Mémoire de Master 1 Mathématiques Appliquées de l'Université de Paris Dauphine

Titre : **A5 - Tarification en assurance IARD avec les GLM et les GAM**

Par : Adnane EL KASMI | Samuel TEBOUL | Antonin AUBRY

Directeur de Mémoire : Pierre Cardaliaguet
Encadrant principal : Christophe DUTANG
Numéro de groupe : 2
Date : Janvier 2021

Confidentialité : ☑ Non ☐ Oui (Durée : ☑ 1 an ☐ 2 ans)

Université Paris-Dauphine – MIDO – Bureau B536, Place du Maréchal de Lattre de Tassigny, 75016 PARIS

Résumé

La tarification constitue l'un des cœurs du métier de l'actuariat. Ce mémoire aborde les méthodes utilisées en tarification en assurance IARD (incendie, accidents et risques divers), et plus spécifiquement en tarification automobile. L'enjeu est de comprendre les méthodes classiques utilisées : modèles linéaires généralisés (GLMs) et modèles additifs généralisés (GAMs).

Il est pour cela indispensable de se pencher sur les grands principes de la tarification afin de mettre en exergue un cadre général commun à toutes ces méthodes.Il sera ensuite nécessaire de comparer ces méthodes l'une après l'autre, comme il est expliqué plus tard dans l'introduction.

Dans la première partie, nous présentons les données et les traitements appliqués sur celles-ci (freMPL3 et freMPL4 réunis) à l'aide des techniques exploratoires usuelles (statistiques descriptives, ACP, AFC).

La deuxième partie met en avant les algorithmes dits linéaires, et principalement les modèles linéaires généralisés (GLM). Les GLM nous permettent de prédire la fréquence et la sévérité des sinistres et vont nous permettre de trouver une prime pure pour les polices étudiées.

La troisième partie présente les modèles additifs généralisés, et en particulier leur méthode de calibration.
Ces modèles vont nous permettre de modéliser d'une façon différente la fréquence et la sévérité des sinistres, puis de sélectionner le modèle GAM le plus adapté.

La quatrième partie aborde le choix de tarification. Nous expliquerons le choix d'utiliser une prime avec chargement. Celle-ci sera calculée via une analyse par simulation, afin que dans 99% des cas la somme des primes soit supérieure à la charge totale des sinistres du portefeuille de données. Enfin, nous allons comparer les résultats obtenus avec chacun des modèles.

Pour chaque police d'assurance, la prime est fonction de variables dites de tarification permettant de segmenter la population en fonction de son risque. Il est usuel d'utiliser une approche fréquence/sévérité ou une approche indemnitaire pour modéliser le coût annuel d'une police d'assurance. Sur les données utilisées dans ce projet, on utilisera cette dernière approche car on ne dispose pas des montants individuels de sinistre.

Mots-clefs

Tarification, modèles linéaires généralisés, GLMs, modèles additifs généralisés, GAMs, prime pure, fréquence des sinistres, sévérité des sinistres, simulation, portefeuille.

Abstract

Pricing is the heart of the actuarial profession. This thesis discusses the methods used in property and casualty insurance pricing (fire, accidents and various risks), and more specifically in automobile pricing. The challenge is to understand the classical methods used : generalized linear models (GLMs) and generalized additive models (GAMs).

It is therefore essential to look at the main principles of pricing in order to highlight a general framework common to all these methods.It will then be necessary to compare these methods one after another, as explained later in the introduction.

In the first part, we present the data and the treatments applied to the automobile base (freMPL3 and freMPL4 combined) available, using the usual exploratory techniques (descriptive statistics, PCA, AFC).

The second part highlights the so-called linear algorithms, and mainly generalized linear models (GLM). The GLMs will predict the frequency and severity of claims and then allow us to find a pure premium for the policies studied.

The third part presents generalized additive models, and in particular their calibration method. These models will allow us to model the frequency and severity of claims in a different way, then to select the most suitable GAM model.

The fourth part deals with the choice of pricing. We will explain the choice to use a premium with loading. This will be calculated via what-if analysis, so that in 99% of cases the sum of the premiums is greater than the total claims burden of the data portfolio. Finally, we will compare the results obtained with each of the models.

For each insurance policy, the premium is a function of so-called pricing variables that make it possible to segment the population according to its risk. It is usual to use a frequency / severity approach or an indemnity approach to model the annual cost of an insurance policy. On the data used in this project, we will use this last approach because we do not have the individual amounts of loss.

Keyword

Pricing, generalized linear models, GLMs, generalized additive models, GAMs, pure premium, frequency of claims, severity of claims, simulation, portfolio.

Remerciements

Nous tenons tout d'abord à remercier notre tuteur Christophe Dutang pour le temps qu'il nous a consacré pour répondre à nos diverses questions ainsi que pour l'aide qu'il nous a apporté lorsque nous avons rencontré des problèmes dans ce mémoire.

Nous remercions aussi le corps enseignant de Dauphine, puisque la plupart des notions abordées dans leur cours ont été mises en pratique dans de ce mémoire. En effet, sans un enseignement de qualité de leur part, l'écriture de ce mémoire n'aurait pas été possible.

Pour finir, nous remercions l'Université Paris-Dauphine car elle nous a permis d'avoir accès à des documents de recherches disponibles à la bibliothèque universitaire et ainsi, a pu faciliter au maximum nos recherches.

Table des matières

Introduction

Le système assurantiel est un système de production inversée.
L'assureur fixe son prix (sa prime) avant de connaître son coût (le montant total des sinistres déclarés). Ce système engendre de grandes problématiques, il faut déterminer un montant de prime assez bas afin de rester attractif, mais assez élevé pour de ne pas faire faillite.

Un contrat d'assurance est un contrat par lequel l'assureur s'engage à verser une prestation en cas de réalisation d'un risque, moyennant le paiement d'une prime ou cotisation.
L'élément si particulier du contrat d'assurance est son caractère aléatoire. L'assureur doit donc mettre en place des méthodes de prédiction de cet aléa, et déterminer une prime couvrant l'ensemble de ses frais prédits.

A travers ce mémoire, nous aborderons plusieurs méthodes de prédiction de la prime assurantielle. Le but de ce projet est de déterminer un tarif en assurance IARD à travers deux méthodes distinctes : les Modèles Linéaires Généralisés (GLM) et les Modèles Additifs Généralisés (GAM). Ces derniers sont une extension des GLM proposé par McCullagh et Nelder (1989) en considérant une approche non-paramétrique pour le prédicteur, voir Hastie et Tibshirani (1990). Plus précisément, pour un GLM, le prédicteur est une fonction linéaire des variables explicatives tandis que pour un GAM la fonction est non-paramétrique.

Afin de déterminer une prime assurantielle, nous aurons à notre disposition deux bases de données, freMPL3 et freMPL4, contenant un ensemble d'informations sur des assurés et leurs sinistres.
Ces bases de données seront réunies en une base freMPL34 afin d'avoir un grand nombre de données pour notre étude.

Nous débuterons par une analyse statistique de notre base de données. Puis nous continuerons avec la présentation des GLM et le calcul d'une prime pure par cette méthode. Ensuite, nous nous tournerons vers la présentation des modèles GAM, et nous déterminerons une seconde prime pure. Enfin, nous effectuerons une partie sur le choix de la prime, en opposant les deux modèles et en déterminant une prime avec chargement via une analyse par simulation.

Chapitre 1

Études statistiques

Afin de tarifer numériquement nos contrats, nous allons devoir étudier les jeux de données qui ont été mis à notre disposition. Durant cette partie, le but va être de déterminer la loi du coût des sinistres ainsi que la loi du fait d'avoir ou non un sinistre.

Nous disposons de deux bases de données, freMPL3 et freMPL4, disponibles dans le package R : CASdatasets.

1.1 Exploration des données freMPL3 et freMPL4

Les deux bases de données freMPL3 et freMPL4 possèdent les mêmes variables mais avec des observations différentes. Les deux bases de données freMPL3 et freMPL4 ont été réunies pour effectuer notre analyse de données, on nommera cette nouvelle base freMPL34. Nous avons ensuite scinder la base de données en deux parties, une partie apprentissage représentant 80 % des données et une partie test, comprenant 20 % des données. La base de données apprentissage nous permettra de calibrer les modèles GLM et GAM, alors que la base test nous permettra de valider les modèles, et de déterminer une tarification adaptée.

1.2 Présentation des variables

D'abord nous avons commencé par l'installation du package R CASdatasets. Pour mieux comprendre les variables de la base de données freMPL3 et freMPL4, nous avons introduit un tableau de présentation des variables afin de mieux les visualiser et étudier chaque variable une à une.

La base police freMPL3 possède 30 595 observations et 23 variables et la base sinistre freMPL4 possède 36 295 observations et 23 variables.

Par conséquent, notre variable réponse, le montant déclaré (noté « ClaimAmount » dans la base de données), n'est autre que le coût du sinistre : il s'agit du montant des dégâts matériels causés par le sinistre et estimé par l'expert.

Dans ce contexte, nous soulignons la présence d'une variable booléenne nommée : « ClaimInd », qui est un indicateur de réclamation, est intervenu suite à la survenance de la réclamation :
modalité « 1 » pour oui et modalité « 0 » pour non.

Nos bases de données contiennent : 15 variables qualitatives (dont 11 polytomiques et 4 dichotomiques), 2 variables temporelles, et 6 variables quantitatives (dont 4 discrètes et 2 continues) :

Intitulé de la variable	Type de variable	Valeurs de la variable	Descriptif
VehMaxSpeed	Qualitative (polytomique)	"1-130 km/h" à "200-220 km/h "	Vitesse maximum du vehicule
SocioCateg	Qualitative (polytomique)	"CSP1" à "CSP99"	Catégorie sociale
VehUsage	Qualitative (polytomique)	"Private", "Professional" ...	Usage du vehicule
Deductype	Qualitative (polytomique)	"Majorized", "Normal Partially" ...	Type de franchise
VehBody	Qualitative (polytomique)	"bus", "cabriolet" ...	Carosserie du vehicule
VehPrice	Qualitative (polytomique)	"A" jusqu'à "Z", "Z1"	Prix du vehicule
VehEngine	Qualitative (polytomique)	"carburation direct", "electric" ...	Type de moteur du vehicule
VehEnergy	Qualitative (polytomique)	"diesel", "electric" ...	Type d'energie du vehicule
VehClass	Qualitative (polytomique)	"0", "A", "B", "H", "M1" et "M2"	Classe du vehicule
Garage	Qualitative (polytomique)	"Collective garage", "None"	Type de garage
VehAge	Qualitative (polytomique)	"0" à "10+"	Âge du vehicule

TABLE 1.1: Variables qualitatives (polytomiques).

Intitulé de la variable	Type de variable	Valeurs de la variable	Descriptif
ClaimInd	Qualitative (dichotomique)	{0,1}	Indicateur de la réclamation
Gender	Qualitative (dichotomique)	{Male, Female}	Le genre
MariStat	Qualitative (dichotomique)	{Alone, Other}	Statut marital
HasKmLimit	Qualitative (dichotomique)	{0,1}	Limite de kilométrage

TABLE 1.2: Variables qualitatives (dichotomiques).

Intitulé de la variable	Type de variable	Valeurs de la variable	Descriptif
RecordBeg	Temporelles	JJ-MM-AAAA	Date début du contrat
RecordEnd	Temporelles	JJ-MM-AAAA	Date fin du contrat

TABLE 1.3: Variables temporelles.

Intitulé de la variable	Type de variable	Valeurs de la variable	Descriptif
Exposure	Quantitative (continue)	[0,1]	Exposition au risque dans l'année
LicAge	Quantitative (discrète)	Entier	Âge du permis de conduire en mois
BonusMalus	Quantitative (discrète)	Entier de 50 à 250	Bonus si > 100 et Malus si < 100
DrivAge	Quantitative (discrète)	Entier de 18 à 97	Âge du conducteur
ClaimAmount	Quantitative (continue)	Réel	Le montant déclaré
RiskVar	Quantitative (discrète)	Entier de 1 à 20	Variable du risque

TABLE 1.4: Variables quantitatives (discètes et continues).

1.3 Vue d'ensemble des données

Afin de mieux comprendre la constitution de notre jeu de données, nous allons brièvement décrire nos variables et visualiser graphiquement la fréquence de sinistralité et le coût de l'expertise en fonction des différentes modalités.

Nos bases de données freMPL3 et freMPL4 disposent de la même data.frame.

- Visualisation du data frame :

```
'data.frame':   30595 obs. of  23 variables:
 $ Exposure   : num  0.583 0.2 0.083 0.375 0.5 0.499 0.218 0.75 0.249 0.75 ...
 $ LicAge     : int  366 187 169 170 224 230 169 232 241 298 ...
 $ RecordBeg  : Date, format: "2004-06-01" "2004-10-19" "2004-07-16"  ...
 $ RecordEnd  : Date, format: NA NA "2004-08-16" NA ...
 $ VehAge     : Factor w/ 9 levels "0","1","10+",..: 4 1 2 2 5 5 8 6 7 4 ...
 $ Gender     : Factor w/ 2 levels "Female","Male": 1 2 1 1 2 2 2 1 1 2 ...
 $ MariStat   : Factor w/ 2 levels "Alone","Other": 2 1 2 2 2 2 2 2 2 2 ...
 $ SocioCateg : Factor w/ 37 levels "CSP1","CSP16",..: 1 23 1 1  ...
 $ VehUsage   : Factor w/ 4 levels "Private","Private+trip to office",..: 3 2 ...
 $ DrivAge    : int  55 34 33 34 53 53 32 38 39 43 ...
 $ HasKmLimit : int  0 0 0 0 0 0 0 0 0 0 ...
 $ DeducType  : Factor w/ 4 levels "Majorized","Normal",..: 2 2 2 2 1 ...
 $ BonusMalus : int  72 80 63 63 72 68 50  ...
 $ VehBody    : Factor w/ 9 levels "bus","cabriolet",..: 6 4 5  ...
 $ VehPrice   : Factor w/ 26 levels "A","B","C","D",..: 4 11 12 ...
 $ VehEngine  : Factor w/ 6 levels "carburation",..: 5 2 2 2 2 2 5 5 5 5 ...
 $ VehEnergy  : Factor w/ 4 levels "diesel","eletric",..: 4 1 1 1  ...
 $ VehMaxSpeed: Factor w/ 10 levels "1-130 km/h","130-140 km/h",..: 5 6 6 6 ...
 $ VehClass   : Factor w/ 6 levels "0","A","B","H",..: 3 5 5 5 1 1 3 2 2 6 ...
 $ RiskVar    : int  15 20 17 17 19 19 19 19 19 10 ...
 $ ClaimAmount: num  0 0 0 0 0 0 0 0 0 0 ...
 $ Garage     : Factor w/ 3 levels "Collective garage",..: 2 2 2 3 2 ...
 $ ClaimInd   : int  0 0 0 0 0 0 0 0 0 0 ...
```

D'après le tableau ci-dessous, nous avons une vue d'ensemble de la base freMPL3 :

Exposure Min. :0.0010 1st Qu. :0.1910 Median :0.4160 Mean :0.4474 3rd Qu. :0.6730 Max. :1.0000	**VehMaxSpeed** 160-170 km/h :5297 170-180 km/h :4830 180-190 km/h :4677 150-160 km/h :3863 190-200 km/h :3672 200-220 km/h :3331 (Other) :4925	**RecordBeg** Min. :2004-01-01 1st Qu. :2004-01-01 Median :2004-03-01 Mean :2004-04-14 3rd Qu. :2004-07-12 Max. :2004-12-31	**RecordEnd** Min. :2004-01-02 1st Qu. :2004-04-01 Median :2004-07-01 Mean :2004-07-04 3rd Qu. :2004-10-01 Max. :2004-12-31 NA's :14143
SocioCateg CSP50 :17608 CSP60 : 3882 CSP55 : 2661 CSP1 : 1827 CSP48 : 853 CSP42 : 776 (Other) : 2988	**VehAge** 2 :4902 1 :4723 0 :4722 3 :3856 4 :3348 6-7 :2926 (Other) :6118	**ClaimInd** Min. :0.00000 1st Qu. :0.00000 Median :0.00000 Mean :0.04076 3rd Qu. :0.00000 Max. :1.00000	**VehEnergy** diesel : 9438 eletric : 6 GPL : 2 regular :21149
VehPrice K : 2976 J : 2829 L : 2529 F : 2417 H : 2397 G : 2392 (Other) :15055	**VehClass** 0 : 759 A :2991 B :9567 H :4894 M1 :7745 M2 :4639	**LicAge** Min. : 0 1st Qu. :163 Median :282 Mean :301 3rd Qu. :425 Max. :940	**ClaimAmount** Min. : 0.0 1st Qu. : 0.0 Median : 0.0 Mean : 75.8 3rd Qu. : 0.0 Max. :23229.8
RiskVar Min. : 1.00 1st Qu. :10.00 Median :15.00 Mean :13.19 3rd Qu. :16.00 Max. :20.00	**BonusMalus** Min. : 50.00 1st Qu. : 50.00 Median : 54.00 Mean : 64.27 3rd Qu. : 76.00 Max. :272.00	**VehBody** sedan :20140 sport utility vehicle : 1858 other microvan : 1679 station wagon : 1629 microvan : 1374 cabriolet : 1343 (Other) : 2572	**DrivAge** Min. :18.00 1st Qu. :34.00 Median :45.00 Mean :46.25 3rd Qu. :57.00 Max. :97.00
VehUsage Private : 9956 Private+trip :13522 Professional : 6523 Professional run : 594	**VehEngine** carburation : 516 direct injection : 7037 electric : 6 GPL : 2 injection :20821 injection overp : 2213	**DeducType** Majorized : 2820 Normal :23226 Partially refunded : 4441 Proportional : 108	**HasKmLimit** Min. :0.0000 1st Qu. :0.0000 Median :0.0000 Mean :0.1094 3rd Qu. :0.0000 Max. :1.0000
Garage Collective garage : 6623 None :20011 Private garage : 3961	**Gender** Female :11570 Male :19025	**MariStat** Alone :7424 Other :23171	

TABLE 1.5: Tableau de vue d'ensemble des données.

1.4 Statistiques descriptives

Tout d'abord nous allons commencer par analyser de facon générale les jeux de données afin de comprendre quelles données sont liées entre elles.
Le jeu de données que nous allons observer sera freMPL34 (fusion de freMPL3 et freMPL4).

L'analyse des données offre une meilleure visualisation du jeu de données et permet de comprendre les éventuelles liaisons qui existent entre les différentes variables. Nous nous intéressons en particulier aux relations qui peuvent exister entre les variables.

1.4.1 Analyse des variables quantitatives

Il convient de vérifier la nature des liaisons qui existent entre les variables quantitatives et la variable réponse "claimAmount". La corrélation est une mesure qui permet de détecter les liaisons linéaires qui peuvent exister entre deux variables quantitatives. Elle est donnée par la formule suivante :

$$r = \frac{cov(x,y)}{\sigma_x \sigma_y}.$$

Où :
→ r représente le coefficient de corrélation entre les variables x et y ;
→ $cov(x, y)$ désigne la covariance des variables x et y ;
→ σ_x, σ_y les écart-types respectifs des deux variables.

Pour rappel :
- Si sa valeur est égale à 1 (ou -1), alors nous pouvons déduire une relation linéaire positive (ou négative) entre les variables en question ;
- Si par contre elle vaut zéro, nous pouvons en déduire qu'il n'y a pas d'association linéaire entre les deux variables.

Dans le tableau ci-dessous, nous retrouvons les coefficients de corrélation entre chacune des variables explicatives quantitatives et la variable à expliquer :

Variable	Coefficient de corrélation de Pearson
ClaimAmount, Exposure	0.062495143
ClaimAmount, LicAge	0.004919227
ClaimAmount, DrivAge	0.007572122
ClaimAmount, HasKmLimit	-0.004751432
ClaimAmount, BonusMalus	0.03387802
ClaimAmount, RiskVar	-0.005774936

TABLE 1.6: Tableau des coefficients de correclation.

La commande **cor()** sur R génère des coefficients de corrélation assez faibles voire négligeables, comme le témoignent les résultats du tableau. Ceci nous amène à mettre en cause l'existence d'une relation linéaire entre les variables étudiées et à nous interroger sur l'existence d'une relation de dépendance non linéaire entre les variables explicatives quantitatives et la variable à expliquer.

