Institut de Statistique, Biostatistique et Actuariat

SEGMENTATION NON-VIE A L'AIDE DU PACKAGE R : "insurancerating"

Mémoire présenté en vue de
l'obtention du master
en sciences actuarielles par :

SOUBGUI FOTSO Brice Boris
KUEMATSA CHE Thierry Bolivard

Membres du jury :

Prof. DENUIT Michel, *Promoteur*
Prof. ARS Pierre, *Lecteur*

Louvain-La-Neuve
Août 2021

Remerciements

Nous tenons tout d'abord à remercier notre promoteur le professeur Michel Denuit et son assistante Ndeye Arame Seck pour le suivi et les conseils tout au long de notre recherche. Nous remercions également nos proches pour leur soutien au cours de notre master, ainsi que les personnes ayant pris le temps de lire et commenter notre travail.

Table des matières

Table des figures

Liste des tableaux

Chapitre 1

Introduction générale

1.1 Introduction générale

Historiquement, l'assurance s'est bâtie sur l'expression de la solidarité d'un ensemble d'individus soumis à des risques identiques. Mais la hausse soudaine du volume de données et l'apparition de nouvelles méthodes et moyens de calcul à la disposition des assureurs ont permis l'évolution de ce secteur. Dans certains domaines de l'assurance à forte concurrence tel que en assurance IARD (incendies, accidents et risques divers), l'individualisation des tarifs est de plus en plus pratiquée. La segmentation est donc un processus statistique consistant à regrouper les observations qui partagent plus ou moins certaines caractéristiques dans un nombre raisonnable de segment, de telle sorte que les individus appartenant à un même segment se ressemblent le plus possible et diffèrent autant que possible de ceux d'un segment différent. L'idée est d'appliquer le même tarif aux individus appartenant au même segment. Si une compagnie d'assurance ne segmente pas son portefeuille et où donc les assurés payent la même prime, les mauvais risques seront plus enclins à se faire assurer dans cette compagnie tandis que les bons risques, jugeant la prime trop chère iront chez la concurrence, ce qui au fil du temps mènera la compagnie à la faillite, car elle n'assurera plus que de mauvais risques. Toutefois, une segmentation des risques trop fine, c'est-à-dire à l'échelle même de l'individu, n'est pas suffisamment robuste pour être effective. Dès lors on comprend l'importance d'une juste classification, voir [Denuit et al., 2007] .

La construction des classes de risque est assez simple quand les facteurs de risque sont tous catégoriels, ce qui n'est pas toujours le cas. Bien que les facteurs de risque continus peuvent être perçus comme des variables catégorielles à plusieurs niveaux (des fois des centaines), il n'est cependant pas pertinent de prendre tous ces niveaux en compte individuellement de peur qu'il en résulte trop de classes de risque, certaines contenant très peu d'observations. On s'interroge de ce fait sur la façon de rendre catégorielles ces variables de manière optimale afin de capturer l'impact exact de chaque variable continue sur la variable réponse, ceci particulièrement dans les modèles linéaires, pour pallier à la catégorisation souvent arbitraire des covariables, qui ne tient pas compte de la variable

réponse. Plusieurs méthodes existent à cet effet, dans le cadre de ce travail, nous nous attarderons sur celle proposée par le package "insurancerating" conçu par l'actuaire Martin Haringa. L'auteur du package s'inspire en grande partie de l'article de [Henckaerts et al., 2018] dans lequel il est détaillé une procédure permettant d'intégrer de façon optimale les facteurs de risques continus et géographiques(latitude et longitude) dans un tarif d'assurance. [Henckaerts et al., 2018] propose de commencer par établir un modèle additif généralisé (GAM) au nombre de sinistre observé et/ou la gravité correspondante, car ces modèles GAMs permettent une modélisation statistique flexible en présence de différents types de facteurs de risque, ensuite il propose de classer les facteurs de risque spatiaux en utilisant l'algorithme des ruptures naturelles de Fisher et les facteurs de risque continus en utilisant des arbres évolutionnaires.

L'objectif principal de ce mémoire est l'évaluation de l'outil de segmentation proposé par Martin Haringa, en l'occurrence le package R insurancerating. Pour le faire nous allons premièrement appliquer cet outil à un jeu de données afin de déterminer les classes qui en découlent, ensuite nous utiliserons différentes méthodes de segmentation afin de juger la qualité de cet outil. Notre travail s'étale sur trois parties, la première partie présente et décrit le package insurancerating ; ici nous analyserons de près les principales fonctions du package, c'est-à-dire celles qui sont en rapport direct avec la notion de segmentation ; nous rappellerons également les fondements théoriques au coeur de ces différentes fonctions. La deuxième partie, plus pratique, consistera en l'application de cet outil à un portefeuille d'assurance TPL(Third party Liability). Dans la troisième partie, nous appliquerons au même portefeuille d'assurance TPL une autre méthode de segmentation, basée sur l'approche du package, mais utilisant un modèle GAM complet, c'est-à-dire prenant en compte tous les facteurs de risque pertinents, ceci dans l'optique d'une comparaison que nous effectuerons au fur et à mesure des résultats obtenus, enfin nous ferons une conclusion.

Chapitre 2

Présentation du package R insurancerating

2.1 Description et objectifs du package insurancerating

Ce package propose des techniques analytiques pouvant être utilisées en tarification assurantielle et particulièrement en assurance IARD, aidant les actuaires à implémenter les GLMs tout en tenant compte des étapes pertinentes pour construire la prime pure sur des jeux de données. D'après l'auteur, le but principal du package insurancerating est de classer les variables continues de telle sorte qu'il en résulte des facteurs de risque catégoriels qui capturent l'effet de la covariable sur la variable réponse de manière précise, tout en étant facile à utiliser dans un modèle linéaire généralisé (GLM). En d'autres termes, le package permet de transformer tour à tour chaque variable continue et spatiale en variable catégorielle (dont chaque niveau de la variable représente un sous groupe d'individus similaires), afin d'intégrer toutes les variables catégorielles obtenues dans un modèle linéaire généralisé, qui est le modèle le plus couramment utilisé par les actuaires en pratique de part la simplicité d'utilisation et l'interprétabilité des résultats qu'elle offre. D'entrée de jeu, on s'interroge sur la pertinence d'une segmentation basée sur les estimations des effets individuels des covariables, car plus on a de variables significatives dans un modèle, plus il est précis (Cours de LACTU2110).
Cependant, le package se démarque de certains outils de R dans le sens où, certaines de ses fonctions permettent de tenir compte automatiquement des poids des observations, matérialisés par la pondération de la somme des expositions lors de l'ajustement du modèle GAM dans le calcul de la prime pure, la pondération par le nombre de sinistres dans le cas d'un modèle de sévérité par exemple. Le package permet de catégoriser automatiquement une covariable en se servant du résultat obtenu d'un modèle de régression. Bien que disposant de plusieurs fonctions qui n'entrent pas directement en relation avec l'objectif du package, insurancerating s'avère aussi utile pour faciliter des opérations statistiques multivariées. Notre recherche étant centrée sur la segmentation, nous évaluerons la capacité du package insurancerating dans le domaine de la segmentation .

2.2 Fonctions principales du package et procédés

L'approche du package est axée sur les données, c'est-à-dire qu'il travaille à modifier de façon optimale la structure de données du portefeuille en fonction d'une variable réponse de telle sorte qu'il n'en résulte que des variables catégorielles. Son but étant d'intégrer de façon optimale les facteurs de risques continus dans la tarification, tout en restant dans un contexte en accord avec les pratiques des compagnies d'assurance, des assurés et aussi du régulateur. En pratique, de nombreuses variables explicatives sont catégorielles et s'utilisent efficacement dans les modèles GLMs après codage au moyen de variables binaires (pour plus de détails voir cours de LACTU2110). Cependant, certaines variables telles que la puissance du moteur, l'âge du conducteur...etc sont continues. Utiliser directement une variable continue dans un modèle linéaire revient à supposer un effet linéaire de cette variable sur l'échelle du score, ce qui n'est pas toujours vrai, par exemple dans les modèles log-linéaires, cela signifierait que la moyenne est contrainte à varier exponentiellement avec la covariable, un tel comportement exponentiel monotone peut ne pas être pris en charge par les données. En plus de ça, traiter une covariable continue comme une variable explicative catégorielle peut induire un grand nombre de paramètres de régression supplémentaires, où il pourrait ne pas avoir reconnaissance d'une possible variation régulière de la moyenne en cette covariable. C'est pour ces nombreuses raisons que le package insurancerating, construit tout d'abord des modèles GAMs qui capturent l'effet de chaque variable continue sur la variable réponse, car les modèles additifs généralisés permettent une modélisation flexible que nous expliquerons dans les sections suivantes. Ceci se fait à l'aide de la fonction `fit_gam` fournit par le package. Ici la variable réponse peut être le nombre de sinistre, la sévérité ou la prime pure.

Comme mentionné plus haut, le but est d'obtenir à partir de ces facteurs de risque continus et spatiaux des facteurs de risque catégoriels pour optimiser leur usage dans le modèle GLM.

Le package insurancerating fournit plusieurs fonctions, dans la section suivante nous décrirons tour à tour le procédé des fonctions `fit_gam`, `construct_tariff_classes` et `fisher`, car ces trois fonctions sont celles qui sont en relation directe avec la notion de segmentation. Pour ce qui est des autres fonctions du package, le lecteur pourra se référer à la documentation CRAN du package.

2.2.1 Procédé de la fonction `fit_gam`

La fonction `fit_gam` est utilisée pour ajuster un modèle additif généralisé (GAM) sur une variable continue, afin de capturer l'effet individuel de ce facteur de risque continu sur la variable réponse pour 03 types de modèles différents : le nombre de sinistre, la sévérité de sinistre et la prime pure. Autant il y a de variables continues dans la base de données autant il y aura de modèles GAMs (via la fonction `fit_gam`), car la fonction `fit_gam` ne prend en argument qu'une seule variable continue, on se demande donc si l'effet d'une seule covariable sur le nombre de sinistre est le même que l'effet marginal de cette variable

compte tenue des autres variables pertinentes sur ce nombre de sinistre.
Cette fonction s'utilise de la manière suivante :

```
fit_gam(data, nclaims, x, exposure, amount = NULL, pure_premium = NULL,
model = "frequency", round_x=NULL)
```

Où les différents arguments sont décrits comme suit :

`data:`	Le jeu de données représentant le portefeuille d'assurance.
`nclaims:`	La variable dans le portefeuille représentant le nombre de sinistre enregistré.
`x:`	Variable continue dans le portefeuille pour laquelle on veut construire des classes.
`amount:`	Variable dans le portefeuille représentant le montant total des différents sinistres déclarés par chaque assuré.
`pure_premium:`	Variable dans le portefeuille représentant le montant de la prime pure.
`model:`	Modèle choisi pour l'ajustement du modèle additif généralisé, soit **frequency**, **severity** ou **burning**.

TABLE 2.2.1: Arguments en entrée de la fonction `fit_gam`

Lorsque le modèle choisi est **frequency**, alors la fréquence de sinistre est déterminée et utilise un modèle de régression de Poisson de type GAM. Le logarithme de l'exposition est mis en **offset** de telle sorte que le nombre de sinistre soit proportionnel à la période d'exposition. Les modèles **severity** et **burning** utilisent un modèle de régression Log-normale de type GAM pour ajuster respectivement le coût moyen des sinistres et le montant de la prime pure.

2.2.1.1 Modèle : `frequency`

Nous remarquons que cette fonction n'est pas flexible dans le choix de la distribution, car elle impose une loi pour la distribution liée à chaque type de modèle, et pourtant, dans certains cas les données sont distribuées différemment.
En effet la distribution de Poisson joue un rôle de premier plan dans la modélisation des données de dénombrement discret, principalement en raison de son adéquation descriptive en tant que modèle lorsque seul le hasard est présent et que la population sous-jacente est homogène. Malheureusement, ce n'est pas une hypothèse réaliste à faire pour modéliser

de nombreux ensembles de données d'assurance réelles. Les mélanges de Poisson (la loi Binomiale-Négative par exemple est un mélange de lois Gamma-Poisson et est celle qu'utilisent le plus souvent les actuaires en pratique) sont des équivalents bien connus de la distribution de Poisson simple pour la description des populations non homogènes. Les populations constituées d'un nombre fini de sous-populations homogènes présentent un intérêt particulier. Dans ces cas, la distribution de probabilité de la population peut être considérée comme un mélange fini de distributions de Poisson.
Le problème de l'hétérogénéité non observée se pose parce que les différences de comportement au volant entre les individus ne peuvent pas être observées par l'actuaire. L'une des conséquences bien connues de l'hétérogénéité non observée dans l'analyse des données de dénombrement est la surdispersion : C'est-à-dire lorsque la variance empirique de la variable de dénombrement est plus grande que la moyenne. Outre ses implications pour la structure des moments d'ordre inférieur des comptages, l'hétérogénéité non observée a des implications importantes pour la structure de probabilité du modèle de mélange qui s'ensuit. Les phénomènes d'excès de zéros et de queues supérieures lourdes dans la plupart des données d'assurance peuvent être considérés comme une implication d'une hétérogénéité non observée . Il est courant de tenir compte de l'hétérogénéité non observée en superposant une variable aléatoire (appelée effet aléatoire) au paramètre moyen de la distribution de Poisson, ce qui donne un modèle de Poisson mixte. Dans un processus mixte de Poisson, la fréquence annuelle prévue des réclamations devient elle-même aléatoire ([Antonio and Valdez, 2012]).

Modèle de Poisson mixte pour le nombre de sinistres : Ici la fréquence moyenne λ de la distribution de Poisson est multipliée par un effet aléatoire positif θ. La fréquence variera au sein du portefeuille en fonction de la variable aléatoire non observable θ.
$E(\theta) = 1$, conditionnellement à θ on a :

$$P(N = k|\Theta = \theta) = p(k|\lambda\theta) = exp(-\lambda\theta)\frac{(\lambda\theta)^k}{k!}, k = 0, 1, ...$$

$$P(N = k) = E(p(k|\lambda\theta)) = \int_0^\infty exp(-\lambda\theta)\frac{(\lambda\theta)^k}{k!}\,\mathrm{d}F_\Theta(\theta)$$

$$E(N) = \int_0^\infty \sum_{k=0}^{\infty} k\ exp(-\lambda\theta)\frac{(\lambda\theta)^k}{k!}\,\mathrm{d}F_\Theta(\theta) = \lambda E(\Theta) = \lambda$$

On se rend donc simplement compte que la rigidité au niveau du choix de la distribution dans le modèle GAM se présente comme une limite à la fonction `fit_gam`.

La fonction `fit_gam` exécutée, la variable réponse qui est le nombre de sinistre pour le modèle de fréquence, le coût des sinistres pour le modèle de sévérité et la prime pure pour le modèle de prime est agrégée pour chaque niveau de la variable explicative continue (en fonction du modèle), comme le montre cet extrait du code R de la fonction `fit_gam` :

```
df<-aggregate(list(nclaims = df$nclaims, exposure = df$exposure,amount =
df$amount),by = list(x = df$x), FUN = sum, na.rm = TRUE,na.action = NULL)
```

Output du `fit_gam` : Pour un modèle de fréquence, la fréquence de sinistre est calculée pour chaque niveau de la covariable comme le rapport entre le nombre de sinistre agrégé sur l'exposition agrégé. Ensuite un modèle Poisson GAM est ajusté sur le nombre de sinistre agrégé (variable réponse) et la covariable (le vecteur des différents niveaux de la covariable). L'objet obtenu à la suite de l'implémentation de la méthode GAM tel que décrite plus haut est ensuite pris comme argument dans la fonction `plot` du package `mgcv` comme le montre la ligne de code suivante :

```
gam0<-plot(gam_x,se=TRUE,rug=FALSE,shift=mean(predict(gam_x)),trans =
  exp)

```

Cette étape permet de construire les éléments utiles à la représentation graphique de l'objet `fit_gam`, ces éléments sont contenus dans un dataframe de 100 observations (valeur par défaut du nombre de point utilisé pour la représentation graphique) classées du minimum au maximum avec un écart constant entre les valeurs, les prédictions pour chaque niveau de la covariables (100 niveaux), ainsi que les bornes supérieures et inférieures à 95% de confiance. La fonction `predict.gam` est utilisée pour calculer les prédictions pour chaque niveau de la covariable, il est important de préciser qu'ici il s'agit de la covariable avec ses niveaux de base et non celle à 100 observations. Enfin un dataframe est créé contenant différents niveaux de la covariable et la prédiction associée.
La fonction `fit_gam` au bout de ce long processus retourne une liste des cinq éléments suivants :

`data:`	Le jeu de données représentant le portefeuille d'assurance.
`Prediction:`	Dataframe résultant de l'application de la fonction `plot` sur l'objet `fit_gam`.
`x:`	Nom de la covariable.
`model:`	Le modèle.
`data:`	Data contenant pour chaque niveau de la covariable, la prédiction, le nombre de sinsitre et l'exposition agrégé par niveau, ainsi que la fréquence pour chaque niveau..
`X_obs:`	différentes observations de la variable continue. -

TABLE 2.2.2: Valeurs résultantes de la fonction `fit_gam`

Le procédé décrit ici est relativement identique qu'il s'agisse d'un modèle de coût total des sinistres ou d'un modèle de prime pure.

2.2.1.2 Modèle :`severity`

Précisons dans ce cas que le nombre de sinistre, l'exposition et le montant des sinistres sont agrégés par rapport aux différents niveaux de la covariable, et on calcule le coût moyen de sinistre comme le rapport entre le montant agrégé des sinistres et le nombre de sinistre. Etant donné que la distribution utilisée est une Log-Normale, comme le montre le bout de code R ci-dessus, on ajuste le logarithme du coût moyen agrégé des sinistres par rapport à la covariable concernée. Ici également, nous remarquons que la fonction `fit_gam` ne prévoit pas des possibilités d'adaptation du modèle en fonction des données. En effet, celui-ci utilise à chaque fois un modèle de régression Log-Normale, sans possibilité d'adapter le modèle à la structure des données. Ceci se matérialise dans la fonction `fit_gam` par l'extrait de code R suivant :

```
gam_x <- mgcv::gam(log(avg_claimsize)~s(x), data = df, family = gaussian,
    weights = nclaims)
```

Le but d'un modèle de sévérité est de produire un modèle multiplicatif à fin d'estimer le coût moyen des sinistres μ_i. Soient :
-$\overline{\mathcal{C}}_i$ le coût moyen de sinistre de l'assuré i
-$\mathcal{C}_{ik}$ le coût du $k^{ième}$ sinistre de l'assuré i
-$\mathcal{S}_i$ le coût total des sinistres pour l'assuré i

Si l'assuré i a à son actif ω_i sinistre, alors

$$\overline{\mathcal{C}}_i = \frac{\mathcal{S}_i}{\omega_i} = \frac{\sum_{k=1}^{\omega_i} \mathcal{C}_{ik}}{\omega_i}$$

En supposant pour l'assuré i que les coûts $\mathcal{C}_{i1}, ..., \mathcal{C}_{i\omega_i}$ soient $i.i.d.$ tout comme les coûts $\mathcal{C}_i$ alors

$$\mathbb{V}[\overline{\mathcal{C}}_i] = \frac{\mathbb{V}[\mathcal{C}_i]}{\omega_i} \Rightarrow \mathbb{E}[\overline{\mathcal{C}}_i] = \mathbb{E}[\mathcal{C}_i]$$

Dans un modèle Log-Normale (les queues de distribution plus épaisses permettent de mieux gérer les sinistres importants), comme celui utilisé par ce package (voir bout de code ci-dessus), on suppose que logarithme népérien du coût moyen des sinistres suit une loi Normale dont la moyenne est le prédicteur $\beta^t x_i$ et la variance σ^2, c'est-à-dire :

$$ln(\overline{\mathcal{C}}_i) \sim \mathcal{N}or(\beta^t x_i, \sigma^2).$$

où β^t est la transposée du vecteur β des coefficients et x le vecteur des variables explicatives du modèle. Alors l'équation de vraisemblance permettant d'estimer les coefficients contenus dans le vecteur $\hat{\beta}$ est donné par :

$$\sum_{i \ t.p. \ n_i>0} \{ln(\overline{\mathcal{C}}_i) - \omega_i \beta^t x_i\} x_i = 0 \quad \Leftrightarrow \quad \sum_{i \ t.p. \ n_i>0} lcres_i x_i = 0$$

où $lcres_i x_i$ est le résidus d'estimation pour les assurés ayant au moins déclaré un sinistre et est donné par :

$$lcres_i = ln(\overline{\mathcal{C}}_i) - \omega_i \beta^t x_i$$

Cette équation exprime également une relation d'orthogonalité entre les variables explicatives et les résidus d'estimation, de plus cette manière de procéder a pour inconvénient le fait que la variable réponse étant le logarithme des montants de sinistres, alors les conclusions inférentielles se rapporteront aux montants de sinistres ainsi transformés, et il n'est pas toujours aisé de repasser aux données initiales ([Charpentier and Denuit, 2005]), car la moyenne ne suit pas formellement une distribution Log-Normale, contrairement à une distribution Gamma qui est plus appropriée dans le sens où est permet de pallier au problème rencontré avec la loi Log-Normale.

Dans le cas où le coût moyen de sinistre de l'assuré i suit une loi Gamma :

$$\overline{\mathcal{C}}_i \sim \mathcal{G}(\mu_i, \omega_i \alpha), \quad alors \quad \mathbb{V}[\overline{\mathcal{C}}_i] = \frac{\mu_i^2}{\omega_i \alpha} \quad et \quad \ln(\mu_i) = \beta^t x_i$$

L'espérance des coûts moyens des sinistres est Log-Normale. Rappelons que dans le cas d'une telle loi (Gamma), on rencontre souvent des problèmes d'optimisation numérique pour la méthode du maximum de vraisemblance et aussi elle ne permet pas de représenter correctement les valeurs extrêmes. Vu que le package ne donne pas la possibilité de choisir une distribution, dans le cas d'un modèle de sévérité, c'est la distribution Log-Normale qui est utilisée par défaut.
On peut également modéliser la sévérité de sinistre en utilisant des distributions telles que la loi Exponentielle, la loi de Pareto...etc.

2.2.1.3 Modèle :`burning`

Dans le cas d'un modèle pour le calcul de la prime pure (`burning cost`), nous remarquons ici aussi que la fonction `fit_gam` utilise uniquement et sans laisser de choix une distribution Log-Normale et pourtant nous savons qu'en pratique il existe des problèmes souvent rencontrés avec cette loi statistique lorsqu'il s'agit d'établir la prime pure.
En effet, soit Y le coût (total) des sinistres sur un an pour chacune des polices du portefeuille et C_k le coût du $k^{ième}$ sinistre. Un très grand nombre de polices n'auront pas de sinistre, la variable Y sera alors nulle pour la plupart des observations. Une loi Gamma ou Gaussienne ne permet pas de modéliser ce genre de comportement ([Denuit et al., 2004]). La loi de Tweedie permet de prendre en compte ce genre de comportement, en rajoutant une mesure de Dirac en 0 à une loi de probabilité de support $\mathbb{R}^+$. La loi de Y est alors une

loi Poisson composée :

$$Y = \sum_{k=1}^{N} C_k$$

avec $N \sim \mathcal{P}(\lambda)$ et $C_k \sim \mathcal{G}(\alpha, \tau)$, toutes variables étant indépendantes, on a que :

$$\mathbb{E}[Y] = \mathbb{E}[N]\mathbb{E}[C_1] = \lambda\tfrac{\alpha}{\tau} \quad \text{et} \quad \mathbb{V}[Y] = \mathbb{E}[N]\mathbb{E}[C_1^2] = \lambda\left(\tfrac{\alpha}{\tau^2} + (\tfrac{\alpha}{\tau})^2\right)$$

Pour un assuré i, la prime pure définie comme l'espérance du coût de sinistre pour l'assuré i, est donné par :

$$prime \;\; pure = \mu_i^{(N)}\mu_i^{(C)}.$$

avec $\mu_i^{(N)} = \mathbb{E}[N_i]$ et $\mu_i^{(C)} = \mathbb{E}[C_i] = \mathbb{E}[\overline{\mathcal{C}}_i]$

2.2.1.4 `fit_gam` et `mgcv::gam` : similitudes ?

Il y a lieu ici de préciser que la méthode d'analyse statistique qu'utilise fonction `fit_gam` du package `insurancerating` est en grande partie calquée sur celle de la fonction `GAM` du package `mgcv` comme nous le montre le bout de code qui suit :

```
gam_x <- mgcv::gam(nclaims~ s(x), data = df, family = poisson(), offset
  = log(exposure))

```

Nous voyons bien que le modèle GAM ajusté ici est celui du package `mgcv` compte tenue des opérations préalables faites sur les données précisées plus haut.

L'implémentation `mgcv` de `gam` représente les fonctions de lissage à l'aide de splines de régression pénalisées qui sont par défaut des splines de régression à plaques minces, car dans la fonction `s()` du bout de code ci-dessus, aucune base de lissage pénalisée n'est précisée, d'où l'usage de la base de lissage par défaut (`bs="tp"`), elle utilise également par défaut des fonctions de base pour ces splines conçues pour être optimales, compte tenu des fonctions de base numériques utilisées. Les termes lisses peuvent être des fonctions d'un nombre quelconque de covariables et l'utilisateur a un certain contrôle sur la façon dont la régularité des fonctions est mesurée. Les termes lissés sont représentés à l'aide de splines de régression pénalisées (ou de lisseurs similaires) avec des paramètres de lissage sélectionnés par des critères tels que : GCV(General Cross Validation), $UBRE$(Un-Biased Risk Estimator), AIC(Akaike Information Criterion), $REML$(Restricted Maximum Likelihood) ou par des splines de régression avec des degrés de liberté fixes (le mélange des deux est autorisé).

2.2.2 Procédé de la fonction `construct_tariff_classes`

Une fois le modèle ajusté grâce à la fonction `fit_gam`, la construction des classes de risque s'effectue grâce à la fonction `construct_tariff_classes`. Celle-ci s'utilise sur R

comme suit :

```
construct_tariff_classes(object,alpha=0,niterations=10000,ntrees =200,
   seed=1)
```

Elle prend en argument :

`object:`	le jeu de données agrégé en fonction des niveaux de la variable continue.
`alpha:`	un paramètre de complexité, utilisé pour contrôler le nombre de classes, une grande valeur de `alpha` aboutira à un nombre restreint de classes tarifaires, la valeur par défaut est 0.
`niterations:`	le nombre d'itération après lequel l'exécution se termine si elle ne converge pas.
`ntrees:`	représente le nombre d'arbre dans la population.
`seed:`	permet d'initialiser le générateur de nombre aléatoire pour qu'il puisse être reproductible.
`X_obs:`	différentes observations de la variable continue.

TABLE 2.2.3: Arguments en entrée de la fonction `construct_tariff_classes`

Cette fonction exécute une succession de tâches pour construire les classes. Tout d'abord , elle reçoit en entrée l'objet obtenu à partir de la fonction `fit_gam` (contenant les éléments suivants : Prediction, x, model, data, X_obs), au cours de l'exécution, le quatrième élément est attribué à un dataframe nommé `new`. Ces données sont constituées de 4 colonnes ; la première contient les différents niveaux de la variable continue, la deuxième contient les prédictions obtenues après l'ajustement du modèle GAM, la troisième contient soit le nombre de sinistre agrégé soit le coût des sinistres agrégé soit la prime pure agrégé selon que le modèle choisit pour l'ajustement dans la fonction `fit_gam` était respectivement `frequency`, `severity` ou `burning` ; ensuite, la colonne représentant les différents niveaux de la variable continue est à son tour mise dans un vecteur appelé `counting`. Maintenant, peut se faire l'arborescence des arbres :
C'est au travers de l'élément nommé `split_x` que la fonction principale pour la construction de classe, la fonction `evtree` du package `evtree` est utilisée comme le montre le bout de code R ci-dessus. Pour construire les classes, `evtree` utilise les prédictions (prédiction des données agrégées) comme variable réponse et les ajuste à la variable continue.

```
split_x <- tryCatch({
        tree_x <- evtree::evtree(pred ~ x, data = new, control = evtree::
   evtree.control(alpha = alpha,
```

```
            ntrees = ntrees, niterations = niterations, seed = seed))
        split_obtained <- get_splits(tree_x)
        unique(floor(split_obtained))
    }, error = function(e) {
        NULL
    })
```

Ce qui en résulte c'est un arbre de classification globalement optimal. En fait le package `evtree` a été mis sur pied car les méthodes d'arbre de classification et de régression couramment utilisées, tel que l'algorithme CART, ne sont optimaux que localement. En effet, les divisions sont choisies pour maximiser l'homogénéité à l'étape suivante uniquement. `evtree` utilise des méthodes d'optimisations globales comme les algorithmes évolutifs. C'est une toute autre façon de rechercher dans l'univers des arbres.

