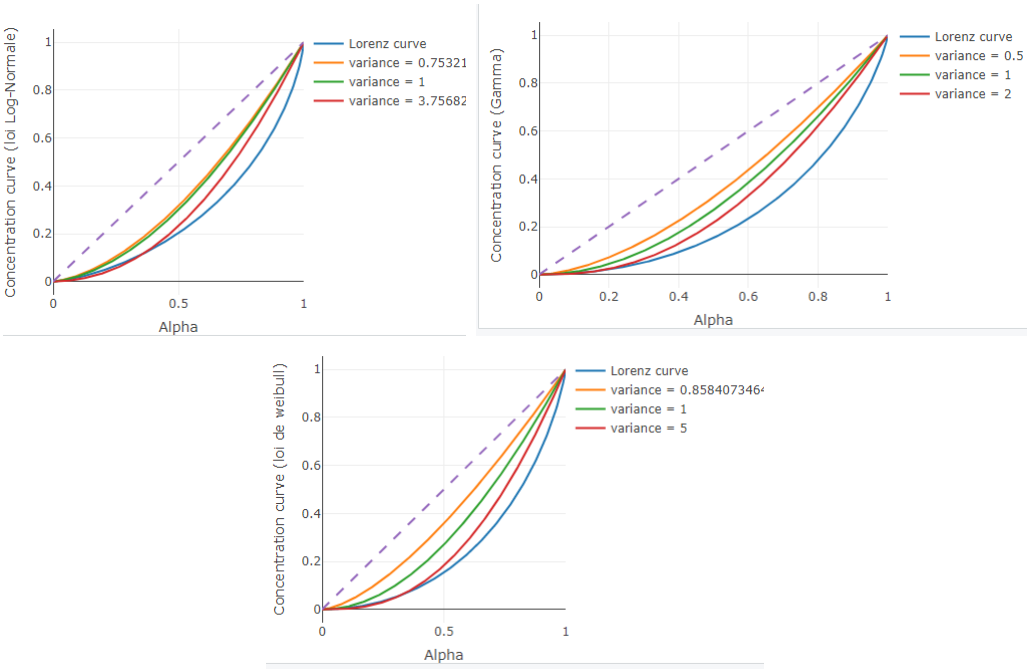

FIGURE 2.4 – Représentation des courbes de concentration et de Lorenz pour $\tau = 0.5$

On remarque que les courbes de concentration ne se coupent pas pour les trois distributions, ceci est dû à l'ordre convexe entre les distributions de $\mu(X)$ considérées.

À partir de ces graphiques, on obtient les tableaux suivants présentant la variation de l'aire entre la courbe de Lorenz et la courbe de concentration suivant les valeurs prises par la variance pour les trois lois. En supposant que les prédicteurs sont distribués de manière identique et ont des performances similaires en termes de dépendance avec la prime réelle, la métrique ABC choisie la plus variable de la prime réelle.

En effet, on constate de ces figures que quel que soit la loi considérée, plus la variance est grande, plus les courbes de concentration et de Lorenz se rapprochent, d'où une réduction des

mesures ABC et ICC. Lorsque la variance est égale à 1 pour les trois lois, la loi Log-Normale fournit la plus petite valeur de ABC, la plus grande valeur de ICC et la plus grande intégrale de la courbe de Lorenz (27.80% contre 25% pour les autres). Ainsi, bien que la loi Log-Normale fournisse les petites valeurs de ABC, son choix ne nous rassure pas, car on voudrait que la courbe de Lorenz soit le plus éloignée possible de la ligne d'égalité.

Couleur	$\pi(X)$	$\mu(X)$	ABC	ICC
Rouge	$\mathcal{LN}or(-\frac{ln2}{2}, \sqrt{ln2})$	$\mathcal{LN}or(-\frac{(1.5\sqrt{ln2})^2}{2}, 1.5\sqrt{ln2})$	4.39%	32.19%
Verte	$\mathcal{LN}or(-\frac{ln2}{2}, \sqrt{ln2})$	$\mathcal{LN}or(-\frac{ln2}{2}, \sqrt{ln2})$	8.78%	36.58%
Orange	$\mathcal{LN}or(-\frac{ln2}{2}, \sqrt{ln2})$	$\mathcal{LN}or(-\frac{(0.9\sqrt{ln2})^2}{2}, 0.9\sqrt{ln2})$	9.85%	37.65%
$ILC = 27.80\%$				

TABLE 2.1 – Variation de ABC et ICC suivant la loi Log-Normale $\tau = 0.5$

Couleur	$\pi(X)$	$\mu(X)$	ABC	ICC
Rouge	$Weibull(1,1)$	$Weibull(1,2)$	4.11%	29.11%
Verte	$Weibull(1,1)$	$Weibull(1,1)$	9.66%	34.66%
Orange	$Weibull(1,1)$	$Weibull(1,0.5)$	15.31%	40.31%
$ILC \approx 25\%$				

TABLE 2.2 – Variation de ABC et ICC suivant la loi Weibull $\tau = 0.5$

Couleur	$\pi(X)$	$\mu(X)$	ABC	ICC
Rouge	$Gam(1,1)$	$Gam(1,2)$	6.33%	31.33%
Verte	$Gam(1,1)$	$Gam(1,1)$	9.66%	34.66%
Orange	$Gam(1,1)$	$Gam(1,0.5)$	13.08%	38.08%
$ILC \approx 25\%$				

TABLE 2.3 – Variation de ABC et ICC suivant la loi Gamma $\tau = 0.5$

Dans de ces tableaux, on suppose que $\pi(X)$ *et* $\mu(X)$ suivent la même distribution.

2.2.4.2 Variation du tau de Kendall suivant les trois lois

Les graphiques suivants présentent la courbe de Lorenz et les courbes de concentration pour différentes valeur du tau de Kendall suivant trois lois (Log-Normale, Weibull et Gamma). Il est question dans ce paragraphe de maintenir la distribution du prédicteur et la distribution de la

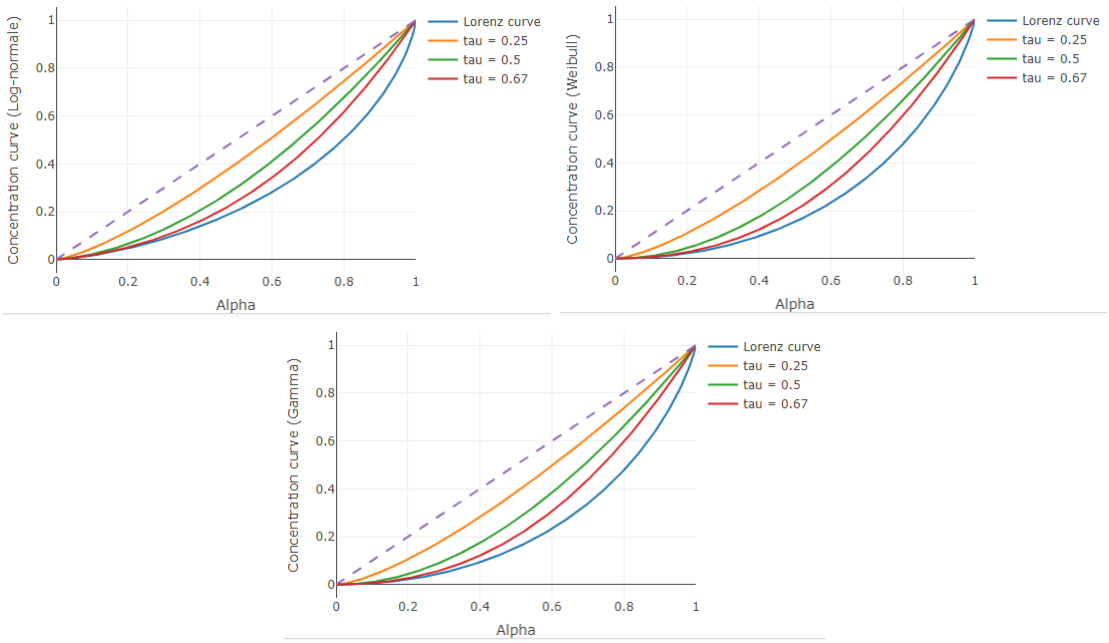

FIGURE 2.5 – Représentation des courbes de concentration et de Lorenz suivant les valeurs de τ

prime réelle, d'isoler l'effet de la force de la dépendance en considérant le copule pour plusieurs valeurs de paramètres correspondant à différents coefficients du tau de Kendall.

Nous notons de ces figure que plus la dépendance est forte, plus les courbes de concentration et de Lorenz se rapprochent, donc une réduction de ABC et ICC pour chaque loi. On constate en plus que les lois $Gam(1,1)$ *et* $Weibull(1,1)$ coïncident pour chaque valeur de τ. Si les plus petites valeurs de ABC s'obtiennent de la loi Log-Normale pour les différentes valeurs de τ considérées, elle fournie les plus grandes valeurs pour ce qui est de l'ICC. Par ailleurs, au-delà de $\tau = 0.67$, on note une divergence des intégrales pour la loi Log-Normale. Comme pour le cas de la variation de la variance, la courbe de Lorenz obtenue par cette dernière loi est la plus proche de la ligne d'égalité.

Couleur	$\pi(X)$	$\mu(X)$	τ	ABC	ICC
Rouge	$\mathcal{LN}or(-\frac{ln2}{2}, \sqrt{ln2}))$	$\mathcal{LN}or(-\frac{ln2}{2}, \sqrt{ln2})$	0.67	4.91%	32.71%
Verte	$\mathcal{LN}or(-\frac{ln2}{2}, \sqrt{ln2})$	$\mathcal{LN}or(-\frac{ln2}{2}, \sqrt{ln2})$	0.5	8.78%	36.58%
Orange	$\mathcal{LN}or(-\frac{ln2}{2}, \sqrt{ln2}))$	$\mathcal{LN}or(-\frac{ln2}{2}, \sqrt{ln2})$	0.25	15.23%	43.03%
$ILC = 27.80\%$					

TABLE 2.4 – Loi Log-Normale et Variation de ABC et ICC suivant les valeurs de τ

Couleur	$\pi(X)$	$\mu(X)$	τ	ABC	ICC
Rouge	$Weibull(1,1)$	$Weibull(1,1)$	0.67	5.26%	30.26%
Verte	$Weibull(1,1)$	$Weibull(1,1)$	0.5	9.66%	34.66%
Orange	$Weibull(1,1)$	$Weibull(1,1)$	0.25	17.04%	42.04%
$ILC \approx 25\%$					

TABLE 2.5 – Loi Weibull et variation de ABC et ICC suivant les valeurs de τ

Couleur	$\pi(X)$	$\mu(X)$	τ	ABC	ICC
Rouge	$Gam(1,1)$	$Gam(1,1)$	0.67	5.26%	30.26%
Verte	$Gam(1,1)$	$Gam(1,1)$	0.5	9.66%	34.66%
Orange	$Gam(1,1)$	$Gam(1,1)$	0.25	17.04%	42.04%
$ILC \approx 25\%$					

TABLE 2.6 – Loi Gamma et variation de ABC et ICC suivant les valeurs de τ

2.2.4.3 Impact de la distribution pour $\tau = 0.5$

La figure suivante montre l'impact de la distribution sur les mesures ABC et ICC pour une valeur fixe du tau de Kendall lorsque la prime $\pi(X)$ suit la loi Gamma de paramètres 1 *et* 1, et la prime $\mu(X)$ suit des distributions variées. D'après cette figure, les courbes de concentration coïncident lorsque $\mu(X)$ suivant $Gam(1,1)$ *et* $Weibull(1,1)$ et on obtient dans ce cas les plus petites valeurs des mesures ABC et ICC.

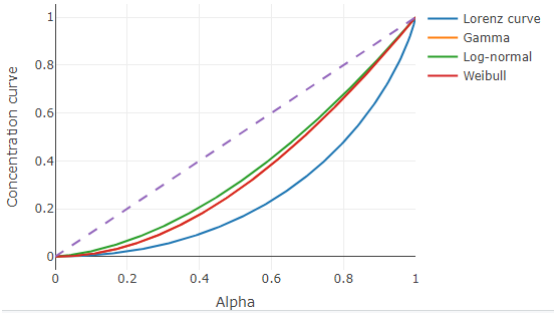

FIGURE 2.6 – Représentation des courbes de concentration et de Lorenz suivant les différentes distributions

Le tableau suivant nous donne plus de visibilité quant à la lecture de la figure et l'évaluation quantitatives des mesures ABC et ICC.

La figure suivante montre l'impact de la distribution sur les mesures ABC et ICC pour une valeur fixe du tau de Kendall, lorsque la prime $\pi(X)$ suit la loi Log-Normale de paramètres $-\frac{ln2}{2}$ *et* $\sqrt{ln2}$, et la prime $\mu(X)$ suit des distributions variées. Comme précédemment, on note

Couleur	$\pi(X)$	$\mu(X)$	ABC	ICC
Rouge	$Gam(1,1)$	$Weibull(1,1)$	9.66%	34.66%
Verte	$Gam(1,1)$	$\mathcal{LN}or(-\frac{ln2}{2}, \sqrt{ln2})$	11.58%	36.58%
Orange	$Gam(1,1)$	$Gam(1,1)$	9.66%	34.66%
$ILC \approx 25\%$				

TABLE 2.7 – Loi Gamma et variation de ABC et ICC suivant les valeurs de τ

une coïncidence des courbes de concentration lorsque $\mu(X)$ suit les lois $Gam(1,1)$ *et* $Weibull(1,1)$. Quand $\mu(X)$ est de même loi que $\pi(X)$, la courbe de concentration est plus éloignée de la courbe de Lorenz.

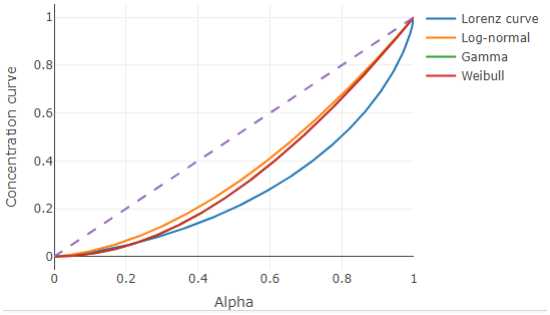

FIGURE 2.7 – Impact de la distribution de $\mu(X)$ sur la variation de ABC

Un résumé du présent cas est représenté par le tableau suivant. Une comparaison de ce tableau et le tableau précédent montre que les plus petites valeurs de ABC sont obtenues lorsque $\pi(X)$ suit la loi Log-Normale de paramètres $-\frac{ln2}{2}$ *et* $\sqrt{ln2}$, mais que $\pi(X)$ suive la loi Log-Normale de paramètres $-\frac{ln2}{2}$ *et* $\sqrt{ln2}$ ou $Gam(1,1)$, les valeurs de ICC restent inchangées.

Couleur	$\pi(X)$	$\mu(X)$	ABC	ICC
Rouge	$\mathcal{LN}or(-\frac{ln2}{2}, \sqrt{ln2})$	$W(1,1)$	6.85%	34.66%
Verte	$\mathcal{LN}or(-\frac{ln2}{2}, \sqrt{ln2})$	$Gam(1,1)$	6.85%	34.66%
Orange	$\mathcal{LN}or(-\frac{ln2}{2}, \sqrt{ln2})$	$\mathcal{LN}or(-\frac{ln2}{2}, \sqrt{ln2})$	8.78%	36.58%
$ILC = 27.80\%$				

TABLE 2.8 – Loi Gamma et variation de ABC et ICC suivant les valeurs de τ

Nous notons de cette section que la loi Gamma est la meilleure, elle permet non seulement d'avoir les plus petites valeurs de ABC, mais aussi une intégrale de la courbe de Lorenz plus petite.

Chapitre 3

Applications : Assurance de la Responsabilité civile automobile en France

Dans ce chapitre, il est question d'une application grâce aux données relatives à l'assurance RC automobile. Les jeux de données sur lesquels nous effectuons nos analyses sont issus de la base de données CASdatast (Computational Acturial Science). Il s'agit des bases freMTPL2freq et freMTPL2sev (French Motor Third Parts Liability datasets). La première nous renseigne sur les caractéristiques de risque collectées pour environ 677 990 polices de la RC automobile, elle compte douze variables :

— Policy ID : numéro de la police ;

— ClaimNb : le nombre de sinistres durant la période d'exposition ;

— Exposure : la période d'exposition pour la police en année ;

— VehPower : la puissance du véhicule ;

— VehAge : l'âge du véhicule en année ;

— DrivAge : l'âge du conducteur en année ;

— VehBrand : la marque du véhicule ;

— VehGas : le type d'essence utilisé dans le véhicule ;

— Region : la région de résidence de l'assuré ;

— Area : le code de la zone de résidence ;

— BonusMalus : le Bonus Malus pour chaque assuré ;

— Density : le nombre d'habitants par km^2 dans la ville où vit le conducteur du véhicule.

Quant à la deuxième base de données, elle a 26639 polices et ne contient que deux variables : policy ID qui est le numéro de la police et ClaimAmount qui est le coût de sinistre.

L'idée de notre étude étant de modéliser les coûts moyens et la fréquence de sinistres en

fonction des caractéristiques d'assurés, il est nécessaire de fusionner les deux bases de données. La fusion est faite grâce à la fonction merge() de R et la variable de fusion est le numéro de la police qui appartient aux deux bases. Pour les polices qui ne figurent pas dans la base freMTPLsev, on leur attribue un coût nul. La base finale de notre étude contient 13 variables et 677 990 polices observations, elle montre par ailleurs que le nombre total de sinistre déclarés est de 34060 correspondant à 5.02% des sinistres. Avant de passer aux différentes analyses, nous effectuerons dans un premier temps une analyse exploratoire des variables d'étude.

3.1 Statistiques descriptives

3.1.1 Variables quantitatives

Une description générale de la base de données montre que les variables n'ont pas de données manquantes. La figure suivante montre les boites à moustaches des différentes variables quantitatives.

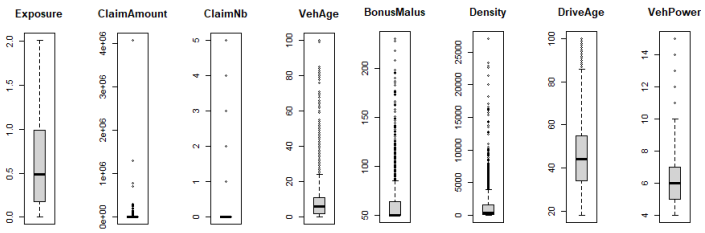

FIGURE 3.1 – Box plot des variables quantitatives

Cette figure montre que chacune des variables quantitatives présente des valeurs aberrantes. Les données peuvent ainsi être traitées dans le but de diminuer ces dernières. Pour cela, nous allons utiliser une fonction définissant une marge pour cette étude. Dans cette étude, les polices d'exposition nulle ne sont pas considérées. Il en est de même celles dont le nombre de sinistres est supérieure à 5. On pose pour une variable quelconque $marge = 1.5 \times IQR$, $a = quantile(variable, 0.05) - marge$ *et* $b = quantile(variable, 0.95) + marge$.
Toute valeur inférieure à a est remplacée par
$quantile(variable, 0.05)$ tandis que toute valeur supérieure à b est remplacée par
$quantile(variable, 0.95)$ de la variable considérée. Cette fonction nous permet de réduire les valeurs aberrantes. Pour le coût moyen, la fonction pmin() de R permet de remplacer tous les coûts moyens supérieur à 10000 par 10000, la même fonction est utilisée pour remplacer toute prime pure observée supérieure à 8000 par 8000.