1.4.2 Liaisons entre les variables quantitatives

Avant de procéder à la modélisation, il est prudent de vérifier qu'il n'existe pas de paires e variables explicatives ayant des liaisons assez fortes entre elles. Nous avons calculé les V de Cramer de toutes les paires de variables explicatives quantitatives. Les résultats détaillés ci-dessous nous permettent de détecter facilement les paires de variables « fortement » corrélées :

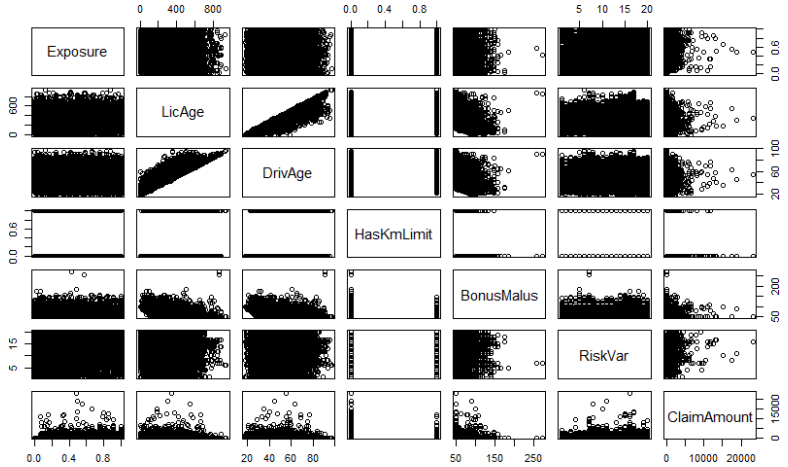

FIGURE 1.1: Représentation des corrélations V Cramer.

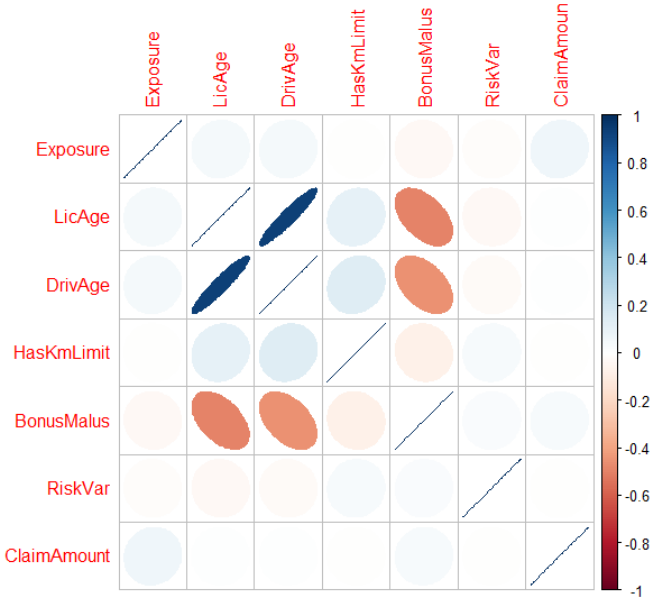

FIGURE 1.2: Représentation du Corrélogramme.

Représentation des corrélations : plus l'ellipse ressemble à un cercle et moins les variables sont corrélées. Plus l'ellipse ressemble à une droite et plus les variables sont corrélées.

On remarque une forte correclation entre **DriveAge** et **LicAge**, donc il ne faut pas garder les deux variables dans notre modèle, sinon on va se retrouver dans le cas du sur-apprentissage de notre modèle.

1.4.3 Analyse des variables qualitatives

La base de données présente 4 variables Dichotomiques, avec 3 variables explicatives et une variable à expliquer : **ClaimInd**.
Elles sont réparties comme présentées ci-dessous :

Variable	Observation	Ont eu un sinistre	En %
Gender	25 358 Female 41 532 Male	637 Female 972 Male	2,51% Female 2,34% Male
HasKmLimit	59 276 égaux à 0 7 614 égaux à 1	1 463 égaux à 0 146 égaux à 1	2,47% égaux à 0 1,92% égaux à 1
MariStat	16 944 Alone 49 946 Other	426 Alone 1 183 Other	2,51% Alone 2,37% Other

FIGURE 1.3: Tableau des Variables dichotomiques.

En moyenne, 2,41 % des assurés ont un sinistre (65 281 fois ClaimInd est égal à 0 et 1609 à 1).
A travers les chiffres du tableau, on peut observer par exemple qu'en moyenne les femmes ont tendance à avoir plus d'accidents que les hommes.

Nous avons à notre disposition 11 variables polytomiques.
Tout d'abord, nous avons recodé plusieurs modalités de ces variables. En effet, certaines présentaient trop peu de données en une modalité, ou trop de modalités pour une seule variable.
Nous expliquons ces changements dans ce tableau :

Variables modifiées	Anciennes modalités	Nouvelles modalités
VehEngine	electric, carburation direct ...	electric et GPL retirés
SocioCateg	CSP1 à CSP 99	Regroupé par dizaines : CSP1, CSP2, CSP3, ...
VehPrice	A jusqu'à Z, et Z1	A jusqu'à Q
VehMaxSpeed	1-130km/h jusqu'à 200-220km/h	1-130km/h et 130-140km/h réunis
VehEnergy	diesel, electric, ...	electric et GPL retirés

FIGURE 1.4: Modification de modalités de variables polytomiques.

Les différentes variables polytomiques et leurs statistiques sur le fait d'avoir un sinistre ou non sont présentés dans les tableaux ci-dessous.
Ils nous éclaire sur le pourcentage de sinistre obtenu selon les modalités des variables qualitatives polytomiques.

Variable	Observation	Ont eu un sinistre	En %
DeducType	6825 Majorized 50014 Normal 6288 Partially refunded 232 Proportional 3515 Refunded	162 Majorized 1176 Normal 212 Partially refunded 2 Proportional 56 Refunded	2,37% Majorized 2,35% Normal 3,37% Partially refunded 0,86% Proportional 1,59% Refunded
VehAge	9258 « 0 » 9017 « 1 » 9468 « 2 » 7773 « 3 » 6979 « 4 » 6058 « 5 » 7139 « 6-7 » 5128 « 8-9 » 6054 « 10+ »	265 « 0 » 252 « 1 » 260 « 2 » 202 « 3 » 199 « 4 » 146 « 5 » 142 « 6-7 » 87 « 8-9 » 55 « 10+ »	2,86% « 0 » 2,79% « 1 » 2,75% « 2 » 2,60% « 3 » 2,85% « 4 » 2,41% « 5 » 1,99% « 6-7 » 1,70% « 8-9 » 0,91% « 10+ »
SocioCateg	3896 csp1 1949 csp2 1053 csp3 5303 csp4 44 644 csp5 9 889 csp6 126 csp7 14 csp9	81 csp1 65 csp2 23 csp3 141 csp4 1 010 csp5 282 csp6 6 csp7 0 csp9	2,08% csp1 3,36% csp2 2,18% csp3 2,66% csp4 2,26% csp5 2,85% csp6 4,76% csp7 0% csp9
VehUsage	21 858 Private 30 059 Private+trip to office 13 729 Professional 1 228 Professional run	489 Private 703 Private+trip to office 377 Professional 39 Professional run	2,24% Private 2,34% Private+trip to office 2,75% Professional 3,18% Professional run

FIGURE 1.5: Statistiques des variables polytomiques 1.

Variable	Observation	Ont eu un sinistre	En %
Garage	13 789 Collective garage 44 904 None 8 184 Private garage	313 Collective garage 1 134 None 161 Private garage	2,27% Collective garage 2,52% None 1,97% Private garage
VehMaxSpeed	3 133 « 1-140 km/h » 3 572 « 140-150km/h » 8 923 « 150-160km/h » 11 586 « 160-170km/h » 10 710 « 170-180 km/h » 9 725 « 180-190 km/h » 7 717 « 190-200 km/h » 6 808 « 200-220 km/h » 4 700 « 220+ km/h »	52 « 1-140 km/h » 91 « 140-150km/h » 250 « 150-160km/h » 297 « 160-170km/h » 244 « 170-180 km/h » 200 « 180-190 km/h » 201 « 190-200 km/h » 179 « 200-220 km/h » 94 « 220+ km/h »	1,66% « 1-140 km/h » 2,55% « 140-150km/h » 2,80% « 150-160km/h » 2,56% « 160-170km/h » 2,28% « 170-180 km/h » 2,06% « 180-190 km/h » 2,60% « 190-200 km/h » 2,63% « 200-220 km/h » 2% « 220+ km/h »
VehBody	319 bus 2 737 cabriolet 2 793 coupe 2 779 microvan 3 450 other microvan 45 131 sedan 3 783 sport utility vehicle 3 384 station wagon 2 498 van	7 bus 77 cabriolet 65 coupe 75 microvan 77 other microvan 1 118 sedan 67 sport utility vehicle 64 station wagon 58 van	2,19 bus 2,81 cabriolet 2,33 coupe 2,70 microvan 2,23 other microvan 2,48 sedan 1,77 sport utility vehicle 1,89 station wagon 2,32 van
VehEnergy	20 064 diesel 46 810 regular	500 diesel 1 008 regular	2,49% diesel 2,15% regular

FIGURE 1.6: Statistiques des variables polytomiques 2.

Variable	Observation	Ont eu un sinistre	En %
VehEngine	2 209 carburation direct 13 830 injection overpowered 45 874 injection 4 961 injection overpowered	15 carburation direct 369 injection overpowered 1 122 injection 102 injection overpowered	0,68% carburation direct 2,67% injection overpowered 2,45% injection 2,06% injection overpowered
VehPrice	2 070 A 4 054 D 5 096 E 5 415 F 5 410 G 5 276 H 4 907 I 6 049 J 6 289 K 5 229 L 4 228 M 3 095 N 2 593 O 2 148 P 5 015 Q	41 A 106 D 130 E 149 F 138 G 112 H 123 I 164 J 141 K 134 L 93 M 96 N 51 O 42 P 88 Q	1,98% A 2,61% D 2,55% E 2,75% F 2,55% G 2,12% H 2,51% I 2,71% J 2,24% K 2,56% L 2,20% M 3,10% N 1,97% O 1,96% P 1,75% Q
VehClass	1 774 0 6 591 A 21 234 B 10 323 H 16 881 M1 10 071 M2	37 0 222 A 519 B 200 H 414 M1 216 M2	2,08% 0 3,37% A 2,44% B 1,94% H 2,45% M1 2,14% M2

FIGURE 1.7: Statistiques des variables polytomiques 3.

1.4.4 Analyse en composantes principales (ACP)

L'analyse en composantes principales (ACP ou PCA en anglais pour principal component analysis), ou selon le domaine d'application la transformation de Karhunen–Loève (KLT), est une méthode de la famille de l'analyse des données et plus généralement de la statistique multivariée, qui consiste à transformer des variables liées entre elles (dites « corrélées » en statistique) en nouvelles variables décorrélées les unes des autres. Ces nouvelles variables sont nommées « composantes principales », ou axes principaux. Elle permet au praticien de réduire le nombre de variables et de rendre l'information moins redondante.

L' ACP est une méthode descriptive permettant de représenter graphiquement l'essentielle de l'information contenu dans le tableau des données quantitatives.

Le principe de l'ACP est de trouver des espaces de petites dimensions sur lesquels les projections des individus minimisent la déformation de la réalité.

Les vecteurs propres sont appelés les axes principaux, le premier axe principal est associé à la plus grande valeur propre, le deuxième axe principal est associé à la deuxième valeur propre, etc... .

La projection des individus sur un axe principal est une nouvelle variable appelée composante principale, la première composante représente les coordonnées des projections des individus sur le premier axe principal, la deuxième composante représente les coordonnées des projections des individus sur le deuxième axe principal, etc ... .

Validité des représentations graphiques :
→ La projection perd le moins d'information possible (vérifier le pourcentage d'inertie expliquée par l'axe et conserver le nombre d'axes nécessaires pour avoir une inertie expliquée correcte)
→ Les variables sont bien représentées si elles sont proches du cercle, par contre celles qui sont proches de l'origine sont peu corrélées avec les axes (pas d'interprétation possible pour ces variables).
→ Les individus sont bien représentés s'ils ne sont pas trop éloignés de l'axe sur lequel on les projette.
→ Eliminer les individus ayant une contribution trop importante dans la construction de l'axe (vérifier la contribution des individus).

Pour réaliser cette analyse, nous avaons scinder la matrice de données. Nous allons utiliser une base de données contenant uniquement les variables quantitatives pour l'ACP. On a réalisé une ACP sur les variables quantitatives (Exposure, LicAge, BonusMalus, DrivAge, RiskVar et ClaimAmount).

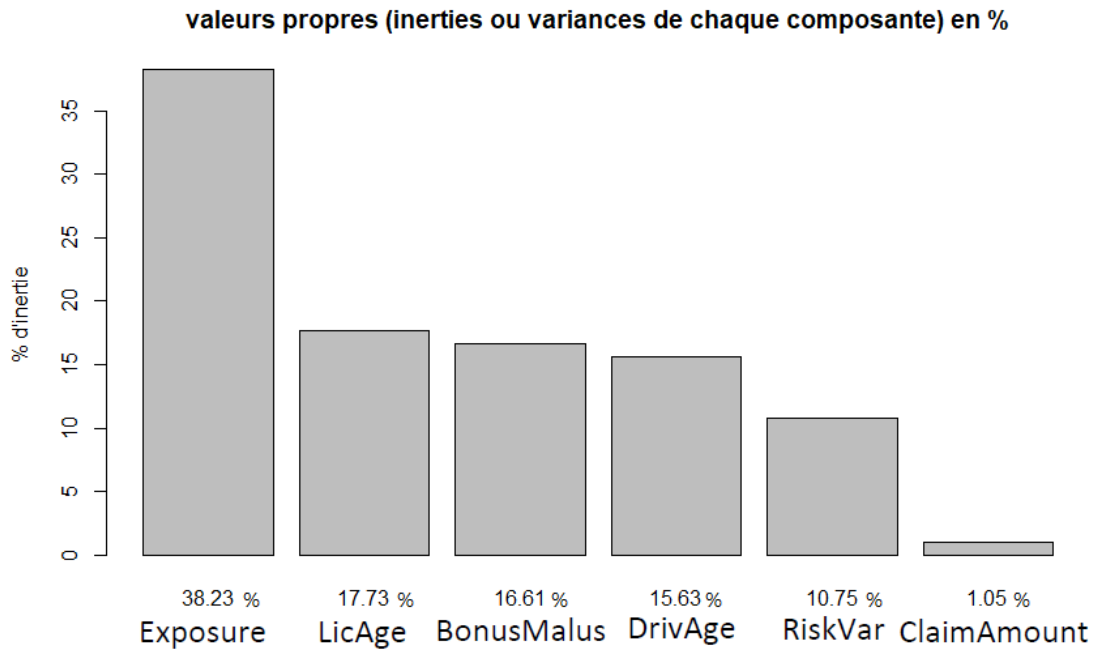

FIGURE 1.8: Valeurs propres en pourcentage.

Les valeurs propres sont calculées sur la matrice de corrélation avec la fonction **valprop** sur R. L'inertie expliquée par la i-ème composante principale, qui est associée à la i-ème plus grande valeur propre.

Chaque valeur propre représente la variance du facteur correspondant. Un facteur est une combinaison linéaire des variables initiales dans laquelle les coefficients sont données par les coordonnées des vecteurs propres (changement de base).

On considère la représentation de ces variables dans le cercle de corrélation ci-dessous :

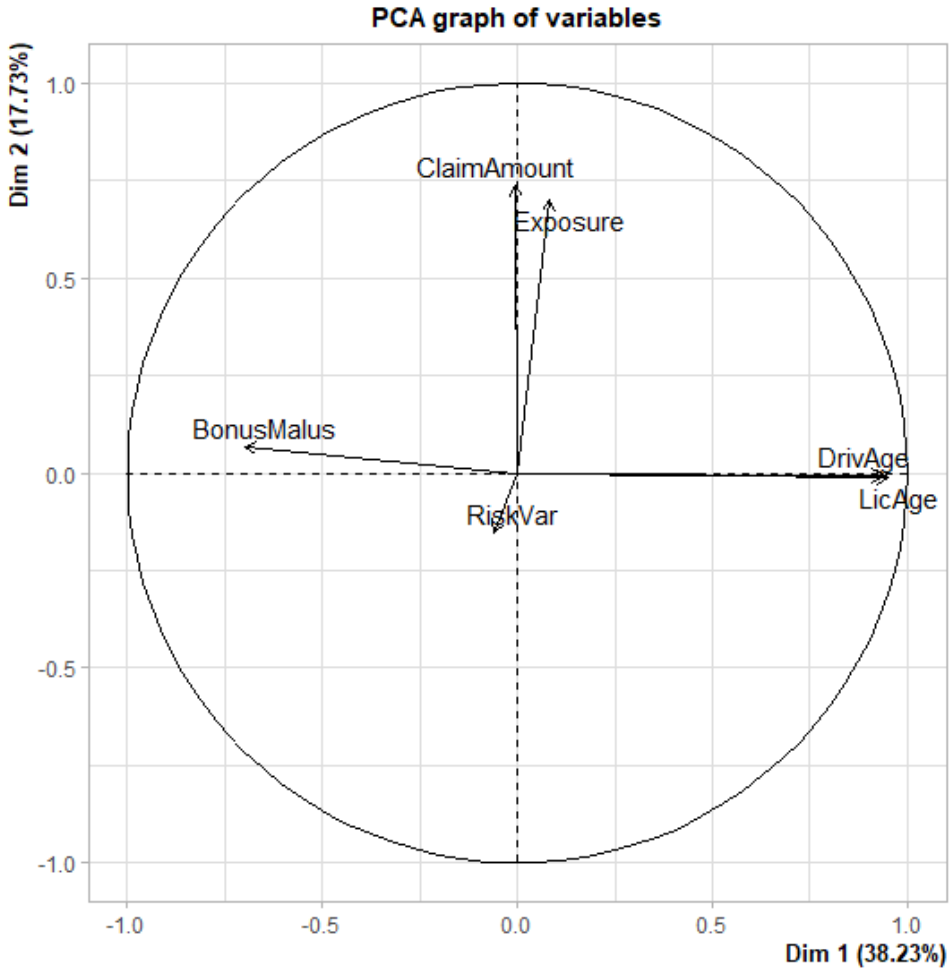

FIGURE 1.9: Cercle de l'ACP.

Interpretation :
→ On distingue 4 groupes : (ClaimAmount,Exposure), (BonusMalus), (DriveAge,LicAge) et (RiskVar).
→ les variables qui ne doivent pas être interprétées sur cette figure sont les variables représentées par des points trop éloignés du cercle des corrélations (proches de 0) : **RiskVar**.
→ Les variables représentées par des points proches du cercle des corrélations et proches entre elles sont fortement corrélées positivement.
→ **DriveAge** et **LicAge** sont fortement corrélées positivement entre elles.
→ **ClaimAmount** et **Exposure** sont corrélées positivement entre elles.
→ Les 4 groupes sont peu corrélées entre elles.
→ Deux variables représentées par des points proches du cercle des corrélations et formant avec 0 un angle droit (ou presque droit) ne sont pas corrélées entre elles (ou sont peu corrélées entre elles).
→ **ClaimAmount** et **(DriveAge,LicAge)** ne sont pas corrélés.
→ **ClaimAmount** et **BonusMalus** sont peu corrélés.
→ Deux variables représentées par des points proches du cercle des corrélations et formant avec 0 un angle plat (ou presque plat) sont fortement corrélées négativement entre elles.
→ On observe que les variables fortement corrélées négativement avec **BonusMalus** sont **DrivAge** et **LicAge**.

1.4.5 Analyse Factorielle des Correspondances (AFC)

L'analyse factorielle des correspondances (AFC ou CA pour correspondence analysis en anglais) est une extension de l'analyse en composantes principales pour analyser l'association entre deux variables qualitatives (ou catégorielles).

L'AFC permet de résumer et de visualiser l'information contenue dans le tableau de contingence formé par les deux variables catégorielles. Le tableau de contingence contient les fréquences formées par les deux variables.

L'AFC retourne les coordonnées des éléments des colonnes et des lignes du tableau de contingence. Ces coordonnées permettent de visualiser graphiquement l'association entre les éléments des lignes et des colonnes d'un un graphique à deux dimensions.

Lors de l'analyse d'un tableau de contingence, une question typique est de savoir si certains éléments lignes sont associés à certains éléments colonnes. L'analyse factorielle par correspondance est une approche géométrique pour visualiser les lignes et les colonnes d'une table de contingence dans un graphique en nuage de points. Ainsi, les positions des points lignes et celles des points colonnes correspondent à leurs associations dans le tableau.

Format des données : Les données doivent être un tableau de contingence.

variables qualitatives dichotomiques (ClaimInd, Gender, MariStat et HasKmLimit) :

Les données correspondent à un tableau de contingence contenant "$Male$" et "$Female$" (Gender), "$Alone$" et "$Other$" (MariStat), "$HasKmLimit = 0$" et "$HasKmLimit = 1$" et leur répartition selon le "ClaimInd" :

	ClaimInd=0	ClaimInd=1
Male	18263	762
Female	11085	483
Alone	7113	311
Other	22235	936
HasKmLimit=0	26111	1138
HasKmLimit=1	3237	109

TABLE 1.7: Tableau de contingence des variables qualitatives dichotomiques.