2.2.2.1 Valeurs résultantes

Cette fonction retourne comme résultat une liste contenant les éléments suivants :

`prediction:`	Le dataframe contenant les valeurs prédites.
`x:`	La variable continue pour laquelle les classes sont construites.
`model:`	Qui peut être soit `frequency`,`severity` et `burning` pour respectivement un modèle de fréquence, la sévérité ou de la prime pure respectivement.
`splits:`	Vecteur contenant les limites des classes tarifaires construites.
`tariff_classes:`	Vecteur contenant les classes auxquelles chaque éléments de la variable `x` appartient.

TABLE 2.2.4: Valeurs résultantes de la fonction `construct_tariff_classes`

2.2.3 Procédé de la fonction `fisher`

Basée sur l'algorithme des ruptures naturelles de Fisher, cette fonction permet de construire des classes pour les facteurs de risque spatiaux numériques continus. Cette fonction utilise l'algorithme developpé par [Fisher, 1958] et discuté par [Slocum et al., 2008] comme l'algorithme de Fisher-Jenks.

Cette fonction s'utilise de la manière suivante :

```
fisher(vec, n = 7, diglab = 2)
```

Ces différents arguments sont décrits comme suit :
`-vec` : Pour la variable numérique continue.
`-n` : Le nombre de classe choisi, 7 est la valeur par défaut.
`-diaglab` : Le nombre de chiffre utilisés dans le formatage des numéros de rupture.
Pendant l'exécution de la fonction `fisher`, la fonction `classIntervals` du package `ClassInt` est appliquée sur la variable spatiale avec la méthode "Fisher" pour obtenir les points de rupture. Ensuite la fonction `cut` est appliquée sur la variable continue prenant les points de rupture obtenu préalablement dans l'argument `breaks`, déterminant ainsi comment est-ce que la covariable sera découpée. Elle retourne un vecteur contenant les différentes classes construites.
Après avoir expliqué le procédé computationnel des différentes fonctions du package ci-dessus, nous présenterons dans la section suivante les différentes théories mathématiques au coeur de celles-ci.

2.3 Fondements Théoriques du package Insurancerating

2.3.1 Modèles additifs généralisés

L'hypothèse de linéarité faite en générale sur la variable réponse est discutable dans certains cas, car les effets non linéaires des facteurs de risque continus tels que l'âge du conducteur doivent être inclus dans l'échelle du score pour capturer son effet. Les covariables, pour efficacement entrer dans les GLMs doivent être transformées optimalement pour qu'on puisse analyser de façon précise leurs impact sur le score. Mais la littérature ne nous fournit pas clairement la façon dont ces covariables doivent être transformées avant d'être utilisées dans les GLMs. Les modèles additifs généralisés interviennent dont pour traiter les facteurs de risque continus de manière flexible. D'après [Denuit et al., 2019], l'effet de l'âge, de la puissance du véhicule ou de la somme assurée du preneur d'assurance peuvent être modélisés au moyen des modèles GAMs, modèle permettant aussi d'analyser les variations des facteurs de risque spatiaux tout en tenant compte des interactions. La particularité des modèles GAMs réside dans le fait qu'ils sont capables de maximiser la qualité de la prédiction d'une variable dépendante à partir de diverses distributions, en estimant des fonctions non-spécifiques (non-paramétriques) des variables explicatives qui sont "liées" à la variable dépendante par une fonction de liaison. Plus précisément, au lieu d'une forme plus ou moins complexe de fonctions paramétriques, [Hastie and Tibshirani, 1990] évoquent divers lissages des nuages de points que nous pouvons appliquer aux valeurs de la variable explicative, avec l'objectif de maximiser la qualité de la prédiction des valeurs de la variable réponse.

Pour résumer, au lieu d'estimer des paramètres simples (comme les poids dans une régression multiple), nous recherchons, dans les modèles additifs généralisés, une fonction généraliste non-spécifiée (non-paramétrique) qui permet de lier la variable à prédire (trans-

formée) aux variables prédictives.
Si Y est la composante aléatoire, alors elle doit appartenir à la famille exponentielle, c'est-à-dire que sa probabilité de masse ou sa fonction de densité est représentée comme suit :

$$f(y|\theta,\phi) = \exp(\frac{y.\theta - b(\theta)}{\phi}).c(y,\phi)$$

Où :
-θ : Le paramètre canonique
-ϕ : Le paramètre de dispersion
-$b(.)$: Une fonction monotone convexe de θ
-$c(.)$: Fonction de normalisation positive
Si par exemple on suppose que Y suit un distribution de Poisson on a :

$$f(y|\mu) = \exp(-\mu)\frac{\mu^y}{y!} = \exp(y\ln\mu - \mu)(y!)^{-1}$$

-θ=$\ln\mu$
$-\mathrm{b}(\theta)$=$\exp(\theta)$ =μ
-ϕ=1
-$c(y,\phi)$=$(y!)^{-1}$

Pour différentes distributions de la famille exponentielle linéaire, les différents paramètres sont résumés dans le tableau ci-dessous :

Distribution	θ	$b(\theta)$	Φ	$E[Y]$	$V(\mu)$
$Bin(1,\mu)$	$ln\frac{\mu}{1-\mu}$	$ln(1+exp(\theta))$	1	μ	$\mu(1-\mu)$
$Poi(\mu)$	$ln\mu$	$exp(\theta)$	1	μ	μ
$Nor(\mu,\sigma^2)$	μ	$\frac{\theta^2}{2}$	σ^2	μ	1
$Gam(\mu,\alpha)$	$-\frac{1}{\mu}$	$-ln(-\theta)$	$\frac{1}{\alpha}$	μ	μ^2
$InvGauss(\mu,\sigma^2)$	$-\frac{1}{2\mu^2}$	$-\sqrt{-2\theta}$	σ^2	μ	μ^3
$Twe(\mu,\phi)$	$\frac{\mu^{1-q}}{1-q}$	$\frac{((1-q)\theta)^{\frac{2-q}{1-q}}}{2-q}$	ϕ	μ	$\mu^q, 1<q<2$

TABLE 2.3.1: Paramètres de quelques distributions de la famille Exponentielle

Dans les modèles GAMs, la composante déterministe qui est le score se calcule comme suit :

$$score_i = \beta_0 + \sum_{j=1}^{k}\beta_j x_{ij} + \sum_{j=k+1}^{n} f_j(x_{ij}) = g(\mu_i)$$

Où k est le nombre de variable catégorielle, m le nombre de variable continue et on a k+m=n, les fonctions $f_j()$, j=k+1,...,n supposées régulières traduisent l'influence des facteurs de risque continues x_{k+1},...,n sur la réponse y, elles sont non spécifiées et sont estimées à partir des données, g est la fonction de lien entre le score et l'espérance de la prédiction de la réponse.
La prédiction μ_i qui est en effet l'espérance conditionnée par les informations de l'assuré est donnée par :

$$score_i = g(\mu_i) \Leftrightarrow \mu_i = g^{-1}(score_i)$$

La fonction g est une fonction strictement monotone et différentiable, g dépend du modèle utilisé dans la régression, pour plus de détails, voir [Denuit et al., 2019]. Dans les modèles additifs généralisés, aucune hypothèse n'est faite sur les fonctions $f_j()$ si ce n'est qu'elles sont continues et différentiables. Les modèles GAMs sont identiques aux modèles GLM si toutes les fonctions f_j sont affines.
Pour estimer la fonction f en un point x, les observations sont pondérées de telle sorte que de grands poids sont assignés à des valeurs proches de x et des petits poids aux valeurs éloignées de x.

Considérons par exemple un modèle additif à un seul facteur pour faire simple, où y_i est la réponse et x_i la variable explicative toutes deux continues, le but est d'estimer la fonction f en utilisant des techniques de lissage linéaire :

$$y_i = f(x_i) + \epsilon_i \ , \quad \epsilon_i \sim \mathcal{N}(0, \sigma^2)$$

En général, la fonction de lissage f est estimée en utilisant la vraisemblance locale ou la décomposition en spline. La méthode Loess et les fonctions de splines cubiques sont très souvent utilisées pour estimer la fonction f. Etant donné que la fonction `fit_gam` utilise sans possiblité de modification les splines de lissage à plaques minces, nous les décrirons dans la sous-section suivante.

2.3.2 Thin-plate smoothing splines

Les splines de lissage à plaques minces (Thin-plate smoothing splines en Anglais) developpé par [Duchon, 1977] sont une généralisation des splines cubiques (voir [Denuit et al., 2019]) à plusieurs variables explicatives. Plus précisement les splines de lissage à plaque minces permettent de modéliser les interactions entre les variables continues, tout comme les splines cubiques, elles essayent d'approximer les valeurs observées autant que possible. Cependant, la fonction estimée est définie en terme de deux ou plusieurs variables explicatives, tel que le spline de lissage corresponde à une surface lissée.
En considérant uniquement des interactions entre deux variables, d'après [Fahrmeir et al., 2013], une spline de lissage à plaque mince peut être définie par :

$$f(x_1, x_2) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \sum_{K}^{k=1} b_k B_k(x_1, x_2)$$

Avec B_k la fonction de base radiale telle que $B_k(x_1, x_2) = \|x - x_k\|^2 \log(\|x - x_k\|)$, centrée autour de $x_k = (x_{1k}, x_{2k})$ et évaluée en $x = (x_1, x_2)$. Les termes $\beta_1 x_1$ et $\beta_2 x_2$ dénotent les effets linéaires des variables x_1 et x_2. Ces splines sont similaires aux splines cubiques, la solution de la version bivariée de la fonction des moindres carrés pénalisés est donnée par :

$$\sum_{i=1}^{n}(y_i - f(x_i))^2 + \lambda \iint \left[\left(\frac{\partial^2 f(x)}{\partial x_1^2} \right)^2 + 2 \left(\frac{\partial^2 f(x)}{\partial x_1 x_2} \right)^2 + \left(\frac{\partial^2 f(x)}{\partial x_2^2} \right)^2 \right] dx_1 dx_2$$

où le paramètre de lissage λ est détermine de manière optimale par validation croisée.

2.3.3 Algorithme des arbres évolutionnaires : la fonction `evtree`

2.3.3.1 `evtree` algorithm

Le package `evtree`, implémente un algorithme évolutionnaire pour l'apprentissage des arbres de classification et de régression globalement optimaux dans R. Le cadre général des algorithmes évolutionnaires a émergé de différents représentants. [Holland, 1992] appelle sa méthode "algorithmes génétiques" [Rechenberg et al., 1973] a inventé des stratégies d'évolution et [Fogel et al., 1966] ont introduit la programmation évolutionniste. Plus récemment, [Koza, 1992] a introduit un quatrième courant et l'a appelé "programmation génétique". Les quatre approches ne diffèrent que par les détails techniques, par exemple l'encodage des solutions individuelles, mais suivent le même schéma général ([Eiben et al., 2003]).

2.3.3.2 Qu'est ce qu'un algorithme évolutionnaire ?

Comme le suggère l'histoire du domaine, il existe de nombreuses variantes d'algorithmes évolutionnaires. L'idée sous-jacente commune à toutes ces techniques est la même : étant donné une population d'individus dans un environnement aux ressources limitées, la compétition pour ces ressources provoque une sélection naturelle (survie du plus apte). Cela entraîne à son tour une augmentation de la condition physique de la population. Étant donné une fonction de qualité à maximiser, nous pouvons créer aléatoirement un ensemble de solutions candidates, c'est-à-dire des éléments du domaine de la fonction. Nous appliquons ensuite la fonction de qualité à ceux-ci en tant que mesure abstraite de la condition physique le plus élevé sera le mieux. Sur la base de ces valeurs de "fitness", certains des meilleurs candidats sont choisis pour semer la prochaine génération. Cela se fait en leur appliquant une recombinaison et/ou une mutation. La recombinaison est un opérateur appliqué à deux ou plusieurs candidats sélectionnés (les soi-disant parents), produisant un ou plusieurs nouveaux candidats (les enfants). La mutation est appliquée à un candidat et entraîne un nouveau candidat. L'exécution des opérations de recombinaison et de mutation sur les parents conduit donc à la création d'un ensemble de nouveaux candidats (la progéniture). Ceux-ci font évoluer leur condition physique, puis rivalisent en fonction de leur condition physique (et éventuellement de leur âge) avec les anciens pour une place dans la génération suivante. Ce processus peut être itéré jusqu'à ce qu'un candidat de qualité

suffisante (une solution) soit trouvé ou qu'une limite de calcul préalablement définie soit atteinte.
Il y a deux forces principales qui forment la base des systèmes évolutifs :

• Les opérateurs de variation (recombinaison et mutation) créent la diversité nécessaire au sein de la population, et facilitent ainsi la nouveauté.

• La sélection agit comme une force augmentant la qualité moyenne des solutions dans la population.

L'application combinée de la variation et de la sélection conduit généralement à l'amélioration des valeurs de fitness dans des populations consécutives. Il est facile de voir ce processus comme si l'évolution optimisait (ou au moins «approximait») la fonction de fitness, en se rapprochant de plus en plus des valeurs optimales au fil du temps. Un autre point de vue est que l'évolution peut être considérée comme un processus d'adaptation. Dans cette perspective, la fitness n'est pas vue comme une fonction objectif à optimiser, mais comme l'expression d'exigences environnementales. Une meilleure adéquation avec ces exigences implique une viabilité accrue, qui se traduit par un plus grand nombre de descendants. Le processus évolutif aboutit à une population de mieux en mieux adaptée à l'environnement. Il convient de noter que de nombreux composants d'un tel processus évolutif sont stochastiques. Par exemple, lors de la sélection, les meilleurs individus ne sont pas choisis de manière déterministe, et généralement même les individus faibles ont une certaine chance de devenir un parent ou de survivre. Au cours du processus de recombinaison, le choix des pièces des parents à recombiner est fait au hasard. De même pour la mutation, le choix des pièces à changer au sein d'une solution candidate, et des nouvelles pièces pour les remplacer, est faite de manière aléatoire.

Pour certains problèmes de modélisation, il avait été constaté que les méthodes de partitionnement récursifs telles que les CART du fait de leur optimalité locale ne sont pas appropriées, et donc une recherche d'une optimalité globale sur l'espace des paramètres de l'arbre a été pensée pour espérer obtenir des modèles beaucoup plus précis. Le défis ici est que l'espace de recherche est généralement énorme et donc il a souvent été compliqué pour les scientifiques d'implémenter un algorithme de recherche dans cet espace. Heureusement les méthodes d'optimisation stochastiques comme les arbres évolutifs sont une possible solution à ce défi.

Les arbres de classifications et de régressions visent à modéliser une variable de réponse Y par un vecteur de P variables explicatives $X = (X_1, ..., X_P)$ où pour les arbres de classification Y est qualitatif et pour les arbres de régression Y est quantitatif. Les méthodes d'arbres partitionnent généralement d'abord l'espace d'entrée X en un ensemble de M régions $R_m (m = 1, ..., M)$ puis construisent un modèle (généralement simple) dans chaque région.
Dans ce qui suit, un modèle d'arbre binaire à M nœuds terminaux (qui a par conséquent M-1 divisions internes) est désigné par :

$$\theta = (\vartheta_1, s_1, ..., \vartheta_{M-1}, s_{M-1})$$

où $\vartheta_r \in \{1, ..., P\}$ sont les variables de fractionnement et les s_r sont les règles de

fractionnement associées pour les nœuds internes $r \in \{1, ..., M-1\}$. Ainsi, le produit de toutes ces combinaisons forme tous les éléments θ potentiels à partir de Θ_M, l'espace des arbres imaginables à M nœuds terminaux. L'espace global des paramètres est alors $\Theta = \bigcup_{M=1}^{M_{max}} \Theta_M$ (ce qui en pratique est souvent réduit en excluant des éléments entraînant des sous-échantillons trop petits).

Algorithme :

1. Initialisez la population (population des solutions candidates).
2. Évaluez chaque individu (solution candidate).
3. Tant que (la condition de résiliation n'est pas satisfaite), faites :
    3a. Sélectionnez les parents.
    3b. Modifier les individus sélectionnés via des opérateurs de variation.
    3c. Évaluer les nouvelles solutions.
    3d. Sélectionnez des survivants pour la prochaine génération.

Initialisation : Ici, la population des solutions candidates est initialisée avec une règle de fractionnement valide, générée de manière aléatoire dans le nœud racine. ϑ_1 est choisie selon une loi de probabilité uniforme. Si X_{ϑ_1} est numérique ou ordinale avec u valeurs uniques, un point de partage s_1 est sélectionné avec une probabilité uniforme parmi les $u-1$ points de partage possibles de X_{ϑ_1} , et si X_{ϑ_1} est nominal et à c catégories, chaque $k = 1, ..., c$ a une probabilité de 50% d'être affecté au nœud-fils gauche ou droit. Dans les cas où tous les k sont alloués au même nœud terminal, l'une des c catégories est allouée à l'autre nœud terminal, pour avoir pour effet de garantir que les deux nœuds terminaux ne soient pas vides. Si cette procédure aboutit à une règle de fractionnement non valide, les deux étapes de sélection de variables de fractionnement aléatoire et de sélection de point de partage sont répétées. Avec la définition de $r = 1$ (le nœud racine) et la sélection de ϑ_1, s_1, l'initialisation est terminée et chaque individu de la population d'arbres est de type $\theta_1 = (\vartheta_1, s_1)$.

Sélection des parents : À chaque itération, chaque arbre est sélectionné une fois pour être modifié par l'un des opérateurs de variation.

Opérateurs de variation :

Il existe de multiple forme d'opérateur de mutations, mais l'algorithme "evtree" n'en utilise que quelques types. Quatre types d'opérateurs de mutation et un opérateur de croisement sont utilisés par notre algorithme. Ces opérateurs sont décrits ici :

Le `Split:`
Sélectionne un nœud terminal aléatoire et lui attribue une règle de fractionnement valide générée aléatoirement. En conséquence, le nœud terminal sélectionné devient un nœud interne et deux nouveaux nœuds terminaux sont générés. Dans les cas où aucune règle de fractionnement valide ne peut être affectée au nœud interne, la recherche d'une règle de fractionnement valide est effectuée sur un autre nœud terminal sélectionné au hasard. Si, après 10 tentatives, aucune règle de fractionnement valide ne peut être trouvée, alors $\theta_i = \theta_{i+1}$.

Le `Prune:`
Choisit un nœud interne aléatoire qui a deux nœuds terminaux comme successeurs et l'échange en un nœud terminal.

`Mutation de règle de division majeure:`
permet de sélectionner un nœud interne aléatoire r et modifie la règle de partage, définie par la variable de partage ϑ_r correspondante, et le point de partage s_r.

`Mutation de règle de division mineure:`
C'est similaire à l'opérateur de `Mutation de règle de division majeure`. Cependant, il ne modifie pas ϑ_r et ne modifie que légèrement le point de partage s_r.

`Crossover`
Des échanges croisés, sélectionnés au hasard entre deux arbres.

Fonction d'évaluation

Une fonction d'évaluation appropriée pour les arbres de classification et de régression maximise la précision des modèles sur les données d'apprentissage et minimise la complexité des modèles.
Pour les arbres de régression, la précision est généralement mesurée par l'erreur quadratique moyenne (MSE). Ici, il est à nouveau couplé à une mesure de complexité de type BIC : En utilisant $N.log(MSE)$ comme fonction de perte et $\alpha.4.(M+1).log(N)$ comme partie de la complexité, la formulation générale du problème d'optimisation peut être écrite comme suit :

$$loss(Y, f(X, \theta)) = N.log(MSE(Y, f(X, \theta)))$$
$$complexity(\theta) = \alpha.4.(M+1).log(N)$$

Sélection des survivants et choix final

L'algorithme utilise une approche de peuplement où chaque solution parent est en concurrence avec sa progéniture la plus similaire pour une place dans la population. En utilisant les paramètres par défaut, l'algorithme se termine lorsque la qualité des meilleurs 5% d'arbres se stabilise pour 100 itérations, mais pas avant 1 000 itérations. Si l'exécution ne converge pas, l'algorithme se termine après un nombre d'itérations spécifié par l'utilisateur. Dans les cas où l'algorithme ne converge pas, un message d'avertissement le précise et l'arbre avec la meilleure qualité selon la fonction d'évaluation est choisi.

2.3.4 Algorithme des ruptures naturelles de Fisher

La notion de groupe homogène est centrale dans le processus de tarification des primes de risque en assurance. La précision des estimations qu'obtiendra l'actuaire découle en partie de l'homogénéité au sein des sous-groupes constitués. Il est à la fois important et complexe de savoir comment une population peut être décomposée en sous-groupes qui s'opposent fortement les uns avec les autres, les individus d'un même groupe étant assez semblables. Ici, l'idée c'est d'identifier les assemblages naturels des observations qui sont similaires les uns des autres tout en maximisant la distance entre les autres assemblages. [Fisher, 1958] a développé un algorithme de regroupement qui fait cela avec des données unidimensionnel. On peut y trouver des similitudes avec le "clustering k-means", mais ici il s'agit d'un algorithme plus simple et plus rapide car il ne fonctionne que sur des données unidimensionnel. Cette procédure est utilisée pour grouper un nombre arbitraire d'élément tel que la variance intra-groupe soit minimisée, la mesure de la précision des points estimés dépendant de l'homogénéité dans les strates où les échantillons ont été pris.
Les groupes sont crées tels que chaque observations $s_u^{(i)}$ de la classe i soit la plus proche possible de la moyenne de sa classe $\bar{s}^{(i)}$. Ainsi ces groupes sont crées en minimisant la mesure D représentant la somme quadratique des distances entre les observations $s_u^{(i)}$ et les moyennes respectives des classes $\bar{s}^{(i)}$:

$$D = \sum_{i=1}^{k} \sum_{u=1}^{n_i} (s_u^{(i)} - \bar{s}^{(i)})^2$$

Ici, i parcourt les différents sous-groupes, u parcourt dans chaque sous-groupes les éléments s'y trouvant, k est le nombre de sous-groupes et n_i le nombre d'éléments du sous-groupes i.

Toutefois, comme dans la méthode k-means clustering, on doit spécifier le nombre de clusters. Le package insurancerating prend par défaut le nombre 7 comme étant le nombre de classe et ne propose pas un méthode optimale pour déterminer ce nombre de groupe homogène dans un jeu de donnée.

```
1 fisher(vec, n = 7, diglab = 2)}
```

Ce choix du nombre de classe peut s'avérer discutable car il ne découle pas d'une analyse préalable. La littérature nous propose par exemple la méthode "Elbow" qui permet

de déterminer le nombre optimal de classe dans un jeu de données pour la classification Kmeans peu importe la dimension du jeu de données :
Elle s'appuie sur la notion d'inertie, on définit cette dernière comme étant la somme des distances euclidiennes entre chaque point et son centroïde associé. Évidemment plus on fixe un nombre initial de "clusters" élevés et plus on diminue l'inertie : les points ont plus de chance d'être proche d'un centroïde.
Tout d'abord, on calcule la somme de la distance au carré entre chaque membre d'un cluster et son centre de gravité. Mathématiquement, pour le cluster i constitué de k individus, on a :

$$SSE_i = \sum_{j=1}^{k} (y_{ij} - \bar{y_i})^2$$

Le nombre optimal de cluster est celui pour lequel la valeur de cette distance quadratique chute brusquement, tout en commençant à se stabiliser comme le montre la figure ci-dessous :

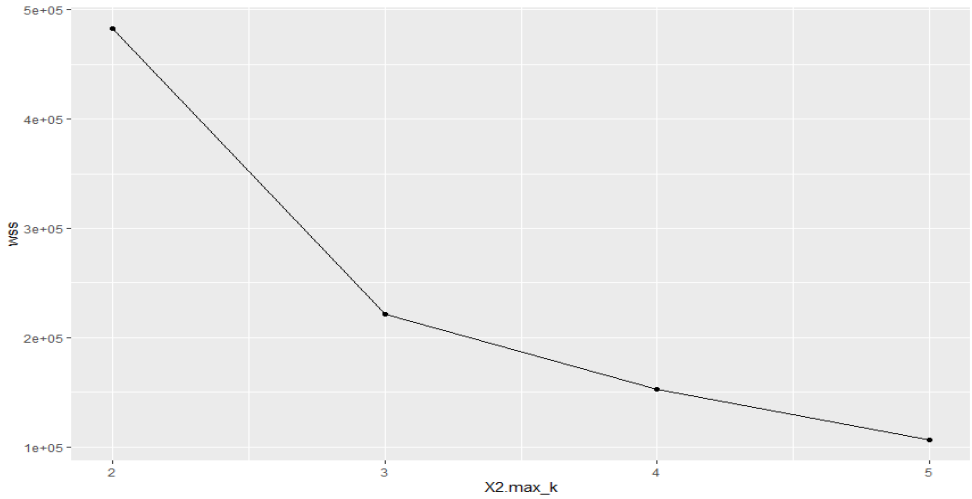


FIGURE 2.3.1: Méthode Elbow

La figure 2.3.1 montre les valeurs de variances intra-groupes en fonction du nombre de classe dans un jeu de données à une seule variable, nous remarquons que le "coude" est formé au point trois, c'est-à-dire que trois est le nombre optimal de classe dans ce jeu de données, connaissant le nombre optimal de classes homogènes, on peut dès à présent appliquer l'algorithme de Fisher en utilisant ce nombre. Toutefois dans notre jeu de données d'application nous ne disposons pas de variables géographiques pour effectuer une telle catégorisation, ne nous permettant pas ainsi d'avoir un avis expérimental sur cette fonction.

2.4 Modèle linéaire généralisé (GLM)

2.4.1 Composantes aléatoire, déterministe et fonction de lien

Présenté pour la première fois par [Nelder and Wedderburn, 1972] et exposé de façon complète par [McCullagh and Nelder, 1989], le GLM est un extension des modèles linéaires simples, permettant de s'affranchir de l'hypothèse de normalité en traitant de manière unifiée les variables réponses appartenant à la famille exponentielle linéaire présentée à la section 2.3.1. La composante aléatoire de ces modèles est identique à celle présentée à la section 2.3.1 dans le cas des modèles additifs généralisés. La composante déterministe et la fonction de lien sont les mêmes compte tenue des fonctions de lissage qui sont linéaires.

2.4.2 Qualité de l'ajustement

2.4.2.1 AIC et BIC

Il existe plusieurs critères permettant de mesurer la qualité de l'ajustement d'un modèle (GLM, GAM,..etc), le critère d'information d'Akaike (AIC) et le critère d'information bayésien (BIC) permettent de pénaliser les modèles en fonction du nombre de paramètres afin de satisfaire le critère de parcimonie. On choisit alors le modèle avec le critère d'information le plus faible. Ces critères sont donnés par :

$$AIC = 2k - 2\log(L)$$

$$BIC = k.log(n) - 2\log(L)$$

où k est le nombre de paramètres dans le modèle et L est la vraisemblance maximisée et n le nombre d'observations. Mais dans les cas des données d'assurance, c'est généralement le critère de déviance communément appelé déviance de poisson qui est utilisé pour mesurer la qualité d'ajustement d'un modèle.