Le tableau suivant présente pour les variables quantitatives, le minimum, le maximum et la moyenne.

	VarAvgClaimAm	ClaimNb	VehAge	DrivAge	Density	Exposure	BonusMa
Min	0.00	0.00	0.00	18.00	1.00	0.0027	50.00
mean	54.34	0.05314	7.008	45.5	1401	0.529	59.58
Max	10000.0	5.00	30.00	100.00	9589	2.01	116.00

TABLE 3.1 – Statistiques descriptives des variables quantitatives

Nous présentons ci-dessous, les histogrammes des différentes variables quantitatives.

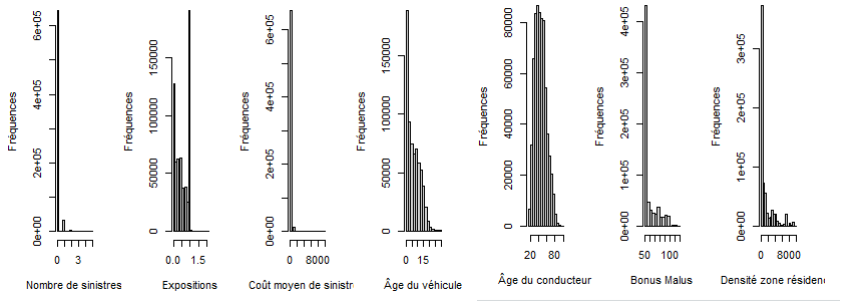

FIGURE 3.2 – Histogrammes des variables quantitatives

Nous notons du graphique précédent que la plupart d'assurés n'ont pas déclaré de sinistres durant la période d'exposition. Pour le coût moyen de sinistres, l'intervalle $[0; 500]$ concentre la majorité de polices tandis que pour l'exposition, l'intervalle $[0.9; 1.0]$ concentre la majorité de polices. Les véhicules nouvellement mis en circulation ont eu plus de sinistres par rapport à d'autres et les conducteurs dont l'âge varie entre 35 et 40 ans ont déclaré plus de sinistres. Quant à la densité, les conducteurs résident dans les régions ayant les plus petites densités concentrent la majorité de polices.

3.1.2 Variables qualitatives

Le graphique suivant est celui des barplot décrivant les variables qualitatives.

La variable Area a six modalités, la zone C concentre la plus grande proportion de polices (28.3%) suivie respectivement des zones D(22.36%), E(20.23%), A(15.22%), B(11.13%) et F(2.65%). Quant au type de gaz, 51.01% de polices sont déclarées par les utilisateurs de véhicules de type diesel tandis que 48.99% sont de type régulier. La marque du véhicule B12 a la plus grande fréquence (24.49%), suivie de la marque B1 (24.00%), tandis que la marque B14

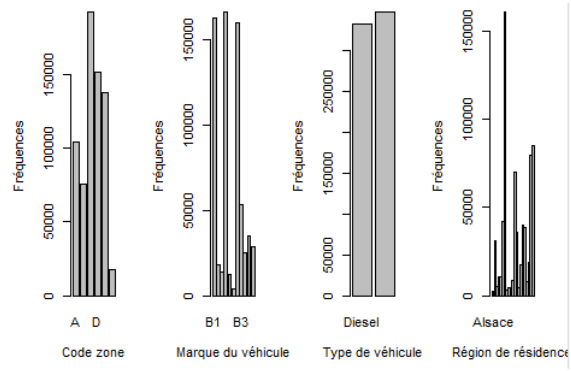

FIGURE 3.3 – Description des variables qualitatives

a la plus faible fréquence (0.6%). Pour la variable Region, elle a 21 modalités. La région du centre a la plus grande fréquence (23.69%), ensuite viennent les régions Rhone-Alpes (12.5%) Provence-Alpes-Cotes-D'Azur (11.9%), Ile-de-France (10.29%), la région Franche-Comte quant à elle a la plus petite fréquence (0.2%).

3.1.3 Corrélation

Dans le but d'étudier la relation entre les variables quantitatives de l'étude, nous allons analyser la corrélation entre ces variables. En effet, cette analyse permet de déterminer si la relation entre deux variables données est linéaire, non linéaire, positive ou négative. Le coefficient de corrélation de Pearson permet de mesurer la dépendance linéaire entre deux variables. Les coefficients de corrélation de Pearson entre certaines variables de l'étude sont donnés dans le tableau ci-dessous. Une capture du résultat obtenu pour notre analyse est présentée ci-dessous.

	VehAge	DrivAge	BonusMalus	Density	ClaimNb	Exposure	AvgClaimAmount
VehAge	1.00	-0.06	0.08	-0.09	-0.02	0.15	0.00
DrivAge	-0.06	1.00	-0.49	-0.03	0.01	0.17	0.00
BonusMalus	0.08	-0.49	1.00	0.11	0.05	-0.19	0.04
Density	-0.09	-0.03	0.11	1.00	0.01	-0.10	0.01
ClaimNb	-0.02	0.01	0.05	0.01	1.00	0.07	0.54
Exposure	0.15	0.17	-0.19	-0.10	0.07	1.00	0.04
AvgClaimAmount	0.00	0.00	0.04	0.01	0.54	0.04	1.00

FIGURE 3.4 – Tableau des coefficients de corrélation linéaire

Nous constatons par exemple que les variables BonusMalus et DrivAge sont négativement corrélées, le coefficient de corrélation entre elles est de 0.49 en valeur absolue. On note de plus qu'il n'existe aucune corrélation entre les variables DrivAge et le coût moyen. Il en est de même pour la variable VehAge et le coût moyen. La plus grande corrélation est celle des variable ClaimNb et AvgClaimAmount (0.54).

3.1.4 Dépendance entre les variables qualitatives

Pour mesurer l'association entre deux variables qualitatives, on utilise le test V de Cramér. Le V de Cramér dépend du Khi2 qui permet d'établir s'il existe un effet entre deux variables qualitatives croisées dans un tableau de contingence. Étant donné deux variables qualitatives X1 et X2 avec plusieurs modalités chacune, ce test permet de comparer l'intensité du lien entre elles. La valeur absolue du test V de Cramér prend ses valeurs entre 0 *et* 1, où 0 correspond au cas d'une absence de lien et 1 au cas d'une colinéarité entre les deux variables. Le tableau suivant présente les valeurs du test V de Cramér pour les variables qualitatives de notre étude.

	Area	VehBrand	VehGas	VehPower	Region
Area	1	0.07340852	0.1314644	0.03976159	0.3184873
VehBrand	0.07340852	1	0.1165859	0.1575166	0.1299584
VehGas	0.1314644	0.1165859	1	0.3187086	0.087421
VehPower	0.03976159	0.1575166	0.3187086	1	0.04361352
Region	0.3184873	0.1299584	0.087421	0.04361352	1

TABLE 3.2 – Les Valeurs du test V de Cramér entre les variables qualitatives

Nous notons de ce tableau une absence de liaison entre les variables Region et VehPower d'une part et entre les variables Area et VehPower d'autre part. Entre les variables VehBrand et Area d'une part et les variables Region et VehGas d'autre part, il y a une liaison très faible. Le plus grand effet est entre les variables VehGas et VehPower où il existe une dépendance modérée, on note également une dépendance modérée entre les variables Area et Region. Nous notons qu'il n'existe pas de colinéarité, ni une forte dépendance entre les variables qualitatives de notre base. Toutes ces variables seront utilisées dans nos différentes régressions.

3.2 Modélisation

Pour effectuer les analyses, notre base de données est divisée en deux, un sous-jeu pour le traitement et un autre pour le test. Le premier contient 80% d'observations et le reste est utilisé pour test.

Dans le but de choisir le modèle qui prédirait le mieux la prime pure d'assurance, nous allons étudier trois modèles à savoir, les modèles GLM (Generalized Linear Models), les modèles GAM (Generalized Additive Models) et les modèles GBM (Gradient Boosting Machine). Les régressions faites dans cette partie se basent sur l'équation (1.2), Y est tantôt le nombre de

sinistres, tantôt le coût moyen de sinistres, et X le vecteur des caractéristiques des assurés. On peut obtenir la prime pure d'assurance en multipliant la fréquence de sinistres par le coût moyen de sinistres.

prime pure=fréquence× Coût moyen. Ainsi, pour chaque modèle cité ci-haut, deux régressions seront faites, l'une pour la fréquence et l'autre pour le coût moyen de sinistres. Notons $\widehat{N}_i$ *et* $\widehat{C}_i$ respectivement la fréquence de sinistres estimée et le coût moyen de sinistres de l'assuré i, alors la prime pure estimée de cet assuré est $\widehat{PP}_i = \widehat{N}_i \times \widehat{C}_i$.

Les déviances utilisées ici sont celles de poisson et de gamma définies respectivement par : $D_{poisson} = 2\sum_{i=n}^{n}[y_i log(\frac{y_i}{\widehat{\mu}_i}) - (y_i - \widehat{\mu}_i)]$ $D_{gamma} = 2\sum_{i=n}^{n}[-log(\frac{y_i}{\widehat{\mu}_i}) + \frac{y_i - \widehat{\mu}_i}{\widehat{\mu}_i}]$, en respectant la condition que si $y_i = 0$, alors, $ylog(y) = 0$. Contrairement à la première partie, les courbes de concentration et de Lorenz de cette partie sont des versions empiriques, elles se construisent à partir des données disponibles. Dans le but d'étudier l'influence des variables explicatives des modèles, nous allons commencer par le GBM qui nous permet d'évaluer l'importance de chaque variable à l'explication de la variable réponse. Les métriques telles que ICC, ILC, ABC et Indice de Gini seront calculés en utilisant la base de données test.

3.2.1 Régressions par le modèle GBM

Le GBM est l'un des modèles permettant d'étudier l'impact de certaines caractéristiques d'assurés sur le coût moyen ou le nombre de sinistres. Par rapport aux modèles GLM, les modèles GBM permettent d'avoir plus d'informations sur les données. Ce qui apporte un plus sur la capacité d'interprétation des résultats du modèle mise en place. L'idée du Boosting est de construire une prédiction puissante grâce à la combinaison de plusieurs prédictions faibles. Dans Denuit et al. (2019), on note qu'il est question de construire une estimation $\widehat{y}_i$ de l'espérance $\mathbb{E}(Y_i)$ connaissant l'information x_i. Cet estimateur est une correspondance non linéaire de $\widehat{x}_i$ à $\widehat{y}_i$ tel que : $\widehat{y}_i = F(\widehat{x}_i\ )$. Le meilleur estimateur est celui qui minimise la moyenne de la fonction de perte dans l'ensemble du portefeuille, il est défini par :

$$F^*(.) = argmin_F \frac{1}{n}\sum_{i=1}^{n}\mathcal{L}(Y_i, F(x_i)).$$

Où $\mathcal{L}$ peut être une fonction de déviances ou toute autre fonction de pénalité bien lisse. $F : \mathcal{X} \to \mathbb{R}$ et il est juste nécessaire d'optimiser les paramètres définis par $F = (F_1, ..., F_n)' = (F(x_1, ..., F(x_n)' \in \mathbb{R}^n$ qui minimisent la perte $\mathcal{L}_L(F) = \frac{1}{n}\sum_{i=1}^{n}\mathcal{L}(Y_i, F_i)$. Trouver une solution globale est un exercice très complexe, et dans le but d'améliorer localement une approximation courante un algorithme itératif est utilisé.

— Initialisation : Choisir un estimateur initial $F_0^*(x) = \rho_0 = argmin_\rho L(Y_i, \rho)$.

— Itération : pour $m = 1, \cdots, M$,

- Calcul les réponses $R_{m,i}$ pour $i = 1, ..., n$ de la fonction de perte : $R_{m,i} = -\frac{\partial L(Y_i, F_i)}{\partial F_i}|_{F_i = F_m^*(x_i)}$;
- Ajuster un modèle de régression $\widehat{g} : \mathcal{X} \to \mathbb{R}$ pour les réponses $(R_{m,i})_{i=1,...,n}$ aux couples $(x_i, v_i)_{i=1,\cdot,n}$ des variables et expositions.
- Calculer ρ_m tel que : $\rho_m = argmin_\rho \frac{1}{n} \sum_{i=1}^{n} L(Y_i, \widehat{F}_{m-1}(X_i) + \rho \widehat{g}_m(x_i))$;
- Mettre à jour : $\widehat{F}_{m+1}(x) = \widehat{F}_m(x) + \rho_m \widehat{g}_m(x)$.

— Retourner le l'estimateur : $x \to \widehat{F}(x) = \widehat{F}_M(x)$.

3.2.1.1 Estimations et importance des variables

Pour les modélisations, nous utilisons le package gbm3 de Florian Pechon et optons pour une validation croisée avec $cv_folds = 5$, pour le modèle de fréquence, nous avons considéré quadruplet $(numtrees, shrinkage, interactiondepth, minnumobsinnode) = (5000, 0.01, 5, 100)$ comme les hyperparamètres et le nombre d'arbres optimal est trouvé grâce à la fonction gbmt-performance du package gbm3. Pour la modélisation du coût moyen,quadruplet $(numtrees, shrinkage, interactiondepth, minnumobsinnode) = (1500, 0.01, 5, 100)$ est utilisé pour hyperparamètres, et comme précédemment, le nombre d'arbres optimal est obtenu grâce à la fonction gbmt-performance.

La régression GBM sur le nombre de sinistres montre que toutes les variables explicatives considérées ont chacune une importance non nulle à l'explication de cette variable. L'ordre d'importance de ces variables est le suivant : VehAge (41.65%), DrivAge (13.23%), VehBrand (12.99%), Region (11.26%), logdensity(6.39%), VehPower (6.35%), VehGas (4.65%), et Area(3.45%). En ce qui concerne l'explication du coût moyen, la variable la plus importante est Region (25.73%) suivies respectivement de VehBrand (6.92%), DrivAge(15.05%), logdensity(13.71%), VehAge(11.80%) , Area(8.0%), VehPower(7.12%) et VehGas(1.66%).

La figure suivante montre l'importante des différentes variables explicatives (modèle de fréquence à gauche et celui du coût moyen à droite) à l'explication du coût de sinistres. Grâce aux régressions effectuées, on obtient la prime estimée par le modèle GBM, une description de cette prime montre qu'elle varie entre 14.65 *et* 2812.02 avec une moyenne de 166.03.

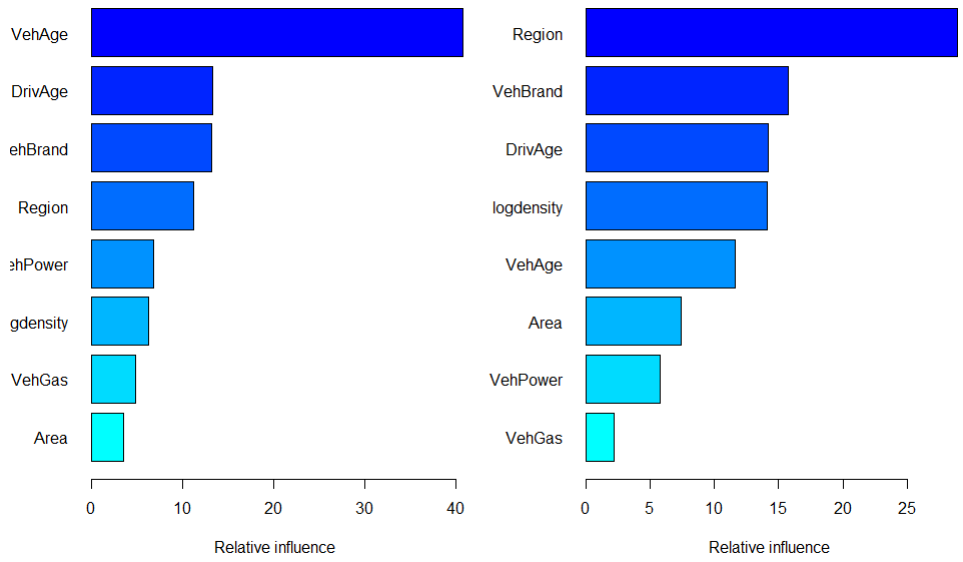

FIGURE 3.5 – Importance des variables explicatives

3.2.1.2 Interaction entre les variables

La statistique H de Friedman permet d'étudier l'interaction entre les variables explicatives du modèle. Nous allons calculer cette statistique pour nos deux modèles afin de tenir compte des éventuelles interactions dans les modèles GLM et GAM des sections suivantes. Le graphique suivant présente cette statistique pour les fréquences (à gauche) et le coût (à droite).

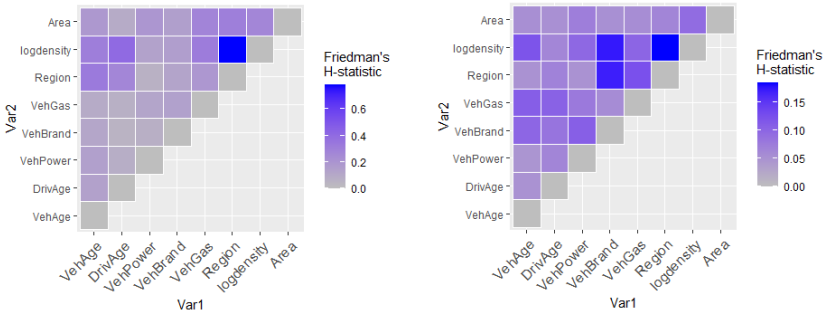

FIGURE 3.6 – La statistique H de Friedma

De cette figure, on note que pour la fréquence, la plus grande interaction s'observe entre les variables logdensity et Region, on note également une interaction entre Region et VehAge mais qui n'est pas aussi forte que la précédente. En ce qui concerne le modèle du coût moyen, les plus grandes interactions s'observent entre les variables : logdensity et Region, logdensity et VehBrand, Region et VehBrand.

Les tableaux suivants présentent les sommes des valeurs observées et les sommes des valeurs prédites pour chaque modèle GBM.

Ce tableau laisse voir que le modèle GBM sous estime le nombre de sinistres, mais lorsque

GBM	$\sum y_i$	$\sum \widehat{\pi}(X_i)$	$\sum y_i - \sum \widehat{\pi}(X_i)$
train	28754	28731.99	22.01444
test	7276	7192.997	83.00265

TABLE 3.3 – Somme des primes observées et somme des primes prédites (GBM) suivant la fréquence

.

la base de traitement est utilisée pour la prédiction, la différence entre la somme des observés et la somme des prédits est faible par rapport à la différence obtenue lorsque la prédiction se fait sur la base de test. Ceci n'est pas surprenant car c'est la base de traitement qui a été utilisée pour la régression.

GBM	$\sum y_i$	$\sum \widehat{\pi}(X_i)$	$\sum y_i - \sum \widehat{\pi}(X_i)$
train(Cout>0)	29395786	29230234	165552.4
test(Cout>0)	7435996	7440680	−4683.738
test	7435996	199672447	−192236451

TABLE 3.4 – Somme des primes observées et somme des primes prédites (GBM) suivant le coût moyen.