Tableaux de contingence :

Le tableau de contingence ci-dessus n'est pas très gros. Par conséquent, il est facile d'inspecter et d'interpréter visuellement les profils des lignes et des colonnes : Il est évident qu'il y a plus de "$ClaimInd = 1$" en proportion pour les femmes que pour les hommes et toujours en proportion moins de "$ClaimInd = 1$" pour les vehicules au kilométrage limité.

Graphique du tableaux de contingence :

Le tableau de contingence peut être visualisé via un graphique.

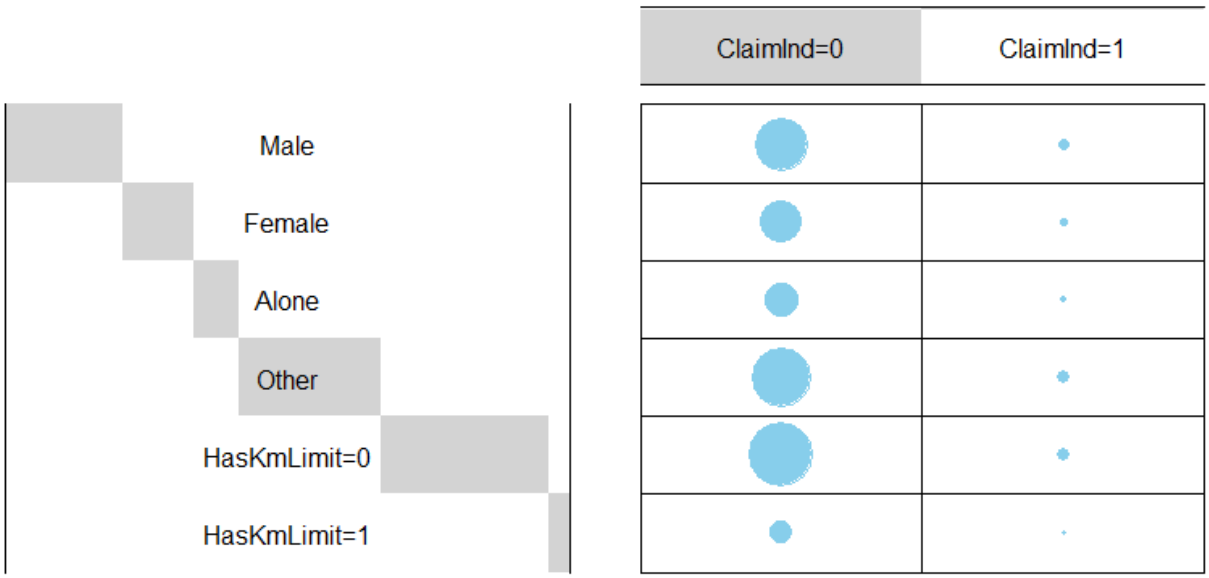

FIGURE 1.10: Graphique du tableaux de contingence.

Test de χ^2 d'indépendance :

Pour un petit tableau de contingence, vous pouvez utiliser le test du χ^2 pour évaluer s'il existe une dépendance significative entre les catégories des lignes et des colonnes :

```
        Pearson's Chi-squared test with Yates' continuity correction

data:  tableau
X-squared = 0.7.2925, df = 5, p-value = 0.1998
```

Les variables de ligne et de colonne ne sont pas statistiquement significativement associées car (p-value $> 5\%$).

Valeurs propres / Variances :

L'examination des valeurs propres permet de déterminer le nombre d'axes principaux à considérer. Les valeurs propres correspondent à la quantité d'informations retenue par chaque axe. Elles sont grandes pour le premier axe et petites pour l'axe suivant.

Les valeurs propres et la proportion de variances expliquées (capturées) par les différents axes :

```
       eigenvalue variance.percent cumulative.variance.percent
Dim.1 1.741862e-05              100                         100
```

On dispose d'un seul axe donc on ne peut pas extraire le graphe de l'AFC pour les variables qualitatives dichotomiques.

Chapitre 2

Présentation des Modèles linéaires généralisés (GLMs)

But : Meilleure prédiction de $y|x$.

Problème : On ne peut pas traiter le cas d'une variable réponse y qualitative dans le modèle linéaire classique (classification).

Solution : Introduire de nouveaux modèles mais en gardant la linéarité en x, $g(E[y|x]) = x^T\beta$.

2.1 Introduction

On dispose de n observations indépendantes $(y_i, x_i)_{i=1,...,n}$ òu $x_i \in \mathbb{R}^r$ et $y_i \in \mathbb{Y} \subset \mathbb{R}$ pour tout $i \in \{1, ..., n\}$. On définit la matrice *design* X comme suit :

$$X = \begin{pmatrix} x_1^1 & \cdots & x_1^r \\ \vdots & & \vdots \\ x_n^1 & \cdots & x_n^r \end{pmatrix} = (X^1, ..., X^r) \text{ et } Y = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}.$$

ou $X^k, k \in \{1, ..., r\}$ sont les variables explicatives. Le modèle linéaire classique permet de décrire une relation linéaire entre la variable d'intérêt et des covariables. Cependant, utiliser le modèle linéaire requiert le respect des postulats **[P1]**,**[P2]**,**[P3]** et **[P4]**.

Tout d'abord la relation linéaire entre y et les covariables (**[P1]** : $E[y_i] = x_i\beta$) implique que y puisse prendre ses valeurs dans $\mathbb{R}$ tout entier. Si la variable $y_i \in [0, 1]$ ou $y_i \in \{0, 1\}$, et que nous utilisons le modèle linéaire classique alors nous prédirons des valeurs dans $\mathbb{R}$ ce qui semble aberrent.

Par ailleurs, nous avons supposé que les observations sont la réalisation d'une variable gaussienne (**[P4]**). Cette hypothèse permet d'écrire des tests et des intervalles de cofiance de niveau exact.

Grâce au théorème central limite, il est possible d'étendre les résultats sur les tests au cas de résidus non-gaussien. Cependant, si la variable y prend un nombre fini ou dénombrable de valeurs, cette hypothèse de gaussianité n'est plus tenable. Par ailleurs, (**[P3]**) supposait que les y_i étaient tous de même variance, ainsi que (**[P2]**) supposait que les erreurs sont de variance constante, ce qui peut ne pas être le cas en général.

2.2 Définition du modèles linéaire généralisé (GLM)

Le modèle linéaire généralisé (GLM) est une extension du modéle linéaire permettant de s'affranchir des postulats **[P1]**,**[P2]** et **[P3]** et de traiter des observations dont la loi de probabilité appartient à une famille de lois élargie.

Soient $y_1, y_2, ..., y_n$ n observations indépendantes d'une variable quantitative. D'autre part, pour chaque observation i, on dispose de p variables explicatives $(x_i^1, x_i^2, ..., x_i^p)$ réelles. On cherche à "expliquer" y_i comme une fonction des $(x_i^1, x_i^2, ..., x_i^p)$.

Le modèle linéaire généralisé est la donnée d'une loi de probabilité pour les y_i et d'une fonction g appelée fonction de lien telle que :

$$g(E[y_i|x_i]) = x_i^T \beta.$$

Cela permet d'établir une relation non linéaire entre l'espérance de la variable à expliquer et les variables explicatives et d'envisager des observations de nature variée comme des données de présence/absence, des taux de succés pour des traitements, des données de comptage d'espéces, ou encore des durées de vie ou autres variables positives dissymétriques.

Comme dans le cas du modèle linéaire, les variables explicatives peuvent être quantitatives (régression), qualitatives (anova) ou les deux (ancova).

Les GLM ont fait leur apparition dans Nelder et Wedderburn [1972]. Ils sont adaptés à de nombreuses problématiques et sont d'utilisation courante dans le domaine de la statistique et de l'actuariat (cf. Denuit et Charpentier [2005]). La théorie des GLMs bénéficie d'un avantage par rapport aux modèles linéaires classiques : le caractère normal de la variable à expliquer Y n'est plus imposé, seule l'appartenance à une famille exponentielle est indispensable.

2.3 Famille exponentielle naturelle

La famille exponentielle naturelle est une famille de lois de probabilité qui contient entre autres des lois aussi usuelles que la loi normale, la loi de Bernoulli, la loi binomiale, la loi de Poisson, la loi Gamma ... Ces lois ont en commun une écriture sous forme exponentielle qui va permettre d'uniér la présentation des résultats.

Soit f_Y la densité de probabilité de la variable Y. f_Y appartient à la famille exponentielle naturelle si elle s'écrit sous la forme :

$$f_Y(y) = exp(\tfrac{1}{\gamma(\phi)}(y\theta - b(\theta)) + c(y, \phi)).$$

Avec c est une fonction dérivable, b est trois fois dérivable et sa dérivée première b' est inversible. Le paramètre θ réel est appelé paramètre naturel de la loi. ϕ est un paramétre appelé paramétre de nuisance ou de dispersion. Dans ce cas on a :

$$E[Y] = \mu = b'(\theta) \text{ et } V(Y) = b''(\theta)\gamma(\phi).$$

Les lois : Bernoulli/Binomiale, Poisson, Gamma, Gaussienne sont de la familles exponentielle naturelle.

2.4 Choix du modèle et de la fonction de lien

Ecrire un modèle linèaire généralisé requiert le choix de deux éléments :
1. D'abord choisir une loi de probabilité pour les variables aléatoires Y_i au sein de la famille exponentielle naturelle. Ce choix est guidé par la nature du problème.
2. Ensuite, modéliser le lien entre l'espérance des Y_i et les variables explicatives au travers d'une fonction g inversible : $g(E[Y_i|x_i]) = x_i^T\beta$.

Choix de la fonction de lien : Toute bijection de l'espace de $E[Y]$ dans R peut être choisie comme fonction de lien. Cependant, très souvent on choisit comme fonction de lien la fonction qui transforme l'espérance $E[Y]$ en paramétre naturel : $g = (b')^{-1}$, g ainsi définie est appelée fonction de lien canonique.

En pratique on utilise souvent en assurance :
- la fonction de lien **log**, qui permet d'avoir un tarif multiplicatif ;
- la loi de Poisson, la loi binomiale négative ou binomiale pour la fréquence ;
- la loi gamma ou log-normale pour le coût.

Le choix de la fonction de lien est une liberté supplémentaire dans la démarche de modélisation. le choix spécifique de la fonction de lien canonique (ou naturel) est motivé par des considérations théoriques. En effet, il permet d'assurer la convergence de l'algorithme d'estimation utilisé classiquement (algorithme de Newton-Raphson) vers le maximum de vraisemblance. En pratique, si aucune raison de choisir une fonction de lien spécifique ne s'impose, le choix par défaut consiste à choisir la fonction de lien canonique.

Choix de loi de distribution :
- Si $y \in \mathbb{R}$ alors $y \sim N(\mu, \sigma^2)$ (modèle linéaire).
- Si $y \in \mathbb{R}^+$ alors $y \sim Gamma(\alpha, \beta)$ ou $y \sim Exponentiel(\lambda)$.
- Si $y \in \mathbb{N}$ alors $y \sim Poisson(\lambda)$ (regression poissonière).
- Si $y \in \mathbb{E}$ avec $Card(\mathbb{E}) = n$ fini, alors $y \sim Binomiale(n, p)$ (regression logistique).
- Si $y \in \{0, 1\}$ alors $y \sim Bernoulli(p)$ ie $y \sim Binomiale(1, p)$ (regression logistique).

Le Tableau suivant présente quelques modèles linéaires usuels (les plus connus). A chaque choix de la loi de $Y|X = x$ correspond une fonction de lien canonique g(.) qui donne son nom à la regression.

Choix de la loi de $Y\|x$	Bernoulli/ Binomial	Poisson	Gamma	Gausienne
Fonction de lien canonique	$g(\mu) = \text{logit}(\mu) = \log\left(\frac{\mu}{1-\mu}\right)$	$g(\mu) = \log(\mu)$	$g(\mu) = -\frac{1}{\mu}$	$g(\mu) = \mu$
Nom du lien	logit	log	réciproque	identité

TABLE 2.1: GLM usuels. $E[u] = \mu = x_i^T\beta$.

Il existe d'autre fonctions liens non canoniques utilisés en pratique :
→ lien **Probit** : $g(u) = \Phi^{-1}(u)$ ou Φ la fonction de répartition de loi normale centrée réduite.
→ lien **log-log** : $g(u) = log(-log(1-u))$.

Les choix de fonction de lien courants pour les distributions **Gamma** et **Binomiale** :

• Binomial	• Gamma
– Logit: $g(\mu) = \log \frac{\mu}{1-\mu}$ – Probit: $g(\mu) = \Phi^{-1}(\mu)$ – Complementary Log-Log link: $g(\mu) = \log(-\log(\mu))$ – Log: $g(\mu) = \log \mu$	– Inverse: $g(\mu) = \frac{1}{\mu}$ – Log: $g(\mu) = \log \mu$ – Identity: $g(\mu) = \mu$

TABLE 2.2: Liens des les Gamma et binomiale.

2.5 Estimateur du Maximum de Vraisemblance

Soit f_Y la densité de probabilité de la variable Y. f_Y appartient à la famille exponentielle naturelle. La log vraisemblance de $Y \in \mathbb{R}^n$:

$$l(\beta) = \ln(f_{\theta,\gamma(\phi)}(Y)) = \sum_{i=1}^n \ln(f_{\theta_i,\gamma(\phi)}(y_i)) = \sum_{i=1}^n (\frac{1}{\gamma(\phi)}(y_i\theta_i - b(\theta_i)) + c(y_i,\phi)) = \sum_{i=1}^n l_i(\beta).$$

2.6 Equations de vraisemblance

Les équations de vraisemblance sont :

$$\frac{\partial l(\beta)}{\partial \beta_i} = \sum_{i=1}^n (\frac{1}{Var(y_i|x_i)}(y_i - b'(x_i^T\beta))(g^{-1})'(x_i^T\beta)x_{i,j}) = 0\,;\, \forall j \in \{1,...,r\} \text{ avec } \theta_i = x_i^T\beta.$$

Sous forme matricielle, le gradient s'écrit :

$$\nabla l(\beta) = (\frac{\partial l(\beta)}{\partial \beta_1}, ..., \frac{\partial l(\beta)}{\partial \beta_r})^T = 0_r.$$

Lorsque le lien choisit est le lien canonique ou $g^{-1} = b'$, les équations de vraisemblance se simplifient de la facon suivante :

$$\sum_{i=1}^n x_i^j \frac{1}{\gamma(\phi)}(Y_i - b'(x_i\beta)) = \sum_{i=1}^n \frac{1}{\gamma(\phi)} x_i^j (Y_i - E[Y_i|x_i]) = 0 \ , \forall j \in \{1,...,r\}.$$

Les équations de vraisemblance n'ont pas de solution explicite en général, sauf dans le cas : $b'(\mu) = \mu$, ce qui correspond au modèle linéaire gaussien. On a donc recourt à des procédures d'optimisation itératives pour approcher la solution.

On utilise les équations de transcendances : $\widehat{\beta}^{MV}$ est approché par des algorithmes efficaces.

2.7 Algorithme IRLS/Newton-Raphson

Les équations de vraisemblance sont en générales transcendantes. Une solution pour approcher l'e.m.v. est d'utiliser des procédures itératives d'optimisation.

Application de l'algorithme de Newton-Raphson au cas de l'estimation des paramètres β :

On applique le principe précédent à la dérivée $\beta \mapsto \nabla l(\beta)$ (pour trouver un maximum local), et l'algorithme de Newton-Raphson s'écrit de la facon suivante :

1. Choisir un point de départ $\beta^{(0)}$.
2. A l'itération $(k+1)$: calculer
$$\beta^{(k+1)} = \beta^{(k)} + A_k \bigtriangledown \mathcal{L}(\beta^{(k)})$$
avec $A_k = -\left[\mathcal{H}(\mathcal{L})(\beta^k)\right]^{-1}$ la matrice Hessienne de $\mathcal{L}(\beta)$.
3. On s'arrête lorsque $\beta^{(k+1)} \approx \beta^{(k)}$ ou bien $\bigtriangledown\mathcal{L}(\beta^{k+1}) \approx \bigtriangledown\mathcal{L}(\beta^k)$.

Convergence de l'algorithme : la convergence de l'algorithme vers le maximum de la vraisemblance peut être démontré rigoureusement dans le cas de la famille exponentielle avec fonction de lien canonique.

2.8 Tests statistiques

Comme pour le modèle linéaire gaussien, on a besoin d'écrire des tests d'hypothèses par exemple pour tester le modèle globalement, ou encore tester la nullité de certains paramètres. Dans le modèle linéaire gaussien, les lois des statistiques de tests étaient connues. Dans le cas du modèle linéaire généralisé, nous n'aurons qu'une approximation asymptotique de la loi des statistiques de test.

2.8.1 Test de nullité des paramètres

En utilisant le résultat sur la loi asymptotique de $\widehat{\beta_n}$ il est direct d'écrire une statistique de test pour l'hypothèse $H_0 : \beta_k = 0$ vs $H_1 : \beta_k \neq 0$, En effet, la statistique :

$$T(Y_n) = \frac{\widehat{\beta_{n,k}}}{\sqrt{\widehat{s_k^2}}} \text{ avec } \widehat{s_k^2} = [-E[\mathcal{H}(\mathcal{L}(\beta))]]^{-1}_{k,k}.$$

est de loi asymptotique normale centrée réduite sous l'hypothèse H_0. Par conséquent, on rejette H_0 si $|T| > q_{1-\frac{\alpha}{2}}$ avec $q_{1-\frac{\alpha}{2}}$ le quantile de niveau $1-\frac{\alpha}{2}$ d'une loi gaussienne centrée réduite.

2.8.2 Test de Wald

Le test de Wald est un test paramétrique économétrique dont l'appellation vient du mathématicien américain d'origine hongroise Abraham Wald (31 octobre 1902-13 décembre 1950) avec une grande variété d'utilisations.

Test asymptotique de taille α pour $H_0 : \beta_k = 0$ vs $H_1 : \beta_k \neq 0$.

Sous H_0 : $S = T^2 \longrightarrow \chi^2_1$ lorsque $n \to +\infty$.

Peut être décrit pas sa zone de rejet : $R_\alpha = \{S > q^{\chi^2_1}_{1-\alpha}\}$ avec $q^{\chi^2_1}_{1-\alpha}$ le quantile de niveau $1-\alpha$ de la loi χ^2_1.

2.8.3 Deviance

Se faire une idée de la qualité du modèle en se basant sur la vraisemblance est difficile, étant donné qu'elle dépend en autre, de la taille de l'échantillon. On préfere comparer la vraisemblance d'un autre modèle à la vraisemblance d'un modèle de référence : le modèle "parfait" ou le modèle "saturé". Le modèle saturé est le modèle possédant autant de paramètres que d'observations (overfitting n=r) et estimant donc y par $\widehat{y}$, dans ce cadre : $E[\widehat{y_i|x_i}] = y_i$.

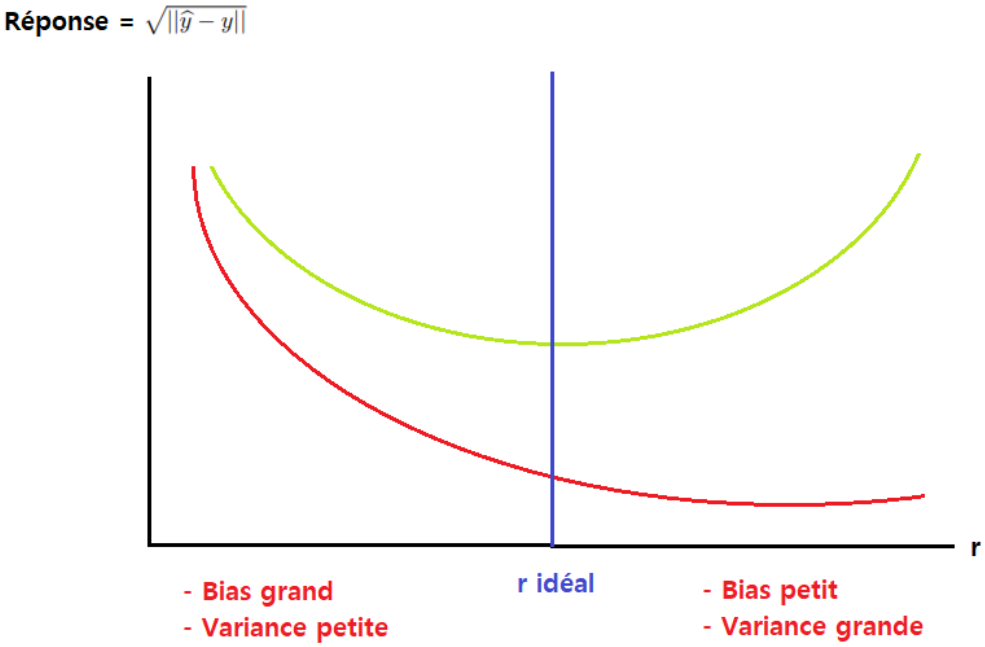

FIGURE 2.1: Graphe de la réponse en fonction du paramètre r

La deviance d'un modèle $[m]$ par rapport au modèle saturé $[sat]$:

$$D_{[m]} = 2(l_{[sat]} - l_{[m]}) \geq 0 \ .$$

Si le modèle $[m]$ a une $l_{[m]}$ proche de $l_{[sat]}$, alors on préfere le modèle $[m]$: on rend $D_{[m]}$ "petit" pour selectionner le modèle $[m]$.