2.4.2.2 Déviance de Poisson

Pour des données d'assurance, le critère de la déviance est meilleur que l'erreur quadratique moyen (MSE), car le MSE de part sa formule fait une hypothèse de normalité alors que les données suivent généralement une distribution de Poisson, du moins dans le cas du nombre de sinistre. Ici on définit la qualité d'ajustement d'un modèle en prenant comme référence le modèle comptant autant de paramètres que d'observations, c'est-à-dire le modèle saturé, le but est de comparer le rapprochement du modèle saturé au modèle estimé qui est caractérisé par $\mu_i = y_i$, pour $i = 1, 2, ..., n.$, le meilleur modèle ajusté sera celui qui est le plus proche du modèle saturé. Ce rapprochement est donc mesuré par la notion de déviance.

Posons $L(y|y)$ et $L(\hat{\mu}|y)$ respectivement la vraisemblance du modèle saturé et celle du modèle ajusté, un modèle ajusté décrit bien les observations si ces deux grandeurs sont proches, c'est-à-dire $L(y|y) \approx L(\hat{\mu}|y)$. La statistique correspondante est la déviance d'échelle D^* définie comme le ratio de test de vraisemblance entre le modèle estimé et le modèle saturé :

$$D^* = D^*(y|\hat{\mu}) = 2 \ln \frac{L(y|y)}{L(\hat{\mu}|y)} \geq 0$$

Le concept de déviance marche au niveau du groupe, ce qui signifie que la $i^{ème}$ observation est l'observation de la $ième$ classe de risque (voir cours LACTU 2110). Un D^* grand signifie fie que le modèle considéré s'éloigne du modèle saturé. Formellement, le modèle considéré sera rejeté si

$$D^* > \chi^2_{n-p-1;1-\alpha}$$

où $\chi^2_{n-p-1;1-\alpha}$ est le quantile à $1-\alpha$ de niveau de confiance d'une distribution de Khi-carré avec $n-p-1$ de degré de liberté, en pratique $\alpha = 5\%$.
Prenons par exemple h l'inverse de la fonction $b^{'}$ tel que $\theta_i = h(\mu_i)$ (voir section 2.3.1).

$$D^*(y|\hat{\mu}) = 2[\ln(L(y|y)) - \ln(L(\hat{\mu}|y))] \ = \frac{2}{\phi} \sum_{i=1}^{n} \omega_i \left(y_i h(y_i) - b(h(y_i)) - y_i h(\hat{\mu}_i) + b(h(\hat{\mu}_i))\right)$$

La déviance non échelonnée $D(y|\hat{\mu})$ diminue le paramètre de dispersion et est donnée par :

$$D(y|\hat{\mu}) = \phi D^*(y|\hat{\mu})$$

Dans le cas d'un modèle de Poisson, on a :

$$D(y|\hat{\mu}) = 2 \sum_{i=1}^{n} \omega_i \left(y_i \ln(y_i) - y_i - y_i \ln(\hat{\mu}_i) + \hat{\mu}_i\right)$$

Au besoin dans ce mémoire, tout en apportant des justifications, nous utiliserons l'un des critères susmentionés.

Chapitre 3

Application du package R insurancerating à un portefeuille d'assurance TPL

3.1 Description de la base de données et sélection des variables

La base de données sur laquelle nous appliquerons différentes méthodes de segmentation est celle utilisée dans le cadre du jeu de tarification (pricing game en Anglais) organisé par l'institut français des actuaires le 05 Novembre 2015. C'est l'une des multiples bases de données fournies par le package R CASdatasets mis sur pieds par Christophe Dutang. Cette base de données intitulée "pg15training", assortie d'une base de données d'application nommée "pg15pricing" (qu'on utilisera pour comparer différents modèles si nécessaire). Elle est constituée initialement de 100 021 polices TPL pour l'assurance automobile privée. Suite à des répétitions de certains numéros de polices, nous avons pris le soin de les éjecter de la base de données, après cette opération le jeu de données ne contient désormais 100 000 polices. Chaque police a été observée au plus un an et contient les caractéristiques de risque du preneur d'assurance et du véhicule assuré. Pour des raisons de confidentialité, la plupart des niveaux catégoriels ont une signification inconnue, ce qui peut d'une certaine manière limiter les analyses que nous effectuerons. Initialement, pg15training contient 20 variables, mais nous présenterons ici que les variables utilisées dans la suite de notre analyse, à savoir :

Type	Variable	Label
variables réponses	Numtppd	Le nombre de sinistres matériels causés aux tiers.
	Numtpbi	Le nombre de sinistres corporels causés aux tiers.
	Indtppd	Le cout total en Euro des sinistres matériels causés aux tiers.
	Indtpbi	Le cout total en Euro des sinistres corporels causés aux tiers .
variables explicatives	Gender	Le genre du conducteur.
	Type	Le type de véhicule (représenté par une unique lettre).
	Age	L'âge du conducteur en année (en France l'âge pour conduire est de 18 ans).
	Poldur	L'âge de la police (en année).
	Adind	Une variable binaire indiquant une couverture matérielle.
	Density	La densité d'habitants (nombre d'habitants par Km^2) dans la ville où vit le conducteur.
	Expdays	Durée de l'exposition (en jours).

TABLE 3.1.1: Les variables de la base de données

Pour des raisons pratiques, nous avons crée les variables `Nbclaims`, `ClaimAmount`, `frequency` et `avg_amount` représentant respectivement pour chaque police le nombre total des sinistres enregistrés, le montant total des sinistres enregistrés, la fréquence de sinistre (nombre total de sinistre enregistré par rapport à l'exposition) et le coût moyen des sinistres enregistrés (coût total de sinistre par rapport au nombre de sinistre). Cependant on constate dans les données brutes qu'une variable généralement catégorielle est enregistrée comme variable continue, il nous revient donc le soin de la transformer en facteur, il s'agit de la variable `Adind`, variable binaire indiquant si l'assuré a pris en supplément la couverture pour les dégâts matériels.

3.1.1 Statistiques descriptives sur la base de données

La figure 3.1.1 montre comment le nombre de sinistre, l'exposition et la sévérité des sinistres sont distribués dans le jeu de données TPL.
La majeure partie des assurés (84.11%) n'a déclaré aucun sinistre pendant leur période d'exposition, un nombre substantiel d'assurés (13.18%) en a déclaré un seul sinistre, d'autres en n'ont déclarés deux sinistres (2.132%) et les autres assurés (0.579%) en ont déclarés au

moins trois et au plus huit sinistres. La plupart des assurés (72.57%) a été exposée pendant une année entière (365 jours) et l'exposition des autres assurés (27.43%) est inférieure à un an. Les assurés ayant une exposition inférieure à un an ont soit racheté leur contrat soit rejoint le portefeuille au cours d'une des années d'enregistrement. La fréquence globale des sinistres du portefeuille, calculée comme le ratio entre le nombre total de sinistres et l'exposition totale, est égale à 21.64%. La sévérité globale moyenne des sinistres du portefeuille, calculée comme le rapport entre le montant total des sinistres et le nombre total de sinistres est égale à 1 692.852 euros.

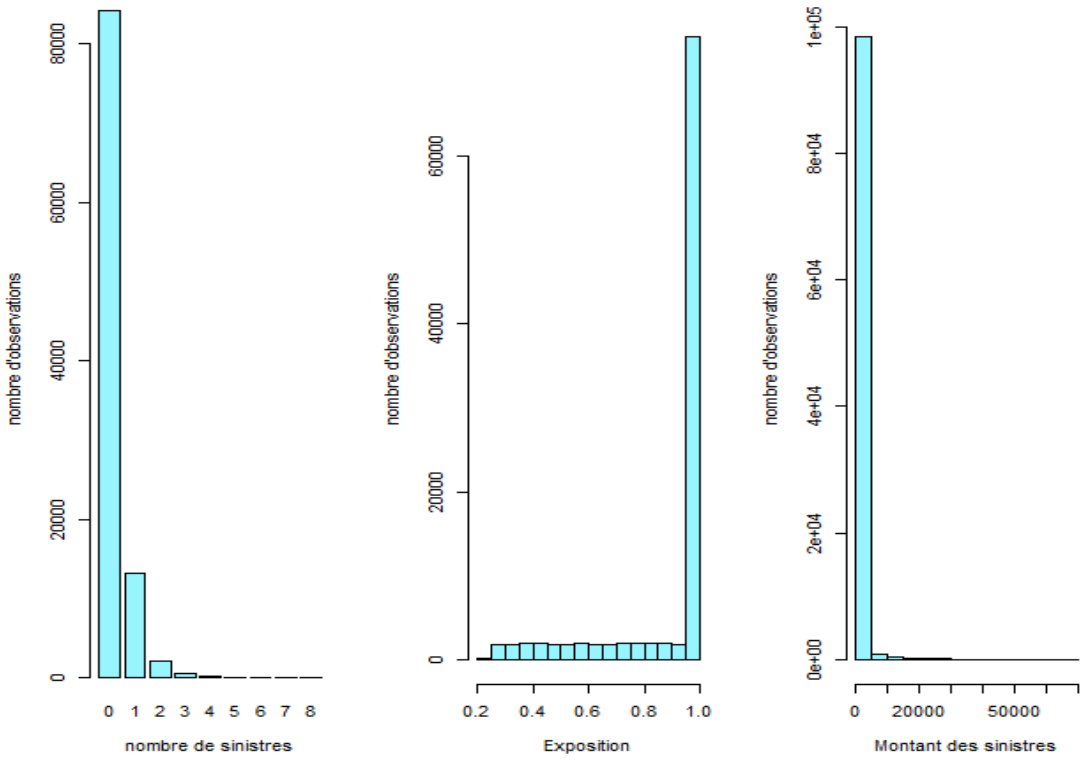


FIGURE 3.1.1: Répartition du portefeuille en fonction du nombre de sinistre, de l'exposition et de la sévérité

La figure 3.1.2 montre la répartition du portefeuille au sein de chaque facteur de risque. Ce jeu de données contient désormais trois facteurs de risque catégoriels :
`Gender, Type, Adind`.
On constate qu'il y'a presque autant d'assuré qui ont une couverture TPL (48.8%) que ceux qui ont pris en plus une couverture complémentaire pour dégâts matériels (51.2%), ce qui peut s'expliquer par le fait qu'environ la moitié des assurés ont des véhicules neufs ou qu'intrinsèquement ils se sentent plus exposés à des accidents. Pour ce qui est de la couverture TPL, seule la responsabilité du conducteur en vers les tiers est couverte. Comme dans

plusieurs pays, cette couverture est obligatoire. La plupart des assurés sont des hommes (63.43%) et conduisent en majorité des véhicules de types A (27.67%).
Le jeu de données que nous utiliserons contient désormais trois facteurs de risque continus : `Age, Poldur, Density`, représentant respectivement l'âge du conducteur, l'âge de la police et la densité de la population de la ville où réside le conducteur.
Une grande partie des assurés (85.167%) a un âge compris entre 20 et 60 ans, ce qui signifie qu'il y'a peu d'anciens conducteurs dans le portefeuille, on y retrouve en majorité (13.73%) des polices de moins d'un an, une plus grande proportion (57.55%) des assurés réside dans des villes dont la densité de la population est comprise entre 14 et 120 km2.

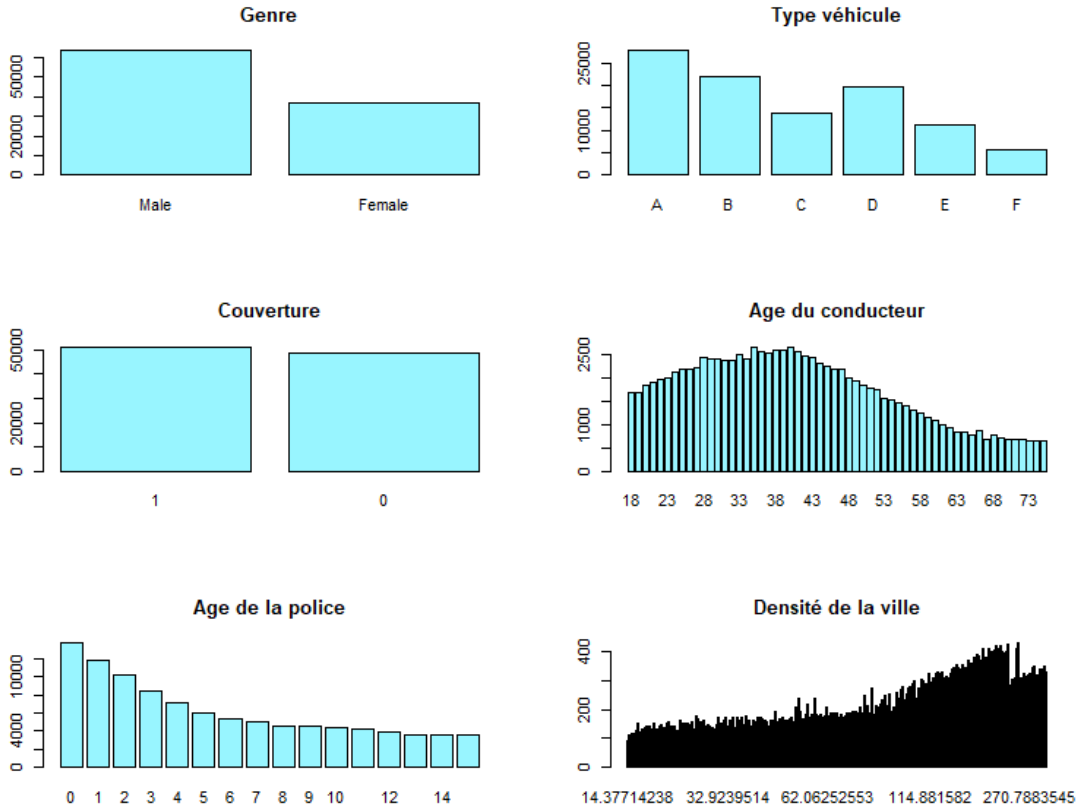


FIGURE 3.1.2: Répartition du portefeuille en fonction des facteurs de risque

3.1.2 Analyses des facteurs de risque relatif à la fréquence de sinistre

3.1.2.1 Analyses univariées

L'analyse de la figure 3.1.3 qui représente la répartition de la fréquence de sinistre observée pour chaque facteurs de risques nous montre que la sinistralité diminue à la fois avec l'âge du conducteur et aussi avec l'âge de la police. Les jeunes conducteurs sont les plus risqués avec une fréquence de sinistre supérieure à 50%. Les conducteurs plus expérimentés (en général plus âgés) sont moins risqués avec une fréquence de sinistre de moins de 10%. On remarque également que les assurés âgés de 25 à 30 ans, avec une fréquence de sinistre de 30%, ont tendance à déclarer plus de sinistre que ceux de 31 à 59 ans (sinistralité d'environ 17%). L'évolution de la fréquence de sinistre en fonction de l'âge de la police peut s'expliquer par le fait que ce sont les vieux conducteurs (expérimentés) qui possèdent les polices les plus vieilles. Le sexe est l'un des critères principal dans la tarification actuarielle, mais certains pays interdisent sont utilisation pour des raisons "d'égalité de sexe". Néanmoins nous remarquons que dans notre portefeuille, les hommes sont plus risqués que les femmes, avec une sévérité de 23.4% contre 18.63% chez les femmes. Nous remarquons également que les assurés ayant juste une couverture TPL déclarent plus de sinistre avec une fréquence de 25.3%.
Le défaut des analyses effectuées dans ce paragraphe est qu'elles peuvent être faussées par des effets de corrélations, voir [Denuit et al., 2007].

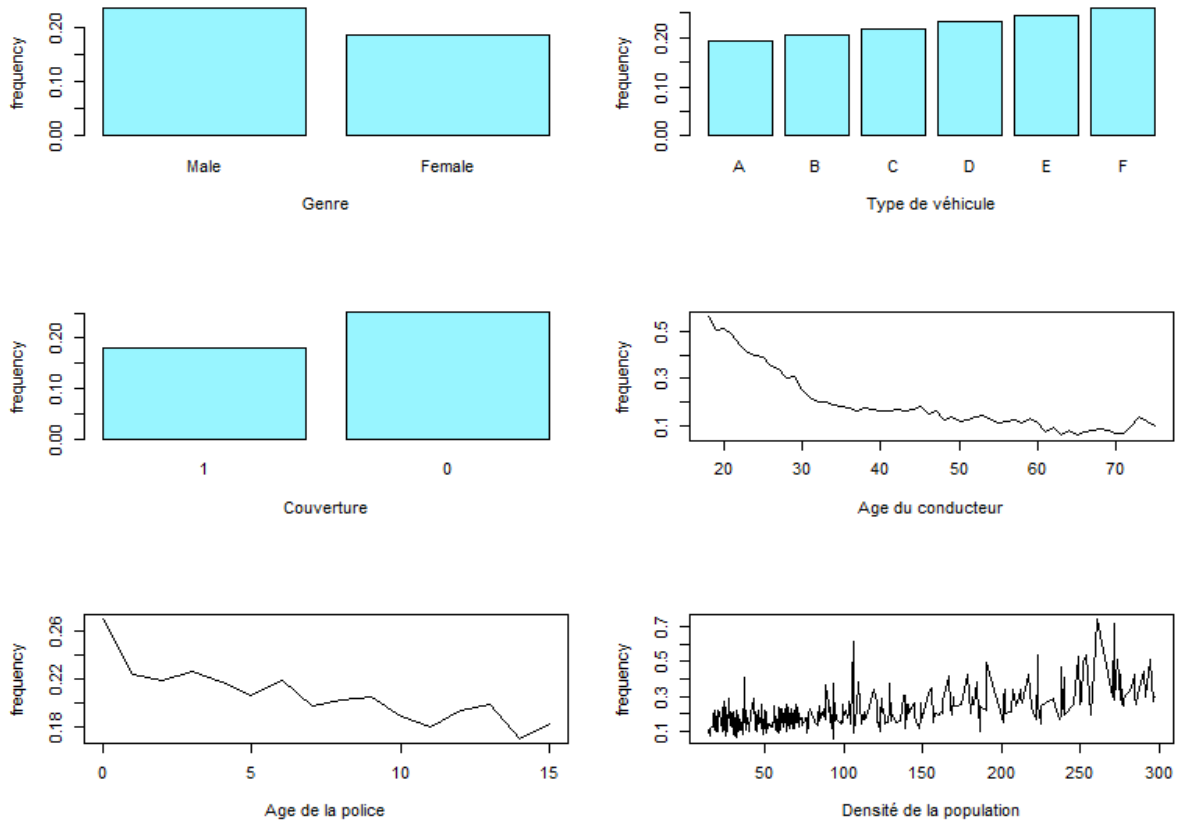


FIGURE 3.1.3: Fréquence de sinistre observée

3.1.2.2 Analyses multivariées

Une analyse bi-dimensionnelle permettrait de visualiser les possibles interactions et corrélations entre les différents facteurs de risque. Le graphique de la figure 3.1.4 nous montre que pour des assurés âgés entre 20 et 24 ans, 31 et 47 ans environ, l'effet de l'âge sur la fréquence de sinistre est différent en fonction que l'assuré soit un homme ou une femme. Pour des polices d'assurance âgées entre 3 et 7 ans, nous remarquons un écart considérable entre la fréquence de sinistre des hommes et des femmes. Cependant, généralement en tarification l'interaction entre le sexe et le genre de l'assuré est celle considérée. De ce fait, nous considérons uniquement cette interaction dans la suite de notre analyse. Intuitivement on supposerait une corrélation entre l'âge du conducteur et l'âge de la police, car comme mentionné plus haut, les conducteurs âgés ont en général des polices également âgées, et donc l'âge de la police semblerait donner une information sur l'âge du conducteur et inversement. Mais le calcul de ce terme de corrélation se révèle n'être que de 5%, trop faible pour conclure que ces deux variables sont corrélées.

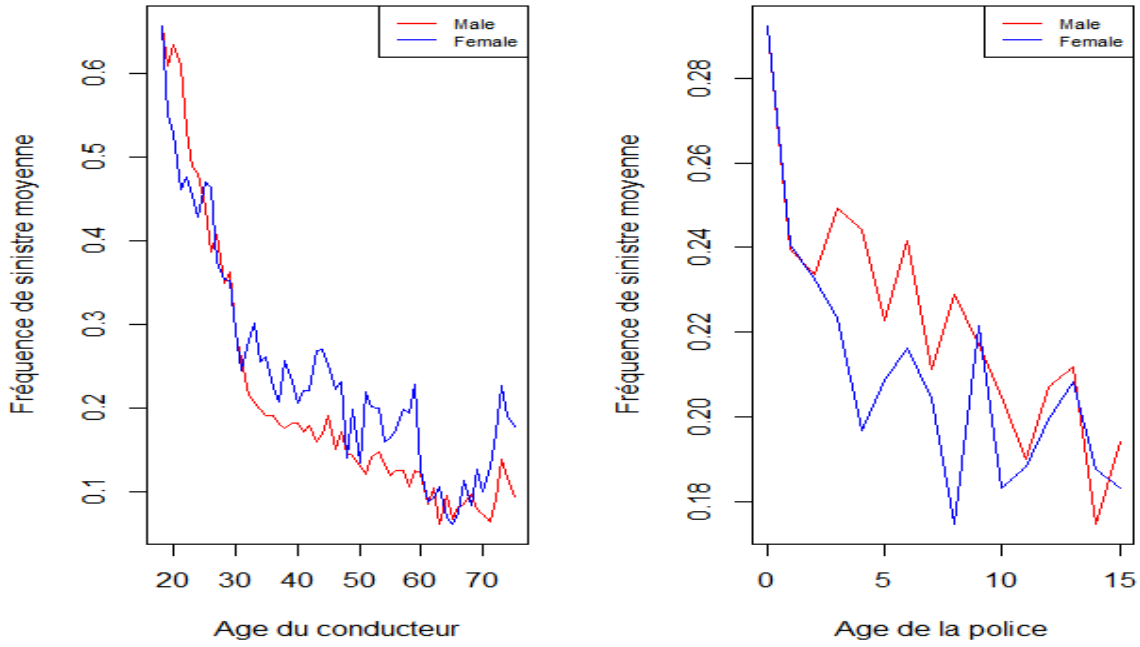


FIGURE 3.1.4: Fréquence de sinistre observée en fonction de l'âge du conducteur et du genre, l'âge de la police et du genre

3.2 Segmentation avec le package insurancerating

Dans ce mémoire, nous utiliserons la méthode de validation non-croisée en vue d'effectuer d'éventuelles comparaison, cette méthode consiste à diviser notre portefeuille de 100 000 observations en deux sous-échantillons, le premier (80%) pour l'apprentissage du modèle et le second (20%) pour tester les différents modèles avec des critères de performance tels que AIC, BIC et la déviance de Poisson. Le modèle est bâti sur l'échantillon d'apprentissage et validé sur l'échantillon de test avec un score de performance de notre choix. Notre portefeuille d'entraînement est donc constitué de 80 000 polices tandis que le portefeuille test est constitué de 20 000 polices. Nous précisons également que pour raisons liées à la taille de notre mémoire et pour des raisons pratiques, il ne sera présenté dans ce travail que le cas d'un modèle de fréquence, car le nombre de sinistre est la principale variable permettant de juger la sinistralité d'un assuré.

3.2.1 Cas du modèle de fréquence de sinistre

Il est question dans cette section de construire des classes de risque homogènes dans le cas d'un modèle de fréquence en utilisant le package insurancerating. Nous commencerons par ajuster dans différents modèles GAMs le nombre de sinistre sur chaque facteurs de

risque (pris individuellement), ensuite par le mécanisme des arbres évolutifs, on utilisera le résultat GAM (objet résultant de la fonction `fit_gam`) pour déterminer les différentes classes relatives (points de rupture) à chacune des covariables concernées.

3.2.1.1 Modèles GAMs

La première étape consiste à estimer la relation non paramétrique existante entre le nombre de sinistre et chacune des variables continues ([Clijsters, 2015]), cela se fait par l'usage de la fonction `fit_gam`.
Il serait tout à fait légitime de s'interroger sur la pertinence d'un effet individuel, qui intuitivement semble moins pertinent que l'éffet marginal d'un modèle global à priori, c'est-à-dire un modèle qui tient compte de toutes les variables catégorielles et continues pertinentes, ceci dans le but de se rapprocher au mieux des valeurs observées. Il s'agit d'ajuster le modèle GAM suivant pour chacun des facteurs de risque continus x, avec le nombre de sinistre N_i étant la variable réponse suivant une distribution de Poisson et a_i l'exposition au risque.

$$E(N_i) = a_i \; exp(\beta_0 + f(x_i))$$

Les estimations de ce modèle GAM sont données par :

$$N_i^{pred} = a_i \; exp(\hat{\beta}_0 + \hat{f}(x_i))$$

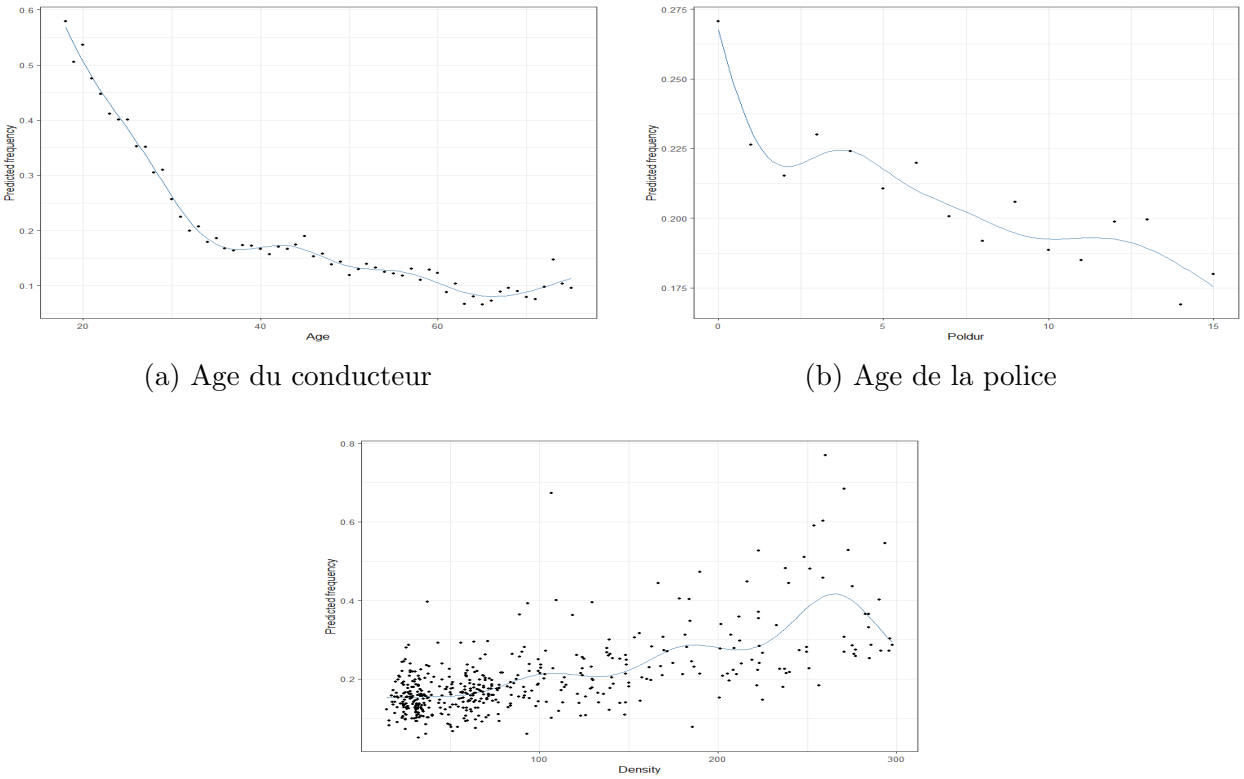
Où $\hat{f}(x_i)$ représente l'estimation de la relation non paramétrique entre le nombre de sinistre et le facteur de risque x.
Nous avons montré à la section 2.2.1.4 que la fonction `fit_gam` utilise les splines à plaques minces qui permettent de modéliser l'interaction entre les variables continues, bien que nous n'ayons ce type d'interaction dans nos données, des analyses ont montré que cette fonction du fait de son paramétrage ne permet pas de tenir compte d'une telle interaction, ou même d'une interaction entre une covariable et une variable catégorielle. Étant donné l'interaction existante entre le sexe et l'âge du conducteur, celle-ci ne sera prise en compte que dans le modèle GLM incluant les variables catégorielles pour des raisons d'interpretabilité. Nous rappelons que l'interaction entre le sexe et l'âge suppose que l'impact de l'âge sur la fréquence de sinistre est différent en fonction qu'on soit un homme ou une femme. De plus, l'usage par défaut des paramètres de la fonction `s()` impose l'utilisation des noeuds basés sur un découpage en valeurs équidistantes de l'étendue de la covariable sans tenir compte de la structure des données, car ces noeuds ne sont pas appropriés pour tous types de covariables. Les nœuds sont généralement situés aux quantiles de la caractéristique x et non à chaque valeur observée de la covariable, l'utilisation de quelques nœuds entraîne une estimation biaisée, tandis que l'utilisation de nombreux nœuds peut aboutir à des estimations instables, c'est-à-dire avec une variance élevée ([Denuit et al., 2019]). Pour un petit nombre de nœuds, la spline résultante n'est pas suffisamment flexible pour capturer la variabilité des données et pour un grand nombre de nœuds, les courbes estimées ont tendance à surajuster les données. Pour surmonter ces difficultés, [Denuit et al., 2019] propose d'utiliser l'approche des splines pénalisées qui est une des méthodes les plus éfficaces en

statistique appliquée. Par exemple, dans le cas d'une variable telle que `Density` (la densité de la population), les valeurs de la variable n'étant pas successives, ces noeuds équidistants ne sont pas adéquats. Il serait bien plus approprié d'utiliser des noeuds basés sur des quantiles de la covariable, c'est-à-dire utiliser pour ce type de covariables des splines pénalisées pour estimer la fonction de lissage.