On remarque également ici que la prédiction sur la base d'apprentissage sous-estime le coût moyen de sinistre. Pour les prédictions avec l'ensemble de test total, la différence est de -192236451 alors qu'elle n'est que de 165552.4 dans l'ensemble de traitement. Ceci n'est pas étonnant puisque l'ensemble de test total contient 96.37% de coûts nuls tandis que la prédiction avec cet ensemble n'a aucun coût nul. Ainsi, une autre prédiction est faite en utilisant juste les polices de la base de test ayant un coût non nul, et la différence trouvée est de -4683.738 qui montre que le GBM sur-estime les coûts. On note ainsi que l'écart entre la somme des primes observées et la somme des primes prédites sera très grand, car, c'est toute la base test qui est utilisée pour la prédiction du coût moyen afin que le produit entre la fréquence prédite et le coût prédit soit possible.

3.2.1.3 Lift et GBM

Pour effectuer l'analyse des diagrammes de lift, il est nécessaire d'avoir une prime de référence à partir de laquelle les comparaisons seront faites. La base de données initiale n'a pas de prime, et nous l'avons obtenu en effectuant le produit de la fréquence de sinistres par le coût

moyen. Les courbes de concentration de la prime estimée par le GBM sont représentées sur le graphique suivant : On remarque que les deux courbes sont proches et la courbe de Lorenz est

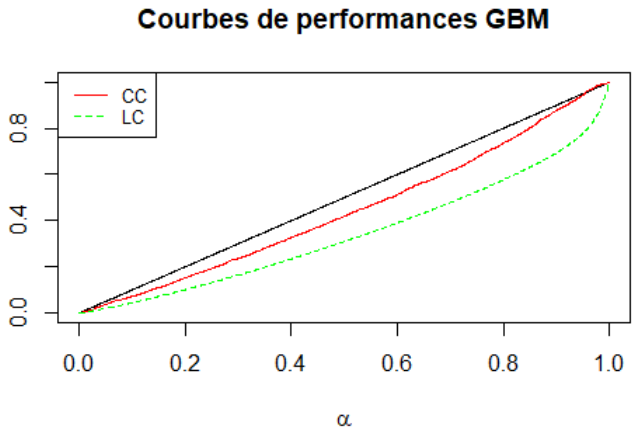

FIGURE 3.7 – Courbes de concentration

en dessous de la ligne de 45-dégrée, tandis que la courbe de concentration est presque confondue à la ligne d'égalité entre 0.95 *et* 1. Les valeurs de ICC, ILC et ABC sont respectivement de 44.4%, 34.32% *et de* 10.12%, l'indice de Gini est de 31.36%. La courbe de Lorenz montre par exemple que 60% d'assurés payent environ 37% des primes, c'est-à-dire qu'un assureur pourrait identifier un portefeuille qui bénéficie de 60% de primes avec seulement 37% de pertes.

Le tableau ci-dessous est celui utilisé pour construire le graphique quantile simple pour la comparaison de la prime pure et la prime GBM. Les colonnes nommées indice sont obtenues en divisant la moyenne des primes dans chaque quantile par la moyenne des primes de la colonne entière, il s'agit juste d'une échelle utilisée pour construire le graphique. Cette méthode est utilisée dans Medina(2016), pour la construction des autres graphiques quantiles, on utilisera également cette méthode.

Déciles	tot_exposition	prime_pure	prime_gbm	indice_prime_pure	indice_prime_gbm
1	7155.405	69.98924	72.49748	0.7265473	0.4328686
2	7155.054	74.49455	92.34561	0.7733163	0.5513780
3	7156.240	78.10580	103.92383	0.8108041	0.6205094
4	7155.883	87.69545	113.60260	0.9103528	0.6782994
5	7155.661	87.80834	122.59881	0.9115247	0.7320141
6	7155.009	91.81654	131.91687	0.9531332	0.7876504
7	7156.381	95.15172	142.44071	0.9877552	0.8504863
8	7155.849	106.09952	155.90059	1.1014026	0.9308527
9	7155.378	122.90827	177.49718	1.2758916	1.0598018
10	7156.120	127.58631	423.10163	1.3244536	2.5262591

FIGURE 3.8 – Exemple de tableau pour la construction d'un diagramme quantile simple

La figure suivante présente le graphique quantile simple permettant de comparer la prime

pure et la prime GBM. La base de données est triée suivant la prime prédite par le GBM, et ensuite les quantiles d'égales expositions sont formées.

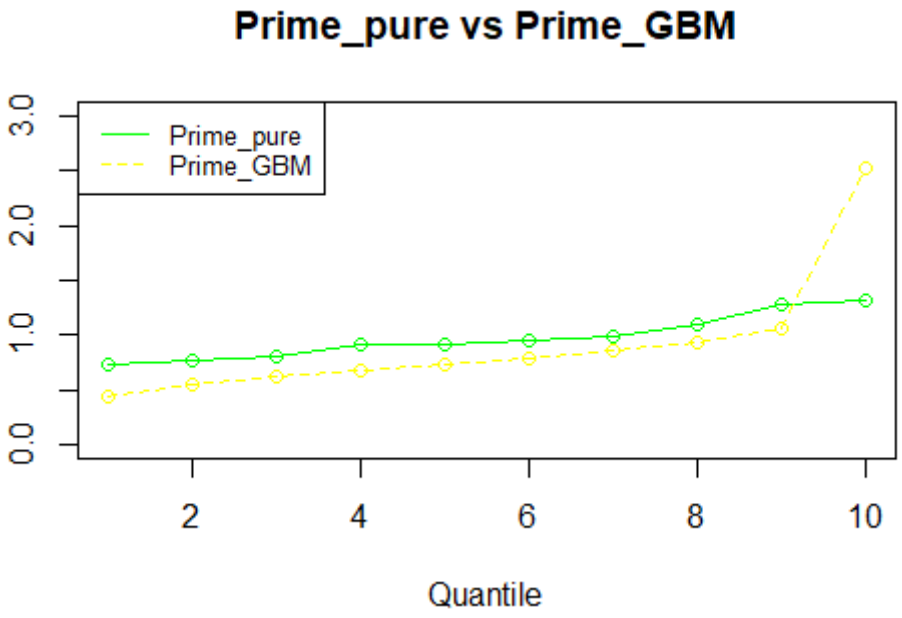

FIGURE 3.9 – Comparaison des primes GBM et pure

On note de cette figure que les deux primes varient dans le même sens lorsque le tri se fait suivant la prime prédite GBM. Ainsi les courbes des deux primes seront toutes croissantes comme on le note dans la note d'étude de l'institut CAS "Study Note : Model Validation and Holdout Data" du 13/01/2019. La notion de lift dans cette note d'étude fait référence à la Tension Tarifaire (différence entre les indices du dernier et du premier quantile pour la prime prédite) qui sera notée par la suite par TT. On constate que la prime prédite par le GBM est inférieure à la prime pure dans les neufs premiers quantiles, les deux courbes se coupent au voisinage de 92% et la prime GBM devient subitement très supérieure à la prime pure. Cette figure confirme qu'il y a une très grande différence entre la somme des primes observées et la somme des primes prédites.

Il ressort des modèles GBM que toutes les variables explicatives ont chacune un impact sur la réponse. Ainsi, pour la suite nous allons considérer toutes ces variables dans les régressions GLM et GAM.

3.2.2 Régressions par le modèle GLM

Un traitement des variables est fait avant les régressions. Pour les variables VehAge et DrivAge, elles sont regroupées en classe dans le but de faciliter la tarification des primes pour des assurés dont l'âge du véhicule ou l'âge du conducteur se trouve dans une même tranche

par exemple. La fonction logarithme est utilisée pour la variable Density pour réduire l'échelle. Pour les variables qualitatives, une description permet de préciser la référence pour chacune d'elle afin d'éviter le choix automatique de la première modalité comme référence. Dans nos régressions, nous allons commencer par la totalité des variables explicatives,puisque le GBM a montrer que chacune d'elles a une importance dans l'explication de la réponse. Ensuite nous allons prendre les interactions entre certaines variables en se basant sur les résultats du GBM.

3.2.2.1 Estimations

Après la première régression de la fréquence, on trouve $AIC = 229002.1$ et on note que la variable AreaGLM n'a pas d'impact significatif sur la fréquence de sinistres. La fonction step() de R exclue cette variable du modèle et l'AIC passe à 229000. On note également que par rapport à la modalité GasDiesel, la modalité GasRegular de la variable VehGas impacte positivement et significativement le nombre de sinistres. Toutes les variables ont chacune au moins deux modalités significatives par rapport à la modalité de référence.

Un nouveau GLM est fait pour prendre en compte les interactions entre les variables Region et DensityGLM d'une part et entre Region et VehAgeGLM d'autre part, ce modèle donne un AIC de 228785.4 qui est bien inférieur à la valeur de l'AIC précédente et toutes les variables ont un impact significatif sur la réponse : ce dernier modèle est alors maintenu comme le modèle de la fréquence.

Pour la modélisation du coût moyen, on ne considère que les polices ayant un coût strictement positif afin de pouvoir utiliser le loi Gamme. Comme précédemment, le premier modèle considère toutes les variables, son AIC est de 331112.5. On note par ailleurs les variables telles que DensityGLM, AreaGLM, VehGas n'ont aucun impact significatif sur le coût moyen de sinistres, pour les autres variables très peu de modalités sont significatives. La fonction step() de R exclue les variables VehBrand, VehGas et AreaGLM et l'AIC passe à 331100, qui est inférieur au précédent. On maintient les autres variables en y ajoutant les interactions entre les variables DensityGLM et Region, puisque la variable VehBrand ne fait plus partie du modèle des coûts, les interactions avec elle ne sont pas considérées. Après la régression, on constate que l'interaction n'améliore pas le modèle, car on trouve un AIC de 331131.2, qui est bien supérieur aux précédents, ainsi le modèle final pour le coût moyen est celui obtenu grâce à la fonction step() de R.

Les tableaux suivants présentent les sommes des valeurs observées et les sommes des valeurs prédites pour chaque modèle GLM

GLM	$\sum y_i$	$\sum \widehat{\pi}(X_i)$	$\sum y_i - \sum \widehat{\pi}(X_i)$
train	28914	28914	0
test	7116	7260.798	−144.7985

TABLE 3.5 – Somme des primes observées et somme des primes prédites (GLM) suivant la fréquence

D'après ce tableau, la somme du nombre de sinistres prédit par le GLM grâce à la base d'apprentissage coïncide avec celle du nombre de sinistres observés dans cette base. Mais, ce modèle sur-estime le nombre de sinistre prédit par l'échantillon test, la différence dans ce cas est d'environ 145 sinistres en faveur du nombre de sinistres prédit.

GLM	$\sum y_i$	$\sum \widehat{\pi}(X_i)$	$\sum y_i - \sum \widehat{\pi}(X_i)$
train(Cout>0)	29499612	29500974	−1362.527
test(Cout>0)	7332170	7318661	13509.03
test	7332170	200243766	−192911596

TABLE 3.6 – Somme des primes observées et somme des primes prédites (GLM) suivant les coûts moyens

Le tableau ci-dessus montre que la base de traitement sur-estime les prédictions du coût contrairement au GBM où cette base sous-estime plutôt les prédictions. Pour les prédictions avec la base de test entière, l'écart entre les deux sommes est très grande comme dans le cas du GBM. On note également que les prédictions faites sur les coût non nuls de la base test sous-estiment les coût contrairement au GBM qui sur-estime avec cette base.

3.2.2.2 Lift et GLM

La figure suivante montre les variations des courbes de performance suivant les valeurs de α.

On constate que la courbe de concentration est convexe dans les neuf premiers quantiles, au voisinage du dernier quantile, elle se rapproche davantage de la bissectrice et la traverse par la suite. La position de la courbe de Lorenz laisse voir qu'il existe bien des inégalités dans la distribution de la prime pure prédite par le modèle GLM. Elle montre par exemple que 60% d'assurés payent environ 38% des primes, c'est-à-dire qu'un assureur pourrait identifier un portefeuille qui bénéficie de 60% de primes avec seulement 38% de pertes. On remarque également que la courbe de Lorenz pour la prime GLM est plus convexe que celle de la prime

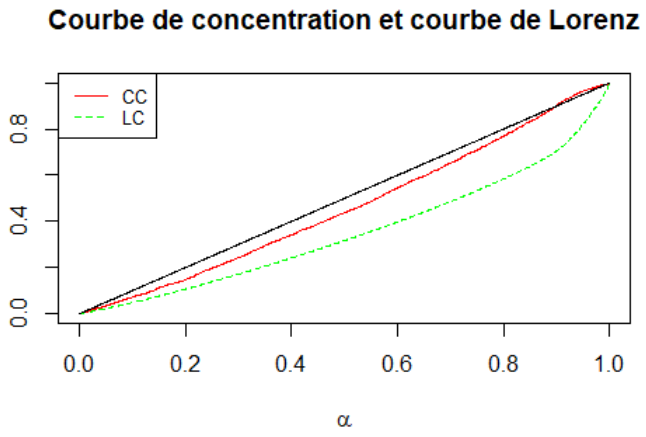

FIGURE 3.10 – Courbes de performance

GBM. La valeur de ICC est de 46.11%, et les valeurs de ILC et de ABC sont respectivement de 35.21% *et* 10.9%. L'indice de Gini dans ce cas est égal à 29.6%.

Une description de la prime prédite par les modèles GLM montre qu'elle varie entre 41.44 *et* 2965.31 avec une moyenne de 169.41.

La figure suivante est le graphique quantile simple permettant de comparer la prime prédite par le GLM avec la prime pure actuelle. Le tri de la base de données utilisée se fait suivant

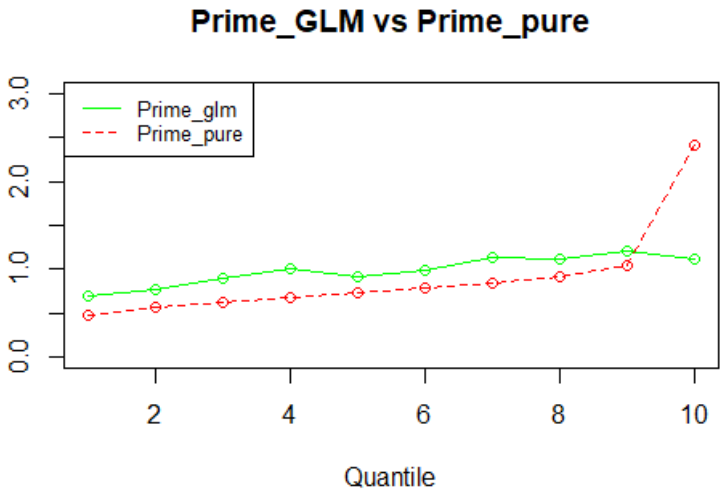

FIGURE 3.11 – Comparaison la prime GLM et la prime pure

la prime prédite GLM, on note un grand écart entre les deux primes dans presque tous les quantiles. Contrairement au cas du GBM où sa prime prédite est restée très proche et inférieure à la prime pure jusqu'au quantile 92%. On constate ici que la prime prédite par le GLM est inférieure à la prime pure dans les neuf premiers quantiles, le plus grand écart entre les deux s'observe dans le dernier quantile où la prime GLM est supérieure à la prime pure. Si la courbe de la prime GLM est strictement croissante. Celle de la prime pure varie un peu, elle est par

exemple décroissante dans le quatrième quantile.

3.2.3 Régressions par le modèle GAM

Dans le modèle GLM, les valeurs de certaines variables ont été regroupées en intervalles, ceci peut négativement impacter les résultats. Par exemple, deux coûts différents mais proches pour la notation peuvent avoir des âges substantiellement différents si les valeurs appartiennent à des intervalles différents. Trouver une bonne subdivision est trop coûteux en temps, l'intervalle devant être suffisamment grand pour atteindre de bonnes précisions des estimations, mais en même temps doivent être petits si l'effet des variables varie beaucoup. Nous allons faire appel à la modélisation grâce aux GAM.

3.2.3.1 Estimations et importance des variables

Dans cette section, nous allons juste utiliser les variables choisies par la fonction step() des deux modèles GLM. Le modèle de fréquence avec interaction a un AIC de 331601.5 tandis que celui sans interaction donne un AIC de 229093.6 qui est inférieur au précédent, le modèle final ne tient pas compte des interactions.

Il ressort des régressions GAM que seules les régions Limousin, Nord-Pas-de-Calai et Auvergne sont significatives pour le modèle de la fréquence. On constate que toutes les autres variables qualitatives ont au moins une modalité significative. De plus, toutes les variables quantitatives sont significatives. Pour ce qui est du modèle de coût moyen, deux modalités sont significatives pour la variable Region (Provence-Alpes-Cotes-D'Azur et Languedoc-Roussillon). A l'exception de la variable logdensity, toutes les variables quantitatives sont significatives pour le modèle des coûts. Bien que les deux modèles soient globalement significatif, on note que les coefficients de détermination ajustés très faible (0.269% pour la fréquence et 0.604% pour le coût). La description de la prime estimée par les modèles GAM montre qu'elle varie de 17.46 à 2242.52 avec une moyenne de 158.99.

Les tableaux suivants présentent les sommes des valeurs observées et les sommes des valeurs prédites pour chaque modèle GBM

Comme dans le cadre du GLM, on note de ce tableau que la somme du nombre de sinistres prédit par le GAM grâce à la base d'apprentissage coïncide avec celle du nombre de sinistres observés dans cette base. Mais, ce modèle sous-estime le nombre de sinistres prédit par l'échantillon test, la différence des deux sommes est d'environ 91 sinistres en faveur du nombre de sinistres observés.

GAM	$\sum y_i$	$\sum \widehat{\pi}(X_i)$	$\sum y_i - \sum \widehat{\pi}(X_i)$
train	28754	28754	0
test	7276	7185.075	90.92524

TABLE 3.7 – Somme des primes observées et somme des primes prédites (GAM) suivant la fréquence.

GAM	$\sum y_i$	$\sum \widehat{\pi}(X_i)$	$\sum y_i - \sum \widehat{\pi}(X_i)$
train(Cout>0)	29395786	29387470	8316.165
test(Cout>0)	7435996	7482595	−46599.03
test	7435996	200616755	−193180759

TABLE 3.8 – Somme des primes observées et somme des primes prédites (GAM) suivant les coûts moyens

Comme dans le cas du GBM, on constate que la base de traitement sous-estime les prédictions du coût. Pour les prédictions avec la base de test entière, l'écart entre les deux sommes est très grande comme dans les cas du GLM et du GBM. On note également que les prédictions faites sur les coût non nuls de la base test sur-estiment les coût contrairement.