La déviance est une quantité positive (aucun modèle ne s'ajuste mieux que le modéle saturé) d'autant plus petite que le modèle est riche et s'ajuste bien. La déviance vaut 0 pour le modèle le plus riche, i.e. le modèle saturé.

2.8.4 Test d'un sous modèle

Contrairement au cas du modèle linéaire gaussien, nous ne pouvons exhiber de statistique dont la loi serait explicite. Nous aurons donc recours au test classique du rapport des maximums de vraisemblance. Notons $l_{[m]}$ la log-vraisemblance du modèle $[m]$.

Test asymptotique de taille α pour $H_0 : [m_0]$ de taille r_0 est adéquat vs $H_0 : [m_1]$ de taille r_1 est adéquat, avec $r_1 > r_0$ posons T la statistique suivante :

$$T = 2(l_{[m_1]} - l_{[m_0]}).$$

Sous certaines hypothèses de régularité des modèles on a :

$$\text{Sous } H_0 : T \longrightarrow \chi^2_{r_1-r_0} \text{ lorsque } n \to +\infty.$$

Peut être décrit par sa zone de rejet : $R_\alpha = \{T > q^{\chi^2_{r_1-r_0}}_{1-\alpha}\}$ avec $q^{\chi^2_{r_1-r_0}}_{1-\alpha}$ le quantile de niveau $1-\alpha$ de la loi $\chi^2_{r_1-r_0}$.

2.9 Qualité d'ajustement

Dans le cas du modèle linéaire, on mesure la qualité d'ajustement du modèle grâce au coéffient de détermination R^2 égal au rapport de la somme des carrés du modèle SCM sur la somme des carrés totale SCT. On rappelle que si R^2 est proche de 1 alors le modèle s'ajuste bien aux données.
Dans le cadre du modèle linéaire généralisé, la décomposition $SCT = SCM + SCR$ n'est plus vraie. Nous avons alors recours au pseudo R^2. Le pseudo R^2 est construit par analogie avec le R^2 du modèle linéaire. Pour cela, on compare la déviance du modèle nul (à un seul paramètre (intercept)) $D_{[m_0]}$ avec celle du modèle qui nous intéresse $[m]$. Plus précisément :

$$pseudoR^2 = R^2_a = \frac{D_{[m_0]} - D_{[m]}}{D_{[m_0]}}.$$

Cette quantité est dans $[0, 1]$. On associe $D_{[m_0]} - D_{[m]}$ à SCM et $D_{[m_0]}$ à SCT. Plus le pseudo R^2 est proche de 1, meilleur est l'ajustement du modèle.

2.10 Choix de modèle

En présence de plusieurs modèles candidats non emboités, un critère de séléction est donné par la déviance. Le modèle qui a la plus mauvaise déviance (la plus forte) est le modèle nul, qui a un seul paramètre. Ce modèle n'a en général aucune utilité car il n'explique rien. Le modèle saturé qui a autant de paramètres que de données possède par définition la meilleure déviance puisqu'elle vaut 0. Ce modèle n'est souvent pas pertinent car il a trop de paramètres. Les déviances de ces deux modèles fournissent les valeurs du pire et du meilleur ajustement possible. Un modèle sera qualifié de bon si sa déviance est proche de celle du modèle saturé et qu'il est construit avec un faible nombre de paramètres. Des critères pénalisés permettent de prendre en compte ces deux contraintes antagonistes.

2.10.1 Critère AIC

Le plus célèbre d'entre eux est le critère **AIC** (Akaike Information Criterion) dont une définition est :

$$AIC(M) = D(M) + 2r_1 + C_{te}.$$

avec r_1 est le rang de la matrice de design X. L'AIC est d'autant plus faible que la log-vraisemblance est élevée et que le nombre de paramètres est petit et permet donc d'établir un ordre sur les modèles en prenant en compte les deux contraintes.

2.10.2 Critère BIC

Le critère **BIC** (Bayesian Information Criterion) qui pénalise plus le sur-ajustement est défini par :

$$BIC(M) = D(M) + \log(n)r_1 + C_{te}.$$

2.11 Modélisation de la fréquence des sinistres

Le nombre de sinistres peut s'observer comme la réalisation d'une variable aléatoire discrète positive suivant une loi de type Poisson, Binomiale Négative ou binomiale. Dans notre cadre, $y_i = "CaimInd"$: $y_i \in \{0, 1\}$ alors naturellement on va utiliser la loi de Bernoulli pour modèliser $y_i = "CaimInd"$.

2.11.1 Regression logistique de $y_i = "ClaimInd_i"$

On a $y_i|x_i \sim Bernoulli(p_i)$ avec $p_i = P(y_i = 1|x_i) = E[y_i|x_i]$, et $q_i = 1 - p_i$, on choisit la fonction lien canonique **logit** : $g(E[y_i|x_i]) = logit(p_i) = log(\frac{p_i}{1-p_i}) = x_i^T\beta$.
Si $\widehat{\beta}$ est un bon estimateur de β alors : $\widehat{p_i} = g^{-1}(x_i^T\widehat{\beta}) = \frac{e^{x_i^T\widehat{\beta}}}{1+e^{x_i^T\widehat{\beta}}}$.

On regroupe les deux bases de données freMPL3 et freMLP4 dans une base freMPL34, 80% de freMPL34 représente notre base d'apprentissage "freMPL34.app" de notre modèle et 20% de freMPL34 représente notre base de test "freMPL34.test" du modèle.
Pour simplifier nos bases au niveau des facteurs, nous avons fait appel à un modèle matriciel "freMPL34.app.reg" et "freMPL34.test.reg" et on a fait quelques transformations au niveau des variables.
Le choix des bases d'apprentissage et de test est fait d'une facon aléatoire.

2.11.2 Apprentissage du modèle linéaire généralisé $y_i = "ClaimInd_i"$

Afin de sélectionner notre modèle optimal, nous avons appliqué l'algorithme "AIC",
stepAIC(GLM.ClaimInd) sur R, on a testé les méthodes Forward, Backward et Stepwise en comparant leur AIC.

On trouve un AIC plus petit que l'AIC du début. Cela nous a permis d'éliminer beaucoup de variables qui n'expliquent pas nos modèles de regression logistiques "GLM.ClaimInd".

Notre modèle optimal "GLM.ClaimInd" $y_i = "ClaimInd_i"$ sous R est :

```
GLM.ClaimInd <- glm(ClaimInd ~ LicAge3 + BonusMalus + VehAge10 + VehAge6 +
                      VehAge8 + SocioCategCSP6 + VehUsageProfessional +
                      VehUsageProfessional_run + DeducTypePartially_refunded +
                      DeducTypeRefunded + VehBodystation_wagon + VehPriceO +
                      VehPriceP + VehPriceQ + VehMaxSpeed140_150_km_h +
                      VehMaxSpeed150_160_km_h + VehMaxSpeed160_170_km_h +
                      VehMaxSpeed170_180_km_h + VehMaxSpeed180_190_km_h +
                      VehMaxSpeed190_200_km_h + VehMaxSpeed200_220_km_h +
                      VehMaxSpeed220_km_h + VehClassA + GaragePrivate_garage,
                      family = binomial(link = 'logit'),
    data = cbind.data.frame(ClaimInd = freMPL34.app$ClaimInd, freMPL34.app.reg))
```

La sortie $summary(GLM.ClaimInd)$ sur R nous donne la valeur des p-value des coefficients, les Deviances residuelles et le AIC :

‘

```
Deviance Residuals:
    Min       1Q   Median       3Q      Max
-0.4873  -0.2427  -0.2144  -0.1830   3.2892


Coefficients:
                               Estimate Std. Error z value Pr(>|z|)
(Intercept)                  -4.6712078  0.2063934 -22.633  < 2e-16 ***
LicAge3                       0.0002195  0.0001282   1.712 0.086873 .
BonusMalus                    0.0054311  0.0014906   3.644 0.000269 ***
VehAge10                     -0.9839766  0.1525900  -6.449 1.13e-10 ***
VehAge6                      -0.3216309  0.1012135  -3.178 0.001484 **
VehAge8                      -0.5227534  0.1298411  -4.026 5.67e-05 ***
SocioCategCSP6                0.2445505  0.0925522   2.642 0.008235 **
VehUsageProfessional          0.2512964  0.0705704   3.561 0.000370 ***
VehUsageProfessional_run      0.5483549  0.1746499   3.140 0.001691 **
DeducTypePartially_refunded   0.3782161  0.0881319   4.291 1.77e-05 ***
DeducTypeRefunded            -0.3529904  0.1565571  -2.255 0.024152 *
VehBodystation_wagon         -0.3669785  0.1525603  -2.405 0.016152 *
VehPriceO                    -0.2745018  0.1666760  -1.647 0.099575 .
VehPriceP                    -0.4625704  0.2034937  -2.273 0.023017 *
VehPriceQ                    -0.4590950  0.1617429  -2.838 0.004534 **
VehMaxSpeed140_150_km_h       0.7663583  0.1986282   3.858 0.000114 ***
VehMaxSpeed150_160_km_h       0.6228162  0.1773355   3.512 0.000445 ***
VehMaxSpeed160_170_km_h       0.6229614  0.1772322   3.515 0.000440 ***
VehMaxSpeed170_180_km_h       0.5518807  0.1846801   2.988 0.002805 **
VehMaxSpeed180_190_km_h       0.3725560  0.1896804   1.964 0.049516 *
VehMaxSpeed190_200_km_h       0.6351366  0.1897964   3.346 0.000819 ***
VehMaxSpeed200_220_km_h       0.7652149  0.1940456   3.943 8.03e-05 ***
VehMaxSpeed220_km_h           0.7932942  0.2358272   3.364 0.000769 ***
VehClassA                     0.4021349  0.0963960   4.172 3.02e-05 ***
GaragePrivate_garage         -0.1755417  0.0961535  -1.826 0.067905 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 12026  on 53498  degrees of freedom
Residual deviance: 11811  on 53474  degrees of freedom
AIC: 11861

Number of Fisher Scoring iterations: 7
```

"Number of Ficher scoring itérations : 7" représente le nombre d'itérations (7) necéssaires pour que l'algorithme Newton-Raphson converge.

Le ratio **residual deviance / ddl** est égal à 11811/53474 , soit 0.22. Ce ratio est inférieur à 1 et permet de mettre en évidence l'absence d'une surdispersion.

2.11.3 Prédiction du modèle linéaire généralisé $y_i = "ClaimInd_i"$

Afin de determiner le seuil s de notre variable de prédiction $\widehat{y_i} = \mathbb{1}_{p_i > s}$, on calcule la probabilité de "ClaimInd=1" dans notre base freMPL3, on trouve : $s = 0.02404522$.

Matrice de confusion sur l'échantillon test

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 6946  123
         1 6093  213

               Accuracy : 0.5353
                 95% CI : (0.5268, 0.5437)
    No Information Rate : 0.9749
    P-Value [Acc > NIR] : 1

                  Kappa : 0.0173

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.53271
            Specificity : 0.63393
         Pos Pred Value : 0.98260
         Neg Pred Value : 0.03378
             Prevalence : 0.97488
         Detection Rate : 0.51933
   Detection Prevalence : 0.52852
      Balanced Accuracy : 0.58332

       'Positive' Class : 0
```

La matrice de confusion est une matrice qui représente :

	$y_i = 0$	$y_i = 1$
$\widehat{y_i} = 0$	VP	FP
$\widehat{y_i} = 1$	FN	VN

Elle résume :
• le nombre de vrais positifs **VP** : prédit "0" et la réference est "0".
• le nombre de vrais négatifs **VN** :prédit "1" et la réference est "1".
• le nombre de faux positifs **FP** :prédit "0" et la réference est "1".
• le nombre de faux négatifs **FN** : prédit "1" et la réference est "0".

Notons que les valeurs dans la matrice de confusion dépendont du seuil s avec lequel on comparera les probabilités estimées par le modèle. Plusieurs indicateurs peuvent être calculés à partir de cette matrice :
• La jutesse (Accuracy) : proportion de prédictions/classifications correctes :
$Accuracy = \frac{VP+VN}{VP+VN+FP+FN} = 0.5353$.
• TVP (sensitivity) : le taux de VP par rapport aux positifs totaux $P = VP + FN$:
$TVP = \frac{VP}{VP+FN} = 0.53271$.
• TVN (specificity) : le taux de VN par rapport aux négatifs totaux $N = VN + FPN$:
$TVN = \frac{VN}{VN+FP} = 0.63393$.

Estimation de l'erreur de prédiction sur l'échantillon test

On trouve que l'erreur de prédiction sur l'échantillon test est estimée à 2.3%.
La matrice de confusion nous indique un problème de prédiction pour les individus ayant un sinistre. Il serait donc possible d'augmenter le seuil s, mais nous l'avons déterminé en calculant la probabilité de survenance d'un sinistre sur la base de données test.

Cependant nous ne nous intéresserons qu'aux probabilités de survenance de sinistre et pas à la prédiction de sa survenance. On peut donc accepter le modèle.

2.12 Modélisation du coût des sinistres

Le choix d'une loi pour modéliser le coût des sinistres est plus délicat que celui pour modéliser leur fréquence. Un phénomène de comptage ou d'occurrence est en effet plus intrinsèquement lié à des phénomènes statistiques que celui de la sévérité d'un sinistre. Ainsi le choix par défaut se porte sur la loi Gamma qui permet de modéliser des données continues positives à queue épaisse ([Murphy,2000]).

2.12.1 Regression Gamma de $y_i = "ClaimAmount_i"$

On a $y_i \in \mathbb{R}^+$ donc $y_i|x_i \sim Gamma(\alpha, \beta)$, sa fonction de densité f_Y est donnée par :

$f_Y(y) = \frac{\beta^\alpha}{\Gamma(\alpha)} y^{\alpha-1} e^{-y\beta}$, $\forall y \in \mathbb{R}^+$ avec $\alpha > 0$ le paramètre de forme et $\beta > 0$ le paramètre d'intensité.

On choisit la fonction lien **log** : $g(E[y_i|x_i]) = log(E[y_i|x_i]) = x_i^T\beta$ alors $E[y_i|x_i] = \exp(x_i^T\beta)$.

On regroupe les deux bases de données freMPL3 et freMLP4 dans une base freMPL34, 80% de freMPL34 représente notre base d'apprentissage "freMPL34.app.sinistre" de notre modèle et 20% de freMPL34 représente notre base de test "freMPL34.test.sinistre" du modèle.

On s'interesse que pour les valeurs $ClaimAmount > 0$.

Pour simplifier nos bases au niveau des facteurs, nous avons fait appel à un modèle matriciel "freMPL34.app.sinistre.reg" et "freMPL34.test.sinistre.reg" et on a fait quelques transformations au niveau des variables. Le choix des bases d'apprentissage et de test est fait d'une facon aléatoire.

2.12.2 Apprentissage du modèle linéaire généralisé $y_i = "ClaimAmount_i"$

Afin de sélectionner notre modèle optimal, nous avons appliqué l'algorithme "AIC",
stepAIC(GLM.ClaimAmount) sur R, on a testé les méthodes Forward, Backward et Stepwise en comparant leur AIC.

On trouve un AIC plus petit que l'AIC du début. Cela nous a permis d'éliminer beaucoup de variables qui n'expliquent pas nos modèles de regression Gamma "GLM.ClaimAmount".

Notre modèle optimal "GLM.ClaimAmount" $y_i = "ClaimAmount_i"$ sous R est :

```
GLM.claimAmount <- glm(ClaimAmount ~ LicAge + BonusMalus + HasKmLimit +
                         RiskVar + VehAge3 + VehAge4 + VehAge6 + VehAge8 +
                         DeducTypePartially_refunded + VehPriceE +
                         VehPriceJ + VehPriceM + VehPriceN + VehPriceQ +
                         VehEnergyregular + VehMaxSpeed140_150_km_h +
                         VehMaxSpeed160_170_km_h + VehMaxSpeed170_180_km_h +
                         VehMaxSpeed180_190_km_h + VehMaxSpeed190_200_km_h +
                         VehMaxSpeed200_220_km_h + VehClassA + VehClassM1 +
                         GarageNone + VehUsagePrivate_trip_to_office
                         + VehBodycabriolet,
                         family = Gamma(link = "log"),
     data = cbind.data.frame(ClaimAmount = freMPL34.app.sinistre$ClaimAmount,
                         freMPL34.app.sinistre.reg),
                         weights = VehBodymicrovan + VehBodycoupe +
                         SocioCategCSP2 + SocioCategCSP4 +
                         SocioCategCSP5 + SocioCategCSP7 +
                         SocioCategCSP9)
```

La sortie *summary*(*GLM.ClaimInd*) sur R nous donne la valeur des p-value des coefficients, les Deviances residuelle et le AIC :

```
Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.8642  -0.7348  -0.0035   0.1025   3.4651

Coefficients:
                                Estimate Std. Error t value Pr(>|t|)
(Intercept)                     6.946819   0.248163  27.993  < 2e-16 ***
LicAge                          0.003006   0.003501   0.859 0.390780
BonusMalus                      0.006118   0.002037   3.003 0.002742 **
HasKmLimit                      0.230598   0.127608   1.807 0.071060 .
RiskVar                         0.009008   0.007069   1.274 0.202905
VehAge3                        -0.144909   0.099146  -1.462 0.144184
VehAge4                        -0.070042   0.100555  -0.697 0.486247
VehAge6                        -0.132736   0.120418  -1.102 0.270608
VehAge8                        -0.136729   0.153503  -0.891 0.373300
DeducTypePartially_refunded    -0.429784   0.103438  -4.155 3.54e-05 ***
VehPriceE                      -0.388167   0.129392  -3.000 0.002770 **
VehPriceJ                       0.112594   0.112877   0.997 0.318776
VehPriceM                      -0.020162   0.152408  -0.132 0.894784
VehPriceN                       0.107055   0.132257   0.809 0.418457
VehPriceQ                       0.966009   0.176923   5.460 6.05e-08 ***
VehEnergyregular               -0.141225   0.076743  -1.840 0.066038 .
VehMaxSpeed140_150_km_h         0.034316   0.153236   0.224 0.822851
VehMaxSpeed160_170_km_h         0.153374   0.102661   1.494 0.135507
VehMaxSpeed170_180_km_h         0.392072   0.113307   3.460 0.000563 ***
VehMaxSpeed180_190_km_h         0.234711   0.121378   1.934 0.053439 .
VehMaxSpeed190_200_km_h         0.202670   0.126338   1.604 0.108999
VehMaxSpeed200_220_km_h         0.265254   0.134355   1.974 0.048634 *
VehClassA                       0.043569   0.115664   0.377 0.706491
VehClassM1                     -0.169240   0.083295  -2.032 0.042445 *
GarageNone                     -0.022106   0.080414  -0.275 0.783448
VehUsagePrivate_trip_to_office  0.044839   0.066531   0.674 0.500502
VehBodycabriolet                0.113712   0.165988   0.685 0.493470
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
(Dispersion parameter for Gamma family taken to be 1.123042)

    Null deviance: 1145.65  on 993  degrees of freedom
Residual deviance:  996.25  on 967  degrees of freedom
AIC: 18611