La figure 3.2.1, montre les différents effets non paramétriques individuels $\hat{f}(x_i)$ respectifs de l'âge du conducteur, de l'âge de la police et de la densité de population sur le nombre de sinistre.
On observe sur la figure 3.2.1 (graphe a) qu'un nombre de sinistre élevé est déclaré par les jeunes conducteurs jusqu'à l'âge de 25 ans. Les assurés qui déclarent le moins de sinistres sont les personnes d'au moins 60 ans, ils détiennent plusieurs années d'expérience de conduite. On remarque également qu'à partir de l'âge de 35 ans, la fréquence de sinistre chute pour autour de 17% jusqu'à l'âge de 45 ans. On observe une tendance similaire lorsqu'on examine l'effet non paramétrique de l'âge de la police d'assurance (graphique b) sur le nombre de sinistre déclaré, ce qui est un peu logique, car les jeunes conducteurs ont généralement de nouvelles polices d'assurance contrairement au conducteurs plus âgés qui peuvent avoir des polices de près de 15 ans d'âge même en supposant qu'ils peuvent changer d'assureur une fois au cours de leurs expériences de conducteur. On observe également un nombre élevé de déclarations pour les assurés ayant des contrats âgés de moins de 5 ans, ce qui peut également se justifier par le fait que l'âge de la police peut dans certains cas être corrélé à l'âge du véhicule et/ou à l'âge du conducteur. De plus, nos données provenant d'une compagnie française, on sait qu'en France le premier contrôle technique périodique est à faire au cours des six mois précédant le quatrième anniversaire de la première mise en circulation, ce qui rend les nouveaux véhicules plus exposés aux sinistres. Le dernier graphique (c) montre que les assurés du portefeuille sont plus représenté dans les villes à faible densité de population, où le fréquence de sinistre est plus faible que celle des assurés résidant dans les villes les plus denses.

3.2.1.2 Construction des classes par le mécanisme les arbres évolutionnaires

Le point de départ ici est basé sur les résultats obtenus des différents modèles GAMs individuels (résultat de la fonction `fit_gam`), c'est-à-dire les différents effets non paramétriques, cette étape consiste à utiliser l'effet non paramétrique de chacune des variables continues, qui sera ajusté sur chacune de celles-ci (variable continue) dans un arbre évolutif. Au coeur de la fonction `construct_tariff_classes`, on recherche les points de rupture dans l'arbre afin de pouvoir segmenter la covariable en catégories homogènes.
Avant de commencer à construire les classes, il est important d'avoir un aperçu sur l'exposition correspondant à chaque niveau de la variable. Par exemple, au travers de la figure 3.2.1, on voit que le nombre de sinistre prédits dans le modèle GAM diminue fortement jusqu'à l'âge de 30 ans, ensuite continue de décroître faiblement, on remarque un pic inférieur à l'âge de 65 ans et un retour à la croissance après 65 ans. Cependant certaines tranches d'âges possèdent peu d'assurés, ce qui leurs donnent moins d'intérêt aux yeux des compa-



(a) Age du conducteur

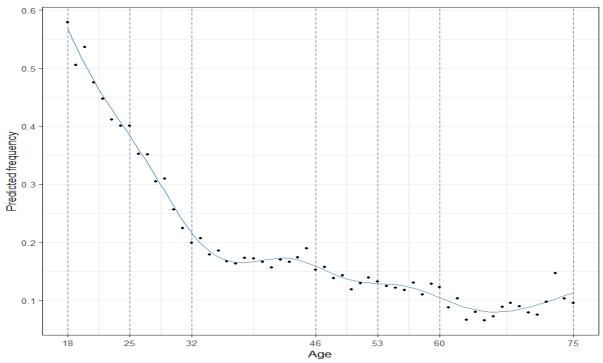
(b) Age de la police

(c) Densité de la ville du conducteur

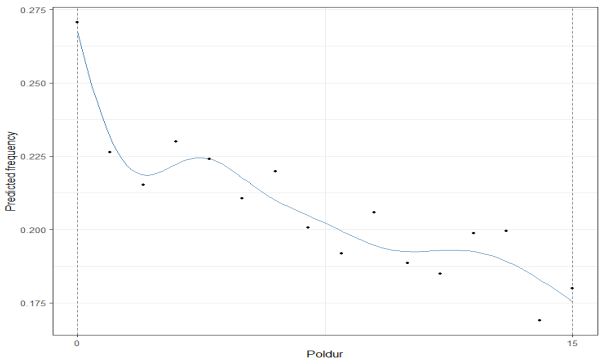
FIGURE 3.2.1: Estimation des effets individuels des covariables sur la fréquence de sinistre

gnies. Ne pas tenir compte des poids de chaque niveau lors de la construction des classes aboutira à un nombre élevé de classe contenant peu d'individu. On se demande si la prise en compte des poids dans la fonction `fit_gam` peut compenser l'absence des poids dans les arbres évolutifs. Ici le poids représente le nombre de fois qu'une observation apparaît dans le jeu de données, en analysant la fonction `construct_tariff_classes`, on se rend compte qu'elle ne tient pas sans toutefois donner la possibilité de tenir compte des poids des différentes niveaux dans le portefeuille. Ainsi l'arbre est construit sur la base de l'estimation GAM sans toutefois tenir compte de l'importance relative des observations dans le portefeuille.

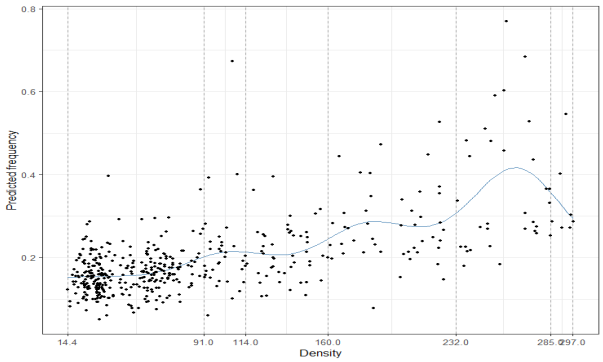
La figure 3.2.2 montre comment les différentes variables sont segmentées. Le tableau ci-dessous présente les différentes classes qui en découlent.



(a) Age du conducteur



(b) Age de la police



(c) Densité de la ville du conducteur

FIGURE 3.2.2: Segmentation des effets individuels des covariables sur la fréquence de sinistre

`Age_class`	effectif	`Density_class`	effectif	`Poldur`	effectif
[18, 25]	12 357	[14.4, 91]	38 426	[0, 15]	80 000
(25, 32]	12 985	(91, 114]	6 440		
(32, 46]	27 765	(114, 160]	12 044		
(46, 53]	10 422	(160, 232]	13 258		
(53, 60]	7 354	(232, 285]	8 026		
(60, 75]	9 126	(285, 297]	1 806		
Total	80 000	Total	80 000	Total	80 000

TABLE 3.2.1: Les segments construits à l'aide du package insurancerating

La variable `Age` a été segmentée en six catégories, la plus peuplée est celle comprenant les assurés âgés de 32 à 46 ans. On remarque que toutes les classes possèdent au minimum 9% du portefeuille d'entraînement, ce qui est assez représentatif. Pour cette variable, le package s'avère assez efficace, car une bonne segmentation devrait aboutir à des segments

contenant au minimum 5% des polices du portefeuille dans chacune des différentes classes ([Henckaerts et al., 2018]).
Pour la variable `Density`, on obtient également six catégories avec des effectifs assez disproportionnés, une classe contenant près de 48% des données, deux classes contenant près 15% des données, deux autres contenant entre 8% et 10% des données, une dernière qui ne comporte que 2% des données, cette classes n'est donc pas assez représentative du portefeuille d'entraînement. Pour la variable `Poldur` on obtient une seule catégorie, ce qui n'est pas un résultat espéré, car ayant analysé les estimations GAM de l'âge de la police sur le nombre de sinistre prédit, la relation existante n'est pas constante et donc le nombre de sinistre varie en fonction de l'âge de la police. Ce résultat se justifie techniquement par le manque de flexibilité de la fonction `construct_tariff_classes`, car cette fonction prenant en argument la fonction `evtree.control`, permettant de fixer les paramètres de contrôle de l'arbre, le paramètre représentant la somme minimale des poids dans un nœud afin d'être pris en compte pour le fractionnement est pris par défaut `(minsplit = 20L)`. Cette valeur étant donc supérieure au nombre de valeurs de la variable `Poldur` (allant de 0 à 15), aucun split ne peut être trouvé, d'où l'obtention d'une seule classe. Ce résultat remet une fois de plus en question l'optimalité de cet outil, sa rigidité ne permettant pas d'adapter différents paramètres en fonction des données. Les variables pertinentes de la nouvelle base de données doivent être intégrées dans un modèle GLM permettant ainsi d'estimer le nombre de sinistre, d'obtenir les différentes classes homogènes au sein du portefeuille assorties de leurs fréquences de sinistre prédites. N'ayant pas pu tenir compte de l'interaction du fait du paramétrage de la fonction `fit_gam`, nous l'incluons donc dans le modèle GLM, à fin d'estimer des coefficients correcteurs en fonction du sexe.

3.2.1.3 Implémentation d'un modèle GLM avec les variables catégorisées

Nous rappelons que le but de notre analyse est de transformer les facteurs de risque continus en variables catégorielles à fin que celles-ci soient facile à implémenter dans un modèle GLM. Dans la section 3.2.1.2, nous avons transformé les covariables en variables catégorielles, ici nous ajustons le nombre de sinistre observé en fonction de toutes variables catégorielles pertinentes dont on dispose désormais dans un modèle linéaire généralisé, ce qui nous permettra d'analyser la significativité de ces catégories créées dans ce contexte et dans une autre section, cela nous permettra de construire des segments globaux (segments basés sur les variables intégrées dans le modèle) de fréquence annuelle prédites dans le portefeuille.
Pour obtenir le meilleur modèle GLM, on effectue une recherche exhaustive de toutes les combinaisons des variables explicatives, ceci se fait en exécutant un algorithme stepwise avec une méthode forward et validation croisée pour sélectionner les meilleures variables pour notre modèle, nous commençons par ajuster un modèle sans aucune variable explicative :

$$glm(Nbclaims \sim 1)$$

Les variables sont ensuite ajoutées une à une jusqu'à ce que le critère de sélection (AIC, BIC ou déviance de poisson) cesse de décroître malgré l'ajout d'une variable supplémentaire.

Pour chacun des ces trois critères, le meilleur modèle GLM contient cinq variables : `Age_class, Density_class, Gender, Type, et Adind`.
Comme attendue, la variable `Poldur` n'est pas sélectionnée dans le modèle final, car ayant un seul niveau, elle ne peut pas être considérée comme pertinente par l'algorithme stepwise. On rajoute ensuite l'interaction entre l'âge et le genre du conducteur et le modèle que nous exécutons se présente comme suit :

```
GLM<-glm(formula =Nbclaims~Density_class+Gender+Type+Adind+Age_class+
    Age_class*Gender
      ,family = poisson(link = "log"),data = training, offset = log(
    Expo))
```

Le résultat du modèle GLM présenté ci-dessous montre que les coefficients estimés pour chacun des segments construits pour la variable `Age_class` et `Density_class` ont tous une `p-value` inférieure à 5%, ce qui montre que ces classes sont significatives dans notre modèle. On remarque également que les termes d'interaction pour les femmes de plus de 32 ans n'est pas significatif, montrant que à cet âge là, le nombre de sinistre déclaré ne dépend pas forcément du genre.

```
(Intercept)                      -2.15982    0.02687 -80.374  < 2e-16 ***
Density_class(91,114]             0.27240    0.03159   8.622  < 2e-16 ***
Density_class(114,160]            0.27159    0.02486  10.926  < 2e-16 ***
Density_class(160,232]            0.55382    0.02192  25.265  < 2e-16 ***
Density_class(232,285]            0.81435    0.02359  34.515  < 2e-16 ***
Density_class(285,297]            0.74332    0.04492  16.548  < 2e-16 ***
GenderFemale                     -0.10796    0.03202  -3.372 0.000746 ***
TypeB                             0.06622    0.02371   2.793 0.005227 **
TypeC                             0.11847    0.02683   4.416 1.00e-05 ***
TypeD                             0.18688    0.02375   7.867 3.63e-15 ***
TypeE                             0.24507    0.02759   8.882  < 2e-16 ***
TypeF                             0.29529    0.03520   8.390  < 2e-16 ***
Adind0                            0.07845    0.01686   4.654 3.25e-06 ***
Age_class[18,25]                  1.10464    0.02556  43.210  < 2e-16 ***
Age_class(25,32]                  0.58270    0.02820  20.661  < 2e-16 ***
Age_class(46,53]                 -0.25148    0.03897  -6.453 1.09e-10 ***
Age_class(53,60]                 -0.37140    0.04681  -7.934 2.12e-15 ***
Age_class(60,75]                 -0.70995    0.04810 -14.759  < 2e-16 ***
GenderFemale:Age_class[18,25]    -0.42344    0.04437  -9.544  < 2e-16 ***
GenderFemale:Age_class(25,32]    -0.28163    0.04948  -5.692 1.26e-08 ***
GenderFemale:Age_class(46,53]     0.05631    0.06720   0.838 0.402061
GenderFemale:Age_class(53,60]     0.08022    0.08174   0.981 0.326377
GenderFemale:Age_class(60,75]     0.11115    0.08703   1.277 0.201526
```

Le terme d'interaction ici permet de corriger la fréquence de sinistre pour les femmes par rapport à celles des hommes (classe de référence), pour les assurées appartenant aux classes `[18, 25]`, `(25, 32]`, les coefficients de corrections sont respectivement -0.42344, -0.28163, montrant ainsi que le nombre de sinistre déclaré par les jeunes conductrices est inférieur au nombre de sinistre déclaré par les jeunes conducteurs. Bien que les coefficients estimés

correspondants aux termes d'interaction pour les conductrices de plus de 46 ans ne sont pas significatifs, ces coefficients sont tout de même positifs et semblent bien corroborer le fait que les conductrices de âgées sont plus risquées que leurs homologues masculin (voir 3.1.4). Pour ces deux variables `Gender` et `Age_class` qui interagissent, les niveaux de référence sont respectivement `Male` et `(32, 46]`, ainsi tout terme d'interaction incluant ces deux niveaux sont nuls. Pour tous les hommes du portefeuille et les femmes âgées de 33 à 46 ans, il n'y a pas de terme de correction.

Pour avoir plus d'informations sur notre modèle, on décide de visualiser l'importance des différents niveaux de chaque variable dans le modèle, ceci à l'aide de la fonction `visreg` du package R visreg. Les traits bleus sur la figure 3.2.3 représentent la fréquence de sinistre correspondant à chaque catégorie prise individuellement, c'est-à-dire l'exponentielle du coefficient estimé pour une catégorie par exemple. Pour une même variable, des catégories ayant approximativement la même fréquence de sinistre sont semblables et peuvent donc être regroupées.

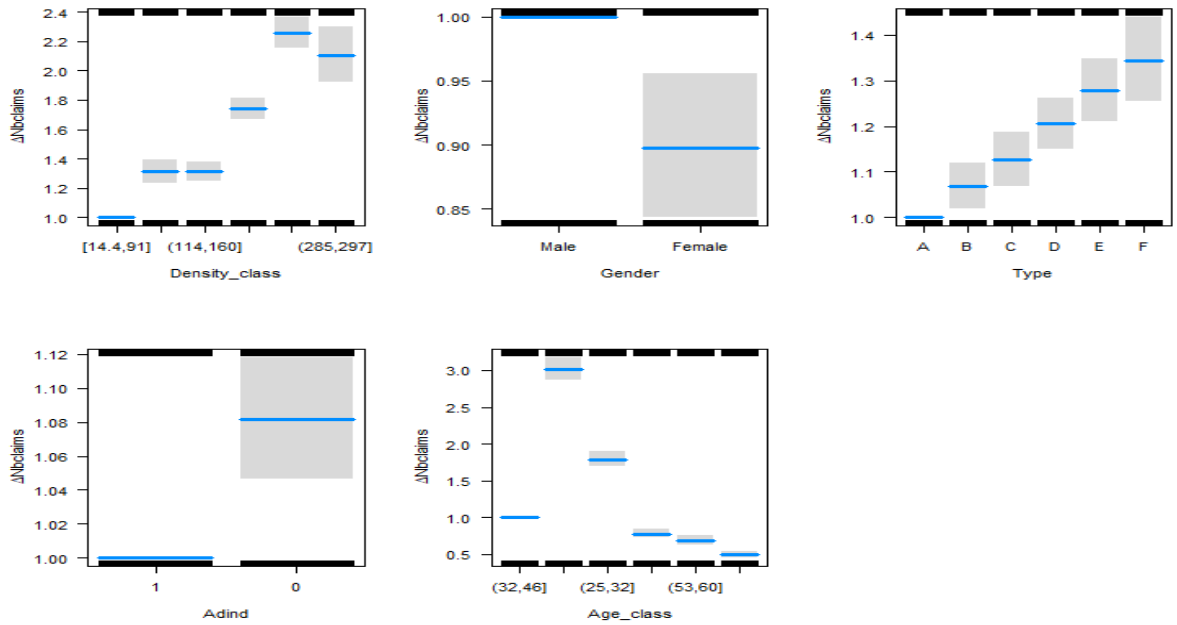


FIGURE 3.2.3: Importance relative des catégories de quelques variables dans le modèle GLM

La figure 3.2.3 montre que pour les variables `Density_class` et `Age_class`, les classes qui ont approximativement les mêmes importances sont respectivement `(91, 114]`, `(114, 160]` et `(46, 53]`, `(53, 60]`. Ce constat remet en cause l'optimalité de notre outil de segmentation car on constate que les classes créées ne sont pas totalement différentes les unes des autres. Étant donné l'objectif de ce mémoire qui est analyser les résultats fournit par ce package, nous conservons la catégorisation telle quelle. Pour les autres variables du

modèle, chaque niveau a une fréquence de sinistre particulière, différente d'un niveau à un autre.
Ainsi, ce modèle GLM nous permet d'estimer le nombre de sinistre, on détermine ensuite la fréquence de sinistre estimée en divisant le nombre de sinistre estimé par le nombre d'exposition, c'est ainsi donc qu'on compare graphiquement ces fréquences observées et prédites par rapport à l'âge du conducteur et à la densité de la population, ce qui nous permettra dans un premier temps de visualiser la catégorisation effectuée sur la fréquence de sinistre au sein du portefeuille et dans un second temps au sein des différents segments globaux obtenus comme le montre la figure 3.2.4 ci-dessous.

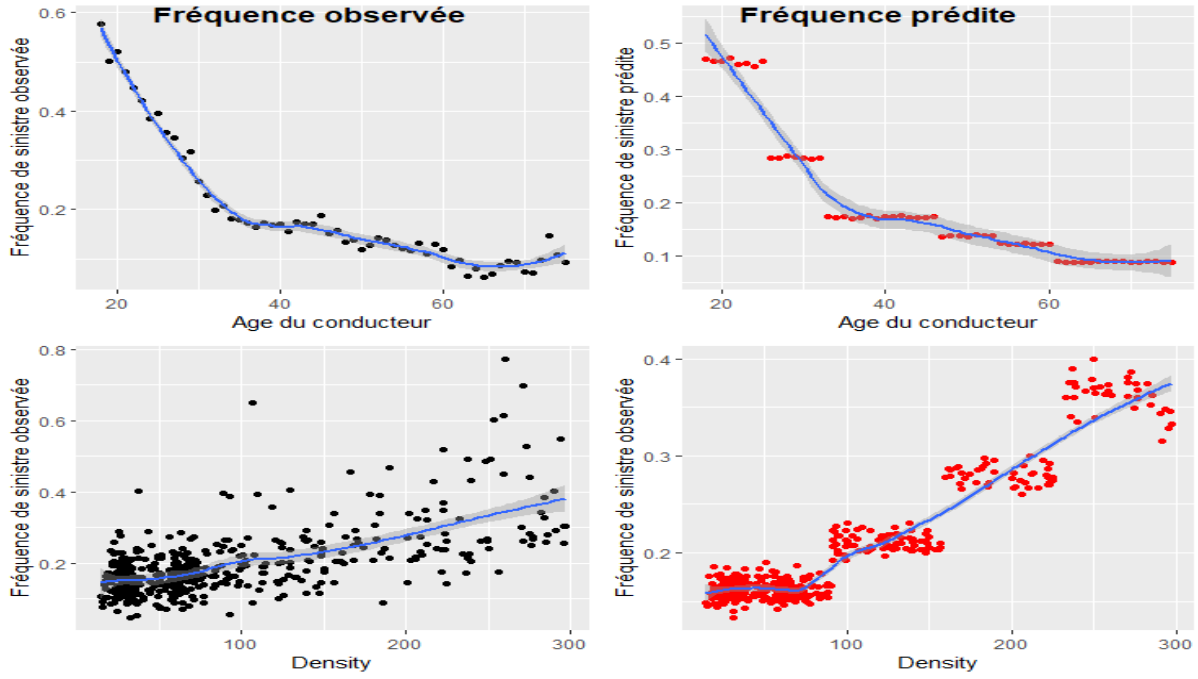


FIGURE 3.2.4: Fréquence de sinistre prédite et observée en fonction de l'âge du conducteur

Ici on se focalise sur les fréquences prédites (graphiques de droite), sans surprise on constate que la fréquence de sinistre prédite en fonction de l'âge du conducteur, forme six regroupements, correspondant exactement aux différentes classes obtenues à l'aide package, une fois de plus, on constate que les assurés appartenant aux classes `(46, 53]` et `(53, 60]` ont une fréquence de sinistre très proches et devrait dans un autre exercice vraisemblablement former une seule classe. Dans le cas de la densité, on remarque à première vue sur le graphique inférieure droit que les six classes préalablement obtenues à la section 3.2.1.2 sont moins visibles graphiquement et moins uniformes que celles de la variable `Age`, on peut également voir qu'il existe des classes peu peuplées, il semblerait se former plutôt quatre regroupements, on a du mal à distinguer la classe `(91, 114]` de la classe `(114, 160]` et `(232, 285]` de la classe `(285, 297]`.

Sur la base du modèle GLM ainsi ajusté et des coefficients estimés y afférents, on détermine la fréquence de sinistre attendue et ainsi aboutir à la création des groupes homogènes.

3.2.1.4 Création de groupes homogènes au sein du portefeuille TPL

Dans le cadre de la segmentation du portefeuille, le modèle que nous avons retenu se présente comme suit :

$$\log(\mathbb{E}(\texttt{Nbclaims})) = \log(\texttt{Expo}) + \beta_0 + \beta_1 \texttt{Gender}_{Female} + \beta_2 \texttt{Type}_B + \beta_3 \texttt{Type}_C + \beta_4 \texttt{Type}_D + \beta_5 \texttt{Type}_E + \beta_6 \texttt{Type}_F + \beta_7 \texttt{Adind}_0 + \beta_8 \texttt{Age\_class}_{[18,25]} + \beta_9 \texttt{Age\_class}_{(25,32]} + \beta_{10} \texttt{Age\_class}_{(46,53]} + \beta_{11} \texttt{Age\_class}_{(53,60]} + \beta_{12} \texttt{Age\_class}_{(60,75]} + \beta_{13} \texttt{Density\_class}_{(91,114]} + \beta_{14} \texttt{Density\_class}_{(114,160]} + \beta_{15} \texttt{Density\_class}_{(160,232]} + \beta_{16} \texttt{Density\_class}_{(232,285]} + \beta_{17} \texttt{Density\_class}_{(285,297]} + \beta_{18} \texttt{Age\_class\_Female}_{[18,25]} + \beta_{19} \texttt{Age\_class\_Female}_{(25,32]} + \beta_{20} \texttt{Age\_class\_Female}_{(46,53]} + \beta_{21} \texttt{Age\_class\_Female}_{(53,60]} + \beta_{22} \texttt{Age\_class\_Female}_{(60,75]}$$

On rappelle que toute variable catégorielle doit être coder à l'aide de variable binaire avant d'être utiliser dans les modèles linéaires. Par exemple, si une variable catégorielle à n niveaux, $n-1$ variables binaires vont être utilisées pour le codage (voir cours LACTU2110). Le but ici est de se servir des coefficients estimés pour faire ressortir au sein de notre portefeuille des groupes d'assurés similaires. On crée donc des variables binaires prenant des valeurs "yes" ou "no", permettant ainsi de savoir si un assuré appartient à une catégorie ou pas et donc si son coefficient estimé pour cette catégorie sera nul ou pas. La création des classes se fait donc en extrayant de façon unique chaque vecteur de variable binaire (yes ou no) assorti d'une fréquence observée, d'une fréquence prédite, de l'exposition et même du nombre de sinistre, représentant ainsi les différents segments globaux dans lesquels les assurés peuvent être repartis.
Obtenu en fonction de l'appartenance de l'assuré aux différentes catégories, les segments globaux donnent la fréquence annuelle de sinistre prédite des différents assurés du portefeuille.
Les résultats obtenus dans le tableau présenté à l'annexe B.4.1 s'obtiennent de la manière suivante : L'idée est de partir de la fréquence de sinistre attendue de la classe de référence. La classe de référence ici est celle pour laquelle les variables binaires représentant l'appartenance de l'assuré à une quelconque catégorie prennent toutes la valeur "no", il s'agit en effet de assurés de sexe masculin ayant un âge compris entre 32 et 46 ans, vivant dans des villes de faible densité de population, conduisant des véhicules de type `A`, ayant pris une couverture complémentaire. Le nombre de sinistre de cette classe est estimé comme suit :

$$exp(\hat{\beta}_0) = 0.1153462.$$

A ça on applique les coefficients de correction équivalents aux différentes autres caractéristiques que peuvent avoir un preneur d'assurance dans ce portefeuille et qui sont estimés par le modèle GLM utilisé.