3.2.3.2 Lift et GAM

Les courbes de performance obtenues à partir des régressions GAM sont représentées sur le graphique suivant :

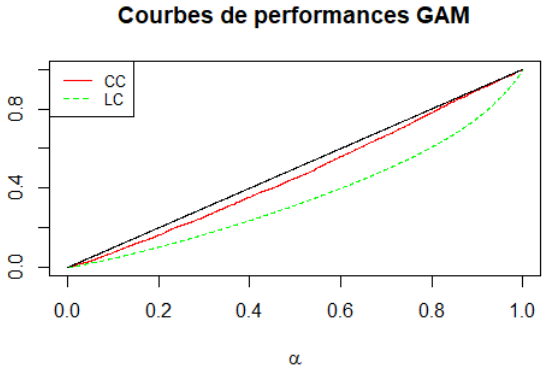

FIGURE 3.12 – Courbes de concentration des primes GAM

On constate à travers ce graphique que la courbe de concentration a la même forme comme celle obtenue par le modèle GBM. Les valeurs d'ICC, ILC et ABC sont respectivement de 46.94%, 35.87% *et* 11.08%. Ainsi, l'aire entre les ces deux courbes est plus grande pour le modèle GAM par rapport à celle du modèle GLM (11.07% *contre* 10.91%). On constate suivant

la courbe de Lorenz que 60% d'assurés payent environ 40% des primes, c'est-à-dire qu'un assureur pourrait identifier un porte feuille qui bénéficie de 60% de primes avec seulement 40% de pertes. Par ailleurs, l'indice de Gini pour le modèle GAM est de 28.27% : il ressort de ces résultats que même si le modèle GLM prédit mieux les primes par rapport au modèle GAM, sa distribution des primes est moins inégalitaire comparativement à la distribution des primes prédites par le GLM et par le GBM.

Comme pour la construction d'un graphique quantile simple, le tableau ci-dessous est un example pour la construction d'un graphique quantile double. La base de données est triée suivant le ratio $\frac{Prime\ GBM}{Prime\ GAM}$ les deux dernières colonnes montrent que les indices des primes GAM et GBM varient en sens opposé.

Déciles	tot_exposition	prime_pure	prime_game	prime_gbm	indice_prime_pure	indice_prime_game	indice_prime_gbm
1	7155.264	81.79886	232.9716	141.7910	0.8491413	1.4070760	0.8466072
2	7155.726	91.51489	161.6678	132.2953	0.9500018	0.9764228	0.7899100
3	7155.463	94.87897	148.8996	132.3484	0.9849238	0.8993073	0.7902270
4	7156.322	99.54497	142.3985	135.4509	1.0333608	0.8600427	0.8087515
5	7155.578	86.53424	139.7042	138.3864	0.8982984	0.8437696	0.8262788
6	7154.956	99.37239	136.6335	139.5707	1.0315692	0.8252238	0.8333499
7	7155.703	112.05667	133.8762	144.2092	1.1632428	0.8085702	0.8610453
8	7156.340	93.99366	132.4577	152.9070	0.9757336	0.8000034	0.9129786
9	7155.889	113.64351	133.5944	169.1010	1.1797156	0.8068686	1.0096699
10	7155.738	98.77253	199.2647	385.7108	1.0253423	1.2034969	2.3030055

FIGURE 3.13 – Exemple de tableau pour la construction des graphiques quantiles doubles

La figure suivante représente les graphiques quantiles permettant de comparer la prime pure actuelle et la prime prédite par le GAM d'une part et par rapport à la prime GLM d'autre part. Pour le graphique quantile simple (panel de gauche), la base de données est triée suivant les valeurs de la prime GAM, et pour le graphique quantile double, le tri se fait suivant le ration $\frac{Prime\ GAM}{Prime\ GLM}$ et la référence est la prime pure. Le graphique de gauche montre que les primes

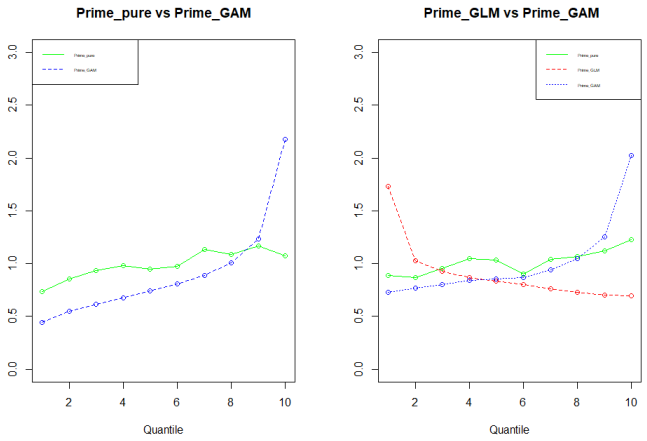

FIGURE 3.14 – Comparaison de la prime GAM à la prime pure et à la prime GLM

pure et GAM se rapprochent davantage à partir du quatrième quantile. La prime prédite par le

GAM reste inférieure à la prime pure jusqu'au $\alpha = 0.85$, à partir duquel la prime GAM passe au-dessus. On constate également que la courbe de la prime pure varie de sens contrairement à celle de la prime prédite qui est strictement croissante. Comme dans les cas du GBm et du GLM, le plus grand écart entre les deux courbes s'observe dans le dernier quantile.

Quant au graphique de droite, le ratio utilisé pour le tri de la base de données est le rapport $\frac{prime\ GAM}{prime\ GLM}$. Ainsi, les plus petites valeurs de ce ratio s'obtiennent lorsque la prime GLM est grande. C'est la raison pour laquelle la courbe de la prime GLM est décroissante et celle de la prime GAM est croissante. Ce graphique favorise la prime GLM par rapport à la prime GAM dans les deuxième, troisième et dixième quantiles. Les deux primes prédites sont indifférentes dans le quatrième quantile et dans le reste de quantiles, la prime GAM est meilleure.

Dans le but de comparer la prime GBM par rapport aux primes GLM et GAM, nous allons effectuer les diagrammes quantiles doubles. La capture suivante présente un exemple de tableau utilisé pour construire un graphique quantile double.

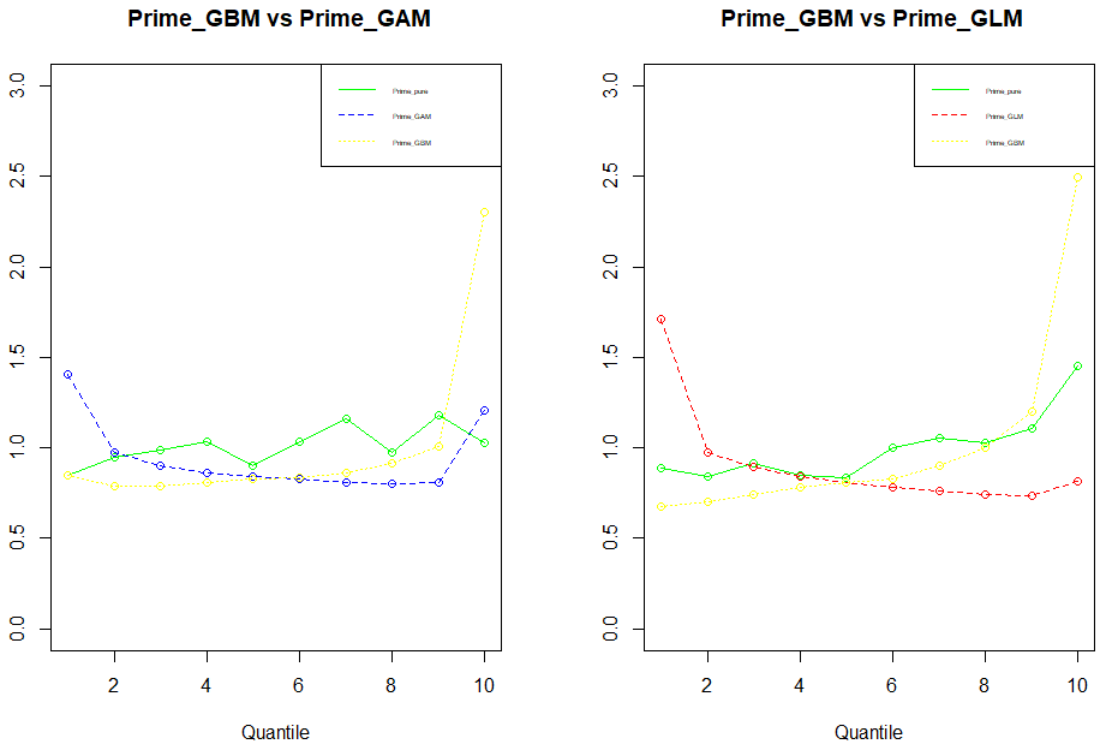

FIGURE 3.15 – Comparaison des primes GBM à la prime GAM et et à la prime GLM

Le panel de gauche permet de comparer des primes prédites par les modèles GAM et GBM, le ratio utilisé pour le tri de la base de données est le rapport $\frac{prime\ GBM}{prime\ GAM}$. On constate que la prime GBM est plus proche de la prime pure que la prime GAM dans les premier, sixième, septième et huitième quantiles. Ce résultat montre que le modèle GBM prédit mieux la prime pure dans ces quantiles. Les deux primes sont indifférentes dans le cinquième quantile, et dans

les autres quantiles, la prime GAM est meilleure par rapport à la prime GBM.

Le panel de droite quant à lui, permet de comparer des primes prédites par les modèles GLM et GBM, le ratio utilisé pour le tri de la base de données est le rapport $\frac{prime\ GBM}{prime\ GLM}$. Ce graphique favorise la prime GLM dans les deuxième, troisième, quatrième et dixième quantiles ; et dans les autres quantiles, la prime GBM est meilleure. Les deux graphiques laissent voir que si la prime GBM dans a tendance à être croissante, les primes GLM et GAM changent de sens dans le dernier quantile.

3.2.4 Une généralisation suivant les modèles

1. **Somme de la prime pure et somme de la prime prédite suivant les modèles**

Modèle	Somme des primes observées	Somme des primes prédites	Différence
GBM	13062714	22710825	−9648111
GLM	12844841	22972421	−10127581
GAM	7332170	22451822	−9389107

TABLE 3.9 – Somme des primes observées et somme des primes prédites suivant les modèles

Comme nous l'avons constaté plus haut un grand écart entre la somme des coûts prédits par des différents modèles et la somme des coûts observés, on note un écart énorme entre la somme des primes prédites et la somme des primes observées. Ceci peut s'expliquer par le fait que la base de données contient plus de 90% de polices ayant une prime pure nulle. En effet, la prime pure a été obtenue en faisant le produit du coût moyen par la fréquence et la plupart des coûts sont nuls. Les primes prédites quant à elles n'ont aucune observation de prime nulle quel que soit le modèle utilisé.

2. **Courbes de performance selon les modèles**

La figure suivante présente les courbes de concentrations d'une part et d'autre part les courbes de Lorenz des différents modèles étudiés.

On constate premièrement que les courbes de concentration des primes prédites par les trois modèles sont proches. Celles du GLM et du GAM sont presque confondues et sont plus proches de la ligne d'indépendance. Pour les courbes de Lorenz, toutes les trois sont presque confondues dans les six premiers quantiles, à partir du septième quantile, celle du GBM est plus éloignée de la ligne d'égalité tandis que celle du GAM est plus proche et celle du GLM est entre les deux précédentes. Ce qui montre que le modèle GBM est celui qui tient plus compte les inégalités.

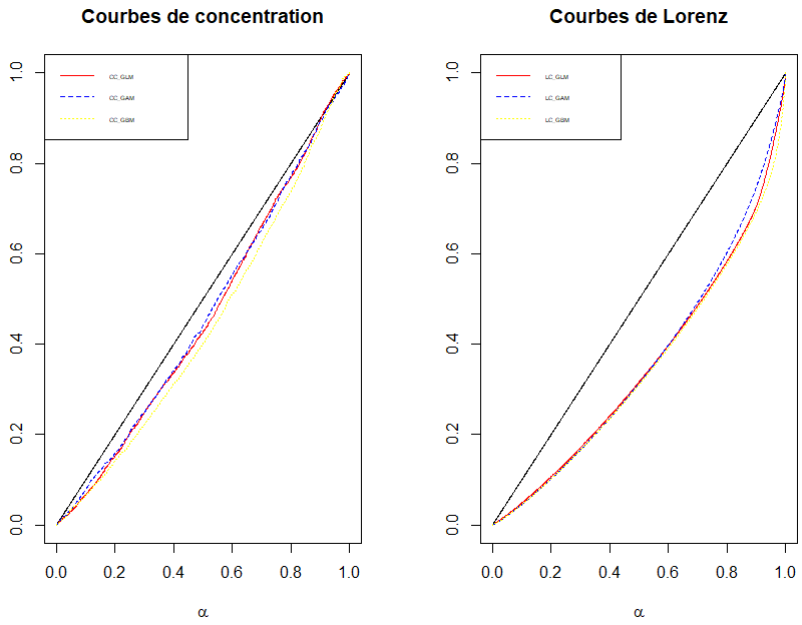

FIGURE 3.16 – Courbes de concentration suivant les modèles

3. **Variation de ICC, ILC et ABC suivant des modèles** La figure suivante montre les variations de ICC, ILC et ABC en fonction de α, et suivant les modèles étudiés à la section précédente. Les plus grandes ICC sont obtenues par le modèle GAM, ensuite

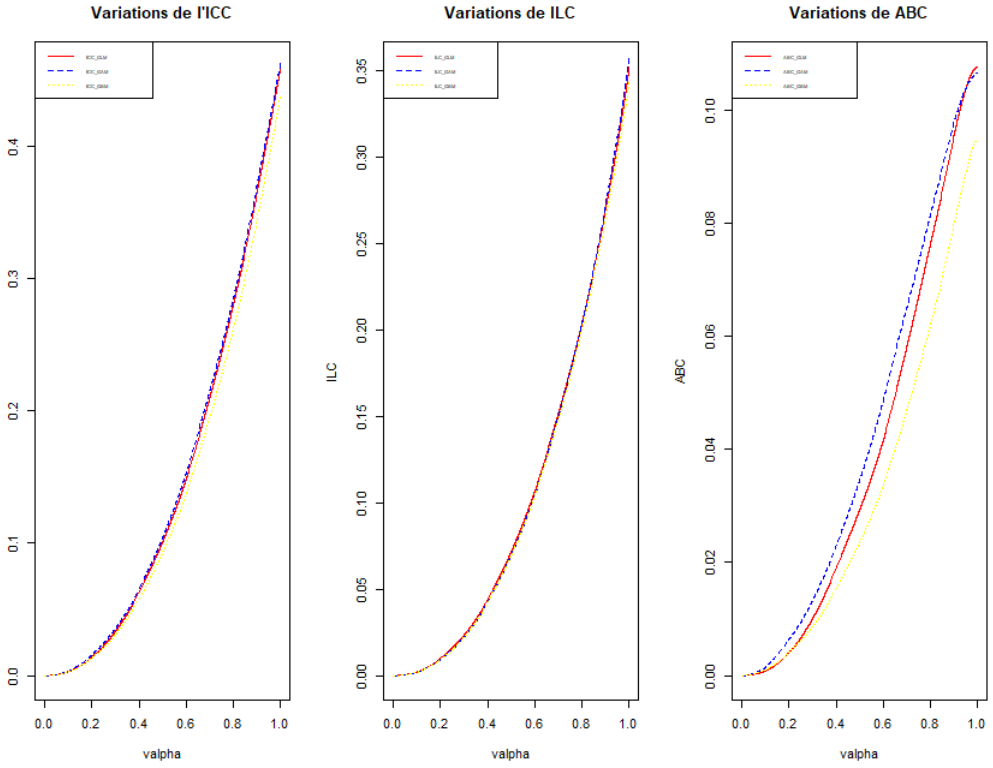

FIGURE 3.17 – Variation de ICC, ILC et ABC en fonction de α

viennent celles du modèle GAM et enfin celles du modèle GBM. Le graphique du milieu montre que les ILC sont presque confondus pour les trois modèles. Quant à l'ABC, le

modèle GBM donnent les plus faibles valeurs et le modèle GAM donne les plus grandes.

4. **Distribution de primes prédites**

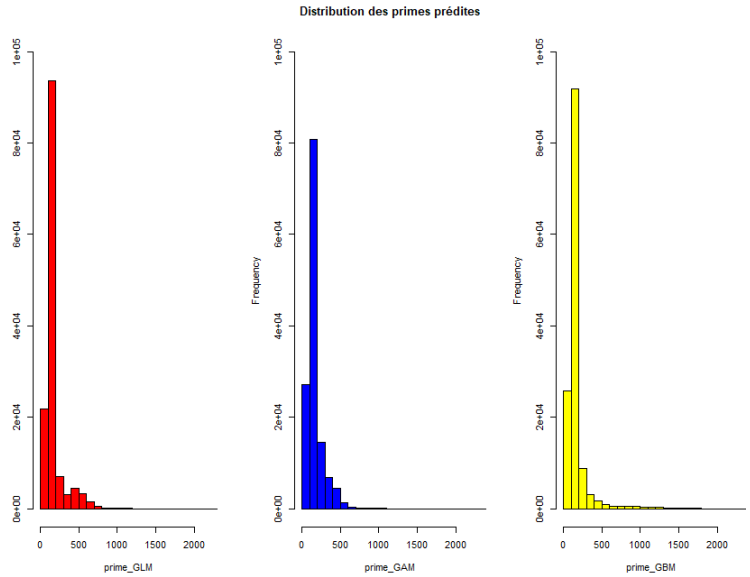

FIGURE 3.18 – Distribution de primes prédites

La figure ci-dessus est celle des histogrammes des primes prédites, on constate que la prime prédite par le modèle GLM est a une forme différente de celles des deux autres modèles. Pour la prime GLM, la première classe concentre moins d'assurés que les premières classes des autres modèles. La deuxième classe $[100; 200]$ est la classe modale pour les trois primes, elle concentre environ 95000 assurés pour le cas du GLM, 81000 assurés pour le cas du GAM et 92000 assurés pour la prime GBM. On note également que très peu d'assurés payent une prime supérieure à 800 dans le cas du GLM, pour le GAM très peu d'assurés payent une prime supérieure à 600, alors que pour le GBM, le nombre d'assurés qui payent une prime de 1200 n'est pas négligeable.

3.3 Choix du meilleur modèle

3.3.1 Choix du modèle selon la qualité d'ajustement

Dans ce paragraphe, nous allons comparer les modèles suivant la déviance et suivant l'erreur quadratique, afin de choisir le modèle qui prédit mieux les coûts de sinistres. Plus la déviance d'un modèle est faible, plus le modèle est meilleur en terme de qualité d'ajustement, il en est de même pour l'erreur quadratique, le meilleur est celui ayant la plus petite erreur. Les tableaux suivants présentent pour chaque modèle, les déviances et les erreurs quadratiques.

Modèles	GLM	GAM	GBM
Fréquence	**0.047146931**	0.04811298	0.04779112
Coût moyen	2221815	2237141	**2205240**
Coût moyen test(coût>0)	**2790169**	2860674	2936167

TABLE 3.10 – Erreur quadratique sur la base de test

Le tableau ci-dessus montre que le meilleur modèle pour la prédiction de la fréquence de sinistres est le modèle GLM, suivis respectivement des modèles GBM et GAM. En ce qui concerne la prédiction du coût moyen, le GBM est le meilleur modèle lorsque les coûts sont prédits en éliminant les coûts nuls dans la base de test, ensuite viennent les modèles GLM et GAM respectivement. Mais si la base de test entière est utilisée pour les prédictions, l'erreur quadratique favorise le GLM tandis que le GBM sera le pire modèle.

Pour le calcul de la déviance du modèle du coût moyen, on exclut les polices de coûts nuls de la base de test pour pouvoir calculer la déviance de Gamma.

Modèles	GLM	GAM	GBM
Fréquence	42943.98	43731.14	**42604.88**
Coût moyen (Coût>0)	**4335.628**	4770.962	4760.944

TABLE 3.11 – Déviance sur la base de test

En se basant sur le critère de déviance, le meilleur modèle pour la fréquence est le modèle GBM, suivis respectivement des modèles GAM et GLM . Quant à la prédiction du coût moyen, le meilleur est le GLM, ensuite on a les modèles GBM et GAM respectivement.