Number of Fisher Scoring iterations: 9
```

"Number of Ficher scoring itérations : 9" représente le nombre d'itérations (9) necéssaires pour que l'algorithme Newton-Raphson converge.

Le ratio **residual deviance / ddl** est égal à 996.25 /967, soit 1.03. Ce ratio est très proche de 1 et permet de mettre en évidence l'absence d'une surdispersion.

2.12.3 Prédiction du modèle linéaire généralisé $y_i = "ClaimAmount_i"$

On trace le graphe des "ClaimAmount" dans notre base de données d'apprentissage et la prédiction du modèle linéaire généralisé pour un échantillion d'apprentissage :

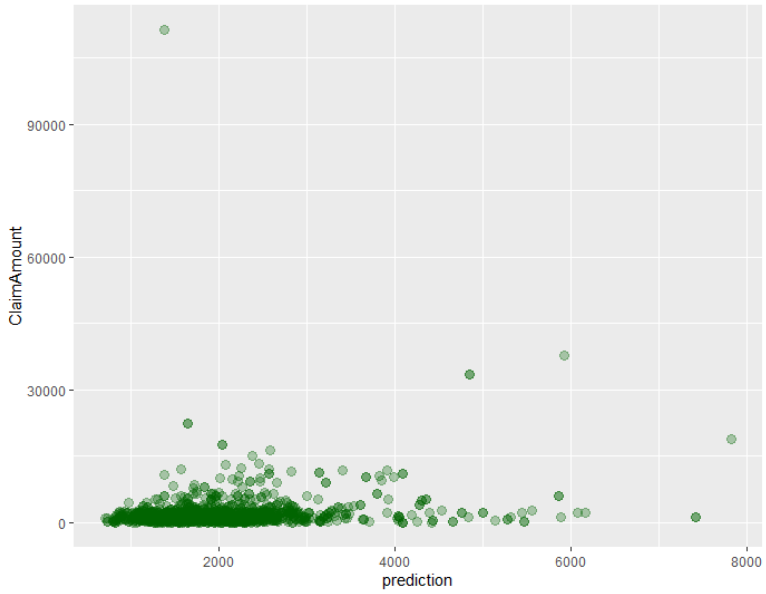

FIGURE 2.2: Graphe de ClaimAmount et prédiction pour l'apprentissage

On remarque que le modèle répond bien aux données d'apprentissage puisqu'il n'y a pas dispertion de points sur le graphe.

On trace le graphe des "ClaimAmount" dans notre base de données de test et la prédiction du modèle linéaire généralisé pour un échantillion de test :

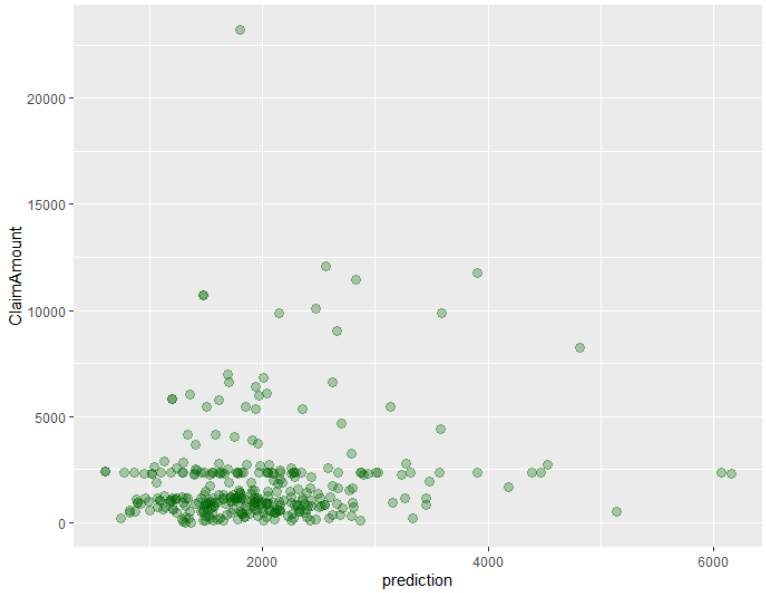

FIGURE 2.3: Graphe de ClaimAmount et prédiction pour le test

On remarque que le modèle répond bien aux données de test puisqu'il n'y a pas dispertion de points sur le graphe.
On trace les points de "ClaimAmount" de notre base de test et les points de prediction sur l'échantillion de test :

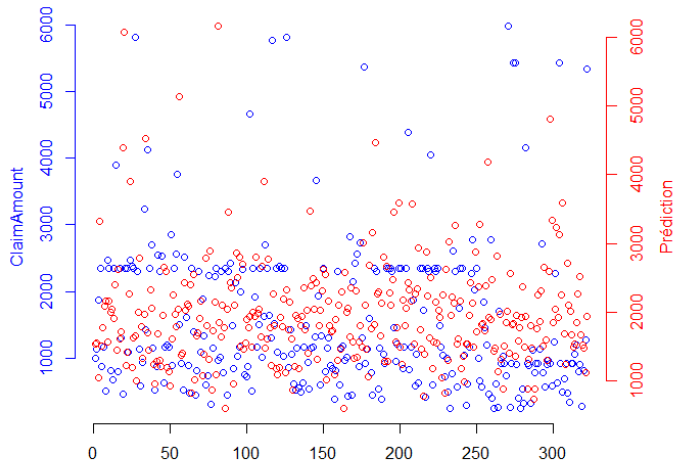

FIGURE 2.4: Graphe de ClaimAmount et prédiction pour le test

On conclut que notre modèle linéaire généralisé s'adapte bien aux données.

Nous pouvons désormais observer la répartition des sinistres déclarés prédits en fonction des différentes classes d'assurés.Pour cela, nous allons regarder des graphiques de type "boxplots".

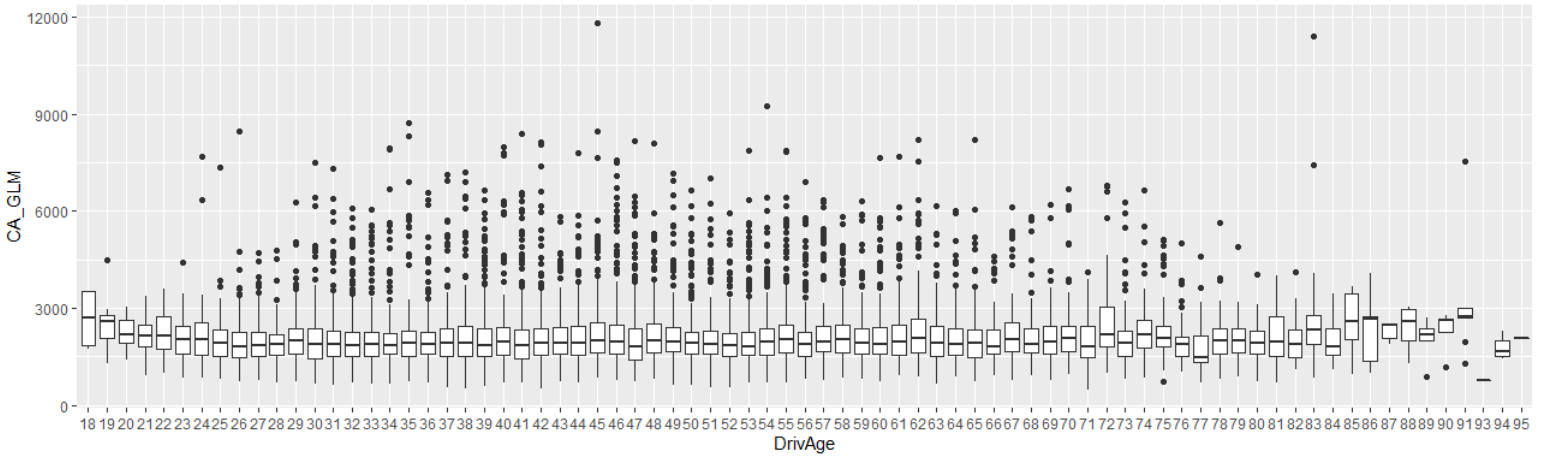

FIGURE 2.5: Boxplot de la prédiction des sinistres déclarés en fonction du nombre d'années de permis de l'assuré.

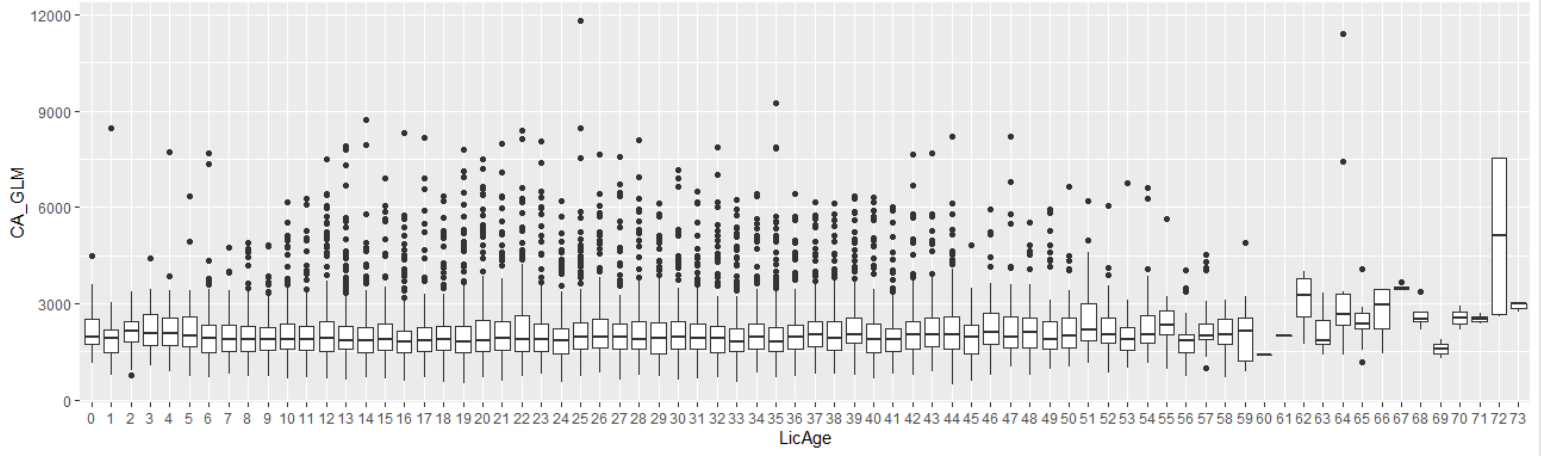

FIGURE 2.6: Boxplot de la prédiction des sinistres déclarés en fonction de l'âge de l'assuré.

On observe que les sinistres déclarés prédits ont tendance à être plus élevés lorsque les individus sont assez âgés, ou très jeunes (cela est moins visible).

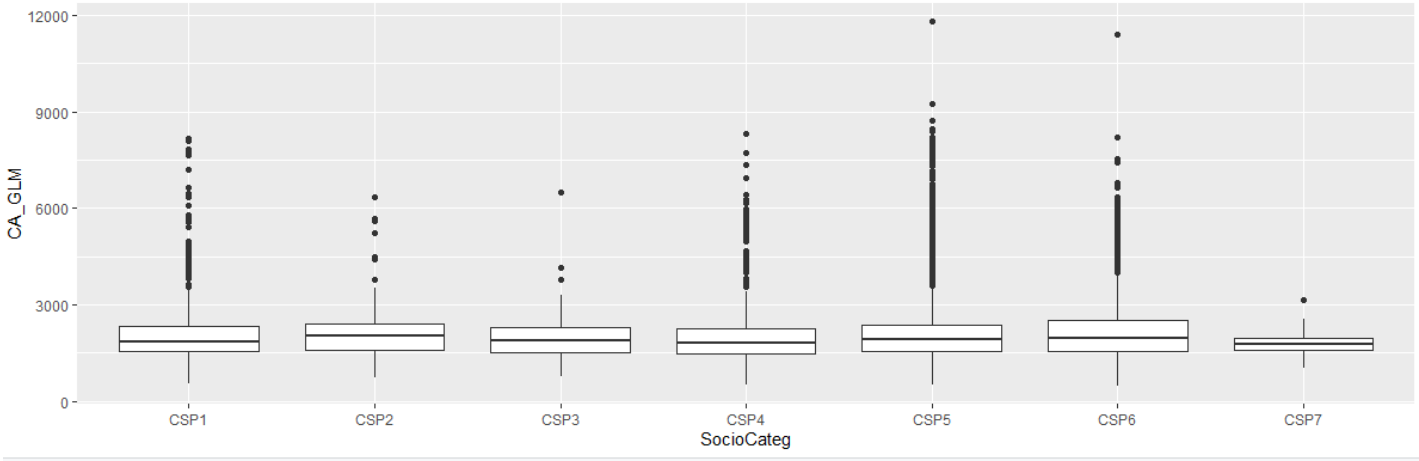

FIGURE 2.7: Boxplot de la prédiction des sinistres déclarés en fonction de la CSP de l'assuré.

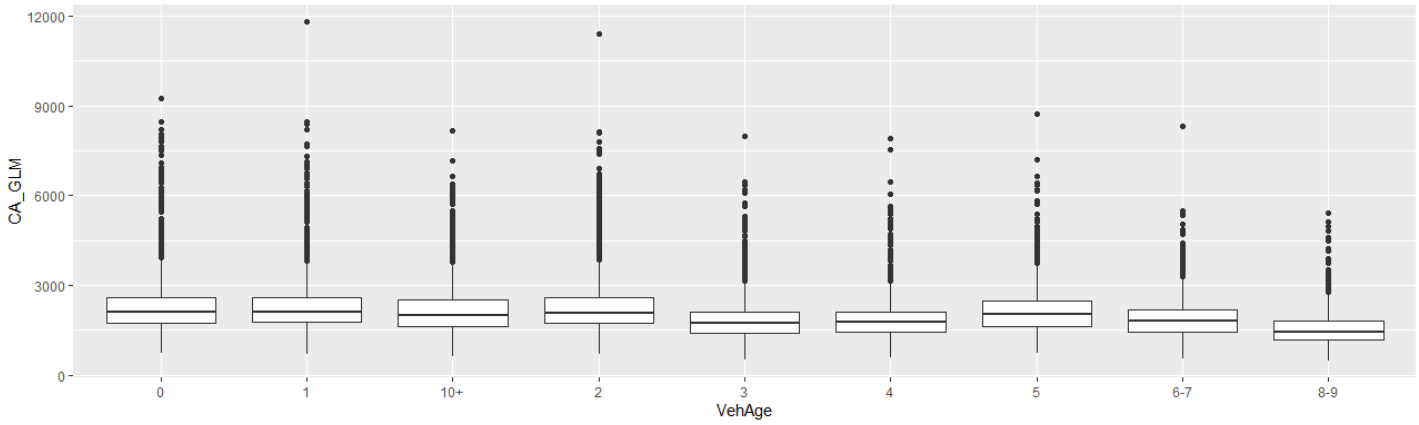

FIGURE 2.8: Boxplot de la prédiction des sinistres déclarés en fonction de l'ancienneté du véhicule de l'assuré.

On voit peu de différences sur la prédiction des sinistres en fonction de la CSP des assurés. La CSP7 semble plus écrasée mais cela est certainement dû au nombre limité de CSP7 dans notre base de donnnées test.
On remarque que les sinistres prédits ont tendance à être un peu plus importants lorsque les véhicules sont récents plutôt que lorsqu'ils sont anciens, mais cette différence n'est pas si grande.

2.13 La prime pure

Dans le domaine de l'assurance, la prime pure correspond au montant de sinistres moyen auquel devra faire face l'assureur. En faisant payer une telle prime à ses assurés, il aura en moyenne un bénéfice nul.

Dans un contexte d'assurance non vie, la prime pure peut être modélisée par une approche fréquence-sévérité, qui repose sur les hypothèses suivantes :
• Les charges des sinistres individuels sont des variables aléatoires indépendantes et identiquement distribuées.
• Les variables représentant le fait d'avoir un sinistre ou non, sont des variables aléatoires indépendantes et identiquement distribuées.
• La fréquence et la sévérité sont supposées indépendantes.

La charge totale des sinistres de l'assuré i est modélisée, sur une période d'assurance donnée, par le produit suivant :

$$X_i = I_i B_i.$$

où B_i désigne le montant du sinistre individuel de l'assuré i et I_i est le fait que l'assuré est un sinistre ($I_i = 1$) ou non ($I_i = 0$). Sous l'hypothèse d'indépendance entre la fréquence et la sévérité, nous obtenons :

$$E[X_i] = E[I_i].E[B_i].$$

Ainsi, la prime actuarielle peut être estimée :

• En calibrant séparément un modèle pour la fréquence "ClaimInd" :

$$E[I_i|X_i = x_i] = (predict.GLM.claimInd)_i = E[y_i|x_i] = \frac{e^{x_i^T \beta}}{1+e^{x_i^T \beta}}.$$

• En calibrant séparément un modèle pour la sévérité "ClaimAmount" :

$$E[B_i|X_i = x_i] = (predict.GLM.claimAmount)_i = E[y_i|x_i] = \exp(x_i^T \beta).$$

Le calcul de la prime pure est donc donnée par :

$$\pi_i = E[X_i] = E[I_i|X_i = x_i].E[B_i|X_i = x_i] = (predict.GLM.claimInd)_i.(predict.GLM.claimAmount)_i.$$

Puisque les "*ClaimInd*" ont été prédit via une loi de Bernouilli :

$$E[I_i] = P(I_i = 1) = (predict.GLM.claimInd)_i.$$

Nous avons ainsi modélisé la prime pure sur la base test freMPL34.test :

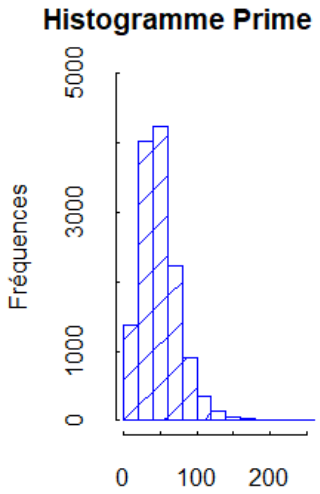

FIGURE 2.9: Histogramme de la prime pure calculée par les GLMs.

Dans le cas de notre base test, la prime pure est répartie entre **2.995** et **839.006**. Elle admet un premier quartile de **30.499**, une moyenne de 49.508 et un troisième quartile de **62.580**.

Nous pouvons observer l'effet des différentes caractéristiques des assurés sur la prime pure, notamment à travers des boxplots.
Ces graphiques nous indiquent, le premier quartile, la moyenne et le troisième quartile de la prime pure en fonction de la modalité de la variable.

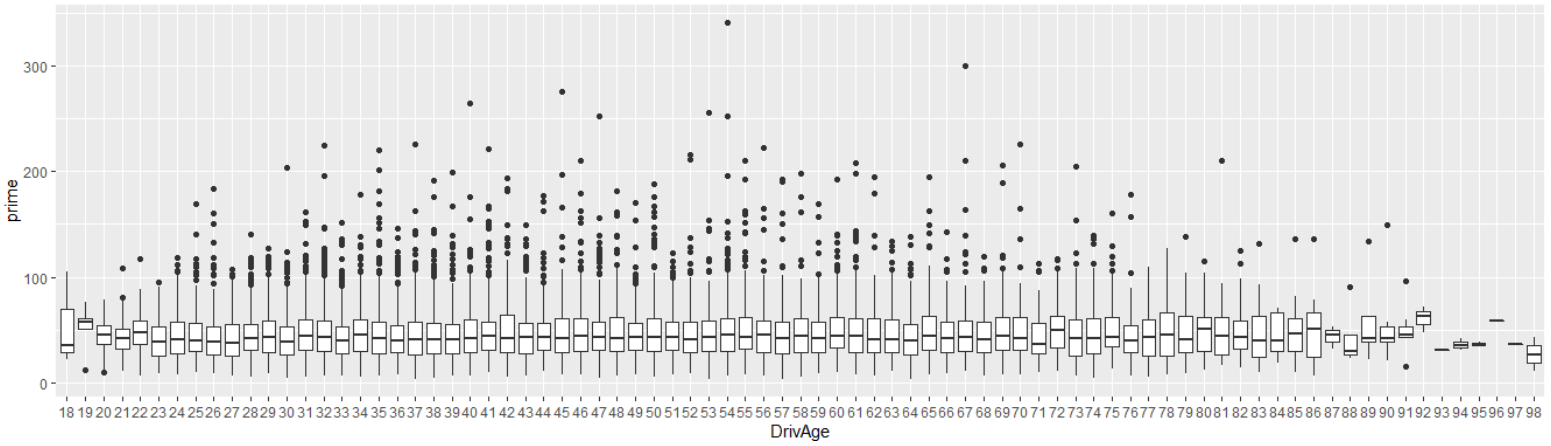

FIGURE 2.10: Boxplot de la prime pure en fonction de l'âge de l'assuré.

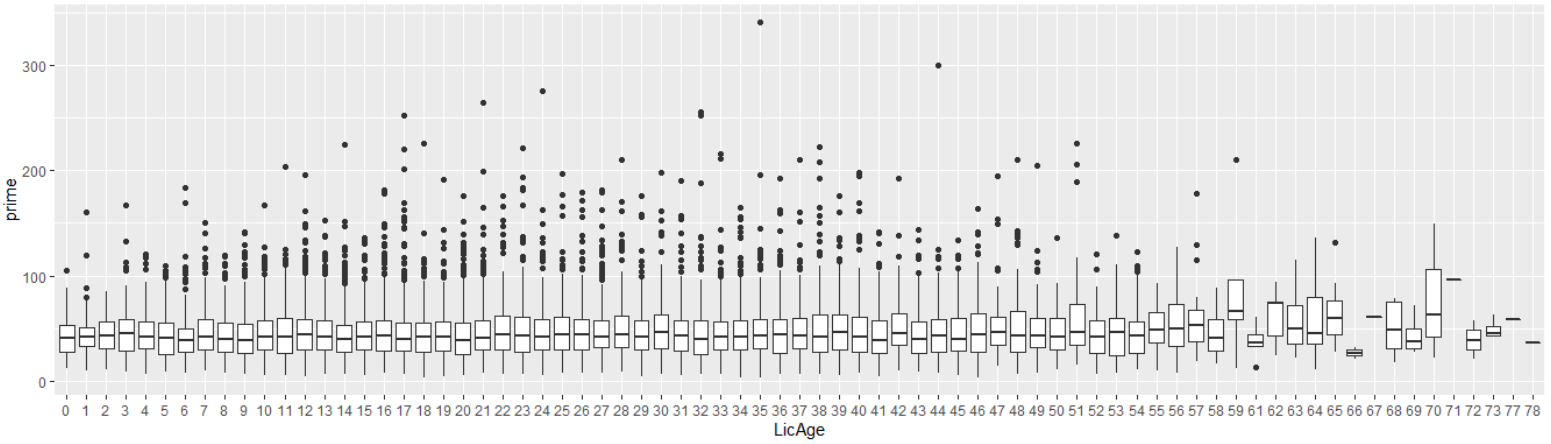

FIGURE 2.11: Boxplot de la prime pure en fonction du nombre d'années de permis de l'assuré.

Les figures 2.11 et 2.12 montrent le montant de prime pure calculé en fonction de l'âge puis du nombre d'années de permis de l'assuré. Les grandes valeurs ne sont pas forcément très représentatives au vu du nombre limité de données.

Il est logique de ne pas avoir beaucoup d'assurés avec plus de 70 années de permis.

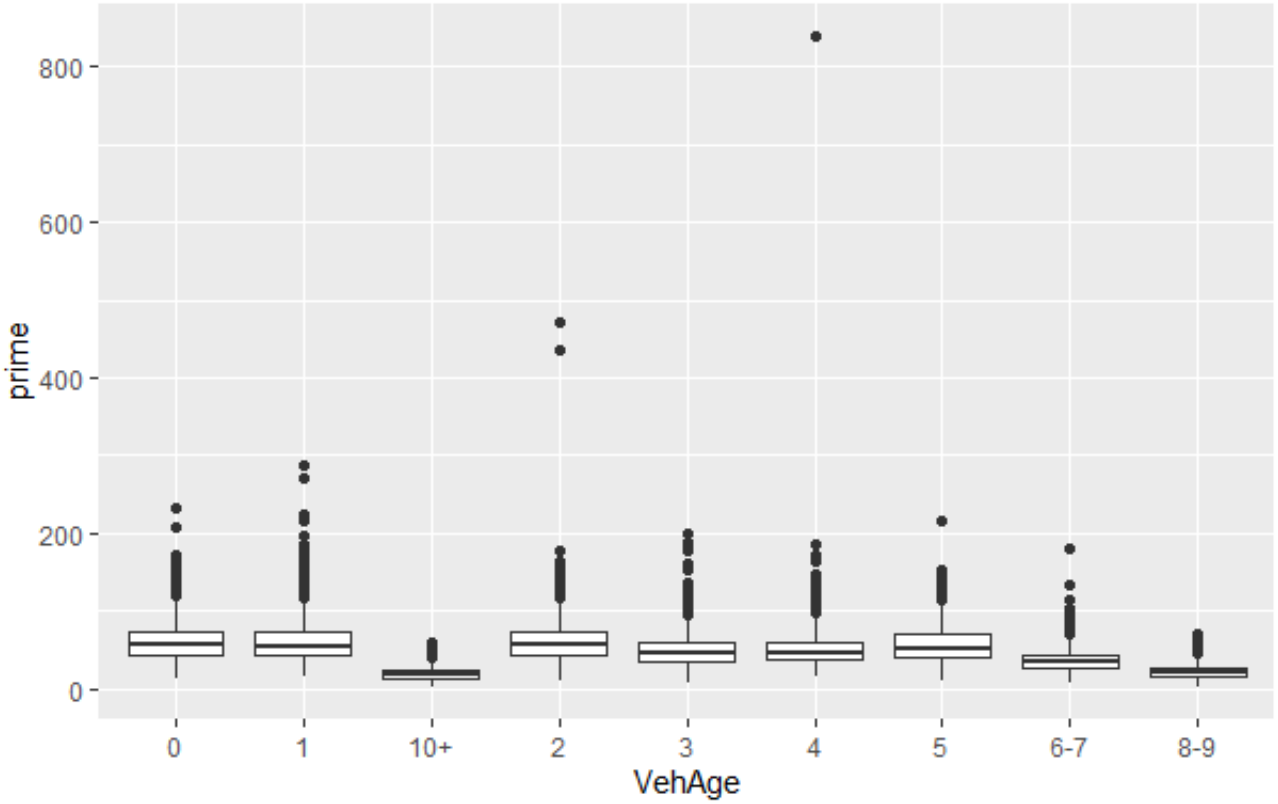

FIGURE 2.12: Boxplot de la prime pure en fonction de l'âge du véhicule de l'assuré.

Le boxplot de la Figure 2.13 est très lisible. On voit aisément que lorsque le véhicule a tendance à être récent, la prime pure est plus importante que lorsqu'il a tendance à être plus ancien.
Cette observation est très logique car un sinistre sur véhicule neuf entraînera des coûts plus importants que sur un véhicule vieillissant.

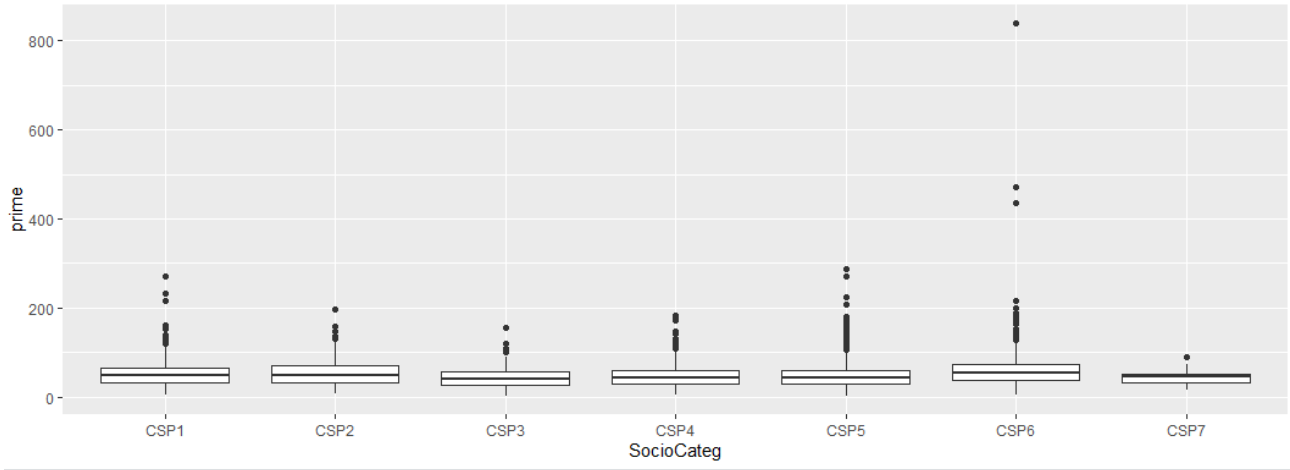

FIGURE 2.13: Boxplot de la prime pure en fonction de la catégorie socio-professionnel de l'assuré.

On observe peu de variations de la prime pure en fonction de la CSP. On pourrait penser qu'il y a moins de valeurs très grandes dans la CSP7 (retraités) mais cela est certainement dû à la sous représentation de cette catégorie par rapport aux autres dans la base de données test.

Chapitre 3

Présentation des modèles additifs généralisés GAMs

But : Meilleure prédiction de $y|x$.

Problème : On ne peut pas représenter des relations non linéaires entre un prédicteur et une variable réponse.

Solution : Introduire de nouveaux modèles additives qui utilises des fonctions qui s'appellent "splines de lissage" .

3.1 Introduction

On dispose de n observations indépendantes $(y_i, x_i)_{i=1,...,n}$ òu $x_i \in \mathbb{R}^r$ et $y_i \in \mathbb{Y} \subset \mathbb{R}$ pour tout $i \in \{1, ..., n\}$. On définit la matrice *design* X comme suit :

$$X = \begin{pmatrix} x_1^1 & \cdots & x_1^r \\ \vdots & & \vdots \\ x_n^1 & \cdots & x_n^r \end{pmatrix} = (X^1, ..., X^r) \text{ et } Y = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}.$$

ou $X^k, k \in \{1, ..., r\}$ sont les variables explicatives. Dans le but de découper de façon optimale les variables continues ou à minima de disposer des arguments pouvant justifier cette segmentation, nous avons eu recours aux modèles additifs généralisés (GAM, Generalized Additive Models, Hastie et Tibshirani).

Les GAM constituent une extension des modèles linéaires généralisés et permettent de prendre compte des effets non linéaires des variables continues explicatives sur le prédicteur linéaire.

En conservant les notations précédentes, nous avons :

$$g(E[y_i|x_i]) = x_i^{DISCRETE}\beta + s(x_i^{CONTINUE}).$$

où $s(.)$ est la fonction de lissage considérée régulière.
Notre but par la suite est de catégoriser les variables explicatives **continues** en fonction de l'influence qu'elles ont sur la variable à expliquer. Le nombre de catégories dépendra de la relation graphique obtenue en utilisant le GAM avec un lissage par splines cubiques.

3.2 Définition du modèles additive généralisé (GAM)

Le modèle additif généralisé (GAM) est un modèle statistique développé par Trevor Hastie et Rob Tibshirani pour fusionner les propriétés du modèle linéaire généralisé avec celles du modèle additif.

Soient $y_1, y_2, ..., y_n$ n observations indépendantes d'une variable quantitative. D'autre part, pour chaque observation i, on dispose de r variables explicatives $(x_i^1, x_i^2, ..., x_i^r)$ réelles. On cherche à "expliquer" y_i comme une fonction des $(x_i^1, x_i^2, ..., x_i^r)$.

Le modèle spécifie une distribution (comme la distribution normale, ou la distribution binomiale) et une fonction de lien g reliant la valeur attendue de la distribution aux prédicteurs, et tentant d'ajuster les fonctions s_i pour satisfaire :

$$g(E[Y_i]) = \beta_0 + s_1(x_i^1) + s_2(x_i^2) + ... + s_p(x_i^p).$$

Les fonctions s_i peuvent être ajustées en utilisant les moyennes non paramétriques ou paramétriques, et fournissant ainsi potentiellement de meilleurs ajustements aux données que les autres méthodes. La méthode est donc très générale - un GAM typique pourrait utiliser une fonction lissante de graphe de dispersion telle que la moyenne pondérée localement pour $s_1(x_i^1)$, et utiliser une modèle facteur pour $s_2(x_i^2)$, etc. En autorisant les ajustements non paramétriques, les GAMs bien conçus permettent de bons ajustements aux données d'apprentissage avec des hypothèses non contraignantes sur les relations réelles, peut-être aux dépens de l'interprétabilité des résultats.