La figure à l'annexe B.4.1 montre les 30 premiers segments globaux obtenus après segmentation à l'aide du package insurancerating suivi de l'implémentation dans un modèle GLM. On a utilisé 17 variables binaires (hormis les interactions) pour coder toutes les variables catégorielles utilisées dans le modèle GLM, cela nous a permis d'obtenir 2 958 segments globaux, matérialisant ainsi les différents groupes homogènes au sein du portefeuille. En se basant des segments globaux (groupes homogènes) obtenus, on remarque que la fréquence prédite moyenne, calculée au sein de ceux-ci (22,95%) est légèrement supérieure à la fréquence de sinistre moyenne réellement observée (21.75%). Le nombre élevé de groupes homogènes obtenus peur se justifier par le fait d'avoir utilisé plusieurs variables catégorielles à plusieurs niveaux pour la modélisation. Et comme nous l'avons dit au début de ce mémoire, l'inconvénient d'avoir plusieurs classes est qu'il peut en résulter des classes contenant très peu d'individus voire pas du tout. Pour ce qui est de la répartition des assurés du jeu de données dans les différentes classes obtenues, on constate que le segment global le plus peuplé contient 1.67% des assurés c'est-à-dire 1 337 assurés, tandis que les segments les moins peuplées n'en possèdent chacun que 0.00125%, c'est-à-dire un seul assuré, on en dénombre plus d'une vingtaine. Le segment le plus peuplée ne correspond à la classe de référence sus-mentionnée. Comme mentionné dans l'analyse descriptive du jeu de données, les jeunes assurés sont ceux qui ont la fréquence de sinistre la plus élevée, c'est également cette remarque que l'on fait lorsqu'on analyse la classe la plus risquée qui est celle des assurés de sexe masculin, âgés entre 18 et 25 ans, vivant dans des villes plutôt peuplées, de densité comprise entre 232 et 285 $habitants/km^2$, conduisant des véhicules de type `F`, n'ayant pas pris de couverture complémentaire .
Pour l'ensemble des segments construits, on remarque que les moyennes des fréquences de sinistre observées et prédites restent très proches contrairement à la variance qui s'amplifie, ce qui montre d'une certaine manière l'hétérogénéité existante entre les différentes classes et d'une autre manière une homogénéité au sein des classes. La figure 3.2.5 nous montre la répartition des fréquences de sinistre observées et prédites au sein des 2 958 segments globaux en fonction de l'âge du conducteur et de la densité de population de la ville du conducteur. Le graphique supérieur droit nous montre pour les fréquences prédites quatre regroupements principaux en fonction de la variable âge du conducteur. Notamment une classe des assurés ayant un âge compris entre 18 et 25 ans, une classe de 25 à 32 ans, une classe de 32 à 46 ans et une dernière de 46 à 75 ans. Cette observation est confortée par la figure 3.2.3 sur le graphe inférieur droit, cette figure montre que les niveaux `(46, 53]`, `(53, 60]`, `(60, 75]` ont une fréquence de sinistre relativement proche. Cette représentation remet en cause d'une certaine façon la classification obtenue par le package insurancerating et laisse paraître une certaine similitude entre ces trois classes qui pourraient être fusionnées en une seule, mais étant donné l'objectif de ce mémoire qui est d'évaluer la segmentation proposée par ce package, nous continuerons nos analyses avec les résultats obtenus dans le but de les comparer plus tard. Pour le cas de la densité, la fréquence de sinistre au sein des différents segments formés est assez éparse, ne permettant pas ainsi de pouvoir statuer visuellement sur des éventuels clusters. On s'attendrait à ce qu'au sein des segments homogènes, on puisse obtenir la même répartition que celle au sein du portefeuille entier.

Ce groupement serait plus cohérent à notre objectif pour une variable préalablement catégorielle, telle que `Type` par exemple.

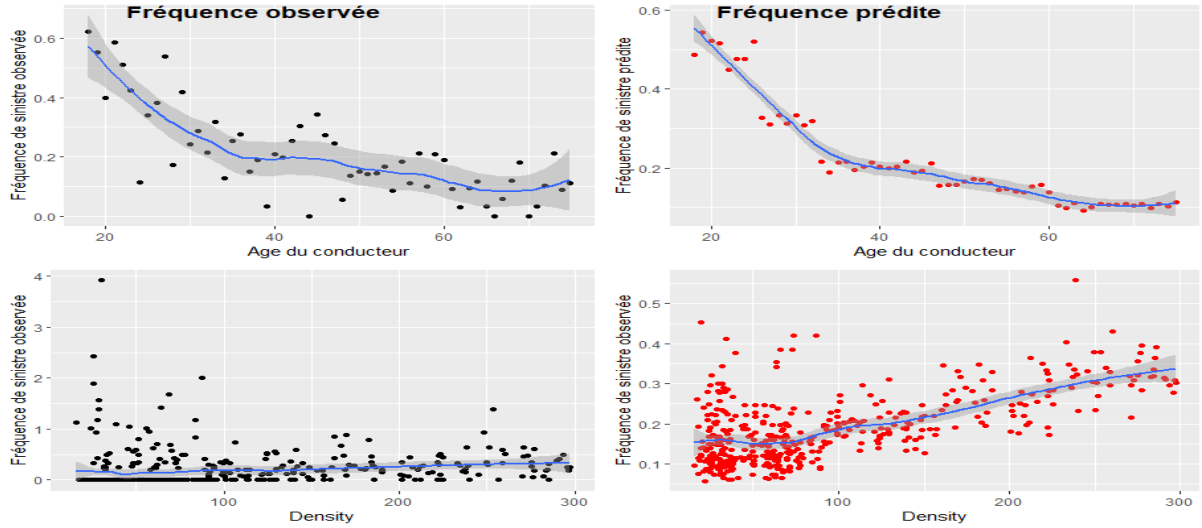


FIGURE 3.2.5: Fréquence observée et fréquence prédite au sein des classes en fonction de l'âge du conducteur et la densité.

Tout au long de ce chapitre nous avons utilisé le package insurancerating pour catégoriser les covariables, ensuite nous les avons intégrées dans un modèle GLM à fin d'estimer le nombre de sinistre des individus et sur la base de cela, déterminer les segments homogènes qui ressortent du portefeuille, également, nous avons jusqu'ici soulevé quelques problèmes lié à ce package, en particulier l'usage des effets individuels pour déterminer les points de rupture au sein des covariables, la non prise en compte des poids dans la régression des arbres évolutifs, la non prise en compte des interactions et le paramétrage rigide des fonctions principales du package. Dans le chapitre qui suivra, nous allons utiliser les effets marginaux d'un modèle GAM complet pour effectuer une segmentation sur ce même portefeuille TPL en vue d'avoir une base d'évaluation du package insurancerating.

Chapitre 4

Segmentation à l'aide d'un modèle GAM complet : application à un portefeuille d'assurance TPL

4.1 Le modèle GAM complet

Le modèle additif développé dans cette partie tient compte de toutes les variables explicatives qui s'avèrent pertinentes dans notre portefeuille.

4.1.1 Modèle de fréquence, effets marginaux dans un modèle GAM complet

Dans cette section, nous nous focalisons sur un modèle de régression flexible pour modéliser le nombre de sinistres. Pour cela, comme dans le cas du package insurancerating, on suppose que le nombre de sinistres observé suit une distribution de Poisson, cette hypothèse est commune à celle faite généralement en tarification assurantielle et est cohérente avec des travaux antérieurs fait sur ce type de données (voir [Denuit et al., 2007]). Rappelons cependant que pour cette approche, on peut néanmoins supposer que le nombre de sinistre suit une autre distribution, par exemple une distribution Binomiale-Négative, c'est dans un soucis de comparabilité que nous faisons cette hypothèse.
Le but de cette approche est d'expliquer le nombre de sinistres observés en utilisant tous les facteurs de risque pertinents à fin de pouvoir construire des classes de risque pour les covariables utilisées. Les variables explicatives utilisées dans cette partie sont les mêmes que celles utilisées dans le modèle GLM développé à la section 3.2.1.3, toujours dans un souci de comparabilité, car la variable `Poldur` pour laquelle une seule catégorie a été obtenue dans la dite section peut dans ce cas ci s'avérer pertinente, mais toutefois, nous n'en disposerons pas. Sans tenir compte de l'interaction entre l'âge et le genre, le modèle GAM que nous développons ici s'écrit comme suit :

$$\log(\mathbb{E}(\texttt{Nbclaims})) = \log(\texttt{Expo}) + \beta_0 + \beta_1 \texttt{Gender}_{Female} + \beta_2 \texttt{Type}_B + \beta_3 \texttt{Type}_C + \beta_4 \texttt{Type}_D +$$

$\beta_5 \texttt{Type}_E + \beta_6 \texttt{Type}_F + \beta_7 \texttt{Adind}_0 + f_1(\texttt{Age}) + f_2(\texttt{Density})$

Le logarithme de l'exposition est inclut dans le modèle en tant que "offset", permettant ainsi que l'espérance du nombre de sinistres soit proportionnelle à la période d'exposition de chaque assuré au sein du portefeuille. Comme nous l'avons vu dans le cours de LACTU2110, ces trois variables catégorielles sont codées à l'aide des variables binaires, et nous prenons comme niveau de référence le niveau ayant la plus grande exposition, notamment les niveaux `Male`, `A` et `1` respectivement pour les variables `Gender`, `Type` et `Adind`.

Les fonctions f_1 et f_2 représentent les effets de lissage univariés pour les facteurs de risque continus. Pour améliorer le modèle, on inclut un terme d'interaction entre l'âge du conducteur et le sexe. L'interaction entre ces deux variables est celle la plus considérée en tarification, elle réduit les critères de sélection des modèles tels que le BIC, le AIC, matérialisant ainsi l'amélioration du modèle. On obtient donc le modèle suivant :

$$\log(\mathbb{E}(\texttt{Nbclaims})) = \log(\texttt{Expo}) + \beta_0 + \beta_1 \texttt{Gender}_{Female} + \beta_2 \texttt{Type}_B + \beta_3 \texttt{Type}_C + \beta_4 \texttt{Type}_D + \beta_5 \texttt{Type}_E + \beta_6 \texttt{Type}_F + \beta_7 \texttt{CalYear}_{2010} + \beta_8 \texttt{Adind}_0 + f_1(\texttt{Age}) + f_2(\texttt{Density}) + f_3(\texttt{Age}, \texttt{Gender})$$

Où f_3 est la fonction de lissage pour le terme d'interaction. Afin de s'assurer de la significativité des variables, nous effectuons un test de significativité à l'aide de la fonction `anova.gam` dont le résultat est présenté ci-dessous :

```
Parametric Terms:
       df Chi.sq  p-value
Gender  1  96.98  < 2e-16
Type    5 139.09  < 2e-16
Adind   1  12.82 0.000343

Approximate significance of smooth terms:
                        edf  Ref.df   Chi.sq p-value
s(Age)              7.73882 9.00000  179.771  <2e-16
s(Density)          7.11133 9.00000 1567.186  <2e-16
s(Age):GenderMale   4.33152 9.00000   32.422  <2e-16
s(Age):GenderFemale 0.07609 9.00000    0.049   0.012
```

On peut s'apercevoir au vue du résultat R ci-dessus et de l'annexe B.2 que toutes les variables du modèle sont significatives. La figure 4.1.1 montre l'impact des différentes variables dans le modèle GAM, le but est de s'assurer de l'importance spécifique des variables dans le modèle tandis que la figure 4.1.2 montre ce même impact en fonction des variables `Age` et `Gender`. On constate directement que les jeunes conductrices sont moins risquées que leurs homologues masculins, contrairement aux conductrices plus âgées (plus de 70 ans) qui sont plus risquées que leurs homologues masculins. On constate également que la répartition du nombre de sinistre prédit en fonction de la densité de la ville du conducteur est différente de celle obtenue à l'aide du package insurancerating, celle-ci est moins stable et ici, les assurés vivant dans la ville la moins peuplée ne sont plus ceux ayant le nombre

de sinistre estimé le plus faible.

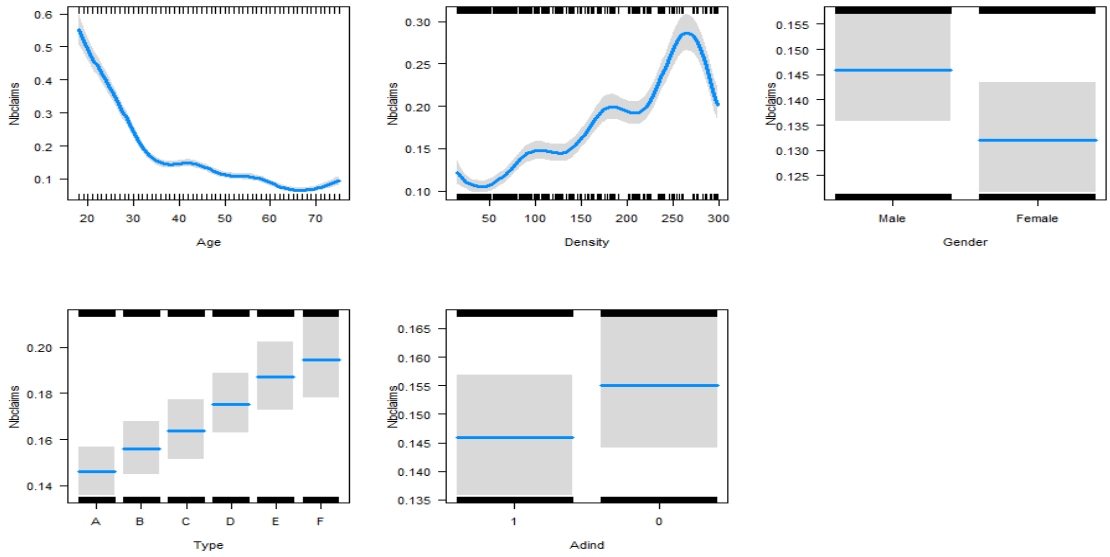


FIGURE 4.1.1: Différents effets du modèle GAM

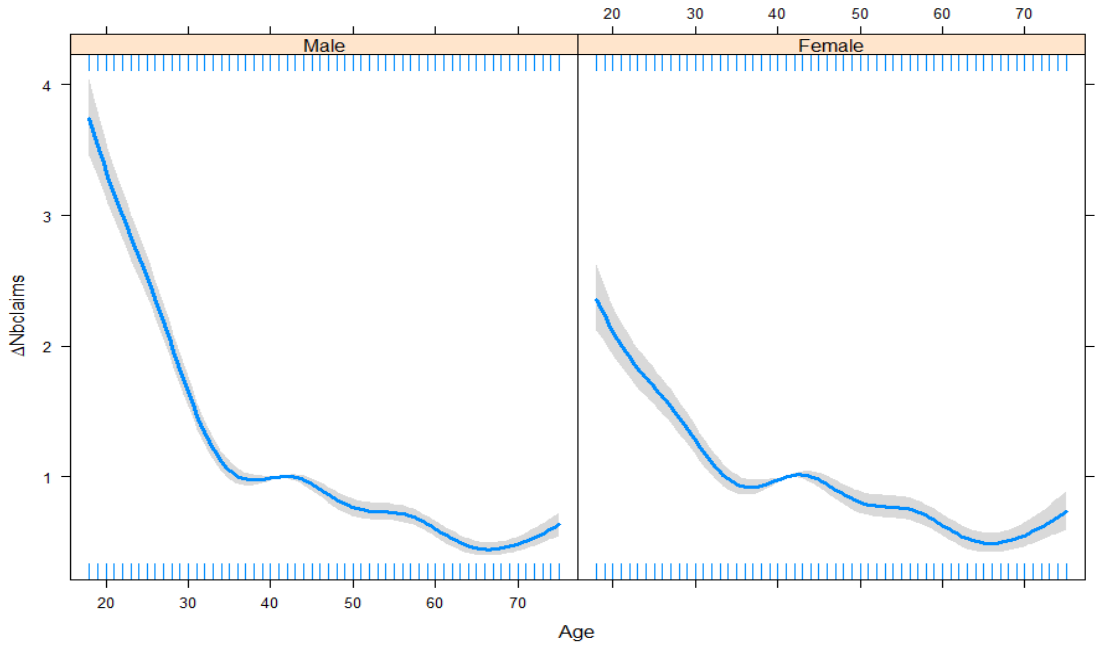


FIGURE 4.1.2: Effets de l'interaction entre le sexe et le genre

La figure 4.1.3 présente les fonctions de lissage $\hat{f}_1$ pour l'âge du conducteur, $\hat{f}_2$ pour la densité de la population, et $\hat{f}_3$ pour l'interaction, le graphique inférieur gauche pour les

hommes et celui de droite pour les femmes. Les traits interrompus représentent l'intervalle de confiance à 95%. Comme communément connu dans ce domaine, les jeunes conducteurs sont les plus risqués, ce qui peut s'expliquer par leur façon de conduire qui est dû à leur manque d'expérience, ce risque diminue avec l'âge pour se stabiliser vers 35 ans. Ensuite il augmente légèrement sans doute dû au fait que les enfants des assurés (entre la fin de la quarantaine et au début de la cinquantaine) commencent à conduire les véhicules de leurs parents. Après l'âge de 50 ans, ce risque recommence à diminuer jusqu'à l'âge de 65 ans, après lequel il augmente encore peut-être dû au fait que les conducteurs les plus âgés ont une santé fragile (problème de vue, vieillesse...etc) et ont moins de réflexes au volant. Cela implique que les personnes les plus âgées dans le portefeuille déclarent plus d'accidents de voiture en vieillissant, le léger élargissement de l'intervalle de confiance pour ceux-ci s'explique par le fait qu'ils sont moins présent dans le portefeuille par rapport à d'autres tranches d'âge. Comme cela a été constaté dans le chapitre précédent lors de l'usage du package insurancerating (bien que n'incluant que la densité comme variable explicative), le nombre de sinistres déclarés augmente avec la densité de la population. Globalement, les assurés résidant dans les villes les moins peuplées déclarent moins de sinistre, car dans les villes les moins peuplées, le trafic de circulation est sans doute moins dense, mais dans certains cas, cela peut inciter les conducteurs à des excès de vitesse et à l'imprudence aboutissant à des accidents. On remarque également une stabilisation du risque entre les villes dont la densité de la population est comprise entre 30 à 50 habitants/km^2, 100 à 130 habitants/km^2, 210 à 230 habitants/km^2.
L'effet d'interaction entre l'âge du conducteur et le genre est représenté par les 02 graphiques inférieurs de la figure 4.1.3, ici les valeurs sur l'axe des ordonnées représentent les différents termes correctifs sur le nombre de sinistre déclarés en fonction de l'âge et du genre du conducteur. Étant donné l'allure de ces deux graphiques, on constate que le terme correctif est presque nul pour toutes les femmes, positif pour les hommes ayant un âge maximal de 35 ans, très proche de zéro pour les hommes ayant un âge compris entre 35 et 65 ans et négatif pour la tranche restante. Ce qui corrobore les observations précédentes : Les jeunes hommes sont plus risqués que les jeunes femmes et les vieilles femmes sont plus risquées que les vieux hommes.

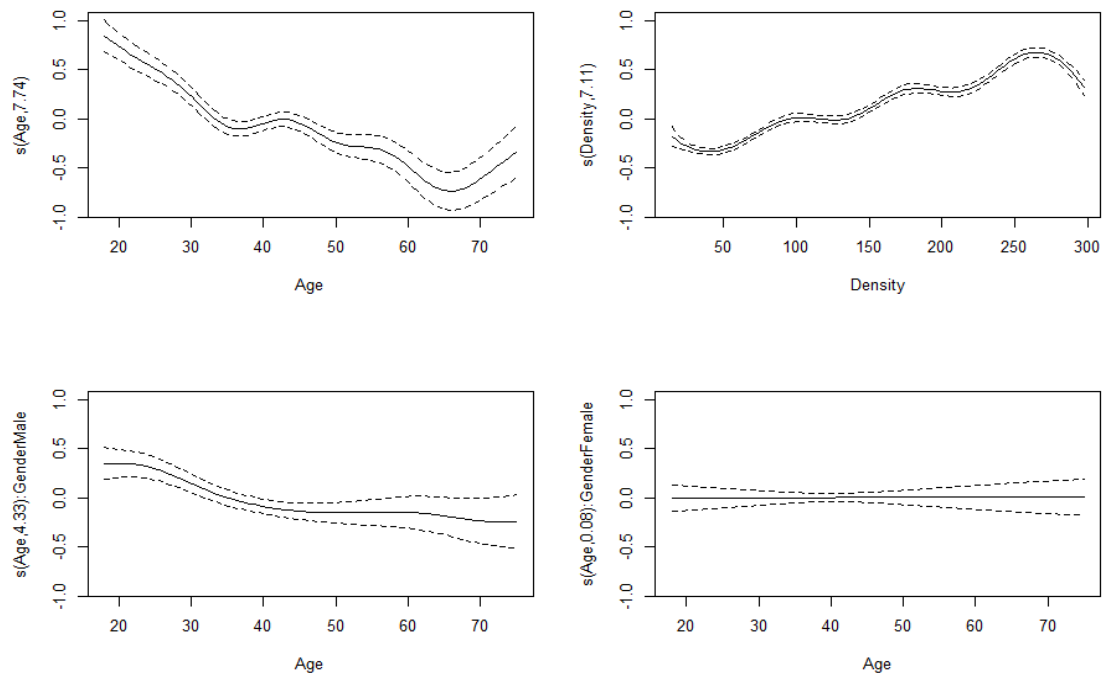


FIGURE 4.1.3: Effets de lissage du modèle GAM

4.1.2 Ajustement de l'effet marginal sur la variable continue dans un arbre évolutif

Ici le but est de construire des classes de risque pour les covariables du modèle en se servant des différents effets de lissage marginaux : $\hat{f}_1(\texttt{Age})$, $\hat{f}_2(\texttt{Density})$ et $\hat{f}_3(\texttt{Age, Gender})$ obtenus dans le modèle GAM complet. La commande R ci dessous est celle nous permettant d'extraire de notre résultat GAM ces différents effets de lissage.

```
f_chapo<-data.frame(predict.gam(GAM,type="terms"))
```

L'objectif de notre approche reste le même que celui du package insuranceting, c'est-à-dire construire des classes où les valeurs consécutives des facteurs de risques continus sont regroupées ensemble. Ainsi, en se basant du procédé du package insurancerating, on utilise un arbre évolutif comme technique de segmentation car ces modèles produisent des fractionnements intuitifs en ligne avec notre exigence de regroupement de valeurs consécutives des variables continues ([Henckaerts et al., 2018]).
On construit un arbre de régression pour chaque effet de lissage ajusté, avec l'effet de lissage moyen comme variable expliquée et la covariable comme variable explicative. L'effet de lissage moyen dans le cas de l'âge du conducteur qu'on ajuste ici est un effet agrégé par rapport à l'âge du conducteur et corrigé, c'est-à-dire l'effet principal sans tenir compte du sexe (graphique supérieur gauche) auquel est ajouté le terme correctif en fonction du

genre (graphique inférieur gauche pour les hommes et celui de droite pour les femmes). Il est important de prendre en compte la composition du portefeuille lorsqu'on décide de segmenter les facteurs de risques. Pour les assurés de plus de 70 ans, l'effet de lissage ajusté augmente (voir figure 4.1.3), mais la figure 3.1.2 montre que le portefeuille ne contient pas beaucoup d'assurés de cette tranche d'âge. Il serait inapproprié d'obtenir plusieurs classes dans cette tranche d'âge, car ça ne concerne qu'une petite portion du portefeuille. Ainsi, la distribution des assurés au sein d'un facteur de risque spécifique est prise en compte en utilisant le nombre d'assurés comme le poids (pris en argument dans la fonction `evtree`). Le tableau ci-dessous montre un extrait de données d'entrée pour l'ajustement dans un arbre évolutif de l'effet de l'âge $\hat{f}_1(\texttt{Age})$ sur la covariable `Age`. Par exemple, pour l'effet de lissage entre $\texttt{Age} = 18$ et $\hat{f}_1(\texttt{Age})$, obtient un poids entier de 1 352.
L'arbre évolutif interprétera ce poids comme si la paire $(\texttt{Age} = 18, \hat{f}_1(18))$ se répète 1 352 fois dans le jeu de données.

Covariable : `Age`	Réponse : $\hat{f}_1(\texttt{Age})$	Poids :ω
18	1.0495	1 352
19	0.9898	1 352
20	0.93386	1 472

TABLE 4.1.1: Extrait des données d'entrée pour l'arbre évolutif qui regroupe $\hat{f}_1(\texttt{Age})$

Pour évaluer la performance (précision) de l'arbre, on utilise la fonction d'évaluation suivante :

$$n.log(MSE) + 4.\alpha.(m+1).log(n)$$

où n est le nombre d'observation dans le portefeuille, m le nombre de noeuds terminaux dans l'arbre, et α le paramètre de réglage, pour plus de détails, voir [Grubinger et al., 2014]. Notons que n est égale à la somme des poids, puisqu'un arbre interprète un poids comme étant le nombre de fois que le couple (réponse, covariable) apparaît dans les données d'entrée. Le premier terme de la fonction d'évaluation mesure la précision de l'arbre via l'erreur quadratique moyenne (MSE), pour $\hat{f}_1(\texttt{Age}_i)$, le MSE est donné par :

$$MSE = \frac{\sum_{i=min(Age_i)}^{max(Age_i)} \omega_{Age_i} (\hat{f}_1(\texttt{Age}_i) - \hat{f}_1^b(\texttt{Age}_i))^2}{\sum_{i=min(Age)}^{max(Age)} \omega_{Age_i}}$$

avec ω_{Age_i} le nombre d'assurés avec $\texttt{Age} = i$ et $\hat{f}_1^b$ la valeur obtenue à partir de l'arbre de régression, cette valeur est obtenue comme la moyenne de $\hat{f}_1(\texttt{Age}_i)$ sur tous les assurés de la tranche d'âge à laquelle $\texttt{Age} = i$ appartient.
Le second terme de cette fonction d'évaluation représente une pénalité sur la complexité de l'arbre, mesurée par m le nombre de noeuds terminaux, ou chaque noeud représente

une classe, le paramètre de réglage α prend en charge le compromis entre précision et complexité.
On considère un ensemble de valeurs possibles pour le paramètre de réglage α, le nombre d'arbre généré ainsi que le nombre d'itérations maximal après lequel la compilation s'arrête. Pour le paramètre de réglage α, on choisit l'ensemble {0,1,5,..., 50,60,..., 100,150,...,950}, pour le nombre d'arbre on choisit l'ensemble {200,500}, et pour le nombre d'itération, on choisit {3000,5000,10000}. On teste 204 différentes combinaisons et on retient celle minimisant la fonction d'évaluation. Pour le cas de l'âge du conducteur, après plus de cinq heures de compilation, la combinaison retenue est celle pour laquelle le paramètre de réglage α_{Age} =700, le nombre d'arbre est 500 et le nombre d'itération est de 3000. Ci-dessous l'extrait du code R que nous avons utilisé.

```
alfa<-c(0,1,seq(5,50,by=5),seq(60,100,by=10),seq(150,950,by=50))
tree<-c(200,500)
n_iteration<-c(3000,5000,10000)
n<-nrow(training)

table<-matrix(NA,ncol=4,nrow=length(alfa)*length(tree)*length(n_iteration
    ))
colnames(table)<-c("Alpha","tree","iteration","Mesure_tree")

m<-1
for(i in 1 : length(alfa)){
  for(j in 1 : length(tree)){
    for(k in 1 : length(n_iteration)){


      tree_x <-evtree(data_age$f_age ~ data_age$Age, data = data_age ,
    weights=data_age$poids,
                      control =evtree.control(alpha=alfa[i],ntrees=tree[j
    ],niterations=n_iteration[k],seed=1))

      y<-data_age[,2]
      u<-predict(tree_x, newdata=data_age, type = "response")
      m<-width(tree_x)
      MSE<-sum(data_age$poids*(y-u)^2)/sum(data_age$poids)
      Mesure_tree<-n*log(MSE)+4*alfa[i]*(m+1)*log(n)
      table[m,1]<-alfa[i]
      table[m,2]<-tree[j]
      table[m,3]<-n_iteration[k]
      table[m,4]<-Mesure_tree

      m<-m+1
    }

  }

}
which.min(table[,4])
```

```
35
36 tree_age <-evtree(data_age$f_age ~ data_age$Age, data = data_age ,
       weights=data_age$poids,
37                  control =evtree.control(alpha=700,ntrees=500,
       niterations=3000,seed=1))
```

Pour la variable `Density`, on exécute le même procédé afin d'obtenir les paramètres optimaux, et il en ressort que : le paramètre de réglage optimal est $\alpha_{Density}$=650, le nombre d'arbre est 200, et le nombre d'itération est de 3000.
Les paramètres optimaux ainsi obtenu, on construit les arbres régression afin de déterminer les différents points de rupture qui nous permettrons d'effectuer la segmentation. La figure 4.1.4 nous montre l'arbre de régression entre l'effet de lissage de l'âge sur la covariable `Age` obtenu à partir du mécanisme d'arbre évolutif, tandis que la figure 4.1.5 montre celui entre l'effet de lissage de la densité de la population de la ville où habite le conducteur et la covariable `Density`. Ces figures montrent qu'ils se forment six groupes homogènes au sein de la variable `Age` qui pourra être catégorisée en fonction des points de rupture dans l'arbre, et pour la variable `Density`, il se forme cinq classes, bien qu'étant le même nombre de classe que celui obtenu à l'aide du package, celles-ci sont plus équilibrées en terme d'effectifs, dans le sens où chaque noeud terminal contient plus des 5% des observations d'entrée.