3.3.2 Choix du modèle selon le lift

Une autre possibilité permettant de choisir le meilleur modèle est d'utiliser les outils de lift tels que l'indice de Gini, l'aire entre les courbes de concentration (ABC), les diagrammes quantiles simple et double. On peut également utiliser les graphiques quantiles pour choisir le meilleur modèle. La notion de lift (qui correspond à la tension tarifaire) telle que définie dans la note d'étude de l'institut CAS (2019) va nous permettre de classer les modèles à partir de leurs graphiques quantiles simples, le meilleur modèle selon ce critère est celui qui a la plus grande tension tarifaire. Le tableau suivant présente chaque modèle avec ses valeurs de l'indice de Gini et de l'ABC. En assurance, on souhaiterait que les primes soient les plus inégalitaires

possible vu que les assurés n'ont pas le même profil, ainsi, plus l'indice de Gini d'un modèle est élevé, mieux ce modèle estime les primes d'assurance. Pour ce qui est de l'ABC, lorsqu'elle diminue, on obtient la meilleure prime prédite $\pi(X)$ approchant la prime pure $\mu(X)$.

Modèles	GLM	GAM	GBM
ABC	10.9%	11.08%	**10.12%**
Indice de Gini	29.6%	28.27%	**31.36%**
TT	1.938019	1.734088	**2.09339**

TABLE 3.12 – ABC, Indice de Gini et lift (TT) suivant les différents modèles

Ce tableau montre que quel que soit le critère utilisé ici, le meilleur modèle est le GBM, il est suivi respectivement des modèles GLM et GAM.

Conclusion

Il était question dans notre étude de classer les modèles de tarification afin de choisir celui qui permettrait à un assureur de déterminer les primes pures d'assurance de manière équitable. Nous avons pour cela étudier trois modèles : GLM, GAM et GBM.

Il ressort de notre étude que le choix du modèle dépend du critère de sélection utilisé. En effet, pour ce qui est de la qualité d'ajustement, l'erreur quadratique favorise le GLM pour la modélisation de la fréquence et le GBM pour la modélisation du coût moyen, le GAM est le pire modèle pour ce critère. La déviance quant à elle favorise le GBM pour la modélisation de la fréquence et le GLM pour la modélisation du coût.

Quant à la capacité des modèles à segmenter les profils de risques d'assurés, l'aire entre les courbe de Concentration et de Lorenz (ABC), l'indice de Gini et le diagramme quantile choisissent le GBM comme le meilleur modèle de tarification des primes pures, et le GAM est dans ces cas le pire des modèles. Ainsi, le modèle GBM serait celui qui prédit le mieux les primes pures. Nous pouvons quand même noter que choisir le GLM serait problématique. Car bien que son interprétation et son utilisation soient faciles, sa modélisation nécessite un pré-traitement avant une régression et il n'est pas toujours évident de trouver le meilleur regroupement pour les variables continues afin d'avoir les bonnes prédictions.

Enfin, il serait mieux d'utiliser les critères lift pour le choix du modèle. En effet, ils permettent selon la capacité à prédire les primes. Puisque nous constatons que pour la qualité d'ajustement, la déviance choisit le GBM pour la prédiction de la fréquence et le GLM pour la prédiction du coût moyen, il devient difficile dans ce cas de choisir celui qui prédit mieux la prime pure qui dépend à la fois de la fréquence prédite et le coût prédit.

BIBLIOGRAPHIE

1. Charpentier, A. (2013). Modélisation des coûts individuels de sinistres. Actuariat IARD - ACT2040, page 7. http ://freakonometrics.hypotheses.org/.
2. Clément, B. (1986). ROC Curve and Lift Curve : GRAPHS F0R GOODNESS OF FIT.
3. Cummings, D. (2009). Value of Lift-A Net Present Value Framework. Actuarial Review.
4. Denuit, M., Sznajder, D., Trufin, J. (2019). Model selection based on Lorenz and concentration curves, Gini indices and convex order. Insurance : Mathematics and Economics 89, 128-139.
5. Denuit, Michel ; Charpentier, Arthur ; Trufin, Julien. Autocalibration and Tweedie-dominance for insurance pricing with machine learning. Insurance : Mathematics and Economics.
6. Denuit, M., Hainaut, D., Trufin, J. (2019) Effective Statistical Learning Methods for Actuaries III. Neural Networks and Extensions. Springer Actuarial Lecture Notes.
7. Frees, E.W., Meyers, G., Cummings, A.D. (2011). Summarizing insurance scores using a Gini index. Journal of the American Statistical Association 106, 1085-1098.
8. Frees, E.W., Meyers, G., Cummings, A.D. (2014). Insurance ratemaking and a Gini index. Journal of Risk and Insurance 81, 335-366.
9. Guirguis I. et Dutang C.(2016). Introduction à l'assurance construction : Analyse de la sinistralité, tarification, provisionnement. Notes basées sur une formation dispensée à l'université du Mans en 2016 et 2017.
10. Leblanc, M.M (2017). Contributions méthodologiques à l'estimation de lois de probabilité pour l'assurance en présence de données rares, Mémoire. Le CNAM.
11. Lecomte, K.(2018). Automatiser la comparaison de modèles : Application sur l'amélioration d'un modèle de fréquence par des techniques de machine learning. ENSAE ParisTech. https ://www.institutdesactuaires.com/docs/mem/7caa6ce8d798a6f115283e57b1549f35.pdf
12. Medina,L. (2016). And The Winner Is... ? How to pick a Better Model. Model Lift. CAS RPM Seminar. Verisk Analytics.
13. Maydeu, O., and Garcıa, F. (2010). Goodness-of-Fit Testing. University of Barcelona,Bacelona, Spain.
14. Meyers, G. (2008). The Value of Lift. Actuarial Review.
15. Paglia, A., Phélippé-Guinvarc'h, M., Lenca, P. (2011). Adaptation de l'algorithme CART pour la tarification des risques en assurance non-vie. In EGC 2011, vol. RNTI-E-20, pp.611-622.

16. Partrat, C. et Besson, J-L (2005). Assurance non-vie-modélisation et simulation, AAA, Economica.

17. Tevet, D.(2013). Exploring Model Lift : Is Your Model Worth Implementing ? Actuarial Review.

18. Tevet, D.(2016). And The Winner Is... ? How to pick a Better Model. Verisk Insurance Solutions.

19. Therond, P-E. (2004). Assurance non-vie : Mesures et comparaison de risques. ISFA.

20. CAS Institute (2019). Model Validation and Holdout Data. Study Note. page 12.

Annexes

Annexe1 : Code R pour les example numériques avec les données du Brésil

```
rm(list=ls())
install.packages("CASdatasets", repos = "http://cas.uqam.ca/pub/",
type="source")
library(CASdatasets)
library(IC2)
library(dplyr)
library(data.table)
library(leaps)
library(datasets)
library(tidyr)
?CASdatasets
data(brvehins1a)
data(brvehins1b)

fusion=rbind(brvehins1a,brvehins1b)
dat=fusion

dat=dat[dat$VehYear > 0&dat$ExposTotal<=2,]
boxplot(dat$VehYear)
dat =dat %>% mutate(
  NbSin=ClaimNbRob+ClaimNbFire+ClaimNbOther+ClaimNbPartColl+ClaimNbTotColl,
  CouSin  =
ClaimAmountFire+ClaimAmountOther+ClaimAmountPartColl+ClaimAmountRob
  +ClaimAmountTotColl
,
Gender=as.factor(Gender),
VehGroup=as.factor(VehGroup),
VehModel=as.factor(VehModel),
DrivAge=as.factor(DrivAge),
State=as.factor(State),
StateAb=as.factor(StateAb),
Area=as.factor(Area)

  )%>%
  select(Gender,DrivAge,VehYear,Area,StateAb,ExposTotal
         ,PremTotal,NbSin,CouSin)
dat=dat[dat$NbSin>0,]
dat$coumoy=dat$CouSin/dat$NbSin
dat=drop_na(dat)
summary(dat)
dat=dat[dat$PremTotal<=10000&dat$PremTotal>0,]
dat$PremTotal[is.na(dat$PremTotal)]=median(dat$PremTotal,na.rm=T)
dat$CouSin

###Distribution des primes####
dat$PremTotal=as.numeric(dat$PremTotal)
dada=dat[dat$PremTotal<=10000,]
par(mfrow=c(1,1))
hist(dada$PremTotal,breaks =
c(0,500,1000,1500,2000,2500,3000,3500,4000,4500,5000,
                                10000)
     , main="Distribution des primes", xlab = "Primes totales",ylab =
"fréquences")
curveLorenz(dada$PremTotal, col="red", xlab="Proportion des polices"
            ,ylab="Distribution des primes", main="courbe de Lorenz")
##########Division de la base de données en deux###############
datglm=data.frame(dat)
```

```
#regression GLM#
datglm$NbSin=as.numeric(datglm$NbSin)
datglm$CouSin=as.numeric(datglm$CouSin)
datglm$Gender=as.factor(datglm$Gender)
datglm$DrivAge=as.factor(datglm$DrivAge)
datglm$VehYear=as.numeric(datglm$VehYear)
datglm$Area=as.factor(datglm$Area)
datglm$ExposTotal=as.numeric(datglm$ExposTotal)

d=sort(sample(nrow(datglm),nrow(datglm)*0.75))
trainglm=datglm[d,]
testglm=datglm[-d,]

reg=glm(formula =CouSin ~ Gender+DrivAge+VehYear+Area,weights=NbSin,
         family = Gamma(link = "log"), data=datglm)
summary(reg)
reg0=glm(formula = NbSin ~ Gender+DrivAge+VehYear+Area,offset =
log(ExposTotal),
        family =poisson(), data=datglm)
summary(reg0)

cousinglm=predict(reg,newdata = testglm,type="response")
pred_freq=predict(reg0,newdata = testglm,type="response")

obserglm=reg$fitted.values
obserglm
prime_glm=pred_freq*cousinglm
summary(prime_glm)
par(mfrow=c(1,2))
hist(prime_glm)
hist(dat$PremTotal)
###GAM#####
install.packages("gam")
library(gam)
library(mgcv)
#premier GAM#
d1=sort(sample(nrow(dat),nrow(dat)*0.75))
train=dat[d1,]
test=dat[-d1,]
game0=gam(data=train,formula =NbSin ~ Gender+DrivAge+
            s(VehYear,bs="cr",k=3)+Area,offset = log(ExposTotal),
          family =poisson(link = "log"),  select= TRUE)

summary(game0)
#deuxième GAM#
game1=gam(data=train,formula = CouSin ~ Gender+DrivAge+
            s(VehYear,bs="cr",k=3)+Area,weights=NbSin,
          family = Gamma(link = "log"),select= TRUE)

summary(game1)

pred_freq_game=predict(game0,test,type = "response")
cousingam=predict(game1,test,type = "response")
summary(cousingam)
Prime_game=pred_freq_game*cousingam
summary(Prime_game)

#Graphiques quantiles simples
```

```
dat_lift_glm=data.frame(testglm,cousinglm)
dat_lift_glm$obs=1:(dim(testglm)[1])
tri= order(dat_lift_glm$cousinglm,
runif(length(dat_lift_glm$cousinglm)))#tri
dat_lift_glm=dat_lift_glm[tri,]
expo_cum=cumsum(dat_lift_glm$ExposTotal)#division en quantile d'égale
exposition
expo_tot=sum(dat_lift_glm$ExposTotal)
seuil=c(0,1:9*expo_tot/10,10.1*expo_tot/10)
dat_lift_glm$quan=cut(expo_cum, breaks = seuil, labels = 1:10)
dat_lift_glm= dat_lift_glm[order(dat_lift_glm$obs),]

rm(tri, expo_cum,expo_tot, seuil)
tapply(dat_lift_glm$ExposTotal, dat_lift_glm$quan, sum)
moy_glm= tapply(dat_lift_glm$cousinglm, dat_lift_glm$quan, mean)/
mean(dat_lift_glm$cousinglm)

moy_priref= tapply(dat_lift_glm$CouSin,dat_lift_glm$quan, mean)/
mean(dat_lift_glm$CouSin)
#par(mar = c(4,4,1,1)+0.1)
summary(moy_glm)
summary(moy_priref)
par(mfrow=c(1,1))
numquan=seq(1,10,by=1)
plot(numquan,moy_glm,xlab="Quantiles",ylab="",type="o",
     ylim=c(0,2),lwd = 1,col="red",main="Coût_GLM vs CouSin")
lines(moy_priref,col="green",type = "o",lty=2)
legend("topleft", legend=c("Coût_glm", "CouSin"),
       col=c("red", "green"), lty=1:2, cex=0.4)

#Diagramme quantile double#

dat_lift_dou=data.frame(test,cousinglm,cousingam)
ratio =dat_lift_dou$cousingam / dat_lift_dou$cousinglm
summary(ratio)
seuil1= c(0.1,quantile(ratio,probs =
c(0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,.9,1))[-c(1,11)],1)
qua = cut(ratio, breaks = seuil1, labels = 1:10)

moy_glm= tapply(dat_lift_dou$cousinglm, qua, mean)/
mean(dat_lift_dou$cousinglm)
moy_gam=tapply(dat_lift_dou$cousingam, qua, mean)/
mean(dat_lift_dou$cousingam)
moy_actu= tapply(dat_lift_dou$CouSin, qua, mean)/mean(dat_lift_dou$CouSin)
summary(moy_actu)
summary(moy_gam)
summary(moy_glm)
plot(x=1:10,moy_actu,xlab="Quantiles",ylab="",type="o",ylim=c(0.0,2.5),
     lwd = 1,col="green",main=" Prime_GLM vs Prime_GAM")
lines(moy_glm,col="red",type = "o",lty=2)
lines(moy_gam,col="blue",type = "o",lty=3)
legend("topleft", legend=c("Coût_reference","Coût_GLM","Coût_GAM"),
       col=c( "green","red","blue"), lty=1:3, cex=0.35)
```