Le surapprentissage peut être un problème avec les GAMs. Le nombre de paramètres lissants peut être spécifié, et ce nombre devrait être raisonnablement petit, certainement très en dessous des degrés de liberté offerts par les données. La validation croisée peut être employée pour détecter et/ou réduire les problèmes de surapprentissage survenant avec les GAMs ou avec d'autres méthodes statistiques. D'autres modèles tels que les GLMs sont quelquefois préférables aux GAMs sauf dans le cas où celles-ci améliorent substantiellement la capacité prédictive dans le domaine appliqué.

3.3 Splines de lissage

Les splines sont utilisées pour représenter numériquement des contours complexes. Leur mise en œuvre est simple. Elles sont fréquemment employées dans les logiciels de dessin ou de conception graphique ; leur usage y a été généralisé par Pierre Bézier avec les B-splines.

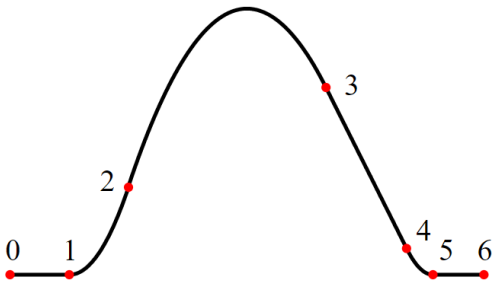

FIGURE 3.1: Exemple de spline quadratique.

‘

L'idée du lissage par splines est de découper la plage de la fonction à ajuster en sous-intervalles, puis d'ajuster sur chaque sous-intervalle une fonction simple, en prenant des précautions pour le raccordement aux points de jonction.

Dans le modèle additive généralisé, une spline de lissage s est une combinaison linéaire de fonctions de base b (basis functions) avec des poids β estimés en fonction des données.

une spline avec **k** fonctions de base :

$$s(x_i) = \sum_{j=1}^{k} \beta_j b_i(x_i).$$

Par défaut, le le modèle additive généralisé utilise des spines en plaque mince (thin plate regression spline) avec **k=10**.

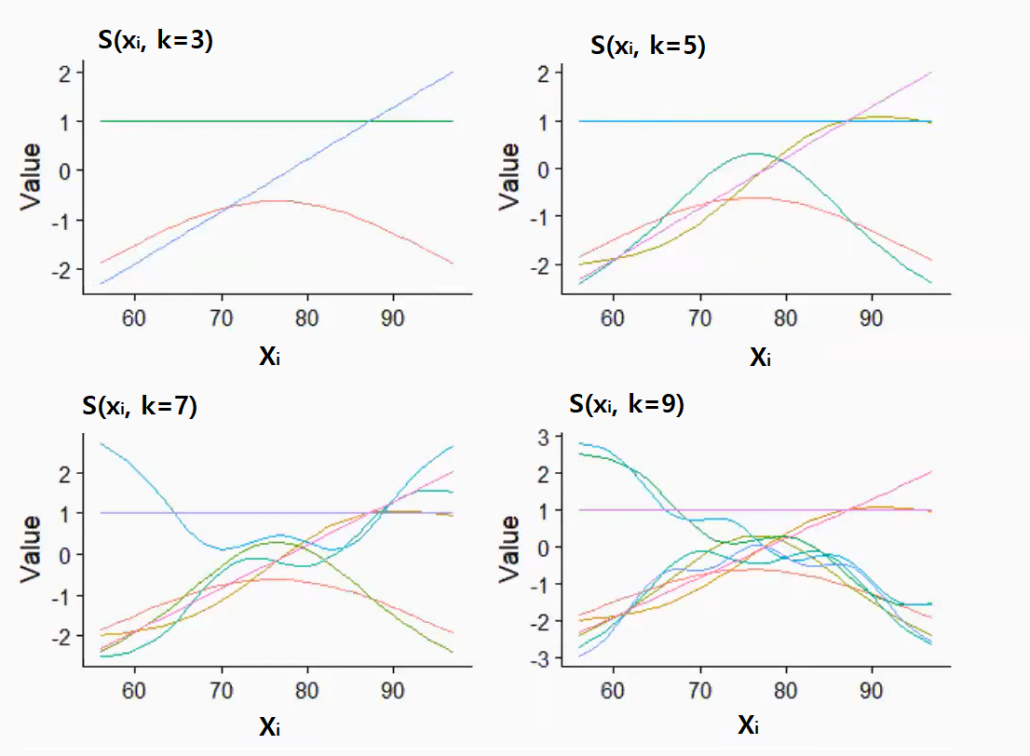

FIGURE 3.2: Nombre de fonctions de base en fonction de **k**.

‘

3.4 Paramètre de lissage

Pour éviter le sur-ajustement, les paramètres de splines $s(x_i)$ sont estimés en maximisant une version modifiée de la log-vraisemblance l, en ajoutant une pénalité proportionnelle à la "rugosité" de la spine, mesurée par sa dérivée seconde :

$2l(\beta) - \sum_i \lambda_i \int s''(x_i)^2 dx_i$, λ_i est le paramètre du lissage, noté **sp** dans le GAM.

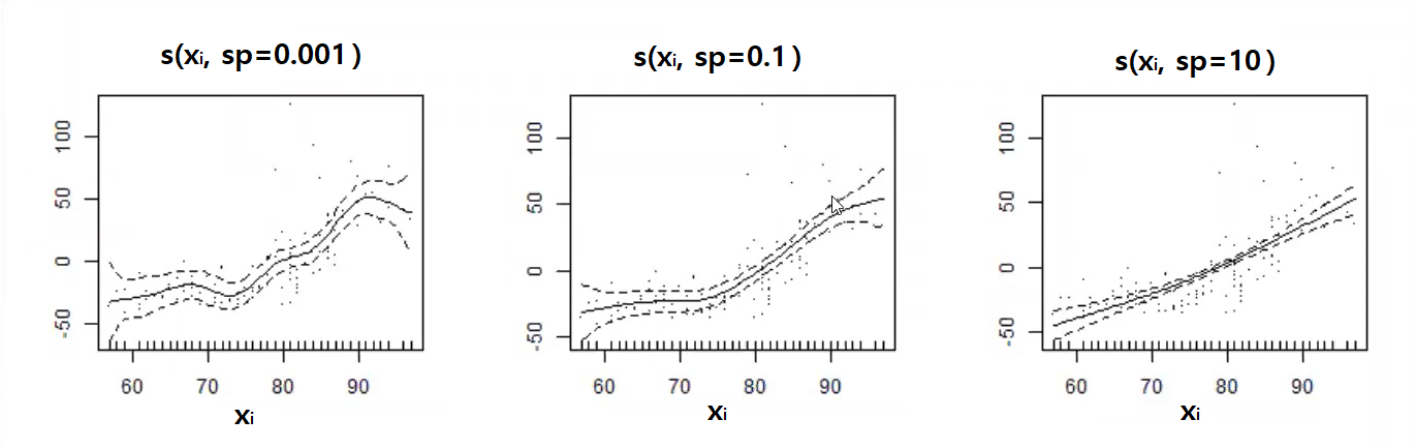

FIGURE 3.3: Exemple de differents paramètres de lissage **sp**.

Pour cet exemple, on remarque que pour **sp=0.001** on a une courbure qui est beaucoup moins lisse, et qui peut entrainer le sur-ajustement des données. Par contre, pour **sp=10**, on pénalise beaucoup la courbure, on est presque à une ligne droite, et cela peut entrainer le sous-ajustement des données. Un paramètre de lissage **sp=0.1** entre les deux situations précédentes, présente un meilleur compromis entre le sur-ajustement et le sous-ajustement.

En pratique :
• On choisit un nombre de bases **k** élevé pour permettre la representation de plusieurs formes fonctionnelles.
• On utilise un algorithme qui choisit λ (**sp**) pour atteindre un compromis optimal entre sous-ajustement et sur-ajustement.

Plusieurs algorithmes sont disponibles pour la fonction **gam** pour R, le maximum de vraisemblance restreint **(method = "REML")** est d'avantage recommandé.

3.5 Concurvité

La concurvité est un diagnostique qu'on peut appliquer à notre modèle additif généralisé qui représente l'équivalent non paramétrique de la colinéarité pour les relations non linéaires.

La concurvité se produit lorsqu'un terme lisse dans un modèle peut être approximé par un ou plusieurs des autres termes lisses du modèle. C'est souvent le cas lorsqu'un lissage de l'espace est inclus dans un modèle, avec des lissages d'autres covariables qui varient également plus ou moins doucement dans l'espace. De même, cela a tendance à être un problème dans les modèles incluant un lissage du temps, ainsi que des lissages d'autres covariables variant dans le temps.

La concurvité peut être considérée comme une généralisation de la colinéarité et pose des problèmes similaires d'interprétation. Cela peut également rendre les estimations quelque peu instables (de sorte qu'elles deviennent sensibles à des détails de modélisation apparemment inoffensifs, par exemple).

3.6 Modéliser les interactions

3.6.1 Intéraction entre spline et facteur

Dans R, **bam** est une version de **gam** adaptée aux modèles additifs généralisés plus complexes avec beaucoups de données, ce qui réduit l'utilisation de la mémoire et le temps de calcul.

Avec $s(x_i, by = facteur)$ on spécifie qu'une spline differente doit être ajustée pour chacun des facteurs. Chaque spline a une moyenne de 0, donc le terme séparé + **facteur** représente les differences de réponse moyenne entre les facteurs.

3.6.2 Intéraction avec un paramètre de lissage commun

Avec $s(x_i, facteur, bs = "fs")$ (**fs** : factor-smooth interaction), on estime des splines séparés pour chaque facteur, mais ces splines doit partager le même paramètre de lissage λ.

3.6.3 Intéraction entre deux variables numériques

$te(x_i, x_j)$, (**te** : tensor product), définit une spline en plusieurs dimensions, où le paramètre de lissage est choisi séparément pour chaque dimension.

$s(x_i, x_j)$ définit une spline en plusieurs dimensions avec un paramètre de lissage unique ; utile pour le lissage de données spatiales.

3.7 Choix de modèle

Lorsque les paramètres de lissage sont estimés dans le cadre de l'ajustement du modèle, une grande partie de ce qui serait traditionnellement considéré comme une sélection de modèle a été absorbé dans le processus d'ajustement : l'estimation des paramètres de lissage a déjà été sélectionné parmi une riche famille de modèles de complexité fonctionnelle différente. Cependant, l'estimation des paramètres de lissage ne supprime généralement pas complètement un terme lisse du modèle, car la plupart des pénalités laissent certaines fonctions non pénalisées.

La question de savoir si un terme doit être dans le modèle demeure donc. Une approche simple à ce problème consiste à ajouter une pénalité supplémentaire à chaque terme lisse dans le GAM, ce qui pénalise les composants du lisse qui autrement ne seraient pas pénalisés (et uniquement ceux-ci). Dans les paramètres de haute dimension, il peut être plus judicieux d'essayer cette tâche en utilisant la régularisation Lasso (apprentissage statistique) ou Elastic net . L'amplification effectue également automatiquement la sélection des termes dans le cadre de l'ajustement.

Une alternative consiste à utiliser des méthodes de régression pas à pas traditionnelles pour la sélection du modèle. Il s'agit également de la méthode par défaut lorsque les paramètres de lissage ne sont pas estimés dans le cadre de l'ajustement, auquel cas chaque terme de lissage est généralement autorisé à prendre l'un d'un petit ensemble de niveaux de lissage prédéfinis dans le modèle, et ceux-ci sont sélectionnés dans un mode pas à pas.

Les méthodes pas à pas fonctionnent en comparant itérativement des modèles avec ou sans termes de modèle particuliers, et nécessitent des mesures de l'ajustement du modèle ou de l'importance des termes afin de décider quel modèle sélectionner à chaque étape. Par exemple, nous pourrions utiliser des valeurs p (p-value) pour tester chaque terme d'égalité à zéro afin de décider des termes candidats à

supprimer d'un modèle, et nous pourrions comparer les valeurs du critère d'information Akaike (AIC) pour les modèles alternatifs.

3.8 Vérification du modèle

Comme pour tout modèle statistique, il est important de vérifier les hypothèses du modèle d'un GAM. Les parcelles résiduelles doivent être examinées de la même manière que pour tout GLM.La seule vérification supplémentaire que les GAM introduisent est la nécessité de vérifier que les degrés de liberté choisis sont appropriés.

Cela est particulièrement grave lorsque vous utilisez des méthodes qui n'estiment pas automatiquement la régularité des composants du modèle. Lors de l'utilisation de méthodes avec sélection automatique des paramètres de lissage, il est toujours nécessaire de vérifier que le choix de la dimension de base n'était pas restrictivement petit, bien que si les degrés de liberté effectifs d'une estimation de terme sont confortablement inférieurs à sa dimension de base, cela est peu probable.

Dans tous les cas, vérifier $s(x_i)$ est basé sur l'examen du modèle des résidus par rapport à x_i. Cela peut être fait en utilisant des résidus partiels superposés sur le tracé de $\widehat{s}(x_i)$, ou en utilisant la permutation des résidus pour construire des tests de motif résiduel (comme dans la fonction 'gam.check 'du package R' mgcv').

3.9 Modélisation de la fréquence des sinistres

Le nombre de sinistres peut s'observer comme la réalisation d'une variable aléatoire discrète positive suivant une loi de type Poisson, Binomiale Négative ou binomiale. Dans notre cadre, $y_i = "CaimInd"$: $y_i \in \{0, 1\}$ alors naturellement on va utiliser la loi de Bernoulli pour modèliser $y_i = "CaimInd"$.

3.9.1 Regression logistique de $y_i = "ClaimInd"$

On a $y_i|x_i \sim Bernoulli(p_i)$ avec $p_i = P(y_i = 1|x_i) = E[y_i|x_i]$, et $q_i = 1 - p_i$, on choisit la fonction lien canonique **logit** : $g(E[y_i|x_i]) = logit(p_i) = log(\frac{p_i}{1-p_i}) = x_i^{DISCRETE}\beta + s(x_i^{CONTINUE})$.
Si $\widehat{\beta}$ est un bon estimateur de β alors :

$$\widehat{p_i} = g^{-1}(x_i^{DISCRETE}\widehat{\beta} + s(x_i^{CONTINUE})) = \frac{e^{x_i^{DISCRETE}\widehat{\beta}+s(x_i^{CONTINUE})}}{1+e^{x_i^{DISCRETE}\widehat{\beta}+s(x_i^{CONTINUE})}}.$$

On regroupe les deux bases de données freMPL3 et freMLP4 dans une base freMPL34, 80% de freMPL34 représente notre base d'apprentissage "freMPL34.app" de notre modèle et 20% de freMPL34 représente notre base de test "freMPL34.test" du modèle.
Pour simplifier nos bases au niveau des facteurs, nous avons fait appel à un modèle matriciel "freMPL34.app.reg" et "freMPL34.test.reg" et on a fait quelques transformations au niveau des variables.

Le choix des bases d'apprentissage et de test est fait d'une facon aléatoire.
'

3.9.2 Apprentissage du modèle additif généralisé $y_i = "ClaimInd_i"$

Afin de sélectionner notre modèle optimal, nous avons appliqué l'algorithme "AIC" (stepAIC(GAM.claimInd) sur R) et "REML".
On a testé les méthodes Forward, Backward et Stepwise en comparant leur AIC.

On trouve un AIC plus petit que l'AIC précedant. cela nous a permis d'éliminer beaucoup de variables qui n'expliquent pas nos modèles de regression logistique "GAM.ClaimInd".

Notre modèle optimal "GLM.ClaimInd" $y_i = "ClaimInd_i"$ sous R est :