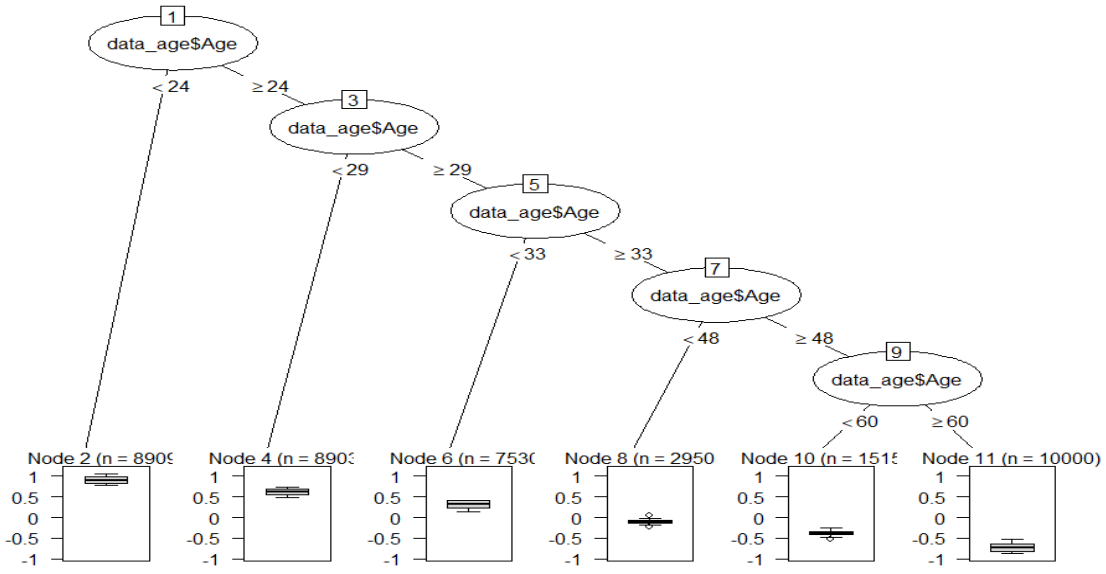


FIGURE 4.1.4: Arbre de régression pour l'effet de lissage de l'âge du conducteur

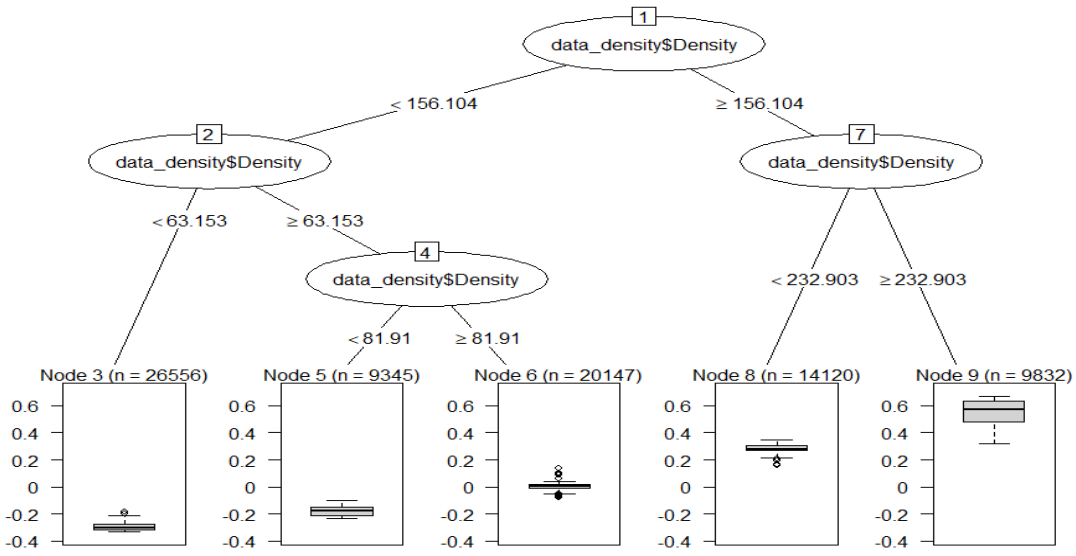


FIGURE 4.1.5: Arbre de régression pour l'effet de lissage de la densité de la ville du conducteur

4.1.3 Analyse des classes obtenues

Les arbres de régression ayant permis d'obtenir les points de rupture au sein des variables `Age` et `Density` nous permettent de segmenter manuellement ces covariables comme montré sur la figure 4.1.6.

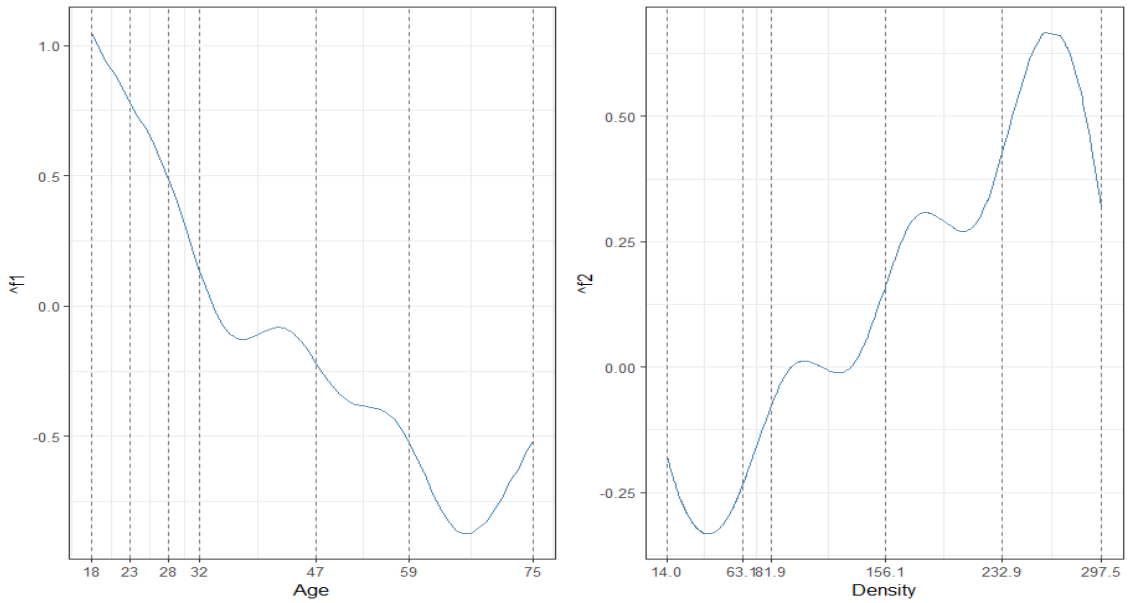


FIGURE 4.1.6: Classification sur les effets marginaux

La figure 4.1.4 nous montre que l'effet marginal $\hat{f}_1(\texttt{Age})$ est divisé en six classes comme dans le cas du package insurancerating tandis que la figure 4.1.6 montre les différentes divisions produites par les arbres évolutifs préférés. Sur le premier graphique, on constate que les assurés ayant un âge compris entre `(32, 47]` sont groupés tous ensemble, ce qui n'est pas très surprenant car graphiquement, $\hat{f}_1(\texttt{Age})$ est relativement stable sur cet intervalle. Les assurés les plus jeunes ont un profil de risque plus élevé et trois classes sont formées pour les assurés âgés de moins de 33 ans : `[18, 23]`, `(23, 28]`, `(28, 32]`. L'effet de lissage $\hat{f}_1(\texttt{Age})$ se stabilise légèrement aux alentours de l'âge de 49 ans et recommence à croître après l'âge de 60 ans, ce qui conduit à la formation de deux classes, `(47, 59]` pour la région de stabilisation et `(59, 75]` pour les assurés les plus âgés.
Sur le deuxième graphique de la figure 4.1.6, l'effet marginal $\hat{f}_2(\texttt{Density})$ est divisé en cinq classes, moins que dans le cas du package insurancerating, une première classe `[14, 63.1]`, la plus peuplée où graphiquement la fréquence de sinistre est plus faible en moyenne ; moins il y a de trafic moins c'est risqué. La classe la moins peuplée est `(63.1, 81.9]`, dans laquelle on observe une croissance de la courbe de l'effet marginal. Les autres classes sont `(81.9, 156.1]`, `(156.1, 233]`, `(233, 298]`.
Les résultats obtenus à partir des arbres évolutifs qui sont des arbres globalement optimaux, nous permettent de transformer les variables continues qui sont `Age` et `Density` en variables catégorielles qu'on nomme comme dans le chapitre précédent `Age_class` et `Density_class` respectivement pour l'âge du conducteur et la densité de la ville d'habitation de ce dernier. Le tableau ci-dessous présente les différentes classes obtenues et leurs effectifs.

`Age_class`	effectif	`Density_class`	effectif
[18, 23]	8 909	[14, 63.1]	26 556
(23, 28]	8 903	(63.1, 81.9]	9 345
(28, 32]	7 530	(81.9, 156.1]	20 147
(32, 47]	29 503	(156.1, 233]	14 120
(47, 59]	15 155	(233, 298]	9 832
(59, 75]	10 000		
Total	80 000	Total	80 000

TABLE 4.1.2: Les segments construits à l'aide du modèle GAM complet

Une bonne construction de classe doit aboutir à des segments contenant au minimum 5% de l'ensemble du portefeuille ([Henckaerts et al., 2018]). Pour ce qui est de notre cas, on constate que chaque classe construite est assez représentative du portefeuille, car elles contiennent au minimum 9% du portefeuille d'entraînement utilisé. Dans ce sens, ces classes sont plus optimales que celles obtenues à l'aide du package Insurancerating, où certaines d'entre elles contenaient moins de 5% de l'ensemble du portefeuille.

Cette catégorisation ainsi effectuée, nous intégrons ces variables dans un modèle GLM afin d'estimer le nombre de sinistre de chaque individu du portefeuille et d'analyser les différents segments qui en ressortent.

4.2 Segmentation et modèle GLM

Il s'agit dans cette section d'implémenter un modèle GLM avec les nouvelles variables catégorisées (`Age_class`, `Density_class`) et toutes autres variables pertinentes à fin de ressortir comme dans le chapitre précédent les différents segments globaux de fréquence de sinistre annuelle qui se forment. Notre approche suppose que les variables à intégrer dans le modèle GLM soient les mêmes que celles utilisées dans le modèle GAM développé ([Henckaerts et al., 2018]) dans la section précédente, bien évidemment, les variables continues sont désormais catégorielles. On définit ainsi les différents niveaux de référence comme ceux étant les plus exposés avant l'exécution du modèle. Nous exécutons le modèle GLM via la commande R suivante :

```
GLM<-glm(formula = Nbclaims~Age_class+Density_class+Type+Adind+Gender+
   Age_class*Gender, family = poisson(link = "log"), data = training,
   offset = log(Expo))
```

Le résultat obtenu après exécution du code R est présenté ci-dessous :

```
Coefficients:
                              Estimate Std. Error z value Pr(>|z|)
(Intercept)                   -2.20547    0.02819 -78.225  < 2e-16 ***
Age_class[18,23]               1.17414    0.02687  43.697  < 2e-16 ***
Age_class(23,28]               0.83900    0.02910  28.832  < 2e-16 ***
Age_class(28,32]               0.43781    0.03479  12.585  < 2e-16 ***
Age_class(47,59]              -0.31774    0.03452  -9.204  < 2e-16 ***
Age_class(59,75]              -0.66040    0.04533 -14.569  < 2e-16 ***
Density_class(63.1,81.9]       0.09866    0.03113   3.170 0.001526 **
Density_class(81.9,156]        0.30392    0.02311  13.150  < 2e-16 ***
Density_class(156,233]         0.58560    0.02358  24.835  < 2e-16 ***
Density_class(233,298]         0.84611    0.02414  35.047  < 2e-16 ***
TypeB                          0.06576    0.02371   2.774 0.005544 **
TypeC                          0.11760    0.02683   4.384 1.17e-05 ***
TypeD                          0.18593    0.02375   7.827 4.99e-15 ***
TypeE                          0.24469    0.02759   8.869  < 2e-16 ***
TypeF                          0.29358    0.03520   8.341  < 2e-16 ***
Adind0                         0.06972    0.01690   4.125 3.70e-05 ***
GenderFemale                  -0.10382    0.03116  -3.331 0.000864 ***
Age_class[18,23]:GenderFemale -0.46190    0.04726  -9.775  < 2e-16 ***
Age_class(23,28]:GenderFemale -0.33260    0.05107  -6.512 7.41e-11 ***
Age_class(28,32]:GenderFemale -0.23369    0.06131  -3.811 0.000138 ***
Age_class(47,59]:GenderFemale  0.07446    0.05947   1.252 0.210542
Age_class(59,75]:GenderFemale  0.07668    0.08241   0.930 0.352132
```

Premièrement, on constate que toutes les variables sont significatives, les coefficients estimés sont également significatifs, à l'exception des différents coefficients estimés pour les termes d'interaction des femmes âgés de plus de 32 ans. Deuxièmement, sachant que les assurés de sexe masculin sont plus risqués, les assurées de sexe féminin, particulièrement les plus jeunes bénéficient d'un terme de correction qui est négatif du fait de leur faible sinistralité par rapport aux hommes. Pour ce qui est de celles âgées de plus de 47 ans, ce terme correctif bien que n'étant pas significatif, est positif et semble montrer que ces assurées sont plus risquées que les assurés de sexe masculin. On rappelle que le coefficient estimé pour le terme d'interaction entre la tranche d'âge `(32, 47]` (niveau de référence) et le genre féminin est nul.
Nous utilisons également la fonction `visreg` du package R **visreg** pour visualiser l'apport individuel de chacune des variables et leurs niveaux respectifs dans le modèle.

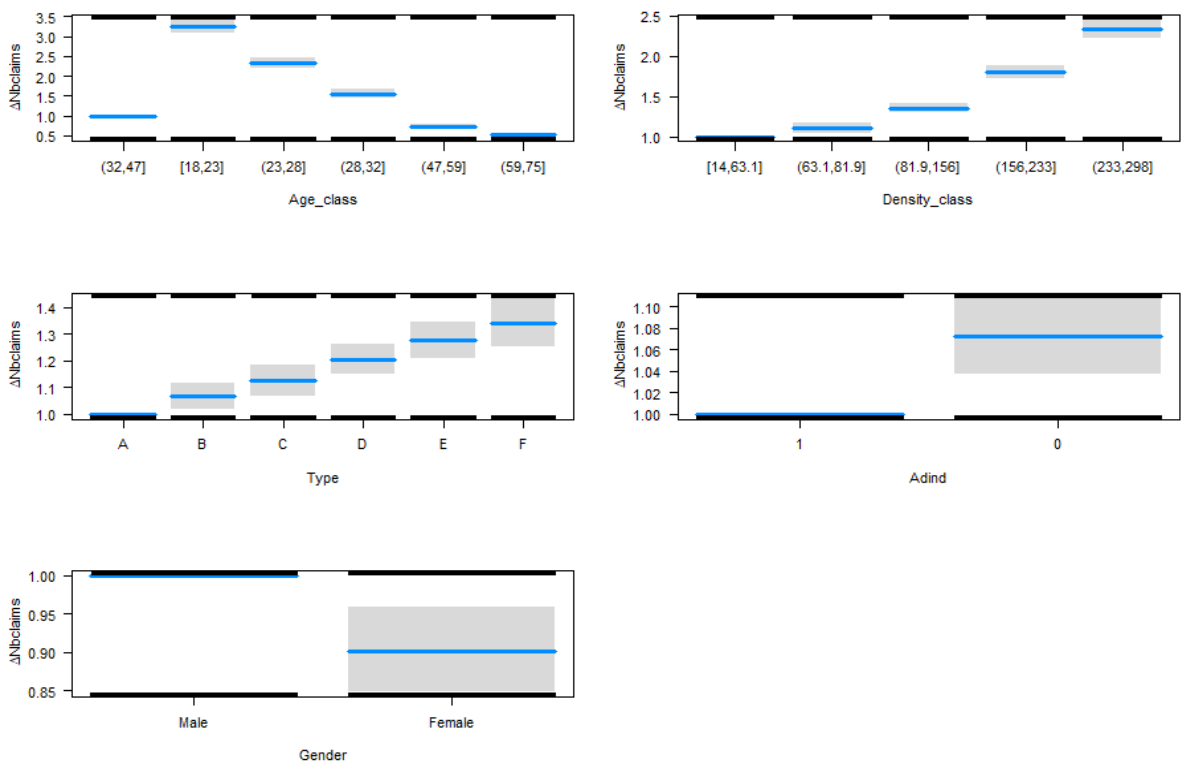


FIGURE 4.2.1: Importance des différentes catégories des variables du modèle GLM

En observant les graphiques de la figure 4.2.1, on constate pour la variable `Age_class` que les niveaux `(47, 59]` et `(59, 75]` ont sensiblement une importance similaire, ce qui est également le cas pour la variable `Density_class` et ses différents niveaux `[14, 63.1]` (niveau de référence) et `(63.1, 81.9]`. Pour chacune des variables mentionnées, ces deux catégories peuvent être regroupées en une seule dans un nouveau modèle GLM dont on

comparera au précédent afin d'en retenir un seul. Notre objectif étant de comparer les résultats issus directement de notre approche à ceux issus du package insurancerating, nous choisissons de nous abstenir de ce regroupement qui plus est émettrait déjà un doute sur l'optimalité globale des arbres évolutifs.

4.2.1 Création des groupes homogènes au sein du portefeuille TPL

Le but dans cette section est de pouvoir ressortir la fréquence de sinistre annuelle prédite au sein du portefeuille connaissant les différentes caractéristiques de l'assuré. Premièrement, pour obtenir la fréquence de sinistre annuelle prédite par le modèle GLM, on divise le nombre de sinistre estimé (`GLM[["fitted.values"]]`) par la durée d'exposition dans le portefeuille. Dans le but de mieux appréhender la segmentation multivariée que nous voulons effectuer, on décide de créer comme dans le chapitre précédent des variables binaires correspondant aux différents niveaux des différentes variables binaires prenant les valeurs `yes` ou `no` en fonction du fait qu'un individu appartienne ou pas à une catégorie. C'est ainsi qu'on utilise 16 variables binaires pour construire le tableau de segmentation multivariée dont les 30 premières lignes sont présentées à l'annexe B.4 de ce mémoire. Le tableau de segmentation construit contient 2 710 segments au total, autrement dit, il existe dans notre portefeuille d'entraînement sur la base du modèle implémenté 2 710 fréquences de sinistre estimées pour les 80 000 individus de notre portefeuille, ceci est issu des différentes combinaisons des niveaux des différentes variables utilisées dans notre modèle GLM (voir section 4.2.1). Ce nombre de segments est inférieur à celui obtenu à l'aide du package insurancerating, ce qui peut s'expliquer par le fait d'avoir obtenu pour la variable `Density` cinq classes contrairement à six comme dans l'approche précédente, ce nombre inférieur pourrait traduire également une homogénéité moins prononcée au sein des différents segments globaux et aussi une faible hétérogénéité entre ces différentes classes.
En observant sur le même graphe la courbe des fréquences annuelles observées et celle des fréquences annuelles estimées comme présenté sur la figure 4.2.2, on constate pour le cas de l'âge du conducteur bien évidemment que les deux courbes ont la même allure et du fait de la catégorisation de la variable `Age_class`, les fréquences prédites sont regroupées telles quelles ont été segmentées, les six classes préalablement formées sont très bien visibles graphiquement. Pour le cas de la variable `Density_class` dans les données du portefeuille d'entraînement, les observations faites sont analogues à celles faites pour la variable `Age_class`, la fréquence de sinistre prédite reste regroupée par tranche de densité de la population, une fois de plus ce résultat montre que les classes `[14, 63.1]` et `(63.1, 81.9]` sont très proches l'une de l'autre et semblent ne former qu'une seule classe. Dans le cas du package, le regroupement par classes de la répartition de la fréquence de sinistre estimée en fonction de la densité a la même allure.

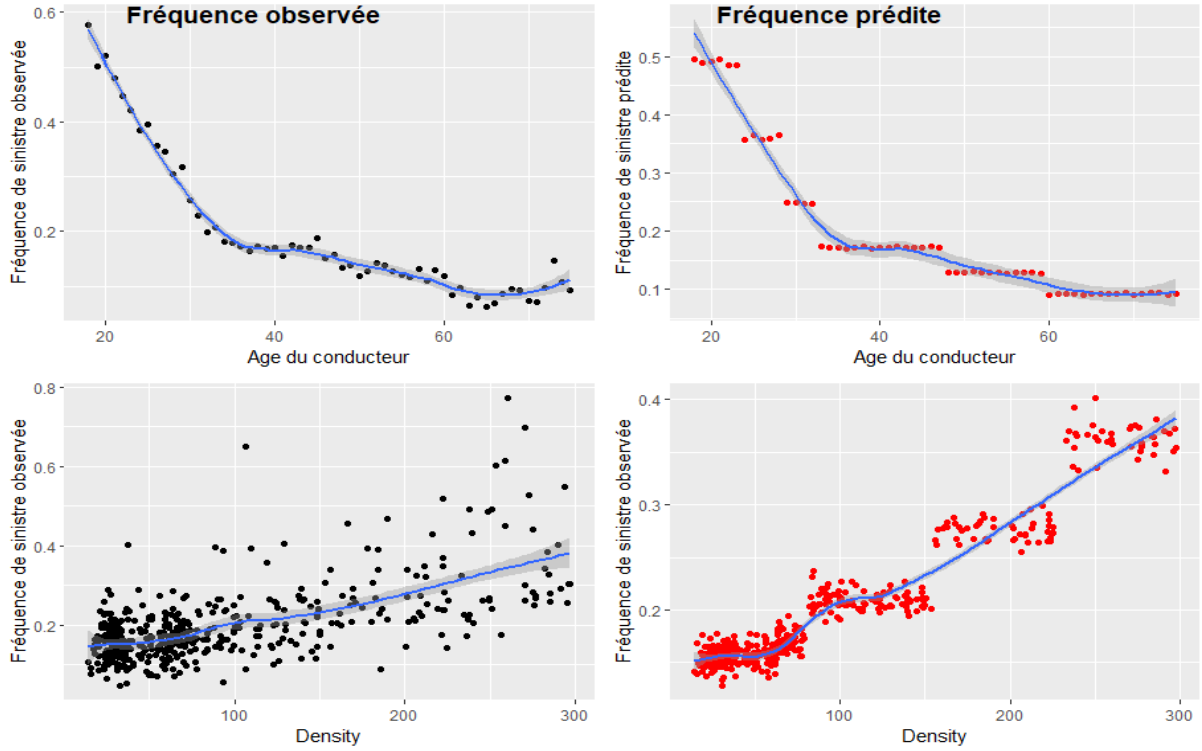


FIGURE 4.2.2: Fréquences observées et fréquences prédites au sein du portefeuille entier

La visualisation des fréquences estimées par le modèle dans l'ensemble du portefeuille ne donne pas vraiment d'information sur la qualité de la segmentation effectuée. Au sein des 2 710 segments globaux obtenus, visualiser ces fréquences là apporterait plus d'information étant donné l'objectif de ce mémoire. La figure 4.2.3 nous montre donc la répartition des fréquences de sinistres observées et prédites au sein des 2 710 segments globaux. Pour le cas de l'âge du conducteur (graphique supérieur gauche), on remarque que même au sein de ces 2 710 segments globaux, la fréquence de sinistre prédite reste regroupée par classe d'âge, même si elles sont moins uniformes que sur le graphique de la figure 4.2.2, on visualise les six classes préalablement constituées, comme dans le cas de la segmentation effectuée à l'aide du package insurancerating. Pour le cas de la densité de la population, la répartition de la fréquence de sinistre reste toujours aussi éparse que ce soit dans le cas de la segmentation effectuée à l'aide du package ou celle effectuée ici à l'aide d'un modèle GAM complet. Ce qui peut s'expliquer par le nombre élevé de niveaux (471) de la variable `Density`.
Toutefois, on se rend compte que les sous groupes formés dans ce cas ci semblent aussi homogènes que ceux obtenus à l'aide du package insurancerating.

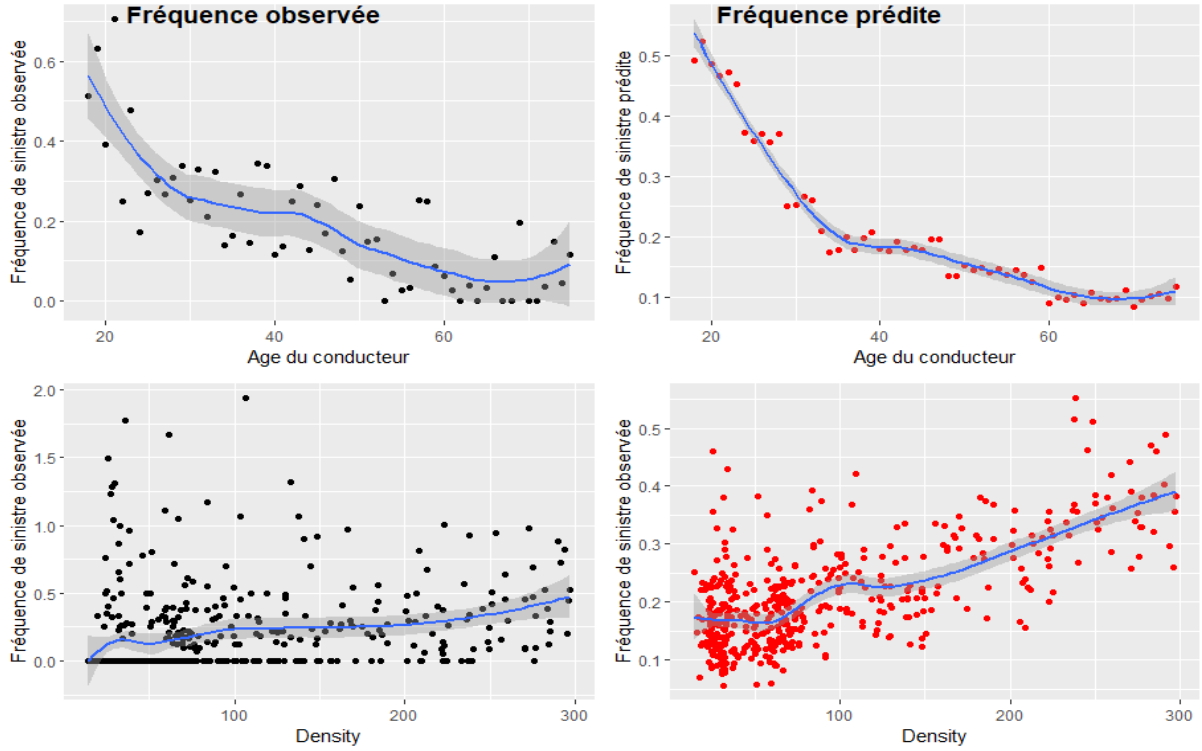


FIGURE 4.2.3: Fréquences observées et fréquences prédites au sein des 2 710 classes

Dans les deux approches effectuées, les différentes classes construites sont très significatives, ainsi, pour mieux juger de la qualité d'ajustement des modèles GLMs basés sur les différentes classes issues des différentes approches, nous utiliserons une base de données "test" fournie à cet égard.