Annexe2 : Code R pour la dépendance des courbes de performance

```
##############Partie théorique mémoire#####################################
rm(list=ls())
library(plotly)
library(copula)

alphas = (0:200)/200
ttaus = c(25,50,75)/100

## Functions

t.integrand = function(v,qy,dC){
  qy(v)*dC(v)
}

####(diPsi) Méthode d'évaluation de la fonction génératrice###
dCu = function(u,v,cop){
  res = diPsi(cop,u,1)/diPsi(cop,pCopula(cbind(u,v),cop),1)
  res[u==0] = 1
  return(res)
}


cc.theoretical = function(a,qy,cop){
  res=rep(0,length(a))
  for(i in 1:length(a))
    res[i] = integrate(t.integrand, 0, 1, qy=qy, dC = function(v)
dCu(v,a[i],cop))$value/
      integrate(qy,0,1)$value
  return(res)
}

data(brvehins1d)
cc.theoretical2 = function(a,qy,dC){
  res=rep(0,length(a))
  for(i in 1:length(a))
    res[i] = integrate(t.integrand, 0, 1, qy=qy, dC = function(v)
dC(v,a[i]))$value/
      integrate(qy,0,1)$value
  return(res)
}

lc.theoretical = function(a,qy){
  res=rep(0,length(a))
  for(i in 1:length(a))
    res[i] = integrate(t.integrand, 0, a[i], qy=qy, dC = function(v) 1)
$value/integrate(qy,0,1)$value
  return(res)
}

abc.integrand = function(v,qx,qy,cop){
  cc.theoretical(v,qy,cop) - lc.theoretical(v,qy=qx)
}

abc.theoretical = function(qx,qy,cop){
  integrate(abc.integrand,0,1,qx=qx,qy=qy,cop=cop)$value
}

abc.integrand2 = function(v,qx,qy,cop){
```

```
  cc.theoretical2(v,qy,cop) - lc.theoretical(v,qy=qx)
}

abc.theoretical2 = function(qx,qy,cop){
  integrate(abc.integrand2,0,1,qx=qx,qy=qy,cop=cop)$value
}



## Variance impact gamma
parm = 1

p1 = plot_ly(x = alphas) %>%
  add_lines(y=lc.theoretical(alphas,function(y) qgamma(y, parm, parm)),
name='Lorenz curve')

parms = c(2,1,0.5)
for(i in 1:length(parms))
  p1 = p1 %>%
  add_lines(y=cc.theoretical(alphas,function(y) qgamma(y, parms[i],
parms[i]), claytonCopula(iTau(claytonCopula(), 0.5))), name =
paste0('variance = ',1/parms[i]))

p1 = p1 %>%
  add_lines(x=c(0,1), y=c(0,1), showlegend=F, line = list(dash=10)) %>%
  layout(xaxis=list(title='Alpha'),yaxis=list(title='Concentration curve
(Gamma)'))

p1
##fin courbe 1############
for(i in parms){
  print(1/i)
  print(abc.theoretical(function(y) qgamma(y, parm, parm), function(y)
qgamma(y, i, i), claytonCopula(iTau(claytonCopula(), 0.5))))
  print(integrate(cc.theoretical,0,1,qy = function(y) qgamma(y, i, i), cop
= claytonCopula(iTau(claytonCopula(), 0.5)))$value)
}
print(integrate(lc.theoretical,0,1,qy = function(y) qgamma(y, parm, parm))
$value)

## Variance impact weibull
parm = 1

p1 = plot_ly(x = alphas) %>%
  add_lines(y=lc.theoretical(alphas,function(y) qweibull(y, parm,parm)),
name='Lorenz curve')

parms1 = c(2,1,0.5)
for(i in 1:length(parms))
  p1 = p1 %>%
  add_lines(y=cc.theoretical(alphas,function(y) qweibull(y,
parms1[i],parms1[i]), claytonCopula(iTau(claytonCopula(), 0.5))),
            name = paste0('variance = ',(gamma(1+2/parms1[i])-(gamma(1+1/
parms1[i]))^2)*(parms1[i])^2))

p1 = p1 %>%
  add_lines(x=c(0,1), y=c(0,1), showlegend=F, line = list(dash=10)) %>%
```

```
  layout(xaxis=list(title='Alpha'),yaxis=list(title='Concentration curve
(loi de weibull)'))

p1
##fin courbe 1############
for(i in parms1){
  print((gamma(1+2/i)-(gamma(1+1/i))^2)*(i)^2)
  print(abc.theoretical(function(y) qweibull(y, parm,parm), function(y)
qweibull(y, i,i), claytonCopula(iTau(claytonCopula(), 0.5))))
  print(integrate(cc.theoretical,0,1,qy = function(y) qweibull(y, i,i), cop
= claytonCopula(iTau(claytonCopula(), 0.5)))$value)
}
print(integrate(lc.theoretical,0,1,qy = function(y) qweibull(y, parm,parm))
$value)


## Variance impact log-normal

s.y = sqrt(log(2))
parm0.5 = 0.9
parm1.5= 1.5
p1 = plot_ly(x = alphas) %>%
  add_lines(y=lc.theoretical(alphas,function(y) qlnorm(y, -(s.y^2)/2,
s.y)), name='Lorenz curve')

  p1 = p1 %>%
    add_lines(y=cc.theoretical(alphas,function(y) qlnorm(y, -
((parm0.5*s.y)^2)/2, parm0.5*s.y), claytonCopula(iTau(claytonCopula(),
0.5))),
              name = paste0('variance = ',2^(0.9^2)-1))
  p1 = p1 %>%
  add_lines(y=cc.theoretical(alphas,function(y) qlnorm(y, -(s.y^2)/2, s.y),
claytonCopula(iTau(claytonCopula(), 0.5))),
            name = paste0('variance = ',1))
  p1 = p1 %>%
  add_lines(y=cc.theoretical(alphas,function(y) qlnorm(y, -
((parm1.5*s.y)^2)/2, parm1.5*s.y), claytonCopula(iTau(claytonCopula(),
0.5))),
            name = paste0('variance = ', 2^(1.5^2)-1))


p1 = p1 %>%
  add_lines(x=c(0,1), y=c(0,1), showlegend=F, line = list(dash=10)) %>%
  layout(xaxis=list(title='Alpha'),yaxis=list(title='Concentration curve
(loi Log-Normale)'))

p1
print(2^(0.9^2)-1)
print(abc.theoretical(function(y) qlnorm(y, -(s.y^2)/2, s.y), function(y)
qlnorm(y, -((parm0.5*s.y)^2)/2, parm0.5*s.y),
claytonCopula(iTau(claytonCopula(), 0.5))))
print(integrate(cc.theoretical,0,1,qy = function(y) qlnorm(y, -
((parm0.5*s.y)^2)/2, parm0.5*s.y), cop =
claytonCopula(iTau(claytonCopula(), 0.5)))$value)
print(1)
print(abc.theoretical(function(y) qlnorm(y, -(s.y^2)/2, s.y), function(y)
qlnorm(y, -(s.y^2)/2, s.y), claytonCopula(iTau(claytonCopula(), 0.5))))
```

```
print(integrate(cc.theoretical,0,1,qy = function(y) qlnorm(y, -(s.y^2)/2,
s.y), cop = claytonCopula(iTau(claytonCopula(), 0.5)))$value)
print(2^(1.5^2)-1)
print(abc.theoretical(function(y) qlnorm(y, -(s.y^2)/2, s.y), function(y)
qlnorm(y, -((parm1.5*s.y)^2)/2, parm1.5*s.y),
claytonCopula(iTau(claytonCopula(), 0.5))))
print(integrate(cc.theoretical,0,1,qy = function(y) qlnorm(y, -
((parm1.5*s.y)^2)/2, parm1.5*s.y), cop =
claytonCopula(iTau(claytonCopula(), 0.5)))$value)
print(integrate(lc.theoretical,0,1,qy = function(y) qlnorm(y, -(s.y^2)/2,
s.y))$value)

##Fin impact variance log-normale



## Dependence impact

parm = 1

p1 = plot_ly(x = alphas) %>%
  add_lines(y=lc.theoretical(alphas,function(y) qgamma(y, parm, parm)),
name='Lorenz curve')

taus = c(0.25,0.5,0.67)

for(i in 1:length(taus))
  p1 = p1 %>%
  add_lines(y=cc.theoretical(alphas,function(y) qgamma(y, parm, parm),
claytonCopula(iTau(claytonCopula(), taus[i]))), name = paste0('tau =
',taus[i]))

p1 = p1 %>%
  add_lines(x=c(0,1), y=c(0,1), showlegend=F, line = list(dash=10)) %>%
  layout(xaxis=list(title='Alpha'),yaxis=list(title='Concentration curve
(Gamma)'))

p1

for(i in taus){
  print(i)
  print(abc.theoretical(function(y) qgamma(y, parm, parm), function(y)
qgamma(y, parm, parm), claytonCopula(iTau(claytonCopula(), i))))
  print(integrate(cc.theoretical,0,1,qy = function(y) qgamma(y, parm,
parm), cop = claytonCopula(iTau(claytonCopula(), i)))$value)
}
print(integrate(lc.theoretical,0,1,qy = function(y) qgamma(y, parm, parm))
$value)


## Dependence impact weibull

parm = 1

p1 = plot_ly(x = alphas) %>%
  add_lines(y=lc.theoretical(alphas,function(y) qweibull(y, parm,parm)),
name='Lorenz curve')
```

```
taus = c(0.25,0.5,0.67)

for(i in 1:length(taus))
  p1 = p1 %>%
  add_lines(y=cc.theoretical(alphas,function(y) qweibull(y, parm,parm),
claytonCopula(iTau(claytonCopula(), taus[i]))), name = paste0('tau =
',taus[i]))

p1 = p1 %>%
  add_lines(x=c(0,1), y=c(0,1), showlegend=F, line = list(dash=10)) %>%
  layout(xaxis=list(title='Alpha'),yaxis=list(title='Concentration curve
(Weibull)'))
p1

#print(abc.theoretical(function(y) qpois(y, parm), function(y) qpois(y,
parm), claytonCopula(iTau(claytonCopula(), 0.24))))
for(i in taus){
  print(i)
  print(abc.theoretical(function(y) qweibull(y, parm,parm), function(y)
qweibull(y, parm,parm), claytonCopula(iTau(claytonCopula(), i))))
  print(integrate(cc.theoretical,0,1,qy = function(y) qweibull(y,
parm,parm), cop = claytonCopula(iTau(claytonCopula(), i)))$value)
}
print(integrate(lc.theoretical,0,1,qy = function(y) qweibull(y, parm,parm))
$value)

#dependance log-normale

p1 = plot_ly(x = alphas) %>%
  add_lines(y=lc.theoretical(alphas,function(y) qlnorm(y, -(s.y^2)/2,
s.y)), name='Lorenz curve')

for(i in 1:length(taus))
  p1 = p1 %>%
  add_lines(y=cc.theoretical(alphas,function(y) qlnorm(y, -(s.y^2)/2, s.y),
claytonCopula(iTau(claytonCopula(), taus[i]))), name = paste0('tau =
',taus[i]))

p1 = p1 %>%
  add_lines(x=c(0,1), y=c(0,1), showlegend=F, line = list(dash=10)) %>%
  layout(xaxis=list(title='Alpha'),yaxis=list(title='Concentration curve
(Log-normale)'))
p1

for(i in taus){
  print(i)
  print(abc.theoretical(function(y) qlnorm(y, -(s.y^2)/2, s.y), function(y)
qlnorm(y, -(s.y^2)/2, s.y), claytonCopula(iTau(claytonCopula(), i))))
  print(integrate(cc.theoretical,0,1,qy = function(y) qlnorm(y, -(s.y^2)/2,
s.y), cop = claytonCopula(iTau(claytonCopula(), i)))$value)
}
print(integrate(lc.theoretical,0,1,qy = function(y) qlnorm(y, -(s.y^2)/2,
s.y))$value)
print(integrate(lc.theoretical,0,1,qy = function(y) qweibull(y,2, 2))
$value)
print(integrate(lc.theoretical,0,1,qy = function(y) qgamma(y,2, 2))$value)
##Distribution impact (gamma,log-normal,weibull)
parm = 1
```

```
s.y = sqrt(log(2))
parm2 = 1.25

p1 = plot_ly(x = alphas) %>%
  add_lines(y=lc.theoretical(alphas,function(y) qgamma(y, parm, parm)),
name='Lorenz curve') %>%
  add_lines(y=cc.theoretical(alphas,function(y) qgamma(y, parm, parm),
claytonCopula(iTau(claytonCopula(), 0.5))), name='Gamma') %>%
  add_lines(y=cc.theoretical(alphas,function(y) qlnorm(y, -(s.y^2)/2, s.y),
claytonCopula(iTau(claytonCopula(), 0.5))), name = 'Log-normal') %>%
  add_lines(y=cc.theoretical(alphas,function(y) qweibull(y, parm ,parm),
claytonCopula(iTau(claytonCopula(), 0.5))), name = paste0('Weibull')) %>%
  add_lines(x=c(0,1), y=c(0,1), showlegend=F, line = list(dash=10)) %>%
  layout(xaxis=list(title='Alpha'),yaxis=list(title='Concentration curve'))

p1

print('gamma')
print(abc.theoretical(function(y) qgamma(y, parm, parm), function(y)
qgamma(y, parm, parm), claytonCopula(iTau(claytonCopula(), 0.5))))
print(integrate(cc.theoretical,0,1,qy = function(y) qgamma(y, parm, parm),
cop = claytonCopula(iTau(claytonCopula(), 0.5)))$value)
print('ln')
print(abc.theoretical(function(y) qgamma(y, parm, parm), function(y)
qlnorm(y, -(s.y^2)/2, s.y), claytonCopula(iTau(claytonCopula(), 0.5))))
print(integrate(cc.theoretical,0,1,qy = function(y) qlnorm(y, -(s.y^2)/2,
s.y), cop = claytonCopula(iTau(claytonCopula(), 0.5)))$value)
print('weibull')
print(abc.theoretical(function(y) qgamma(y, parm, parm), function(y)
qweibull(y, parm, parm), claytonCopula(iTau(claytonCopula(), 0.5))))
print(integrate(cc.theoretical,0,1,qy = function(y) qweibull(y, parm,
parm), cop = claytonCopula(iTau(claytonCopula(), 0.5)))$value)
print(integrate(lc.theoretical,0,1,qy = function(y) qgamma(y, parm, parm))
$value)
## Distribution impact avec weibull

parm = 1
s.y = sqrt(log(2))
parm2 = 1.25

p1 = plot_ly(x = alphas) %>%
  add_lines(y=lc.theoretical(alphas,function(y) qlnorm(y, -(s.y^2)/2,
s.y)), name='Lorenz curve') %>%
  add_lines(y=cc.theoretical(alphas,function(y)qlnorm(y, -(s.y^2)/2, s.y),
claytonCopula(iTau(claytonCopula(), 0.5))), name='Log-normal') %>%
  add_lines(y=cc.theoretical(alphas,function(y) qgamma(y, parm, parm),
claytonCopula(iTau(claytonCopula(), 0.5))), name = 'Gamma') %>%
  add_lines(y=cc.theoretical(alphas,function(y) qweibull(y, parm ,parm),
claytonCopula(iTau(claytonCopula(), 0.5))), name = paste0('Weibull')) %>%
  add_lines(x=c(0,1), y=c(0,1), showlegend=F, line = list(dash=10)) %>%
  layout(xaxis=list(title='Alpha'),yaxis=list(title='Concentration curve'))

p1

print('ln')
print(abc.theoretical(function(y) qlnorm(y, -(s.y^2)/2, s.y), function(y)
qlnorm(y, -(s.y^2)/2, s.y), claytonCopula(iTau(claytonCopula(), 0.5))))
```

```
print(integrate(cc.theoretical,0,1,qy = function(y) qlnorm(y, -(s.y^2)/2,
s.y), cop = claytonCopula(iTau(claytonCopula(), 0.5)))$value)
print('gamma')
print(abc.theoretical(function(y) qlnorm(y, -(s.y^2)/2, s.y), function(y)
qgamma(y, parm, parm), claytonCopula(iTau(claytonCopula(), 0.5))))
print(integrate(cc.theoretical,0,1,qy = function(y) qlnorm(y, -(s.y^2)/2,
s.y), cop = claytonCopula(iTau(claytonCopula(), 0.5)))$value)
print('weibull')
print(abc.theoretical(function(y) qlnorm(y, -(s.y^2)/2, s.y), function(y)
qweibull(y, parm, parm), claytonCopula(iTau(claytonCopula(), 0.5))))
print(integrate(cc.theoretical,0,1,qy = function(y) qweibull(y, parm,
parm), cop = claytonCopula(iTau(claytonCopula(), 0.5)))$value)
print(integrate(lc.theoretical,0,1,qy = function(y) qlnorm(y, -(s.y^2)/2,
s.y))$value)
```