```
GAM.claimInd <- gam(formula = ClaimInd ~ s(LicAge, k=34) + BonusMalus + VehAge10 +
                      VehAge6 + VehAge8 + SocioCategCSP6 + VehUsageProfessional +
                      VehUsageProfessional_run + DeducTypePartially_refunded +
                      DeducTypeRefunded + VehBodystation_wagon + VehPriceO +
                      VehPriceP + VehPriceQ + VehMaxSpeed140_150_km_h +
                      VehMaxSpeed150_160_km_h + VehMaxSpeed160_170_km_h +
                      VehMaxSpeed170_180_km_h + VehMaxSpeed190_200_km_h +
                      VehMaxSpeed200_220_km_h + VehMaxSpeed220_km_h + VehClassA +
                      GaragePrivate_garage,
                      family = binomial("logit"),
    data = cbind.data.frame(ClaimInd = freMPL34.app$ClaimInd, freMPL34.app.reg),
                      method="REML")
```

La sortie *summary*(*GAM.claimInd*) sur R nous donne la valeur des p-value des coefficients, Le score de REML et la deviance expliqueé :

```
Parametric coefficients:
                            Estimate Std. Error z value Pr(>|z|)
(Intercept)                -4.322152   0.140117 -30.847  < 2e-16 ***
BonusMalus                  0.005181   0.001700   3.048 0.002302 **
VehAge10                   -1.008153   0.152090  -6.629 3.39e-11 ***
VehAge6                    -0.329911   0.101129  -3.262 0.001105 **
VehAge8                    -0.536838   0.129592  -4.143 3.44e-05 ***
SocioCategCSP6              0.300521   0.097604   3.079 0.002077 **
VehUsageProfessional        0.235445   0.070523   3.339 0.000842 ***
VehUsageProfessional_run    0.533950   0.174780   3.055 0.002251 **
DeducTypePartially_refunded 0.382399   0.088139   4.339 1.43e-05 ***
DeducTypeRefunded          -0.343600   0.156512  -2.195 0.028138 *
VehBodystation_wagon       -0.370132   0.152573  -2.426 0.015269 *
VehPriceO                  -0.268876   0.166752  -1.612 0.106869
VehPriceP                  -0.458402   0.203512  -2.252 0.024293 *
VehPriceQ                  -0.453892   0.161813  -2.805 0.005031 **
VehMaxSpeed140_150_km_h     0.504574   0.142387   3.544 0.000395 ***
VehMaxSpeed150_160_km_h     0.350835   0.105347   3.330 0.000868 ***
VehMaxSpeed160_170_km_h     0.341104   0.097436   3.501 0.000464 ***
VehMaxSpeed170_180_km_h     0.259117   0.104434   2.481 0.013096 *
VehMaxSpeed190_200_km_h     0.341020   0.110798   3.078 0.002085 **
VehMaxSpeed200_220_km_h     0.469447   0.117013   4.012 6.02e-05 ***
VehMaxSpeed220_km_h         0.494673   0.175706   2.815 0.004873 **
VehClassA                   0.336494   0.090310   3.726 0.000195 ***
GaragePrivate_garage       -0.175548   0.096162  -1.826 0.067919 .
---
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(LicAge)  1.969   2.534   1.176     0.6

R-sq.(adj) =  0.0035   Deviance explained = 1.75%
-REML = 5945.7  Scale est. = 1         n = 53499
```

3.9.3 Verification du modèle additif généralisé $y_i = "ClaimInd_i"$

Visualisé l'effet de LicAge sur l'échelle du logit

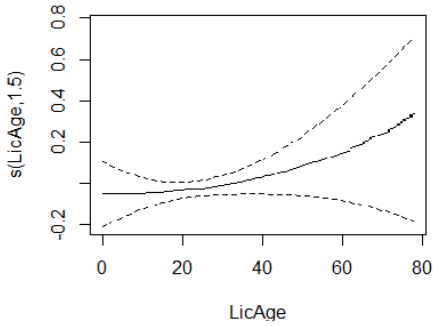

FIGURE 3.4: Sortie R du plot(GAM.ClaimInd)

On observe le spline estimé pour **LicAge** sous l'échelle **logit**. Les intervalles de confiances augmentent lorsqu'on a les valeurs éleves de LicAge, car on a pas beaucoups de données à ce niveau là.

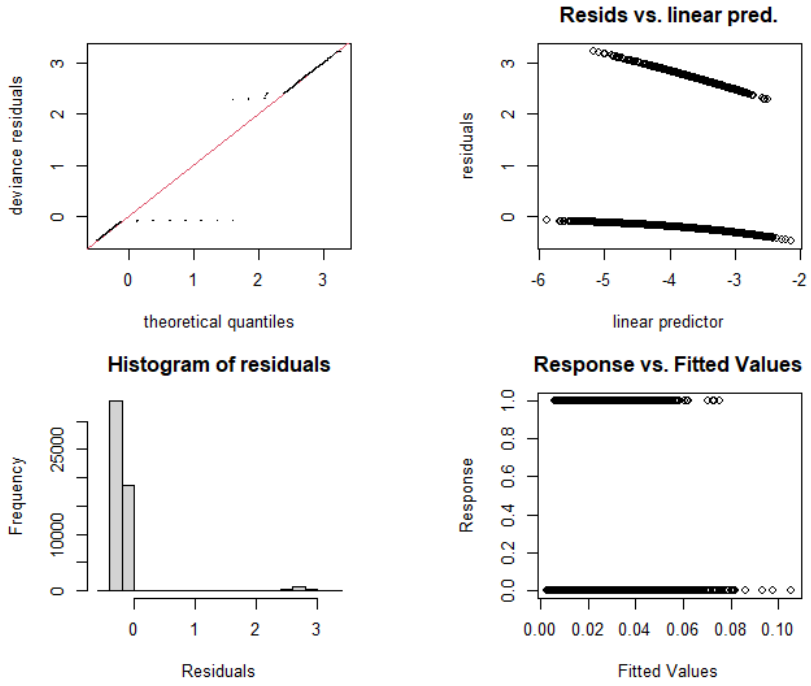

FIGURE 3.5: Sortie R gam.check(GAM.claimInd)

D'après le graphe, on a deux groupes de résidus, des résidus négatifs si la réponse était "1", parce

que on ne peut jammais prédire exactement "1", et des résidus positifs si la réponse était "0", car on ne peut jammais prédire exactement "0".

Vérification des résidus

La fonction **binnedplot** du package arm : résidu moyen par groupes de points, ordonnés en focntion de la prédiction moyenne par groupe, avec un intervalle de prédiction à 95%.

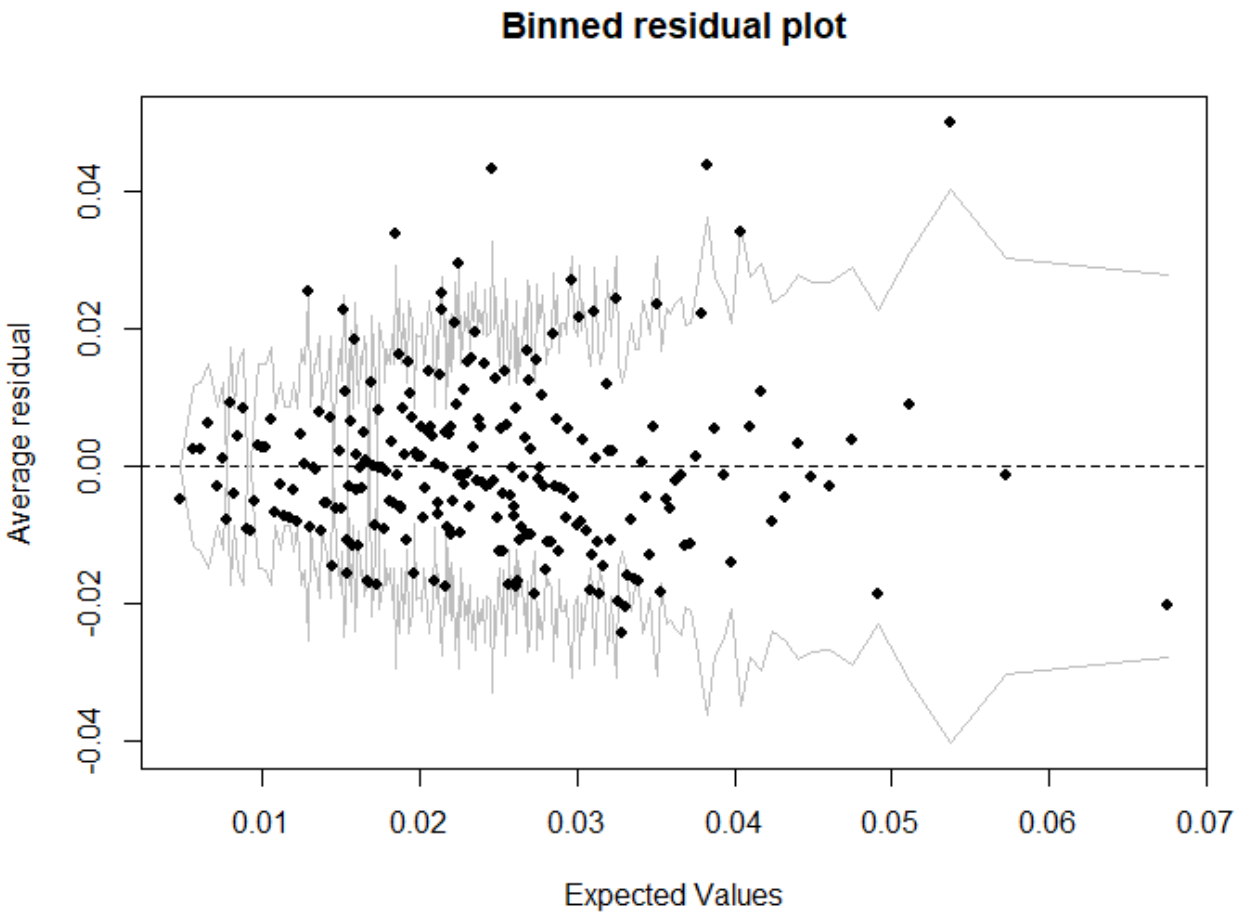

FIGURE 3.6: Sortie R des résidus

On a regroupé toutes nos observations en fonction de la valeur prédite, on remarque que 95% des prédictions groupées sont dans l'intervalle de prédiction, donc on a une bonne description des résidus.

La figure 3.8 affiche le tracé des résidus regroupés produit. Il y a une courbure pour la configuration des résidus regroupés, bien que ce ne soit pas particulièrement extrême.

On observe aussi qu'il a quelques résidus qui ne sont pas dans l'intervalle de prédiction à 95%, cela peu-être expliqué par exemple par l'absence des données pour des varibles, on ne dispose de beaucoup de données pour **LicAge** au niveau des valeurs élevées.

3.9.4 Prédiction du modèle linéaire généralisé $y_i = "ClaimInd_i"$

Afin de determiner le seuil s de notre variable de prédiction $\widehat{y_i} = \mathbb{1}_{p_i > s}$, on calcule la probabilité de "ClaimInd=1" dans notre base freMPL3, on trouve : $s = 0.02404522$.

Matrice de confusion sur l'échantillon test

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 7279  141
         1 5760  195

               Accuracy : 0.5588
                 95% CI : (0.5503, 0.5672)
    No Information Rate : 0.9749
    P-Value [Acc > NIR] : 1

                  Kappa : 0.0152

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 0.55825
            Specificity : 0.58036
         Pos Pred Value : 0.98100
         Neg Pred Value : 0.03275
             Prevalence : 0.97488
         Detection Rate : 0.54422
   Detection Prevalence : 0.55477
      Balanced Accuracy : 0.56930

       'Positive' Class : 0
```

Notons que les valeurs dans la matrice de confusion dépendont su seuil s avec lequel on comparera les probabilités estimées par le modèle. Plusieurs indicateurs peuvent être calculés à partir de cette matrice :
• La jutesse (Accuracy) : proportion de prédictions/classifications correctes :
$Accuracy = \frac{VP+VN}{VP+VN+FP+FN} = 0.5465$.
• TVP (sensitivity) : le taux de VP par rapport aux positifs totaux $P = VP + FN$:
$TVP = \frac{VP}{VP+FN} = 0.55825$.
• TVN (specificity) : le taux de VN par rapport aux négatifs totaux $N = VN + FPN$:
$TVN = \frac{VN}{VN+FP} = 0.58036$.

Estimation de l'erreur de prédiction sur l'échantillon test

On trouve que l'erreur de prédiction sur l'échantillon test est estimé à 2.3%.
La matrice de confusion nous indique un problème de prédiction pour les individus ayant un sinistre. Il serait donc possible d'augmenter le seuil s, mais nous déterminer en calculant la probabilité de survenance d'un sinistre sur la base de données test.Nous avons donc fait :
s = (nombre de "ClaimInd=1") / (nombre total de "ClaimInd=0" et "ClaimInd=1").

Cependant nous ne nous intéresserons qu'aux probabilités de survenance de ce sinistre et pas à la prédiction de sa survenance. Nous acceptons donc le modèle.

3.10 Modélisation du coût des sinistres

Le choix d'une loi pour modéliser le coût des sinistres est plus délicat que celui pour modéliser leur fréquence. Un phénomène de comptage ou d'occurrence est en effet plus intrinsèquement lié à des phénomènes statistiques que celui de la sévérité d'un sinistre. Ainsi le choix par défaut se porte sur la loi Gamma qui permet de modéliser des données continues positives à queue épaisse ([Murphy,2000]).

3.10.1 Modèle de regression Gamma $y_i = "ClaimAmount"$

On a $y_i \in \mathbb{R}^+$ donc $y_i|x_i \sim Gamma(\alpha, \beta)$, sa fonction de densité f_Y est donnée par :

$f_Y(y) = \frac{\beta^\alpha}{\Gamma(\alpha)} y^{\alpha-1} e^{-y\beta}$, $\forall y \in \mathbb{R}^+$ avec $\alpha > 0$ le paramètre de forme et $\beta > 0$ le paramètre d'intensité.

On choisit la fonction lien **log** : $g(E[y_i|x_i]) = log(E[y_i|x_i]) = x_i^{DISCRETE}\beta + s(x_i^{CONTINUE})$
alors $E[y_i|x_i] = \exp(x_i^{DISCRETE}\beta + s(x_i^{CONTINUE}))$.

On regroupe les deux bases de données freMPL3 et freMLP4 dans une base freMPL34, 80% de freMPL34 représente notre base d'apprentissage "freMPL34.app.sinistre" de notre modèle et 20% de freMPL34 représente notre base de test "freMPL34.test.sinistre" du modèle.