4.2.2 Mesure de la qualité de l'ajustement des différents modèles GLMs

Le but de cette section est de comparer la qualité d'ajustement des différents modèles GLMs. Pour ce la on utilisera la base de données de test, représentant les 20% du portefeuille de base (mentionnée à la section 3.1) mis de coté avant l'implémentation des différents modèles GAMs et GLMs en vue d'appliquer les deux approches sur celle-ci et d'en tirer des conclusions sur les segmentations effectuées.
Les différents critères que nous utiliserons pour mesurer l'ajustement des deux modèles GLMs sont : AIC, BIC et déviance de Poisson (voir section 2.4.2.2). On rappelle que pour chacun de ses critères, une valeur élevée indique une faible qualité d'ajustement du modèle. Pour chacun des différents modèles GLMs, le tableau ci-dessous présente les valeurs de ces différents critères d'ajustement.

Modèle GLMs	AIC	BIC	Déviance de Poisson
insurancerating	79 688.78	79 902.44	**25 061.93**
GAM complet	**79 579.27**	**79 783.65**	25 374.67

TABLE 4.2.1: Mesure de la qualité d'ajustement des modèles GLMs

Pour les critères classiques qui sont AIC et BIC, le modèle GLM basé sur la segmentation effectuée à partir d'un modèle GAM complet a une meilleure qualité d'ajustement que le modèle GLM issu du package insurancerating. Pour ce qui est de la déviance de Poisson, le modèle issu du package est meilleur que celui issu du modèle GAM complet. Ainsi les conclusions tirées à ce niveau sont fonction du critère principal pour mesurer la qualité d'ajustement des modèles.

Chapitre 5

Conclusion

Dans cette étude, il était question d'explorer un portefeuille à fin de construire des classes tarifaires à l'aide du package R insurancerating. De façon concrète, il s'agissait d'utiliser le package pour transformer les variables continues de notre portefeuille à fin de les intégrer dans un modèle GLM. Nous avons débuté notre travail en effectuant l'analyse descriptive de notre base de données, qui est une étape importante dans la modélisation des risques car elle fournit des informations sur les individus constituant notre portefeuille. Par la suite, notre base de données a été scindée en deux parties différentes à savoir, une base d'entraînement sur laquelle nos différents modèles ont été implémentés et une base de validation et/ou de test permettant de comparer différents modèles.

Le but de ce mémoire était d'évaluer l'outil de segmentation qu'est le package R insurancerating, l'approche du package étant basé sur des modèles GAMs à une seule covariable dont le but est de les segmenter en ajustant les fonctions de lissage individuels sur la covariable elle-même dans un arbre évolutionnaire, nous avons décider de comparer les résultats du package à ceux basés sur un unique modèle GAM complet, c'est-à-dire incluant toutes les variables jugées pertinentes, dans cette approche comparative, ce sont les effets de lissage marginaux qui sont cette fois-ci ajusté sur la covariable dans un arbre évolutionnaire tout en tenant compte des poids.

Il ressort de cette étude que les deux approches donnent des résultats tout à fait acceptables, raisonnables, et quelques peu similaires en termes de stabilité des résultats au sein des segments globaux obtenus. Pour ce qui est du nombre de classe créée, le package insurancerating aboutit à six classes pour les deux covariables utilisées à savoir l'âge du conducteur et la densité de la population de la ville d'habitation du conducteur, tandis l'approche du modèle GAM complet elle aboutit à respectivement six et cinq classes respectivement pour chacune des deux covariables.

Etant donné l'objectif du package qui est d'intégrer les covariables préalablement catégoriser dans un modèle GLM, nous avons également mesurer la qualité de prédiction des différents modèles GLMs. On obtient sur la base des mesures d'ajustement classiques telles que AIC et BIC que le modèle GLM basé sur les résultats de catégorisation du modèle GAM complet est meilleur que celui basé sur les résultats du package R insurancerating. Mais cependant, en utilisant la déviance de Poisson les préférences sont inversées et le modèle GLM basé

sur les résultats du package est meilleur car ayant une déviance de Poisson plus petite. Il est important de préciser que le paramétrage du package ayant conduit à la non pertinence de la variable `Poldur` représentant l'âge de la police nous a poussé à exclure cette variable du modèle GAM complet qui s'avère pourtant pertinente ici, ceci dans le soucis d'avoir une bonne base de comparaison. L'absence de ce compris dans ce travail aurait sans doute permit d'obtenir de résultats plus concluant. De plus un travail complémentaire serait d'utiliser le Bagging et/ou le Boosting pour améliorer la précision des arbres de régression évolutionnaires.

Bibliographie

[Antonio and Valdez, 2012] Antonio, K. and Valdez, E. A. (2012). Statistical concepts of a priori and a posteriori risk classification in insurance. *AStA Advances in Statistical Analysis*, 96(2) :187–224.

[Charpentier and Denuit, 2005] Charpentier, A. and Denuit, M. (2005). Mathématiques de l'assurance non-vie. *Economica, Paris*.

[Clijsters, 2015] Clijsters, M. (2015). Dealing with continuous variables and geographical information in non-life insurance ratemaking : practical solutions applied to a car insurance data set. *KU Leuven master thesis*.

[Denuit et al., 2004] Denuit, M., Charpentier, A., and Bébéar, C. (2004). *Mathématiques de l'Assurance Non-Vie. Tome I : Principes Fondamentaux de Théorie du Risque*.

[Denuit et al., 2007] Denuit, M., Maréchal, X., Pitrebois, S., and Walhin, J.-F. (2007). *Actuarial modelling of claim counts : Risk classification, credibility and bonus-malus systems*. John Wiley & Sons.

[Denuit et al., 2019] Denuit, M., Trufin, J., and Hainaut, D. (2019). Effective statistical learning methods for actuaries.

[Duchon, 1977] Duchon, J. (1977). Splines minimizing rotation-invariant semi-norms in sobolev spaces. In *Constructive theory of functions of several variables*, pages 85–100. Springer.

[Eiben et al., 2003] Eiben, A. E., Smith, J. E., et al. (2003). *Introduction to evolutionary computing*, volume 53. Springer.

[Fahrmeir et al., 2013] Fahrmeir, L., Kneib, T., Lang, S., and Marx, B. (2013). Regression models. In *Regression*, pages 21–72. Springer.

[Fisher, 1958] Fisher, W. D. (1958). On grouping for maximum homogeneity. *Journal of the American statistical Association*, 53(284) :789–798.

[Fogel et al., 1966] Fogel, L., Owens, A., and Walsh, M. (1966). Artificial intelligence through sim-ulated evolution, j. wiley & sons. *Inc., London, New York, Sidney*.

[Grubinger et al., 2014] Grubinger, T., Zeileis, A., and Pfeiffer, K.-P. (2014). evtree : Evolutionary learning of globally optimal classification and regression trees in r. *Journal of statistical software*, 61(1) :1–29.

[Hastie and Tibshirani, 1990] Hastie, T. and Tibshirani, R. (1990). Exploring the nature of covariate effects in the proportional hazards model. *Biometrics*, pages 1005–1016.

[Henckaerts et al., 2018] Henckaerts, R., Antonio, K., Clijsters, M., and Verbelen, R. (2018). A data driven binning strategy for the construction of insurance tariff classes. *Scandinavian Actuarial Journal*, 2018(8) :681–705.

[Holland, 1992] Holland, J. H. (1992). Genetic algorithms. *Scientific american*, 267(1) :66–73.

[Koza, 1992] Koza, J. R. (1992). *Genetic programming : on the programming of computers by means of natural selection*, volume 1. MIT press.

[McCullagh and Nelder, 1989] McCullagh, P. and Nelder, J. (1989). Generalized linear models ii.

[Nelder and Wedderburn, 1972] Nelder, J. A. and Wedderburn, R. W. (1972). Generalized linear models. *Journal of the Royal Statistical Society : Series A (General)*, 135(3) :370–384.

[Rechenberg et al., 1973] Rechenberg, H., Billard, L., Chamberod, A., and Natta, M. (1973). Champs hyperfins et modele semi-microscopique non-local de l'invar. *Journal of Physics and Chemistry of Solids*, 34(7) :1251–1265.

[Slocum et al., 2008] Slocum, T. A., McMaster, R. M., Kessler, F. C., Howard, H. H., and Mc Master, R. B. (2008). Thematic cartography and geographic visualization.

Annexe A

Code R

A.1 Analyses descriptives

```
library(CASdatasets)
library(descr)
library(dplyr)
library(plyr)
library(corrplot)
#Chargement de la base de donn es
data(pg15training)

#Cr ation des variables Nbclaims et ClaimAmount
pg15training$Nbclaims<-as.integer(pg15training$Numtppd+
   pg15training$Numtpbi)
pg15training$ClaimAmount<-as.integer(pg15training$Indtppd+
   pg15training$Indtpbi)

#suppression es valeurs manquantes
pg15training<-na.omit(pg15training)

#supprimer les assur s ayant une exposition nulle
which(pg15training$Exppdays==0)

#suppressions des doublons
length(unique(pg15training$PolNum)) == nrow(pg15training)
doublons <-which(duplicated(pg15training["PolNum"]))
pg15training<-pg15training[-doublons,]
length(unique(pg15training$PolNum)) == nrow(pg15training)

#suppression des variables inutiles
drop <- c("PolNum","Bonus","Numtppd","Numtpbi","Indtppd","Indtpbi")
training=pg15training[,!(names(pg15training) %in% drop)]
summary(training)
rm(pg15training)

#exposition en ann e , co t moyen de sinistre
```

```
training$Exppdays<-training$Exppdays/365
training$frequency<-training$Nbclaims/training$Exppdays
training$avg_amount<-training$ClaimAmount/training$Nbclaims
training$avg_amount[is.na(training$avg_amount)] <- 0
names(training)[12]<-"Region"
names(training)[14]<-"Expo"

#Transformation des variables qui sont sens es  tre  des facteurs
training$CalYear<-as.factor(training$CalYear)
training$Group1<-as.factor(training$Group1)
training$Adind<-as.factor(training$Adind)


#Figure 1
par(mfrow=c(1,3))
freq(training$Nbclaims,plot = T,xlab="nombre de sinistres",col = "
   cadetblue1",ylab="nombre d'observations")
hist((training$Expo),xlab  = "Exposition", main="",col = "cadetblue1",
   ylab="nombre d'observations")
hist((training$ClaimAmount),xlab = "Montant des sinistres",main="",col =
   "cadetblue1",ylab="nombre d'observations")

#Figure 2
par(mfrow=c(3,2))
barplot(table(training$Gender),main = "Genre",col = "cadetblue1")
barplot(table(training$Type),main = "Type v hicule",col = "cadetblue1")
barplot(table(training$Adind),main = "Couverture",col = "cadetblue1")
barplot(table(training$Age),main = "Age du conducteur",col = "cadetblue1
   ")
barplot(table(training$Poldur),main =  "Age de la police",col = "
   cadetblue1")
barplot(table(training$Density),main = "Densit  de la ville",col = "
   cadetblue1")

#Figure 3
attach(training)
par(mfrow=c(3,2))
w4<-aggregate(frequency~Gender, FUN=mean)
barplot(frequency~Gender, w4,col = "cadetblue1",xlab = "Genre")

w1<-aggregate(frequency~Type, FUN=mean)
barplot(frequency~Type, w1,col = "cadetblue1",xlab = "Type de v hicule")

w8<-aggregate(frequency~Adind, FUN=mean)
barplot(frequency~Adind, w8,col = "cadetblue1",xlab = "Couverture")

x<-tapply(training$frequency,training$Age,mean)
plot(as.numeric(row.names(x)),x, type = "l", main="", xlab = "Age du
   conducteur", ylab = "frequency")

y<-tapply(training$frequency,training$Poldur,mean)
```

```
plot(as.numeric(row.names(y)),y, type = "l",main="", xlab = "Age de la police", ylab = "frequency")

v<-tapply(training$frequency,training$Density,mean)
plot(as.numeric(row.names(v)),v, type = "l",main="", xlab = "Densit  de la population", ylab = "frequency")

#Figure 4
#quelques statistiques par cat gories
#Le genre, l'age et la fr quence
par(mfrow=c(1,2))
female<-training[training$Gender=="Female",]
mean(female$frequency)
var(female$frequency)
female_mean<-tapply(female$frequency,female$Age,mean)

male<-training[training$Gender=="Male",]
mean(male$frequency)
var(male$frequency)
male_mean<-tapply(male$frequency,male$Age,mean)

plot(as.numeric(row.names(male_mean)),male_mean, type = "l",main="", xlab = "Age du conducteur",
     ylab = "Fr quence de sinistre moyenne", col="red")
legend(x = "topright", c("Male", "Female"), col = c("red","blue"),lty = c(1,1), border ="white", cex = .7)
par(new=T)
plot(as.numeric(row.names(female_mean)),female_mean, type = "l", xlab = "", ylab = "", col="blue", axes = F)

#En fonction de l'age de la police et du genre
female<-training[training$Gender=="Female",]
mean(female$frequency)
var(female$frequency)
female_mean<-tapply(female$frequency,female$Poldur,mean)

male<-training[training$Gender=="Male",]# les donn  es pour les hommes
mean(male$frequency)
var(male$frequency)
male_mean<-tapply(male$frequency,male$Poldur,mean)

plot(as.numeric(row.names(male_mean)),male_mean, type = "l",main="", xlab = "Age de la police",
     ylab = "Fr quence de sinistre moyenne", col="red")
legend(x = "topright", c("Male", "Female"), col = c("red","blue"),lty = c(1,1), border ="white", cex = .7)
par(new=T)
plot(as.numeric(row.names(female_mean)),female_mean, type = "l", xlab = "", ylab = "", col="blue", axes = F)

#Moyenne empirique
```

```
mean_Nbclaims=sum(training$Nbclaims)/sum(training$Expo)

#Variance empirique
var_Nbclaims=0
for (i in seq(1,nrow(training))){
  var_Nbclaims=var_Nbclaims+(training$Nbclaims[i]-mean_Nbclaims)^2
}
var_Nbclaims=var_Nbclaims/nrow(training)

k<-var(training$Nbclaims)

#test chi-carr
1-pchisq(nrow(training)*var_Nbclaims/mean_Nbclaims,nrow(training)-1)
#p-valeur <0.05 donc rejet de H0
```

A.2 Segmentation avec le package R insurancerating

```
library(insurancerating)
library(CASdatasets)
library(dplyr)
library(plyr)
library(caret)
library(visreg)
library(ggpubr)
library(ggplot2)

data(pg15training)
pg15training$Nbclaims<-as.integer(pg15training$Numtppd+
   pg15training$Numtpbi)
pg15training$ClaimAmount<-as.integer(pg15training$Indtppd+
   pg15training$Indtpbi)

#supprimer les valeurs manquantes
pg15training<-na.omit(pg15training)

#supprimer les assur s ayant une exposition nulle
which(pg15training$Exppdays==0)

#suppressions des doublons
length(unique(pg15training$PolNum)) == nrow(pg15training)
doublons <-which(duplicated(pg15training["PolNum"]))
pg15training<-pg15training[-doublons,]
length(unique(pg15training$PolNum)) == nrow(pg15training) #pas de
   doublons

#suppression des variables contenues dans le vecteur drop
drop <- c("PolNum","Bonus","Numtppd","Numtpbi","Indtppd","Indtpbi")
training=pg15training[,!(names(pg15training) %in% drop)]
summary(training)
```

```
rm(pg15training)

#exposition en ann e
training$Exppdays<-training$Exppdays/365
training$frequency<-training$Nbclaims/training$Exppdays
training$avg_amount<-training$ClaimAmount/training$Nbclaims
training$avg_amount[is.na(training$avg_amount)] <- 0
names(training)[12]<-"Region"
names(training)[14]<-"Expo"

#transformation des variables qui sont sens es etre des facteurs
training$CalYear<-as.factor(training$CalYear)
training$Group1<-as.factor(training$Group1)
training$Adind<-as.factor(training$Adind)

#conservation des assur s ayant une exposition comprise entre (0,1]
training = subset(training, training$Expo<=1 & training$Expo >0)

#partition du portefeuille en jeu d'entrainement et de test.
set.seed(1234)
datatraining<-createDataPartition(training$Nbclaims, p = .8, list = FALSE
    , times = 1)
head(datatraining)
test<-training[-datatraining,]
training<-training[datatraining,]

####Cat gorisation avec Insurancerating sur le training

###Density
fre_Density<- fit_gam(data=training, nclaims=Nbclaims,x=Density, exposure
    =Expo,model = "frequency")
autoplot(fre_Density, show_observations = TRUE)
clusters_Density<- construct_tariff_classes(fre_Density)
autoplot(clusters_Density, show_observations = TRUE)

training <- training %>%
  mutate(Density_class = clusters_Density$tariff_classes) %>%
  mutate(across(where(is.character), as.factor)) %>%
  mutate(across(where(is.factor), ~biggest_reference(., Expo)))

glimpse(training)
summary(training)

#segementation manuelle de la variable density
###fr quences pr dites
prediction<-clusters_Density[["prediction"]]
###fr quences pr dites
ggplot(clusters_Density[["prediction"]], aes(x=x,y=predicted)) +
  geom_line(color = "steelblue") + theme_bw(base_size = 12) +
  #geom_ribbon(aes(ymin = lwr_95, ymax = upr_95), alpha = 0.12)+
  #scale_x_continuous(breaks = seq(floor(min(prediction$x)), ceiling(max(
```

```
    prediction$x)), by = NULL))+
  scale_x_continuous(breaks = c(14.4,91,114,160,232,285,297))+
  geom_vline(xintercept=c(14.4,91,114,160,232,285,297),linetype = "dashed
    ",color="darkgray")+
  geom_point(data = clusters_Density[["data"]], aes(x = x, y = frequency)
    ,size = 1, color = "black")+
  #geom_point(color="black") +
  xlab("Density") +
  ylab("Predicted frequency")


###Poldur
pre_poldur<- fit_gam(data=training, nclaims=Nbclaims,x=Poldur, exposure=
    Expo,model = "frequency")
autoplot(pre_poldur, show_observations = TRUE)
clusters_poldur<- construct_tariff_classes(pre_poldur)
autoplot(clusters_poldur, show_observations = TRUE)

training <- training %>%
  mutate(Poldur_class = clusters_poldur$tariff_classes) %>%
  mutate(across(where(is.character), as.factor)) %>%
  mutate(across(where(is.factor), ~biggest_reference(., Expo)))

glimpse(training)

###Age
fre_age<- fit_gam(data=training, nclaims=Nbclaims,x=Age, exposure=Expo,
    model = "frequency")
autoplot(fre_age, show_observations = TRUE)
clusters_Age<- construct_tariff_classes(fre_age)
autoplot(clusters_Age, show_observations = TRUE)

training <- training %>%
  mutate(Age_class = clusters_Age$tariff_classes) %>%
  mutate(across(where(is.character), as.factor)) %>%
  mutate(across(where(is.factor), ~biggest_reference(., Expo)))

glimpse(training)
summary(training)

training<-droplevels(training)

##################################################
################## MODELE GLM ####################
##################################################

#d finition des niveaux de ref rence

table(training$Age_class)
training$Age_class<-relevel(training$Age_class,ref ="(32,46]")
table(training$Density_class)
```

```
125 training$Density_class<-relevel(training$Density_class,ref ="[14.4,91]")
126 table(training$Gender)
127 training$Gender<-relevel(training$Gender,ref ="Male")
128 table(training$Adind)
129 training$Adind<-relevel(training$Adind,ref="1")
130 table(training$Type)
131
132
133 GLM<-glm(formula =Nbclaims~Density_class+Gender+Type+Adind+Age_class+
    Age_class*Gender
134       ,family = poisson(link = "log"),data = training, offset = log(
    Expo))
135
136
137 GLM<-step(GLM)
138 dummy.coef(GLM)
139 summary(GLM)
140
141 sum(GLM$fitted.values)/sum(training$Expo)
142 sum(training$Nbclaims)/sum(training$Expo)
143
144
145 #observation de l'impact des variables sur le resultat du GLM
146 par(mfrow=c(2,3))
147 visreg(GLM, type="contrast", scale="response")
148
149 ###Variable Density
150 training$Density91_114<-ifelse(training$Density >91 & training$Density <=
     114, "yes", "no")
151 training$Density114_160<-ifelse(training$Density >114 & training$Density
    <= 160, "yes", "no")
152 training$Density160_232<-ifelse(training$Density >160 & training$Density
    <= 232, "yes", "no")
153 training$Density232_285<-ifelse(training$Density >232 & training$Density
    <= 285, "yes", "no")
154 training$Density285_297<-ifelse(training$Density >285 & training$Density
    <= 297.5, "yes", "no")
155
156 ##Variable Age
157 training$Age18_25<-ifelse(training$Age >=18 & training$Age <= 25, "yes",
    "no")
158 training$Age25_32<-ifelse(training$Age >25 & training$Age <= 32, "yes", "
    no")
159 training$Age46_53<-ifelse(training$Age >46 & training$Age <= 53, "yes", "
    no")
160 training$Age53_60<-ifelse(training$Age >53 & training$Age <= 60, "yes", "
    no")
161 training$Age60_75<-ifelse(training$Age >60 & training$Age <= 75, "yes", "
    no")
162
163 ###variable genre
```

```
training$GenderFemale<-ifelse(training$Gender=="Female", "yes", "no")

#variable Adind
training$Adind0<-ifelse(training$Adind==0, "yes", "no")

##variable type
training$TypeB<-ifelse(training$Type=="B", "yes", "no")
training$TypeC<-ifelse(training$Type=="C", "yes", "no")
training$TypeD<-ifelse(training$Type=="D", "yes", "no")
training$TypeE<-ifelse(training$Type=="E", "yes", "no")
training$TypeF<-ifelse(training$Type=="F", "yes", "no")


##construction des classes
training$freq_yearly_pred<-GLM[["fitted.values"]]/training$Expo
training$freq_yearly_obs<-training$frequency
training$fitted_values<-GLM$fitted.values #nombre de sinistre pr dit
training$Nbclaim<-training$Nbclaims
training$Exposure<-training$Expo
##cr ation des classes
class<-training[!duplicated(training$freq_yearly_pred), ]

####Tableau des classes
tableau_segmentation<-class[,-c(1:21)]
write.csv2(tableau_segmentation,"tab.csv")

mean(class$freq_yearly_pred)
mean(class$freq_yearly_obs)
sum(class$fitted_values)/sum(class$Expo)
sum(class$Nbclaims)/sum(class$Expo)

stat_class<-data.frame(table(training$freq_yearly_pred))

##representations graphiques

#en fonction de l'age du conducteur et de la densit

#fr quence observ e
par(mfrow=c(1,1))
freq<-training$freq_yearly_obs
age<-training$Age
freq_data<-aggregate(freq~age, FUN=mean)

graphiq1<-ggplot(freq_data, aes(x=age,y=freq),xlab=) +
  geom_point() +
  xlab("Age du conducteur") +
  ylab("Fr quence de sinistre observ e") +
  geom_smooth(span = .5)
graphiq1

###fr quences pr dites
```

```
freq<-training$freq_yearly_pred
age<-training$Age
freq_data<-aggregate(freq~age, FUN=mean)
graphiq2<-ggplot(freq_data, aes(x=age,y=freq)) +
  geom_point(color="red") +
  xlab("Age du conducteur") +
  ylab("Fr quence de sinistre pr dite") +
  geom_smooth(span = .5)
graphiq2

#fr quence observ e
freq<-training$freq_yearly_obs
density<-training$Density
freq_data<-aggregate(freq~density, FUN=mean)

graphiq3<-ggplot(freq_data, aes(x=density,y=freq),xlab=) +
  geom_point() +
  xlab("Density") +
  ylab("Fr quence de sinistre observ e") +
  geom_smooth(span = .5)
graphiq3

###fr quences pr dites
freq<-training$freq_yearly_pred
density<-training$Density
freq_data<-aggregate(freq~density, FUN=mean)

graphiq4<-ggplot(freq_data, aes(x=density,y=freq),xlab=) +
  geom_point(color="red") +
  xlab("Density") +
  ylab("Fr quence de sinistre observ e") +
  geom_smooth(span = .5)
graphiq4

ggarrange(graphiq1, graphiq2,graphiq3,graphiq4,
          labels = c("Fr quence observ e", "Fr quence pr dite" ),
          ncol = 2, nrow = 2)

#les fr quences obsev es par classes
freq<-class$freq_yearly_obs
age<-class$Age
freq_data<-aggregate(freq~age, FUN=mean)

graphiq5<-ggplot(freq_data, aes(x=age,y=freq),xlab=) +
  geom_point() +
  xlab("Age du conducteur") +
  ylab("Fr quence de sinistre observ e") +
  geom_smooth(span = .5)
graphiq5

#les fr quences pr dites par classes
```

```
freq<-class$freq_yearly_pred
age<-class$Age
freq_data<-aggregate(freq~age, FUN=mean)
graphiq6<-ggplot(freq_data, aes(x=age,y=freq)) +
  geom_point(color="red") +
  xlab("Age du conducteur") +
  ylab("Fr quence de sinistre pr dite") +
  geom_smooth(span = .5)
graphiq6


#les fr quences observ es par classes
freq<-class$freq_yearly_obs
density<-class$Density
freq_data<-aggregate(freq~density, FUN=mean)

graphiq7<-ggplot(freq_data, aes(x=density,y=freq),xlab=) +
  geom_point() +
  xlab("Density") +
  ylab("Fr quence de sinistre observ e") +
  geom_smooth(span = .5)
graphiq7

###fr quences pr dites
freq<-class$freq_yearly_pred
density<-class$Density
freq_data<-aggregate(freq~density, FUN=mean)

graphiq8<-ggplot(freq_data, aes(x=density,y=freq),xlab=) +
  geom_point(color="red") +
  xlab("Density") +
  ylab("Fr quence de sinistre observ e") +
  geom_smooth(span = .5)
graphiq8


###frequences pr dites et observ es par classes cote    cote
ggarrange(graphiq5, graphiq6,graphiq7, graphiq8,
          labels = c("Fr quence observ e", "Fr quence pr dite" ),
          ncol = 2, nrow = 2)


##################################################
########Qualit  d'ajustement des mod les##########
##################################################

coupure1<-c(14.4,91,114,160,232,285,297.3852)
test$Density_class <- cut(test$Density, breaks = coupure1, include.lowest
    = TRUE)
table(test$Density_class)

```

```
coupure2<-c(18,25,32,46,53,60,75)
test$Age_class <- cut(test$Age, breaks = coupure2, include.lowest = TRUE)
table(test$Age_class)

### Calcul de la scaled deviance
test$prediction<-predict.glm(GLM,test, type = "response")

for(i in 1:length(test)){

  test$log[i]=ifelse(test$Nbclaims[i]==0,test$prediction[i],
   test$Nbclaims[i] * log(test$Nbclaims[i]/test$prediction[i]) -(
   test$Nbclaims[i] - test$prediction[i]))

}

dev_poi=2*sum(test$log*test$Expo)
AIC(GLM)
BIC(GLM)
rm(list=ls())
```