Annexe3 : Code R pour l'étude de cas

```
rm(list=ls())

install.packages("CASdatasets", repos = "http://cas.uqam.ca/pub/",
type="source")
install.packages("dplyr")
library(CASdatasets)
library(IC2)
library(dplyr)
library(data.table)
library(leaps)
library(datasets)
library(tidyr)
library(mice)
library(keras)
library(tis)
library(locfit)
if(!require("parallel")) install.packages("parallel")
?CASdatasets
data(freMTPL2freq)
data(freMTPL2sev)
D=freMTPL2sev %>% group_by(IDpol) %>%
  summarise(ClaimAmount=mean(ClaimAmount))#Coût moyen de sinistre
A=data.frame(IDpol=D$IDpol,AvgClaimAmount=D$ClaimAmount)

C=freMTPL2freq
C$test=C$IDpol %in% A$IDpol
M=subset(C[C$test==FALSE,])
M$AvgClaimAmount=0
N=data.frame(IDpol=M$IDpol,AvgClaimAmount=M$AvgClaimAmount)
data=rbind(N,A)

dat=merge(freMTPL2freq,data,name="IDpol")
dat=data.frame(dat)
dat=dat[dat$Exposure>0,]
dat$ClaimNb=as.numeric(dat$ClaimNb)
dat$Exposure=as.numeric(dat$Exposure)
dat$Density=as.numeric(dat$Density)
dat$BonusMalus=as.numeric(dat$BonusMalus)
dat$VehAge=as.numeric(dat$VehAge)
#dat$VehPower=as.factor(dat$VehPower)
dat$VehBrand=as.factor(dat$VehBrand)
dat$DrivAge=as.numeric(dat$DrivAge)
dat$VehGas=as.factor(dat$VehGas)
dat$Area=as.factor(dat$Area)
dat$Region=as.factor(dat$Region)
#dat=dat[dat$Exposure<=1,]########On conserve juste les expositions non
nuls et inférieurs à 1
dat=dat[dat$ClaimNb<=5 ,]
dat$freqsin=as.numeric(dat$ClaimNb/dat$Exposure)
#Recherche des valeurs manquantes
summary(dat) #pas de valers manquantes
###########Satistiques descriptives###############
#Boites à moustaches#
par(mfrow=c(1,4))
boxplot(dat$Exposure,main="Exposure")
boxplot(dat$AvgClaimAmount,main="ClaimAmount")
boxplot(dat$ClaimNb,main="ClaimNb")
```

```
boxplot(dat$VehAge,main="VehAge")#Cette variable doit être traîtée car elle
contient beaucoup de valeurs abérates
boxplot(dat$BonusMalus,main="BonusMalus")#Pareil
boxplot(dat$Density,main="Density")#egalement
boxplot(dat$DrivAge,main="DriveAge")
boxplot(dat$VehPower,main="VehPower")


############Nombres et pourcentage de sinistres deposés############
claimpercent <- dat %>% group_by(ClaimNb) %>% summarize(count = n()) %>%
mutate(percent = count / sum(count) * 100)
nb_sini_depo=sum(claimpercent$count)-claimpercent$count[1]
pourcen_sin_dep=sum(claimpercent$percent)-claimpercent$percent[1]

#Traitement des valeurs abérantes#
aberante=function(var, na.rm = TRUE) {
  qnt=quantile(var, probs=c(0.05, 0.95), na.rm = na.rm)
  marge= 1.5*IQR(var, na.rm = na.rm)
  var1= var
  var1[var < (qnt[1] - marge)]=  quantile(var,0.05)
  var1[var> (qnt[2] +marge)]=quantile(var,0.95)
  var1 }


dat$VehAge=aberante(dat$VehAge)
dat$DrivAge=aberante(dat$DrivAge)
dat$BonusMalus=aberante(dat$BonusMalus)
dat$Density=aberante(dat$Density)
dat$freqsin=aberante((dat$freqsin))
dat$prime_pure=as.numeric((dat$ClaimNb/dat$Exposure)*(dat$AvgClaimAmount))

dat =dat %>% mutate(
  AvgClaimAmount=pmin(10000,AvgClaimAmount),
  prime_pure=pmin(8000,prime_pure)
) %>%

  select(AvgClaimAmount,ClaimNb,Exposure,BonusMalus,VehPower,VehAge,Density,DrivAge,Veh
         VehBrand,Area,VehGas,Region,prime_pure,freqsin)
summary(dat)
##Histogrammes des variables quantitatives###
par(mfrow=c(1,1))
#hist(dat$prime)
hist(dat$ClaimNb,xlab = "Nombre de sinistres",ylab = "Fréquences",main="")
hist(dat$Exposure,xlab = "Expositions",ylab = "Fréquences",main="")
hist(dat$AvgClaimAmount,xlab = "Coût moyen de sinistres",ylab =
"Fréquences"
     ,main="")
#hist(dat$VehPower, xlab = "Puissance du véhicule",ylab = "Fréquences",
main="")
hist(dat$VehAge, xlab="Âge du véhicule",ylab = "Fréquences",main="")
hist(dat$DrivAge,xlab = "Âge du conducteur",ylab = "Fréquences", main="")
hist(dat$BonusMalus, xlab="Bonus Malus",ylab = "Fréquences", main="")
hist(dat$Density, xlab = "Densité zone résidence",ylab = "Fréquences",
main="")
##Descrition des variables qualitatives##
par(mfrow=c(1,4))
barplot(table(dat$Area),xlab="Code zone",ylab = "Fréquences", main="")
```

```
barplot(table(dat$VehBrand),xlab="Marque du véhicule",ylab = "Fréquences",
main="")
barplot(table(dat$VehGas), xlab="Type de véhicule",ylab = "Fréquences",
main="")
#par(mfrow=c(1,1))
barplot(table(dat$Region),xlab = "Région de résidence",ylab = "Fréquences",
main="")
prop.table(table(dat$Area))
prop.table(table(dat$VehBrand))
prop.table(table(dat$VehGas))
prop.table(table(dat$Region))
#Correlation entre variables explicatives quantitatives#
pear_corr=round(cor(dat[,c("VehAge","DrivAge"
                           ,"BonusMalus","Density","ClaimNb","Exposure","AvgClaimAmount
                    method =  c("pearson")),2)
pear_corr


spear_corr=round(cor(dat[,c("VehAge","DrivAge"
                            ,"BonusMalus","Density","ClaimNb","Exposure","AvgClaimAmoun
                     method =  "spearman"),2)
spear_corr

#V de Cramer por les variables qualitatives

library(DescTools)
matable1=table(dat$Area,dat$VehBrand)
CramerV(matable1)#0.07340852
matable2=table(dat$Area,dat$VehGas)
CramerV(matable2)#0.1314644
matable3=table(dat$Area,dat$Region)
CramerV(matable3)#0.3184873
matable4=table(dat$Area, dat$VehPower)
CramerV(matable4)#0.03976159
matable5=table(dat$VehBrand,dat$VehGas)
CramerV(matable5)#0.1165859
matable6=table(dat$VehBrand,dat$Region)
CramerV(matable6)#0.1299584
matable7=table(dat$VehBrand,dat$VehPower)
CramerV(matable7)#0.1575166
matable8=table(dat$VehGas,dat$Region)
CramerV(matable8)#0.087421
matable9=table(dat$VehGas,dat$VehPower)
CramerV(matable9)# 0.3187086
matable10=table(dat$Region,dat$VehPower)
CramerV(matable10)#0.04361352
tab=table(dat$Area,dat$Area)
CramerV(tab)

#Division de la base de données en deux sous bases: traitement et test#
dat$logdensity=log(dat$Density)
d=sort(sample(nrow(dat),nrow(dat)*0.8))#base de fréquence
train=dat[d,]
test=dat[-d,]
##############################################################################
##########################    Modèle GBM
##############################
##############################################################################
```

```
install.packages("gbm3")
install_github("gbm-developers/gbm3")
#library(gbm)
library(caret)
library(gbm3)
#Modélisation de la fréquence#########
set.seed(123)
gbm_cv = gbmt(ClaimNb ~  offset(log(Exposure))
+VehPower+VehAge+DrivAge+logdensity+Area
              +VehGas+VehBrand+Region,
              data = train,
              distribution = gbm_dist("Poisson"),
              train_params = training_params(num_trees = 5000,
                                             shrinkage = 0.01,
                                             interaction_depth = 5,
                                             min_num_obs_in_node = 100,
                                             num_train = 1*nrow(train)),
              is_verbose = TRUE,
              keep_gbm_data = TRUE,
              cv_folds = 5,
              par_details = gbmParallel(num_threads = 3))

#######nombre d'arbre optimal########
best_iter1 = gbmt_performance(gbm_cv, "cv")
best_iter1
############modélisation du coût###############
train2=train[train$AvgClaimAmount>0,]
cout_cv_gbm = gbmt(AvgClaimAmount ~ VehPower+VehAge+DrivAge+logdensity+Area
                     +VehGas+VehBrand+Region,data = train2,
                     distribution = gbm_dist("Gamma"),
                     train_params = training_params(num_trees = 1500,
                                                    shrinkage = 0.01,
                                                    interaction_depth = 5,
                                                    min_num_obs_in_node =
100,
                                                    num_train =
1*nrow(train2)),
                     is_verbose = TRUE,
                     keep_gbm_data = TRUE,
                   cv_folds = 5,
                     par_details = gbmParallel(num_threads = 3))
best_iter2=gbmt_performance(cout_cv_gbm,"cv")
best_iter2

par(mfrow=c(1,2))
summary(gbm_cv)
summary(cout_cv_gbm)


pred_freq_gbm=predict(gbm_cv,newdata = test,type = "response"
                      ,n.trees = best_iter1)
pred_sev_gbm=predict(cout_cv_gbm,newdata = test,type = "response",n.trees =
best_iter2)
prime_gbm=pred_freq_gbm*pred_sev_gbm
summary(prime_gbm)

###########Dependance partielle#########
par(mfrow=c(2,4))
```

```
for (i in 1:8){
  plot(gbm_cv, var_index = i, num_trees = best_iter2, type="response")
}

##########alcul de la statistique de Friedman's H  entre les variables#
   #pour la fréquance######
par(mfrow=c(1,1))
res = matrix(NA, 8,8)
for (i in 1:7){
  for (j in (i+1):8){
    res[i,j] = interact(gbm_fit_obj = gbm_cv, data=test, var_indices =
c(i,j), best_iter1)
  }
}
diag(res) = 0
row.names(res) = c("VehAge", "DrivAge", "VehPower", "VehBrand","VehGas",
"Region", "logdensity","Area" )
colnames(res) = row.names(res)
library(reshape2)
interact_melt<-melt(res,na.rm = TRUE)
library(ggplot2)
ggplot(data = interact_melt, aes(x=Var1, y=Var2, fill=value)) +
  geom_tile(color = "white")+
  scale_fill_gradient2(low = "white", mid = "gray", high =
"blue",name="Friedman's\nH-statistic") +
  theme(axis.text.x = element_text(angle = 45, vjust = 1,
                                   size = 12, hjust = 1))+
  coord_fixed()###########une forte interaction entre region et logdensity,
ensuite entre logdensity et DrivAge
#pour le coût moyen
###########Dependance partielle#########
par(mfrow=c(2,4))
for (i in 1:8){
  plot(cout_cv_gbm, var_index = i, num_trees = best_iter2, type="response")
}

##########alcul de la statistique de Friedman's H  entre les variables#
#pour la fréquance######
res = matrix(NA, 8,8)
for (i in 1:7){
  for (j in (i+1):8){
    res[i,j] = interact(gbm_fit_obj = cout_cv_gbm, data=test, var_indices =
c(i,j), best_iter2)
  }
}
diag(res) = 0
row.names(res) = c("VehAge", "DrivAge", "VehPower", "VehBrand","VehGas",
"Region", "logdensity","Area" )
colnames(res) = row.names(res)
library(reshape2)
interact_melt<-melt(res,na.rm = TRUE)
library(ggplot2)
ggplot(data = interact_melt, aes(x=Var1, y=Var2, fill=value)) +
  geom_tile(color = "white")+
  scale_fill_gradient2(low = "white", mid = "gray", high =
"blue",name="Friedman's\nH-statistic") +
  theme(axis.text.x = element_text(angle = 45, vjust = 1,
                                   size = 12, hjust = 1))+
```

```
  coord_fixed()
######Vérification de l'égalité des espérance des prime observées et
prédites
test2=test[test$AvgClaimAmount>0,]
sum(test$ClaimNb)-sum(pred_freq_gbm*test$Exposure)
sum(train$ClaimNb)-sum(exp(gbm_cv$fit)*train$Exposure)
sum(train2$AvgClaimAmount)-sum(predict(cout_cv_gbm,newdata = train2,type =
"response",n.trees =  best_iter2))
sum(test2$AvgClaimAmount)-sum(predict(cout_cv_gbm,newdata = test2,type =
"response",n.trees =  best_iter2))
sum(test$AvgClaimAmount)-sum(predict(cout_cv_gbm,newdata = test,type =
"response",n.trees =  best_iter2))
diff_esp_gbm=sum(test$prime_pure)-sum(prime_gbm)

#***********Coûts observés****#

observgbm=exp(cout_cv_gbm$fit)
observgbmf=exp(gbm_cv$fit)
#********Prédiction des coûts*********#
lambgbm=predict(cout_cv_gbm,newdata = test2,type = "response",n.trees =
best_iter2)
lambgbmf=pred_freq_gbm*test$Exposure
#Qualité d'ajustement#
###Erreur d'estimation#####

gbmMSE_test=mean((test2$AvgClaimAmount-lambgbm)^2)
gbmMSE_test
gbmMSE_testf=mean((test$freqsin-lambgbmf)^2)
gbmMSE_testf

hist(test$AvgClaimAmount)
(length(test$AvgClaimAmount)-length(test2$AvgClaimAmount))/
length(test$AvgClaimAmount)
#Deviance#
devgbm= 2*sum((-log((test2$AvgClaimAmount)/lambgbm)+(test2$AvgClaimAmount-
lambgbm)/lambgbm))
devgbm
devgbmf= 2*sum((ifelse(test$ClaimNb==0,0,test$ClaimNb*log( test$ClaimNb/
lambgbmf)))-
                 (test$ClaimNb-lambgbmf))
devgbmf

##LIFT#

#Estimation des courbes de performance#
CC = function(alpha, pi, y){
  q = quantile(pi,alpha)
  sum(y * (pi<=q))/sum(y)
}

LC = function(alpha, pi){
  q = quantile(pi,alpha)
  sum(pi * (pi<=q))/sum(pi)
}
#LIFT GBM#
#Estimation des courbes de performance#
valpha= seq(0,1,by=0.001)
CC_prime_gbm = Vectorize(function(a)
```

```
  CC(a,pi=prime_gbm,y=test$prime_pure))(valpha)
LC_prime_gbm=Vectorize(function(a)
  LC(a,prime_gbm))(valpha)
par(mfrow=c(1,1))
plot(valpha,valpha,lwd=1,xlim=c(0,1),ylim=c(0,1),type ="l",
     xlab=expression(alpha),ylab="",main="Courbes de performances GBM")
lines(valpha,CC_prime_gbm,col="red")
lines(valpha,LC_prime_gbm,col="green",lty=2)
legend("topleft", legend=c("CC", "LC"),
       col=c("red", "green"), lty=1:2, cex=0.8)

#ICC par la méthode des trapèzes#

ICC_gbm=sum(lintegrate(valpha,CC_prime_gbm,xint=valpha))
ILC_gbm=sum(lintegrate(valpha,LC_prime_gbm,xint=valpha))
#ABC
dif_gbm=CC_prime_gbm-LC_prime_gbm
ABC_gbm=sum(lintegrate(valpha,dif_gbm,xint = valpha))

#Index de GINI
I_gbm=1-2*ILC_gbm
ICC_gbm
ILC_gbm
ABC_gbm
I_gbm
#Graphiques quantiles simples
dat_lift_gbm=data.frame(test,prime_gbm)
dat_lift_gbm$obs=1:(dim(dat_lift_gbm)[1])
tri_gbm= order(dat_lift_gbm$prime_gbm,
runif(length(dat_lift_gbm$prime_gbm)))#tri
dat_lift_gbm=dat_lift_gbm[tri_gbm,]
expo_cumgbm=cumsum(dat_lift_gbm$Exposure)#division en quantile d'égale
exposition
expo_totgbm=sum(dat_lift_gbm$Exposure)
seuil_gbm=c(0,1:9*expo_totgbm/10,10.1*expo_totgbm/10)
dat_lift_gbm$quan_gbm=cut(expo_cumgbm, breaks = seuil_gbm, labels = 1:10)
dat_lift_gbm= dat_lift_gbm[order(dat_lift_gbm$obs),]
moy_ppgb= tapply(dat_lift_gbm$prime_pure, dat_lift_gbm$quan_gbm, mean)/
mean(dat_lift_gbm$prime_pure)
moy_gbm= tapply(dat_lift_gbm$prime_gbm,dat_lift_gbm$quan_gbm, mean)/
mean(dat_lift_gbm$prime_gbm)
summary(moy_ppgb)
summary(moy_gbm)
lift_gbm=max(moy_gbm)-min(moy_gbm)
par(mfrow=c(1,1))
numquan=seq(1,10,by=1)
plot(numquan,moy_ppgb,xlab="Quantile",ylab="",type="o",
     ylim=c(0.0,3),lwd = 1,col="green",main="Prime_pure vs Prime_GBM")
lines(moy_gbm,col="yellow",type = "o",lty=2)
legend("topleft", legend=c("Prime_pure", "Prime_GBM"),
       col=c("green", "yellow"), lty=1:2, cex=0.8)

###########tableau des indices pour les quantiles simple

tab1=data.frame(seq(1,10))
colnames(tab1)="Déciles"
tab1$tot_exposition=tapply(dat_lift_gbm$Exposure, dat_lift_gbm$quan_gbm,
sum)
```

```
tab1$prime_pure=tapply(dat_lift_gbm$prime_pure, dat_lift_gbm$quan_gbm,
mean)
tab1$prime_gbm=tapply(dat_lift_gbm$prime_gbm, dat_lift_gbm$quan_gbm, mean)
tab1$indice_prime_pure=tab1$prime_pure/mean(dat_lift_gbm$prime_pure)
tab1$indice_prime_gbm=tab1$prime_gbm/mean(dat_lift_gbm$prime_gbm)
##############################################################################
#########################    Modèle GLM
##############################
##############################################################################
datglm=data.frame(dat)
datglm$AreaGLM<- as.integer( datglm$Area)
datglm$VehPowerGLM  <- as.factor(pmin( datglm$VehPower , 9 ))
VehAgeGLM  <- cbind(c( 0 : 110 ), c( 1 , rep( 2 , 10 ), rep( 3 , 100 )))
datglm$VehAgeGLM  <- as.factor( VehAgeGLM[datglm$VehAge+ 1 , 2 ])
datglm[,"VehAgeGLM"] <- relevel( datglm[,"VehAgeGLM"], ref = "2" )

DrivAgeGLM  <- cbind(c( 18 : 100 ), c(rep( 1 , 21 - 18 ), rep( 2 , 26 - 21
),
                                      rep( 3 , 31 - 26 ), rep( 4 , 41 -31
),
                                      rep( 5 , 51 - 41 ),
                                      rep( 6 , 71 - 51 ), rep( 7 , 101 - 71
)))
datglm$DrivAgeGLM  <- as.factor( DrivAgeGLM[datglm$DrivAge- 17 , 2 ] )
datglm[,"DrivAgeGLM"] <- relevel( datglm[,"DrivAgeGLM"], ref ="5" )
datglm$BonusMalusGLM  <- as.integer(pmin( datglm$BonusMalus , 150 ))
datglm$DensityGLM  <- as.numeric(log( datglm$Density ))
datglm[,"Region"] <-relevel(datglm[,"Region"], ref="Centre")
datglm[,"VehBrand"] <-relevel(datglm[,"VehBrand"], ref="B12")

dc=sort(sample(nrow(datglm),nrow(datglm)*0.8))
train_data=datglm[dc,]
test_data=datglm[-dc,]
set.seed(1234)
freq0=glm(ClaimNb~VehPowerGLM+VehAgeGLM
         +DensityGLM+AreaGLM+VehGas+VehBrand+Region+
           DrivAgeGLM
         ,family = poisson(link = "log"),offset=log(Exposure)
         ,data = train_data)
summary(freq0)
AIC(freq0)

step(freq0)
set.seed(1234)
freq=glm(ClaimNb~VehPowerGLM+VehAgeGLM
         +DensityGLM+VehGas+VehBrand+Region+
           DrivAgeGLM
         ,family = poisson(link = "log"),offset=log(Exposure)
         ,data = train_data)
summary(freq)
AIC(freq)

set.seed(1234)
freq_inter=glm(ClaimNb~VehPowerGLM+VehAgeGLM
         +DensityGLM+VehGas+VehBrand+Region+
           DrivAgeGLM+Region:DensityGLM+Region:VehAgeGLM
         ,family = poisson(link = "log"),offset=log(Exposure)
         ,data = train_data)
```

```
summary(freq_inter)
AIC(freq_inter)

train_data2=train_data[train_data$AvgClaimAmount>0,]
train_data2[,"DrivAgeGLM"] <- relevel( train_data2[,"DrivAgeGLM"], ref ="6"
)
train_data2[,"VehPowerGLM"] <- relevel( train_data2[,"VehPowerGLM"], ref
="6" )
train_data2[,"VehBrand"] <-relevel(train_data2[,"VehBrand"], ref="B2")
set.seed(1234)
cout0=glm(AvgClaimAmount~VehPowerGLM+VehAgeGLM
          +DensityGLM+AreaGLM+VehGas+VehBrand+Region+
            DrivAgeGLM
          ,family = Gamma(link = "log")
          ,data = train_data2)
summary(cout0)
AIC(cout0)
step(cout0)
set.seed(1234)
cout_inter=glm(AvgClaimAmount~VehPowerGLM + VehAgeGLM + DensityGLM +
AreaGLM +
                 VehGas + Region + DrivAgeGLM+DensityGLM:Region
         ,family = Gamma(link = "log")
         ,data = train_data2)
summary(cout_inter)
AIC(cout_inter)
set.seed(1234)
cout=glm(AvgClaimAmount~VehPowerGLM+VehAgeGLM+DensityGLM+Region+DrivAgeGLM
         ,family = Gamma(link = "log"),data = train_data2)
summary(cout)
AIC(cout)
pre_freq_glm=predict.glm(freq,newdata = test_data,type="response")
pre_sev_glm=predict.glm(cout,newdata = test_data,type="response")
prime_glm=pre_freq_glm*pre_sev_glm
summary(prime_glm)

#***********Coûts observés****#
obprob=cout$fitted.values
#********Prédiction des coûts*********#
test_data2=test_data[test_data$AvgClaimAmount>0,]
lambdaK=predict.glm(cout,test_data,type = "response")
lambdaK1=predict.glm(cout,test_data2,type = "response")
###Erreur d'estimation#####
glmMSE_test2=mean((test_data2$AvgClaimAmount-lambdaK1)^2)
glmMSE_test2
glmMSE_test=mean((test_data$AvgClaimAmount-lambdaK)^2)
glmMSE_test

lambdaKf=predict.glm(freq,test_data,type = "response")
glmMSE_testf=mean((test_data$freqsin-lambdaKf)^2)
glmMSE_testf
glmMSE_trainf-glmMSE_testf

#Vérification des sommes#
sum(train_data$ClaimNb)-sum(predict(freq,train_data,type = "response"))
sum(test_data$ClaimNb)-sum(lambdaKf)
sum(train_data2$AvgClaimAmount)-sum(predict(cout,train_data2,type =
"response"))
```

```
sum(test_data2$AvgClaimAmount)-sum(predict(cout,test_data2,type =
"response"))
sum(test_data$AvgClaimAmount)-sum(predict(cout,test_data,type =
"response"))
sum(test_data$prime_pure)-sum(prime_glm)
#Deviance#
devglm= 2*sum((-log(test_data2$AvgClaimAmount/lambdaK1)+
                 (test_data2$AvgClaimAmount-lambdaK1)/lambdaK1))
devglm
devglmf= 2*sum((ifelse(test_data$ClaimNb==0,0,test_data$ClaimNb*log(
test_data$ClaimNb/lambdaKf)))-
                 (test_data$ClaimNb-lambdaKf))
devglmf
deviance(cout)
deviance(freq)

CC_prime_glm = Vectorize(function(a)
  CC(a,pi=prime_glm ,y=test_data$prime_pure))(valpha)
LC_prime_glm=Vectorize(function(a)
  LC(a,pi=prime_glm))(valpha)

par(mfrow=c(1,1))
plot(valpha,CC_prime_glm,lwd=1,xlim=c(0,1),ylim=c(0,1),type ="l",col="red",
     xlab=expression(alpha),ylab="", main="Courbe de concentration et
courbe de Lorenz")
#lines(valpha,CC_prime_glm,col="red")
lines(valpha,LC_prime_glm,col="green",lty=2)
lines(valpha,valpha)
legend("topleft", legend=c("CC", "LC"),
       col=c("red", "green"), lty=1:2, cex=0.8)
#ICC par la méthode des trapèzes#

ICC=sum(lintegrate(valpha,CC_prime_glm,xint=valpha))
ILC=sum(lintegrate(valpha,LC_prime_glm,xint=valpha))
#ABC
diffe=CC_prime_glm-LC_prime_glm
ABC=sum(lintegrate(valpha,diffe,xint = valpha))

#Index de GINI
I=1-2*ILC

#Graphiques quantiles simples
dat_lift=data.frame(test_data,prime_glm)
dat_lift$obs=1:(dim(dat_lift)[1])
tri= order(dat_lift$prime_glm, runif(length(dat_lift$prime_glm)))#tri
dat_lift=dat_lift[tri,]
expo_cu=cumsum(dat_lift$Exposure)#division en quantile d'égale exposition
expo_to=sum(dat_lift$Exposure)
seuil=c(0,1:9*expo_to/10,10.1*expo_to/10)
dat_lift$quan=cut(expo_cu, breaks = seuil, labels = 1:10)
dat_lift= dat_lift[order(dat_lift$obs),]
tapply(dat_lift$Exposure, dat_lift$quan, sum)
moy_glm= tapply(dat_lift$prime_glm, dat_lift$quan, mean)/
mean(dat_lift$prime_glm)

moy= tapply(dat_lift$prime_pure,dat_lift$quan, mean)/
mean(dat_lift$prime_pure)
summary(moy_glm)
```

```
summary(moy)

numquan=seq(1,10,by=1)
plot(numquan,moy,xlab="Quantile",ylab="",type="o",
     ylim=c(0,3),lwd = 1,col="green",main="Prime_GLM vs Prime_pure")
lines(moy_glm,col="red",type = "o",lty=2)
legend("topleft", legend=c("Prime_glm", "Prime_pure"),
       col=c("green", "red"), lty=1:2, cex=0.8)
lift_glm=max(moy_glm)-min(moy_glm)

################################################################################
##########################    Modèle GAM
##############################
################################################################################

install.packages("gam")
library(gam)
library(mgcv)
memory.limit()
memory.limit(size=8080*4)
train$logdensity=log(train$Density)
game01_inter=gam(data=train,formula =ClaimNb ~offset(log(Exposure))
+VehPower+ s(DrivAge)
           + s(VehAge)+VehBrand+Region+s(logdensity)
+VehGas+Region:logdensity
           , family = poisson(),select= TRUE)

summary(game01_inter)
AIC(game01_inter)
#deuxième GAM#

train2=train[train$AvgClaimAmount>0,]
game11_inter=gam(data=train2,formula =AvgClaimAmount~VehPower+ s(DrivAge)
           + s(VehAge)+Region+s(logdensity)+Region:logdensity
           , family = Gamma(link="log"),select= TRUE)

summary(game11_inter)
AIC(game11_inter)

game01=gam(data=train,formula =ClaimNb ~offset(log(Exposure))+VehPower+
s(DrivAge)
           + s(VehAge)+VehBrand+Region+s(logdensity)+VehGas
           , family = poisson(),select= TRUE)

summary(game01)
AIC(game01)
#deuxième GAM#

train2=train[train$AvgClaimAmount>0,]
game11=gam(data=train2,formula =AvgClaimAmount~VehPower+ s(DrivAge)
           + s(VehAge)+Region+s(logdensity)
           , family = Gamma(link="log"),select= TRUE)

summary(game11)
AIC(game11)
test$logdensity=log(test$Density)
pred_freq_game=predict.gam(game01,test,type = "response")/test$Exposure
pred_sev_game=predict.gam(game11,test,type = "response")
```

```
Prime_game=pred_freq_game*pred_sev_game
summary(Prime_game)

########Vérification des sommes####
sum(train$ClaimNb)-sum(predict.gam(game01,train,type = "response"))
sum(test$ClaimNb)-sum(predict.gam(game01,test,type = "response"))
sum(train2$AvgClaimAmount)-sum(predict.gam(game11,train2,type =
"response"))
sum(test2$AvgClaimAmount)-sum(predict.gam(game11,test2,type = "response"))
sum(test$AvgClaimAmount)-sum(predict.gam(game11,test,type = "response"))
sum(test$prime_pure)-sum(Prime_game)
##############Coût############

lambgam=predict.gam(game11,test,type = "response")
gamMSE_test=mean((test$AvgClaimAmount-lambgam)^2)
gamMSE_test

lambgam1=predict.gam(game11,test2,type = "response")
gamMSE_test1=mean((test2$AvgClaimAmount-lambgam1)^2)
gamMSE_test1
############Frequence############

lambgamf=predict.gam(game01,test,type = "response")
gamMSE_testf=mean((test$freqsin-lambgamf)^2)
gamMSE_testf

#Deviance#
devgam= 2*sum((-log((test2$AvgClaimAmount)/lambgam1)+(test2$AvgClaimAmount-
lambgam1)/lambgam1))
devgam
devgamf= 2*sum((ifelse(test$ClaimNb==0,0,test$ClaimNb*log( test$ClaimNb/
lambgamf)))-
                 (test$ClaimNb-lambgamf))
devgamf
deviance(game01)
deviance(game11)

#LIFT GAM#
#Estimation des courbes de performance#

CC_prime_game = Vectorize(function(a)
  CC(a,pi=Prime_game,y=test$prime_pure))(valpha)
LC_prime_game=Vectorize(function(a)
  LC(a,Prime_game))(valpha)

par(mfrow=c(1,1))
plot(valpha,CC_prime_game,lwd=1,xlim=c(0,1),ylim=c(0,1),type
="l",col="red",
     xlab=expression(alpha),ylab="",main="Courbes de performances GAM")
#lines(valpha,CC_prime_game,col="red")
lines(valpha,LC_prime_game,col="green",lty=2)
lines(valpha,valpha)
legend("topleft", legend=c("CC", "LC"),
       col=c("red", "green"), lty=1:2, cex=0.8)
#ICC par la méthode des traèzes#
ICC_game=sum(lintegrate(valpha,CC_prime_game,xint=valpha))
ILC_game=sum(lintegrate(valpha,LC_prime_game,xint=valpha))
#ABC
```

```
dif_gam=CC_prime_game-LC_prime_game
ABC_game=sum(lintegrate(valpha,dif_gam,xint = valpha))
#Index de GINI
I_game=1-2*ILC_game
ICC_game
ILC_game
ABC_game
I_game
#Graphiques quantiles simples
dat_lift_ga=data.frame(test,prime_glm,Prime_game)
dat_lift_ga$obs=1:(dim(dat_lift_ga)[1])
tri_ga= order(dat_lift_ga$Prime_game,
runif(length(dat_lift_ga$Prime_game)))#tri
dat_lift_ga=dat_lift_ga[tri_ga,]
expo_cum=cumsum(dat_lift_ga$Exposure)#division en quantile d'égale
exposition
expo_tot=sum(dat_lift_ga$Exposure)
seuil_ga=c(0,1:9*expo_tot/10,10.1*expo_tot/10)
dat_lift_ga$quan_ga=cut(expo_cum, breaks = seuil_ga, labels = 1:10)
dat_lift_ga= dat_lift_ga[order(dat_lift_ga$obs),]
moy_pp= tapply(dat_lift_ga$prime_pure, dat_lift_ga$quan_ga, mean)/
mean(dat_lift_ga$prime_pure)
moy_gam= tapply(dat_lift_ga$Prime_game,dat_lift_ga$quan_ga, mean)/
mean(dat_lift_ga$Prime_game)
summary(moy_pp)
summary(moy_gam)
lift_gam=max(moy_gam)-min(moy_gam)
par(mfrow=c(1,2))
numquan=seq(1,10,by=1)
plot(numquan,moy_pp,xlab="Quantile",ylab="",type="o",
     ylim=c(0.0,3.0),lwd = 1,col="green",main="Prime_pure vs Prime_GAM")
lines(moy_gam,col="blue",type = "o",lty=2)
legend("topleft", legend=c("Prime_pure", "Prime_GAM"),
       col=c("green", "blue"), lty=1:2, cex=0.4)
#Diagramme quantile double#
dat_lift_ga=data.frame(test,prime_glm,Prime_game)
dat_lift_ga$ratio_gamglm =dat_lift_ga$Prime_game / dat_lift_ga$prime_glm
dat_lift_ga$obs=1:(dim(dat_lift_ga)[1])
tri_ga= order(dat_lift_ga$ratio_gamglm,
runif(length(dat_lift_ga$ratio_gamglm)))#tri
dat_lift_ga=dat_lift_ga[tri_ga,]
expo_cum=cumsum(dat_lift_ga$Exposure)#division en quantile d'égale
exposition
expo_tot=sum(dat_lift_ga$Exposure)
seuil_ga=c(0,1:9*expo_tot/10,10.1*expo_tot/10)
dat_lift_ga$quan_ga=cut(expo_cum, breaks = seuil_ga, labels = 1:10)
dat_lift_ga= dat_lift_ga[order(dat_lift_ga$obs),]
tapply(dat_lift_ga$Exposure, dat_lift_ga$quan_ga, sum)
moy_gampp= tapply(dat_lift_ga$prime_pure, dat_lift_ga$quan_ga, mean)/
mean(dat_lift_ga$prime_pure)
moy_gamglm=tapply(dat_lift_ga$prime_glm, dat_lift_ga$quan_ga, mean)/
mean(dat_lift_ga$prime_glm)
moy_gamdou= tapply(dat_lift_ga$Prime_game, dat_lift_ga$quan_ga, mean)/
mean(dat_lift_ga$Prime_game)
summary(moy_gampp)
summary(moy_gamglm)
summary(moy_gamdou)
plot(x=1:10,moy_gampp,xlab="Quantile",ylab="",type="o",ylim=c(0.0,3.0),
```

```
     lwd = 1,col="green",main=" Prime_GLM vs Prime_GAM")
lines(moy_gamglm,col="red",type = "o",lty=2)
lines(moy_gamdou,col="blue",type = "o",lty=3)
legend("topright", legend=c("Prime_pure","Prime_GLM","Prime_GAM"),
       col=c( "green","red","blue"), lty=1:3, cex=0.4)

#Diagramme quantile double#
###########GAM vs GBM##############
dat_lift_gbm=data.frame(test,prime_gbm,Prime_game)
dat_lift_gbm$ratio_gamgbm =dat_lift_gbm$prime_gbm / dat_lift_gbm$Prime_game
summary(ratio_gamgbm)
dat_lift_gbm$obs=1:(dim(dat_lift_gbm)[1])

tri_ga= order(dat_lift_gbm$ratio_gamgbm,
runif(length(dat_lift_gbm$ratio_gamgbm)))#tri

dat_lift_gbm=dat_lift_gbm[tri_ga,]
expo_cum=cumsum(dat_lift_gbm$Exposure)#division en quantile d'égale
exposition
expo_tot=sum(dat_lift_gbm$Exposure)
seuil_ga=c(0,1:9*expo_tot/10,10.1*expo_tot/10)
dat_lift_gbm$quan_ga=cut(expo_cum, breaks = seuil_ga, labels = 1:10)
dat_lift_gbm= dat_lift_gbm[order(dat_lift_gbm$obs),]
tapply(dat_lift_gbm$Exposure, dat_lift_gbm$quan_ga, sum)
#dat_lift_gbm$quan_gbm=cut(expo_cumgbm, breaks = seuil_gbm, labels = 1:10)
#seuil_gamgbm= c(0.1,quantile(ratio_gamgbm,probs =
c(0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,.9,1))[-c(1,11)],1)
#qua_gamgbm = cut(ratio_gamgbm, breaks = seuil_gamgbm, labels = 1:10)

moy_Ppgbm= tapply(dat_lift_gbm$prime_pure, qua_gamgbm, mean)/
mean(dat_lift_gbm$prime_pure)
moy_gamgbm=tapply(dat_lift_gbm$Prime_game, qua_gamgbm, mean)/
mean(dat_lift_gbm$Prime_game)
moy_gbmdou= tapply(dat_lift_gbm$prime_gbm, qua_gamgbm, mean)/
mean(dat_lift_gbm$prime_gbm)
summary(moy_Ppgbm)
summary(moy_gamgbm)
summary(moy_gbmdou)
par(mfrow=c(1,2))
plot(x=1:10,moy_Ppgbm,xlab="Quantile",ylab="",type="o",ylim=c(0.0,3.0),
     lwd = 1,col="green",main=" Prime_GBM vs Prime_GAM")
lines(moy_gamgbm,col="blue",type = "o",lty=2)
lines(moy_gbmdou,col="yellow",type = "o",lty=3)
legend("topright", legend=c("Prime_pure","Prime_GAM","Prime_GBM"),
       col=c( "green","blue","yellow"), lty=1:3, cex=0.4)

###############tableaux des indices###########

tab=data.frame(seq(1,10))
colnames(tab)="Déciles"
tab$tot_exposition=tapply(dat_lift_gbm$Exposure, dat_lift_gbm$quan_ga, sum)
tab$prime_pure=tapply(dat_lift_gbm$prime_pure, qua_gamgbm, mean)
tab$prime_game=tapply(dat_lift_gbm$Prime_game, qua_gamgbm, mean)
tab$prime_gbm=tapply(dat_lift_gbm$prime_gbm, qua_gamgbm, mean)
tab$indice_prime_pure=tab$prime_pure/mean(dat_lift_gbm$prime_pure)
tab$indice_prime_game=tab$prime_game/mean(dat_lift_gbm$Prime_game)
tab$indice_prime_gbm=tab$prime_gbm/mean(dat_lift_gbm$prime_gbm)
```

```
##########GBM vs GLM##############
dat_lift_gbm=data.frame(test,prime_gbm,prime_glm)
ratio_glmgbm =dat_lift_gbm$prime_gbm / dat_lift_gbm$prime_glm
summary(ratio_glmgbm)
seuil_glmgbm= c(0.1,quantile(ratio_glmgbm,probs =
c(0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,.9,1))[-c(1,11)],1)
qua_glmgbm = cut(ratio_glmgbm, breaks = seuil_glmgbm, labels = 1:10)

moy_ppgbm= tapply(dat_lift_gbm$prime_pure, qua_glmgbm, mean)/
mean(dat_lift_gbm$prime_pure)
moy_glmgbm=tapply(dat_lift_gbm$prime_glm, qua_glmgbm, mean)/
mean(dat_lift_gbm$prime_glm)
moy_gbmdou= tapply(dat_lift_gbm$prime_gbm, qua_glmgbm, mean)/
mean(dat_lift_gbm$prime_gbm)
summary(moy_ppgbm)
summary(moy_glmgbm)
summary(moy_gbmdou)
plot(x=1:10,moy_ppgbm,xlab="Quantile",ylab="",type="o",ylim=c(0.0,3.0),
     lwd = 1,col="green",main=" Prime_GBM vs Prime_GLM")
lines(moy_glmgbm,col="red",type = "o",lty=2)
lines(moy_gbmdou,col="yellow",type = "o",lty=3)
legend("topright", legend=c("Prime_pure","Prime_GLM","Prime_GBM"),
       col=c( "green","red","yellow"), lty=1:3, cex=0.4)

#Courbe de Lorenz de la prime pure #
LC_prime_pure=Vectorize(function(a)
  LC(a,pi=test$prime_pure))(valpha)
#Différente courbes de concentration#
par(mfrow=c(1,2))
plot(valpha,valpha,lwd=1,xlim=c(0,1),ylim=c(0,1),type ="l",lty=1,
     xlab=expression(alpha),ylab="",main = "Courbes de concentration")
lines(valpha,CC_prime_glm,col="red",lty=1)
lines(valpha,CC_prime_game,col="blue",lty=2)
lines(valpha,CC_prime_gbm,col="yellow",lty=3)
legend("topleft", legend=c("CC_GLM","CC_GAM","CC_GBM"),
       col=c("red","blue","yellow"), lty=1:5, cex=0.4)
#Différentes courbes de Lorenz#
plot(valpha,valpha,lwd=1,xlim=c(0,1),ylim=c(0,1),type ="l",lty=1,
     xlab=expression(alpha),ylab="",main = "Courbes de Lorenz")
lines(valpha,LC_prime_glm,col="red",lty=1)
lines(valpha,LC_prime_game,col="blue",lty=2)
lines(valpha,LC_prime_gbm,col="yellow",lty=3)
#lines(valpha,LC_prime_pure,col="green",lty=4)
legend("topleft", legend=c("LC_GLM","LC_GAM","LC_GBM"),
       col=c("red","blue","yellow"), lty=1:5, cex=0.4)



#distribution des primes suivants les modèles#
par(mfrow=c(1,3))

hist(prime_glm[prime_glm>0 & prime_glm<=2300], xlab = "prime_GLM",main
="",col = "red",ylim =c(0, 100000),xlim = c(0,2300),
     breaks =seq(0,2300,by=100) )
hist(Prime_game,xlab = "prime_GAM",col = "blue",ylim =c(0, 100000),xlim =
c(0,2300)
```

```
     ,main="Distribution des primes prédites")
hist(prime_gbm,xlab = "prime_GBM",main="",col="yellow",ylim =c(0,
100000),xlim = c(0,2300))


#Variations de ICC,ILC et ABC suivant les valeurs de alpha"
vec_ICC=seq(0,1,by=0.01)
vec_ILC=seq(0,1,by=0.01)
vec_ABC=seq(0,1,by=0.01)
for ( i in 1:length(valpha)){
  vepsilon=seq(0,valpha[i],by=0.005)
  epsi_CC_prime_glm = Vectorize(function(a)
    CC(a,pi=prime_glm,y=test_data$prime_pure))(vepsilon)
  vec_ICC[i]=sum(lintegrate(vepsilon,epsi_CC_prime_glm,xint = vepsilon))
  epsi_LC_prime_glm = Vectorize(function(a)
    LC(a,pi=prime_glm))(vepsilon)
  vec_ILC[i]=sum(lintegrate(vepsilon,epsi_LC_prime_glm,xint = vepsilon))
  vec_ABC[i]=vec_ICC[i]-vec_ILC[i]
}

vec_ICC_ga=seq(0,1,by=0.01)
vec_ILC_ga=seq(0,1,by=0.01)
vec_ABC_ga=seq(0,1,by=0.01)
for ( i in 1:length(valpha)){
  vepsilon=seq(0,valpha[i],by=0.005)
  epsi_CC_prime_game = Vectorize(function(a)
    CC(a,pi=Prime_game,y=test$prime_pure))(vepsilon)
  vec_ICC_ga[i]=sum(lintegrate(vepsilon,epsi_CC_prime_game,xint =
vepsilon))
  epsi_LC_prime_game = Vectorize(function(a)
    LC(a,pi=Prime_game))(vepsilon)
  vec_ILC_ga[i]=sum(lintegrate(vepsilon,epsi_LC_prime_game,xint =
vepsilon))
  vec_ABC_ga[i]=vec_ICC_ga[i]-vec_ILC_ga[i]
}
vec_ICC_gbm=seq(0,1,by=0.01)
vec_ILC_gbm=seq(0,1,by=0.01)
vec_ABC_gbm=seq(0,1,by=0.01)
for ( i in 1:length(valpha)){
  vepsilon=seq(0,valpha[i],by=0.005)
  epsi_CC_prime_gbm = Vectorize(function(a)
    CC(a,pi=prime_gbm,y=test$prime_pure))(vepsilon)
  vec_ICC_gbm[i]=sum(lintegrate(vepsilon,epsi_CC_prime_gbm,xint =
vepsilon))
  epsi_LC_prime_gbm = Vectorize(function(a)
    LC(a,pi=prime_gbm))(vepsilon)
  vec_ILC_gbm[i]=sum(lintegrate(vepsilon,epsi_LC_prime_gbm,xint =
vepsilon))
  vec_ABC_gbm[i]=vec_ICC_gbm[i]-vec_ILC_gbm[i]
}



par(mfrow=c(1,3))

plot(valpha,vec_ICC,type="l",ylab="ICC",main = "Variations de
l'ICC",lty=1,col="red")
lines(valpha,vec_ICC_ga,col="blue",lty=2)
```

```
lines(valpha,vec_ICC_gbm,col="yellow",lty=3)

legend("topleft", legend=c("ICC_GLM","ICC_GAM","ICC_GBM"),
       col=c("red","blue","yellow"), lty=1:5, cex=0.4)

plot(valpha,vec_ILC,type ="l",ylab="ILC",lty=1,
     main = "Variations de ILC",col="red")
lines(valpha,vec_ILC_ga,type ="l",col="blue",lty=2)
lines(valpha,vec_ILC_gbm,type ="l",col="yellow",lty=3)
legend("topleft", legend=c("ILC_GLM","ILC_GAM","ILC_GBM"),
       col=c("red","blue","yellow"), lty=1:5, cex=0.4)

plot(valpha,vec_ABC,type ="l",ylab="ABC",main = "Variations de
ABC",lty=1,col="red")
lines(valpha,vec_ABC_ga,type ="l",col="blue",lty=2)
lines(valpha,vec_ABC_gbm,type ="l",col="yellow",lty=3)
legend("topleft", legend=c("ABC_GLM","ABC_GAM","ABC_GBM"),
       col=c("red","blue","yellow"), lty=1:5, cex=0.4)


##############################################################################
#########################    FIN
#######################################
##############################################################################
```