A.3 Modèle GAM complet et segmentation

```
#Chargement de la base de donn es
data(pg15training)

#Cr ation des variables Nbclaims et ClaimAmount
pg15training$Nbclaims<-as.integer(pg15training$Numtppd+
   pg15training$Numtpbi)
pg15training$ClaimAmount<-as.integer(pg15training$Indtppd+
   pg15training$Indtpbi)

#suppression es valeurs manquantes
pg15training<-na.omit(pg15training)

#supprimer les assur s ayant une exposition nulle
which(pg15training$Exppdays==0)

#suppressions des doublons
length(unique(pg15training$PolNum)) == nrow(pg15training)
doublons <-which(duplicated(pg15training["PolNum"]))
pg15training<-pg15training[-doublons,]
length(unique(pg15training$PolNum)) == nrow(pg15training)

#suppression des variables inutiles
drop <- c("PolNum","Bonus","Numtppd","Numtpbi","Indtppd","Indtpbi")
training=pg15training[,!(names(pg15training) %in% drop)]
summary(training)
rm(pg15training)

```

```
#exposition en ann e , co t moyen de sinistre
training$Exppdays<-training$Exppdays/365
training$frequency<-training$Nbclaims/training$Exppdays
training$avg_amount<-training$ClaimAmount/training$Nbclaims
training$avg_amount[is.na(training$avg_amount)] <- 0
names(training)[12]<-"Region"
names(training)[14]<-"Expo"

#Transformation des variables qui sont sens es  tre  des facteurs
training$CalYear<-as.factor(training$CalYear)
training$Group1<-as.factor(training$Group1)
training$Adind<-as.factor(training$Adind)

#conservation des assur s ayant une exposition comprise entre (0,1]
training = subset(training, training$Expo<=1 & training$Expo >0)

#partition du portefeuille en jeu d'entrainement et de test.
set.seed(1234)
datatraining<-createDataPartition(training$Nbclaims, p = .8, list = FALSE
    , times = 1)
head(datatraining)
test<-training[-datatraining,]
training<-training[datatraining,]

#d finition des niveaux de ref rence
table(training$Gender)
training$Gender<-relevel(training$Gender,ref ="Male")

table(training$Adind)
training$Adind<-relevel(training$Adind,ref="1")

table(training$Type)

#Impl mentation du mod le gam complet avec interaction
GAM<-gam(Nbclaims~ s(Age)+s(Density,bs="ps")+Gender+Type+s(Age,by=Gender)
    +Adind,
          family=poisson(link = "log"),offset = log(Expo), data=training,
    select=TRUE)

anova.gam(GAM)
dummy.coef(GAM)
summary(GAM)
par(mfrow=c(2,2))
plot(GAM)

#to see quantiles used in Density variable
#GAM$smooth[[2]]$xp

#Importance des variables dans le mod le gam
par(mfrow=c(2,3))
visreg(GAM, type="conditional", scale="response")
```

```
visreg(GAM, xvar="Age", by="Gender", scale="response", type="contrast")


##Recherche des classes

#Extraction des effets marginaux dans le mod le GAM complet
f_chapo<-data.frame(predict.gam(GAM,type="terms"))
f_chapo<-cbind(training,f_chapo[,4:7])


s_age<-f_chapo$s.Age.+f_chapo$s.Age..GenderMale+f_chapo$s.Age..
   GenderFemale

f_chapo<-cbind(f_chapo,s_age)

data_age<-aggregate(f_chapo$s_age~f_chapo$Age,FUN = mean)

data_age_poids<-aggregate(f_chapo$s_age~f_chapo$Age,FUN = length)

data_age<-cbind(data_age, data_age_poids$`f_chapo$s_age`)

names(data_age)<-c("Age","f_age","poids")

#Ensembles de param tre pour la recherche de l'arbre optimal
alfa<-c(0,1,seq(5,50,by=5),seq(60,100,by=10),seq(150,950,by=50))
tree<-c(200,500)
n_iteration<-c(3000,5000,10000)
n<-nrow(training)

table<-matrix(NA,ncol=4,nrow=length(alfa)*length(tree)*length(n_iteration
   ))
colnames(table)<-c("Alpha","tree","iteration","Mesure_tree")

#algorithme de d termination des param tres de l'arbre optimal pour la
   variable Age
m<-1
for(i in 1 : length(alfa)){
  for(j in 1 : length(tree)){
    for(k in 1 : length(n_iteration)){


      tree_x <-evtree(data_age$f_age ~ data_age$Age, data = data_age ,
   weights=data_age$poids,
                      control =evtree.control(alpha=alfa[i],ntrees=tree[j
   ],niterations=n_iteration[k],seed=1))

      y<-data_age[,2]
      u<-predict(tree_x, newdata=data_age, type = "response")
      nod<-width(tree_x)
      MSE<-sum(data_age$poids*(y-u)^2)/sum(data_age$poids)
      Mesure_tree<-n*log(MSE)+4*alfa[i]*(nod+1)*log(n)
```

```
      table[m,1]<-alfa[i]
      table[m,2]<-tree[j]
      table[m,3]<-n_iteration[k]
      table[m,4]<-Mesure_tree

      m<-m+1
    }

  }

}
which.min(table[,4])

#Ajustement de l'arbre optimal pour la variable Age chez les hommes
tree_age <-evtree(data_age$f_age ~ data_age$Age, data = data_age ,
    weights=data_age$poids,
                  control =evtree.control(alpha=700,ntrees=500,
    niterations=3000,seed=1))

par(mfrow=c(1,1))
plot(tree_age)

#Cat gorisation de la variable Age en la variable Age_class
coupure<-c(18,23,28,32,47,59,75)
training$Age_class <- cut(training$Age, breaks = coupure, include.lowest
    = TRUE,right = TRUE)
table(training$Age_class)

#Representation graphique du d coupage effecu  sur l'effet marginal de
    l' ge
graphiq_age<-ggplot(data_age, aes(x=Age,y=f_age)) +
  geom_line(color = "steelblue") + theme_bw(base_size = 12) +
  scale_x_continuous(breaks = coupure)+
  geom_vline(xintercept=coupure,linetype = "dashed",color="gray48")+
  xlab("Age") +
  ylab("^f1")


#####################################################
####################Cas de la densit #############
#####################################################
data_density<-aggregate(f_chapo$s.Density.~f_chapo$Density,FUN = mean)

#Calcul des poids
data_density_poids<-aggregate(f_chapo$s.Density.~f_chapo$Density,FUN =
    length)

data_density<-cbind(data_density,data_density_poids$`f_chapo$s.Density.`)

names(data_density)<-c("Density","f_density","poids")

```

```
#algorithme de d termination des param tres de l'arbre optimal pour la variable Density

n<-nrow(f_chapo)
table_density<-matrix(NA,ncol=4,nrow=length(alfa)*length(tree)*length(n_iteration))
colnames(table_density)<-c("Alpha","tree","iteration","Mesure_tree")

m<-1
for(i in 1 : length(alfa)){
  for(j in 1 : length(tree)){
    for(k in 1 : length(n_iteration)){


      tree_x <-evtree(data_density$f_density ~ data_density$Density, data = data_density , weights=data_density$poids,
                      control =evtree.control(alpha=alfa[i],ntrees=tree[j],niterations=n_iteration[k],seed=1))

      y<-data_density[,2]
      u<-predict(tree_x, newdata=data_density, type = "response")
      nod<-width(tree_x)
      MSE<-sum(data_density$poids*(y-u)^2)/sum(data_density$poids)
      Mesure_tree<-n*log(MSE)+4*alfa[i]*(nod+1)*log(n)
      table_density[m,1]<-alfa[i]
      table_density[m,2]<-tree[j]
      table_density[m,3]<-n_iteration[k]
      table_density[m,4]<-Mesure_tree

      m<-m+1
    }

  }

}
which.min(table_density[,4])

#Ajustement de l'arbre optimal pour la variable Density
tree_x_density <-evtree(data_density$f_density ~ data_density$Density, weights = data_density$poids,data = data_density,
                        control =evtree.control(alpha=550,ntrees=200,niterations=3000,seed=1))
plot(tree_x_density)

#Cat gorisation de la variable Density en la variable Density_class
coupure<-c(14,63.1,81.9,156.1,232.9,297.5)
training$Density_class <- cut(training$Density, breaks = coupure, include.lowest = TRUE)
table(training$Density_class)

#Representation graphique du d coupage effecu  sur l'effet marginal de
```

```
   la densit
graphiq_density<-ggplot(data_density, aes(x=Density,y=f_density)) +
  geom_line(color = "steelblue") + theme_bw(base_size = 12) +
  scale_x_continuous(breaks = coupure)+
  geom_vline(xintercept=coupure,linetype = "dashed",color="gray48")+
  xlab("Density") +
  ylab("^f2")

#Representation des deux effets marginaux segmenter sur le m me graphe
ggarrange(graphiq_age, graphiq_density,
          labels = c("", "" ),
          ncol = 2, nrow = 1)


#####################################################
####################    GLM   ####################
#####################################################

#D finition des classes de ref rence
table(training$Type)

table(training$Age_class)
training$Age_class<-relevel(training$Age_class,ref="(32,47]")

table(training$Density_class)
#training$Density_class<-relevel(training$Density_class,ref="[14,63.1]")
#table(training$Density_class)

GLM<-glm(formula = Nbclaims~Age_class+Density_class+Type+Adind+Gender+
   Age_class*Gender,
         family = poisson(link = "log"),data = training, offset = log(
   Expo))
GLM<-step(GLM)
dummy.coef(GLM)
summary(GLM)
AIC(GLM) #79579.27
BIC(GLM) #79783.65

#Importance des variables dans le mod le GLM ajut
par(mfrow=c(3,2))
visreg(GLM, type="contrast", scale="response")

#nombre de sinistre moyen observ  et pr dit
sum(training$Nbclaims)/sum(training$Expo)
sum(GLM$fitted.values)/sum(training$Expo)

##tableau de segmentation

##Variable Age
training$Age18_23<-ifelse(training$Age >=18 & training$Age <= 23, "yes",
   "no")
```

```
training$Age23_28<-ifelse(training$Age >23 & training$Age <= 28, "yes", "
    no")
training$Age22_32<-ifelse(training$Age >28 & training$Age <= 32, "yes", "
    no")
training$Age47_59<-ifelse(training$Age >47 & training$Age <= 59, "yes", "
    no")
training$Age59_75<-ifelse(training$Age >59 & training$Age <= 75, "yes", "
    no")

###Variable Density
training$Density63_82<-ifelse(training$Density >63.1 & training$Density
    <= 81.9, "yes", "no")
training$Density82_156<-ifelse(training$Density >81.9 & training$Density
    <= 156.1, "yes", "no")
training$Density156_233<-ifelse(training$Density >156.1 &
    training$Density <= 232.9, "yes", "no")
training$Density233_298<-ifelse(training$Density >232.9 &
    training$Density <= 298, "yes", "no")

###variable genre
training$GenderFemale<-ifelse(training$Gender=="Female", "yes", "no")

##variable type
training$TypeB<-ifelse(training$Type=="B", "yes", "no")
training$TypeC<-ifelse(training$Type=="C", "yes", "no")
training$TypeD<-ifelse(training$Type=="D", "yes", "no")
training$TypeE<-ifelse(training$Type=="E", "yes", "no")
training$TypeF<-ifelse(training$Type=="F", "yes", "no")

#variable Adind
training$Adind0<-ifelse(training$Adind==0, "yes", "no")

##Construction des classes

#Calcul de la fr quence de sinistre annuelle pr dite et observ e
training$freq_yearly_pred<-GLM[["fitted.values"]]/training$Expo
training$freq_yearly_obs<-training$frequency

training$fitted_values<-GLM$fitted.values #nombre de sinistre pr dit
training$Nbclaim<-training$Nbclaims
training$Exposure<-training$Expo

#cr ation des classes
class<-training[!duplicated(training$freq_yearly_pred),]

#Tableau de segmentation multivari e
tableau_segmentation<-class[,-c(1:20)]
write.csv2(tableau_segmentation,"tab_gam.csv")

##Repr sentations graphiques
par(mfrow=c(1,1))
```

```
300
301 #repartition des fr quences annuelles observ es et pr dites au sein du
        portefeuille entier
302
303 #En fonction de l' ge  du conducteur
304
305 #Fr quence  observ e
306 freq<-training$freq_yearly_obs
307 age<-training$Age
308 freq_data<-aggregate(freq~age, FUN=mean)
309
310 graphiq1<-ggplot(freq_data, aes(x=age,y=freq),xlab=) +
311   geom_point() +
312   xlab("Age du conducteur") +
313   ylab("Fr quence de sinistre  observ e") +
314   geom_smooth(span = .5)
315
316 #Fr quences  pr dites
317 freq<-training$freq_yearly_pred
318 age<-training$Age
319 freq_data<-aggregate(freq~age, FUN=mean)
320 graphiq2<-ggplot(freq_data, aes(x=age,y=freq)) +
321   geom_point(color="red") +
322   xlab("Age du conducteur") +
323   ylab("Fr quence de sinistre  pr dite") +
324   geom_smooth(span = .5)
325
326 #En fonction de la densit
327
328 #Fr quence  observ e
329 freq<-training$freq_yearly_obs
330 density<-training$Density
331 freq_data<-aggregate(freq~density, FUN=mean)
332
333 graphiq3<-ggplot(freq_data, aes(x=density,y=freq),xlab=) +
334   geom_point() +
335   xlab("Density") +
336   ylab("Fr quence de sinistre  observ e") +
337   geom_smooth(span = .5)
338
339
340 #Fr quence  pr dite
341 freq<-training$freq_yearly_pred
342 density<-training$Density
343 freq_data<-aggregate(freq~density, FUN=mean)
344
345 graphiq4<-ggplot(freq_data, aes(x=density,y=freq),xlab=) +
346   geom_point(color="red") +
347   xlab("Density") +
348   ylab("Fr quence de sinistre  observ e") +
349   geom_smooth(span = .5)
```

```
350
351 ggarrange(graphiq1, graphiq2, graphiq3, graphiq4,
352           labels = c("Fr quence  observ e", "Fr quence  pr dite" ),
353           ncol = 2, nrow = 2)
354
355 #R partiation des fr quences annuelles observ es et pr dites au sein
      des 2902 segments
356
357 #En fonction de l' ge  du conducteur
358
359 #Fr quence  obsev e par classes
360 freq<-class$freq_yearly_obs
361 age<-class$Age
362 freq_data<-aggregate(freq~age, FUN=mean)
363
364 graphiq5<-ggplot(freq_data, aes(x=age,y=freq),xlab=) +
365   geom_point() +
366   xlab("Age du conducteur") +
367   ylab("Fr quence de sinistre  observ e") +
368   geom_smooth(span = .5)
369
370
371 #Fr quence  pr dite par classes
372 freq<-class$freq_yearly_pred
373 age<-class$Age
374 freq_data<-aggregate(freq~age, FUN=mean)
375 graphiq6<-ggplot(freq_data, aes(x=age,y=freq)) +
376   geom_point(color="red") +
377   xlab("Age du conducteur") +
378   ylab("Fr quence de sinistre  pr dite") +
379   geom_smooth(span = .5)
380
381 #En fonction de la densit
382
383 #Fr quence  observ e par classes
384 freq<-class$freq_yearly_obs
385 density<-class$Density
386 freq_data<-aggregate(freq~density, FUN=mean)
387
388 graphiq7<-ggplot(freq_data, aes(x=density,y=freq),xlab=) +
389   geom_point() +
390   xlab("Density") +
391   ylab("Fr quence de sinistre  observ e") +
392   geom_smooth(span = .5)
393 graphiq7
394
395 #Fr quence  pr dite V
396 freq<-class$freq_yearly_pred
397 density<-class$Density
398 freq_data<-aggregate(freq~density, FUN=mean)
399
```

```
graphiq8<-ggplot(freq_data, aes(x=density,y=freq),xlab=) +
  geom_point(color="red") +
  xlab("Density") +
  ylab("Fr quence de sinistre observ e") +
  geom_smooth(span = .5)
graphiq8

ggarrange(graphiq5, graphiq6, graphiq7, graphiq8,
          labels = c("Fr quence observ e", "Fr quence pr dite" ),
          ncol = 2, nrow = 2)

##################################################
########Qualit  d'ajustement des mod les##########
##################################################


coupure1<-c(14,63.1,81.9,156.1,232.9,297.5)
test$Density_class <- cut(test$Density, breaks = coupure1, include.lowest
    = TRUE)
table(test$Density_class)

coupure2<-c(18,23,28,32,47,59,75)
test$Age_class <- cut(test$Age, breaks = coupure2, include.lowest = TRUE)
table(test$Age_class)


### Calcul de la scaled deviance
test$prediction<-predict.glm(GLM,test, type = "response")

for(i in 1:length(test)){

  test$log[i]=ifelse(test$Nbclaims[i]==0,test$prediction[i],
   test$Nbclaims[i] *
                       log(test$Nbclaims[i]/test$prediction[i]) -(
   test$Nbclaims[i] - test$prediction[i]))

}

dev_poi=2*sum(test$log*test$Expo) #25374.67
AIC(GLM) #79579.27
BIC(GLM) #79783.65
```

Annexe B

Résultats

B.1 Coefficients GLM "insurancerating"

```
Full coefficients are

(Intercept):             -2.358619
Age_class:                 (39,69]      [18,25]      (25,32]      (32,39]
        (69,75]
                         0.0000000    1.3496639    0.8179298    0.2978261
     -0.3756097
Gender:                       Male       Female
                         0.00000000  -0.06497603
Density_class:            [14.4,91]      (91,98]     (98,248]    (248,285]
      (285,297]
                         0.0000000    0.1854066    0.4165401    0.8706760
      0.7440470
Type:                          B&C            A          D&E            F
                         0.00000000  -0.08039662   0.13248431   0.21605729
CalYear:                      2009         2010
                         0.0000000    0.0767979
Adind:                           1            0
                         0.00000000   0.07140392
Age_class:Gender:    (39,69]:Male [18,25]:Male (25,32]:Male (32,39]:Male
   (69,75]:Male (39,69]:Female [18,25]:Female (25,32]:Female (32,39]:
   Female (69,75]:Female
                         0.00000000   0.00000000   0.00000000   0.00000000
     0.00000000     0.00000000    -0.47341118    -0.31100580
   -0.05759581     0.22608090
```

B.2 Résultats modèle GAM complet

```
Formula:
Nbclaims ~ s(Age) + s(Density, bs = "ps") + Gender + Type +
    s(Age, by = Gender) + Adind
Parametric coefficients:
             Estimate Std. Error z value Pr(>|z|)

(Intercept)  -1.78588    0.01985 -89.951  < 2e-16 ***
GenderFemale -0.19068    0.01936  -9.848  < 2e-16 ***
TypeB         0.06652    0.02371   2.805 0.005026 **
TypeC         0.11594    0.02683   4.322 1.55e-05 ***
TypeD         0.18263    0.02376   7.688 1.49e-14 ***
TypeE         0.24754    0.02759   8.973  < 2e-16 ***
TypeF         0.28791    0.03520   8.178 2.88e-16 ***
Adind0        0.06064    0.01694   3.581 0.000343 ***

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

Approximate significance of smooth terms:
                        edf Ref.df   Chi.sq p-value
s(Age)              7.73882      9  179.771  <2e-16 ***
s(Density)          7.11133      9 1567.186  <2e-16 ***
s(Age):GenderMale   4.33152      9   32.422  <2e-16 ***
s(Age):GenderFemale 0.07609      9    0.049   0.012 *
---
Signif. codes:  0    ***    0.001    **    0.01    *    0.05    .    0.1
           1
R-sq.(adj) =  0.0938   Deviance explained = 11.2%
UBRE = -0.34565  Scale est = 1         n = 80000
```

B.3 Tableau de segmentation avec insurancerating

Density91_114	Density114_160	Density160_232	Density232_285	Density285_297	Age18_25	Age25_32	Age46_53	Age53_60	Age60_75	GenderFemale	Adind0	TypeB	TypeC	TypeD	TypeE	TypeF	freq_yearly_pred	freq_yearly_obs	fitted_values	Nbclaim	Exposure
no	no	no	no	no	yes	no	no	no	no	no	yes	no	yes	no	no	no	0,423900868	1	0,423900868	1	1
no	no	no	no	no	yes	no	no	no	no	no	no	no	no	no	yes	no	0,444810332	1	0,444810332	1	1
no	no	yes	no	no	no	no	no	no	no	yes	yes	no	no	no	yes	no	0,248965577	2	0,248965577	2	1
no	no	no	no	no	no	yes	no	no	no	yes	no	no	no	yes	no	no	0,16866604	1	0,16866604	1	1
no	no	no	no	no	yes	no	no	no	no	yes	yes	no	yes	no	no	no	0,249160633	1	0,249160633	1	1
no	yes	no	no	no	no	no	no	no	no	no	no	no	no	yes	no	no	0,182436225	1	0,182436225	1	1
no	no	yes	no	no	no	yes	no	no	no	no	yes	no	yes	no	no	no	0,437631783	1	0,437631783	1	1
no	no	yes	no	no	no	no	no	no	yes	yes	yes	no	no	no	no	no	0,107066089	1	0,107066089	1	1
yes	no	no	no	no	yes	no	no	no	no	yes	yes	no	no	no	no	no	0,290621551	1	0,290621551	1	1
no	no	no	no	no	yes	no	no	no	no	no	yes	no	no	yes	no	no	0,453910999	1	0,453910999	1	1
yes	no	no	no	no	yes	no	no	no	no	no	yes	no	no	yes	no	no	0,596034725	1	0,596034725	1	1
no	no	no	no	no	no	yes	no	no	no	yes	yes	no	no	no	no	no	0,151334376	1	0,151334376	1	1
no	no	no	no	yes	yes	no	no	no	no	no	yes	no	no	yes	no	no	0,954529211	1	0,954529211	1	1
yes	no	no	no	no	yes	no	no	no	no	yes	yes	no	no	yes	no	no	0,350337546	1	0,350337546	1	1
no	no	no	no	no	no	yes	no	no	no	no	yes	no	yes	no	no	no	0,251528317	0	0,251528317	0	1
no	no	no	no	no	no	no	no	yes	no	yes	yes	no	no	no	yes	no	0,106944565	0	0,094052617	0	0,87945
no	no	no	yes	no	no	no	no	no	yes	yes	no	no	no	yes	no	no	0,154842296	0	0,154842296	0	1
no	no	no	no	no	no	yes	no	no	no	yes	yes	yes	no	no	no	no	0,16169505	0	0,16169505	0	1
no	no	no	no	yes	no	no	no	no	no	no	no	no	no	no	no	yes	0,325886661	0	0,325886661	0	1
no	yes	no	no	no	no	no	no	no	no	no	no	no	no	yes	no	no	0,182436225	0	0,10996156	0	0,60274
no	no	no	yes	no	no	no	no	no	no	yes	no	no	no	no	no	no	0,233768651	1	0,233768651	1	1
no	no	yes	no	no	no	no	no	no	no	no	yes	no	yes	no	no	no	0,244369255	0	0,125866904	0	0,51507
no	no	yes	no	no	no	no	no	no	no	no	no	no	no	no	yes	no	0,256423087	0	0,209353644	0	0,81644
no	no	yes	no	no	no	no	no	no	no	yes	no	no	no	yes	no	no	0,21716885	0	0,21716885	0	1
no	no	no	yes	no	no	no	no	no	no	no	no	no	no	no	yes	no	0,332739784	0	0,332739784	0	1
no	no	no	no	no	no	no	no	no	no	no	no	no	no	no	no	yes	0,154970155	0	0,104445639	0	0,67397
no	yes	no	no	no	no	no	no	no	no	no	no	no	yes	no	no	no	0,170374532	0	0,170374532	0	1
no	no	no	yes	no	no	no	no	no	no	no	no	yes	no	no	no	no	0,278247988	0	0,278247988	0	1
no	no	no	no	no	no	yes	no	no	no	yes	yes	no	no	yes	no	no	0,182430083	0	0,117954794	0	0,64658

FIGURE B.3.1: Tableau de segmentation au sein du portefeuille TPL

B.4 Tableau de segmentation GAM complet

Age18_23	Age23_28	Age22_32	Age47_59	Age59_75	Density63_82	Density82_156	Density156_233	Density233_298	GenderFemale	TypeB	TypeC	TypeD	TypeE	TypeF	Adind0	freq_yearly_pred	freq_yearly_obs	fitted_values	Nbclaim	Exposure
no	yes	no	no	no	yes	no	no	no	no	no	yes	no	no	no	yes	0,339429439	1	0,339429439	1	1
yes	no	no	no	no	no	no	no	no	no	no	no	no	yes	no	no	0,455374621	1	0,455374621	1	1
no	no	no	no	no	no	no	yes	no	yes	no	no	no	yes	no	yes	0,244319547	2	0,244319547	2	1
no	yes	no	no	no	no	no	no	no	yes	no	no	yes	no	no	no	0,198500786	1	0,198500786	1	1
yes	no	no	no	no	yes	no	no	no	yes	no	yes	no	no	no	yes	0,269533168	1	0,269533168	1	1
no	no	no	no	no	no	yes	no	no	no	no	no	yes	no	no	no	0,179853542	1	0,179853542	1	1
no	yes	no	no	no	no	no	yes	no	no	no	yes	no	no	no	yes	0,552364328	1	0,552364328	1	1
no	no	no	no	yes	no	no	yes	no	yes	no	no	no	no	no	yes	0,106705017	1	0,106705017	1	1
no	yes	no	no	no	no	yes	no	no	yes	no	no	no	no	no	yes	0,239488035	1	0,239488035	1	1
no	yes	no	no	no	no	no	no	no	no	no	no	yes	no	no	yes	0,329290532	1	0,329290532	1	1
yes	no	no	no	no	no	yes	no	no	no	no	no	yes	no	no	yes	0,623912435	1	0,623912435	1	1
no	no	yes	no	no	no	no	no	no	yes	no	no	no	no	no	yes	0,130621927	1	0,130621927	1	1
yes	no	no	no	no	no	no	no	yes	no	no	no	yes	no	no	yes	1,072981825	1	1,072981825	1	1
yes	no	no	no	no	no	yes	no	no	yes	no	no	yes	no	no	yes	0,354350805	1	0,354350805	1	1
no	yes	no	no	no	no	no	no	no	no	no	yes	no	no	no	yes	0,307540715	0	0,307540715	0	1
no	no	no	no	yes	yes	no	no	no	yes	no	no	no	yes	no	yes	0,083748267	0	0,073652585	0	0,879452
no	no	no	no	yes	no	no	no	yes	yes	no	no	yes	no	no	no	0,155522028	0	0,155522028	0	1
no	yes	no	no	no	no	no	no	no	yes	yes	no	no	no	no	yes	0,188735247	0	0,188735247	0	1
no	no	no	no	no	no	no	no	yes	no	no	no	no	no	yes	no	0,344460119	0	0,344460119	0	1
no	no	no	no	no	no	yes	no	no	no	no	no	yes	no	no	no	0,179853542	0	0,108404875	0	0,60274
no	no	no	no	no	no	no	no	yes	yes	no	no	no	no	no	no	0,23149841	1	0,23149841	1	1
no	no	no	no	no	no	no	yes	no	no	no	yes	no	no	no	yes	0,238700975	0	0,122947351	0	0,515068
no	no	no	no	no	no	no	yes	no	no	no	no	no	yes	no	no	0,252794411	0	0,206391053	0	0,816438
no	no	no	no	no	no	no	yes	no	yes	no	no	yes	no	no	no	0,21486268	0	0,21486268	0	1
no	no	no	no	no	no	no	no	yes	no	no	no	no	yes	no	no	0,328023274	0	0,328023274	0	1
no	no	no	no	no	no	no	no	no	no	no	no	no	no	yes	no	0,14780147	0	0,099614141	0	0,673973
no	no	no	no	no	no	yes	no	no	no	no	yes	no	no	no	no	0,167974119	0	0,167974119	0	1
no	no	no	no	no	no	no	no	yes	no	yes	no	no	no	no	no	0,274282574	0	0,274282574	0	1
no	yes	no	no	no	no	no	no	no	yes	no	no	yes	no	no	yes	0,21283474	0	0,137613695	0	0,646575

FIGURE B.4.1: Tableau de segmentation au sein du portefeuille TPL via modèle GAM complet