On s'interesse que pour les valeurs $ClaimAmount > 0$.

Pour simplifier nos bases au niveau des facteurs, nous avons fait appel à un modèle matriciel "freMPL34.app.sinistre.reg" et "freMPL34.test.sinistre.reg" et on a fait quelques transformations au niveau des variables. Le choix des bases d'apprentissage et de test est fait d'une facon aléatoire.

3.10.2 Apprentissage du modèle additif généralisé $y_i = "ClaimAmount_i"$

Afin de sélectionner notre modèle optimal, nous avons appliqué l'algorithme "AIC" et "RMEL" (stepAIC(GAM.ClaimAmount) sur R), on a testé les méthodes Forward, Backward et Stepwise en comparant leur AIC.

On trouve un AIC plus petit que l'AIC précedant. Cela nous a permis d'éliminer beaucoup de variables qui n'expliquent pas nos modèles linéaires généralisés "GAM.ClaimAmount".

Notre modèle optimal "GAM.ClaimAmount" $y_i = "ClaimAmount_i"$ sous R est :

```
GAM.claimAmount <- gam(ClaimAmount ~ s(LicAge, k=62)+ s(BonusMalus) + HasKmLimit +
                         s(RiskVar) + VehAge3 + VehAge4 + VehAge6 + VehAge8 +
                         DeducTypePartially_refunded +  VehPriceE + VehPriceJ +
                         VehPriceM + VehPriceN + VehPriceQ + VehEnergyregular +
                         VehMaxSpeed140_150_km_h + VehMaxSpeed160_170_km_h +
                         VehMaxSpeed170_180_km_h + VehMaxSpeed180_190_km_h +
                         VehMaxSpeed190_200_km_h + VehMaxSpeed200_220_km_h +
                         VehClassA + VehClassM1 + GarageNone +
                         VehUsagePrivate_trip_to_office + VehBodycabriolet,
                         family = Gamma(link = "log"),
        data = cbind.data.frame(ClaimAmount = freMPL34.app.sinistre$ClaimAmount,
                         freMPL34.app.sinistre.reg),
                         weights = VehBodymicrovan + VehBodycoupe +
                         SocioCategCSP2 + SocioCategCSP4 +SocioCategCSP5 +
                         SocioCategCSP7 + SocioCategCSP9, method = "REML")
```

La sortie *summary*(*GAM.claimAmount*) sur R nous donne la valeur des p-value des coéfficients, le score de REML et la deviance expliquée :

```
Parametric coefficients:
                               Estimate Std. Error t value Pr(>|t|)
(Intercept)                     7.52950    0.09790  76.907  < 2e-16 ***
HasKmLimit                      0.22699    0.10636   2.134 0.033029 *
VehAge3                        -0.13995    0.08281  -1.690 0.091292 .
VehAge4                        -0.08245    0.08371  -0.985 0.324863
VehAge6                        -0.14824    0.10042  -1.476 0.140164
VehAge8                        -0.13333    0.12775  -1.044 0.296814
DeductypePartially_refunded    -0.41719    0.08628  -4.835 1.50e-06 ***
VehPriceE                      -0.38947    0.10771  -3.616 0.000311 ***
VehPriceJ                       0.11977    0.09396   1.275 0.202677
VehPriceM                      -0.03965    0.12691  -0.312 0.754765
VehPriceN                       0.12103    0.11026   1.098 0.272570
VehPriceQ                       0.93524    0.14732   6.348 3.04e-10 ***
VehEnergyregular               -0.13103    0.06420  -2.041 0.041467 *
VehMaxSpeed140_150_km_h         0.05293    0.12760   0.415 0.678345
VehMaxSpeed160_170_km_h         0.15209    0.08550   1.779 0.075495 .
VehMaxSpeed170_180_km_h         0.40692    0.09446   4.308 1.78e-05 ***
VehMaxSpeed180_190_km_h         0.24506    0.10116   2.422 0.015559 *
VehMaxSpeed190_200_km_h         0.21396    0.10518   2.034 0.042140 *
VehMaxSpeed200_220_km_h         0.27826    0.11190   2.487 0.013023 *
VehClassA                       0.05344    0.09654   0.554 0.579982
VehClassM1                     -0.16661    0.06934  -2.403 0.016419 *
GarageNone                     -0.01963    0.06692  -0.293 0.769279
VehUsagePrivate_trip_to_office  0.04576    0.05539   0.826 0.408889
VehBodycabriolet                0.10188    0.13823   0.737 0.461275
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Approximate significance of smooth terms:
                edf Ref.df      F  p-value
s(LicAge)     1.006  1.011  1.155 0.281221
s(BonusMalus) 1.054  1.105 13.798 0.000199 ***
s(RiskVar)    2.329  2.925  2.752 0.058602 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

R-sq.(adj) =  0.114   Deviance explained = 13.5%
-REML = 8518.5  Scale est. = 0.77738   n = 1272
```

Plus la p-valeur est petite, plus la variable a une grande probabilité d'être importante, d'après la sorte *summary*, le lissage de **LicAge, BonusMalus** et **RiskVar** est validé.

D'après le graphe suivant, on remarque que le modèle fit bien par rapport au variables continues RiskVar, LicAge et BonusMalus.

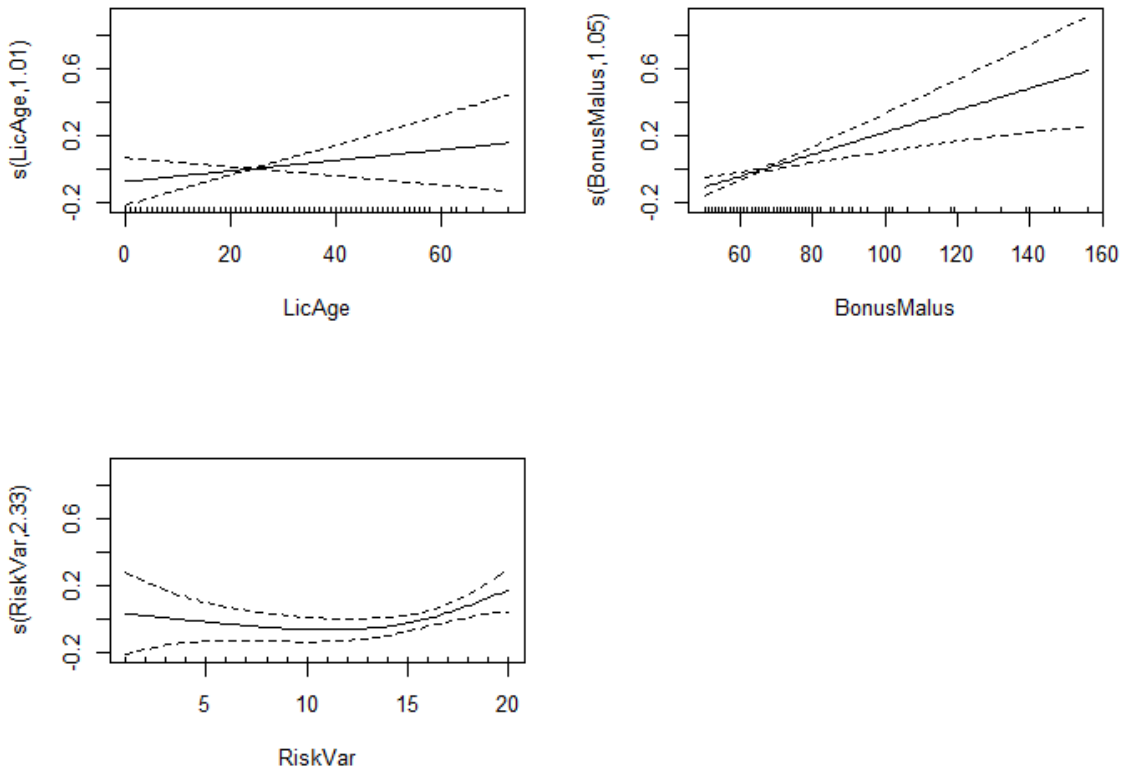

FIGURE 3.7: Sortie R des ajustement

Pour verifier la qualité du modèle, on regardre si le choix de **k** est bien validé :

```
Method: REML   Optimizer: outer newton
full convergence after 9 iterations.
Gradient range [-0.001489986,0.01204624]
(score 8518.484 & scale 0.7773812).
Hessian positive definite, eigenvalue range [0.0009076406,610.3785].
Model rank =  103 / 103

Basis dimension (k) checking results. Low p-value (k-index<1) may
indicate that k is too low, especially if edf is close to k'.

                 k'   edf k-index p-value
s(LicAge)     61.00  1.01    0.89   0.045 *
s(BonusMalus)  9.00  1.05    0.90   0.150
s(RiskVar)     9.00  2.33    0.96   0.850
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

On remarque que les **k'** et les **edf** sont loin donc on peut validé nos choix des valeurs **k**.

3.10.3 Verification du modèle additif généralisé $y_i = "ClaimAmount_i"$

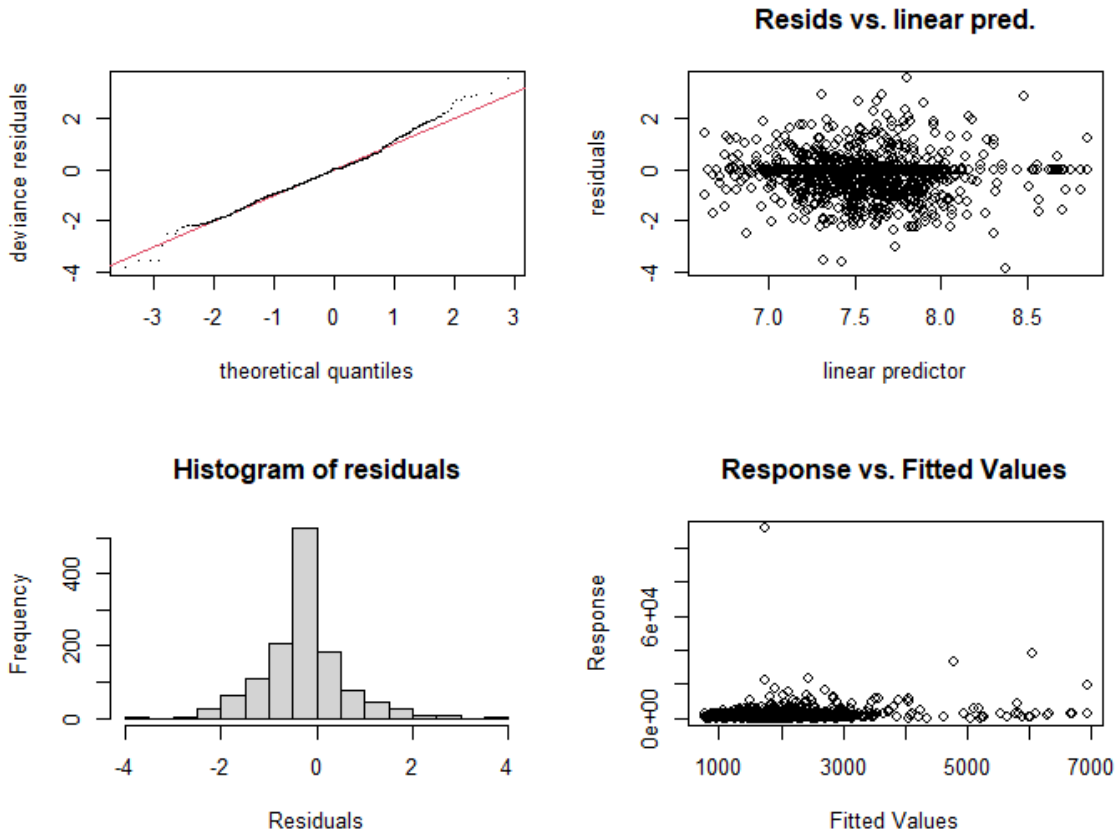

FIGURE 3.8: Sortie R gam.check(GAM.claimAmount)

D'après la sortir R $gam.check(GAM.claimAmount)$:

• Pour le graphe qui représente les déviances residuelles en fonction des quantiles théoriques, on on obtient approximativement une droite par rapport la première bissectrice des axes.

• Pour le graphe qui représente les résidus en fonction de la linearité du prédicteur, le nuage de points est centré autour de 0.

• Pour l'histograme des résidus, on remarque qu'il est symétrique.

• Pour le graphe qui représente la réponse, le nuage de points est quasiment centré autour de 0.

Alors le modèle additif généralisé est vérifié.

Vérification des résidus

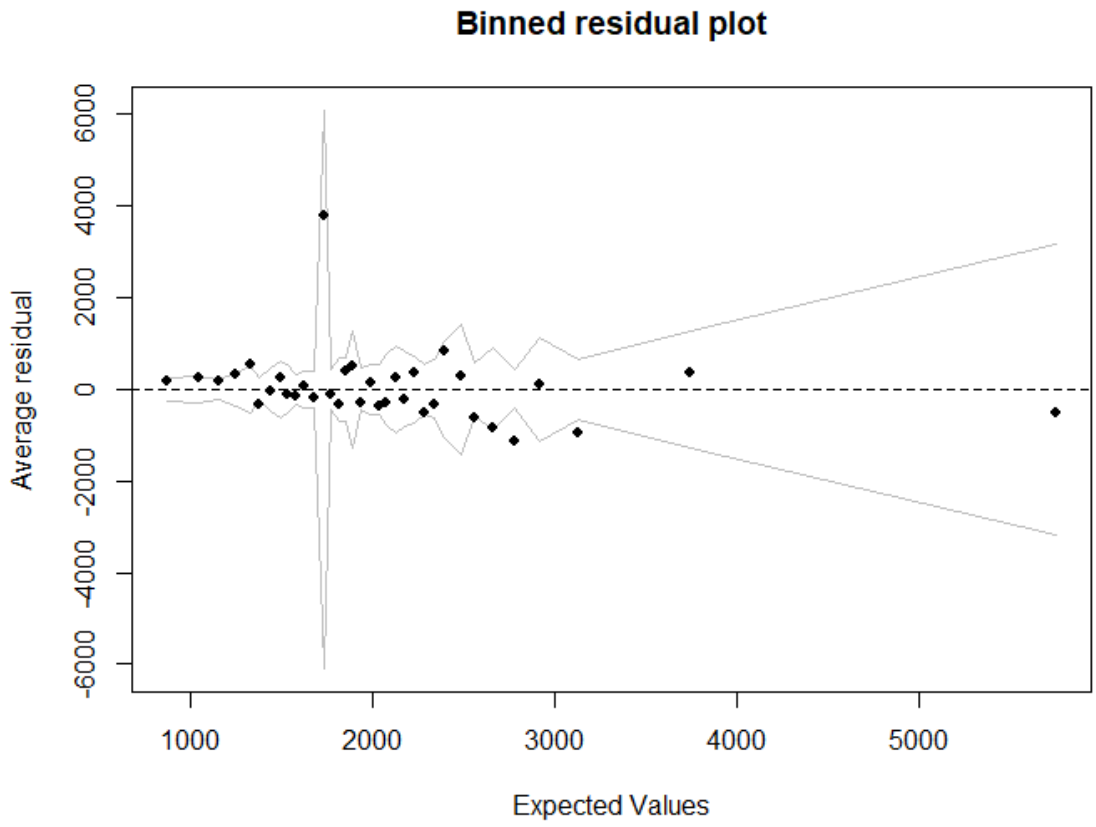

FIGURE 3.9: Sortie R des résidus

On a regroupé toutes nos observations en fonction de la valeur prédite, on remarque que 95% des prédictions groupées sont dans l'intervalle de prédiction, donc on a une bonne description des résidus.

La figure 3.15 affiche le tracé des résidus regroupés produit. Il y a une courbure pour la configuration des résidus regroupés, bien que ce ne soit pas particulièrement extrême.

On observe aussi qu'il n'y a pas résidus qui ne sont pas dans l'intervalle de prédiction à 95%, alors le modèle s'ajuste bien aux données.

3.10.4 Prédiction du modèle additif généralisé $y_i = "ClaimAmount_i"$

On trace le graphe des "ClaimAmount" dans notre base de données d'apprentissage et la prédiction du modèle additif généralisé pour un échantillion d'apprentissage :

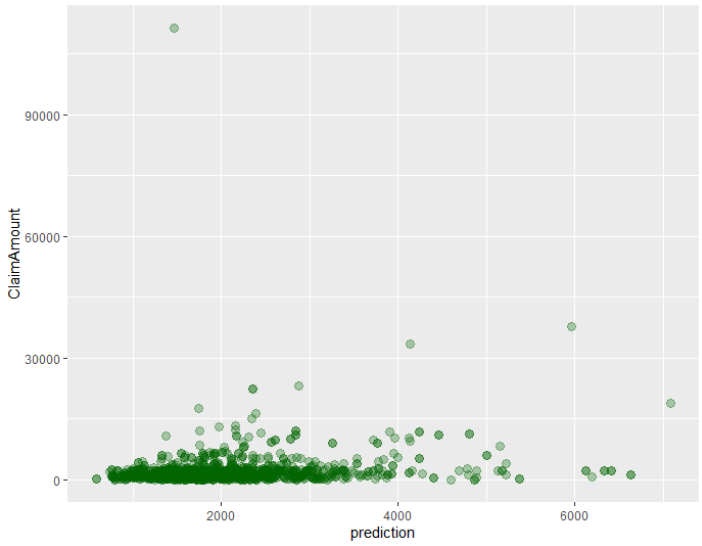

FIGURE 3.10: Graphe de ClaimAmount et prédiction pour l'apprentissage

On remarque que le modèle répond bien au données d'apprentissage puisqu'il y a pas dispertion de points sur le graphe.

On trace le graphe des "ClaimAmount" dans notre base de données de test et la prédiction du modèle linéaire additif pour un échantillion de test :

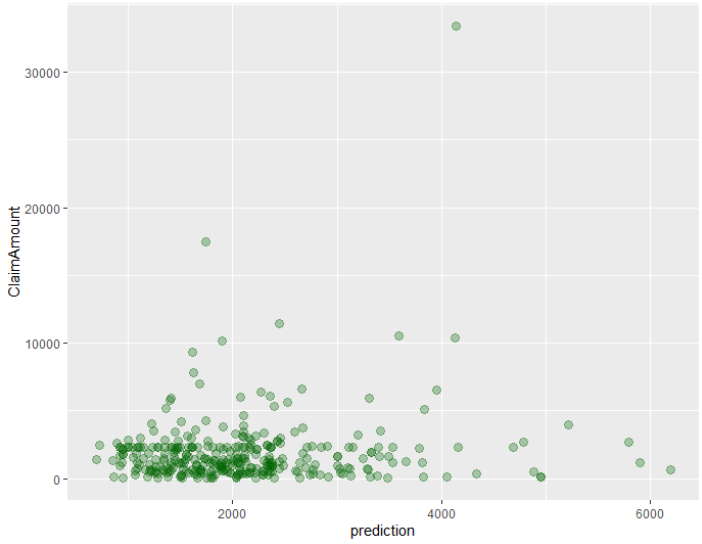

FIGURE 3.11: Graphe de ClaimAmount et prédiction pour le test

On remarque que le modèle répond bien au données de test puisqu'il y a pas dispertion de points sur le graphe.

On trace les points de "ClaimAmount" de notre base de test et les points de prediction sur l'échantillion de test :

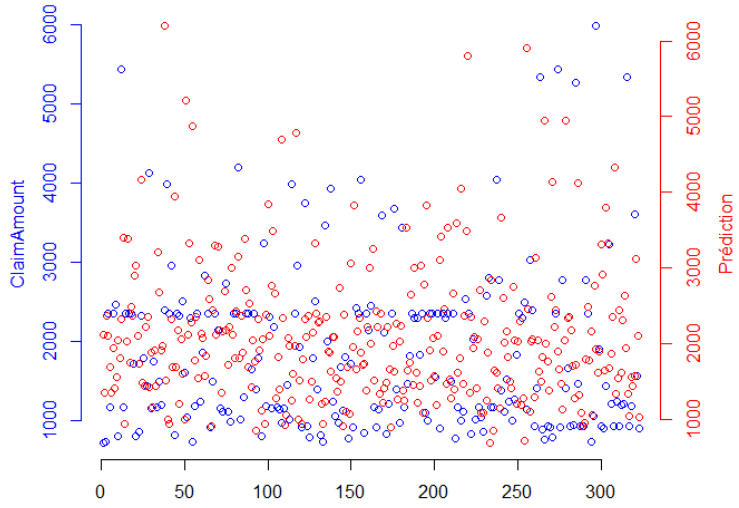

FIGURE 3.12: Graphe de ClaimAmount et prédiction pour le test

On conclut que notre modèle additif généralisé s'adapte bien aux données.

Nous pouvons désormais observer la répartition des sinistres déclarés prédits en fonction des différentes classes d'assurés.

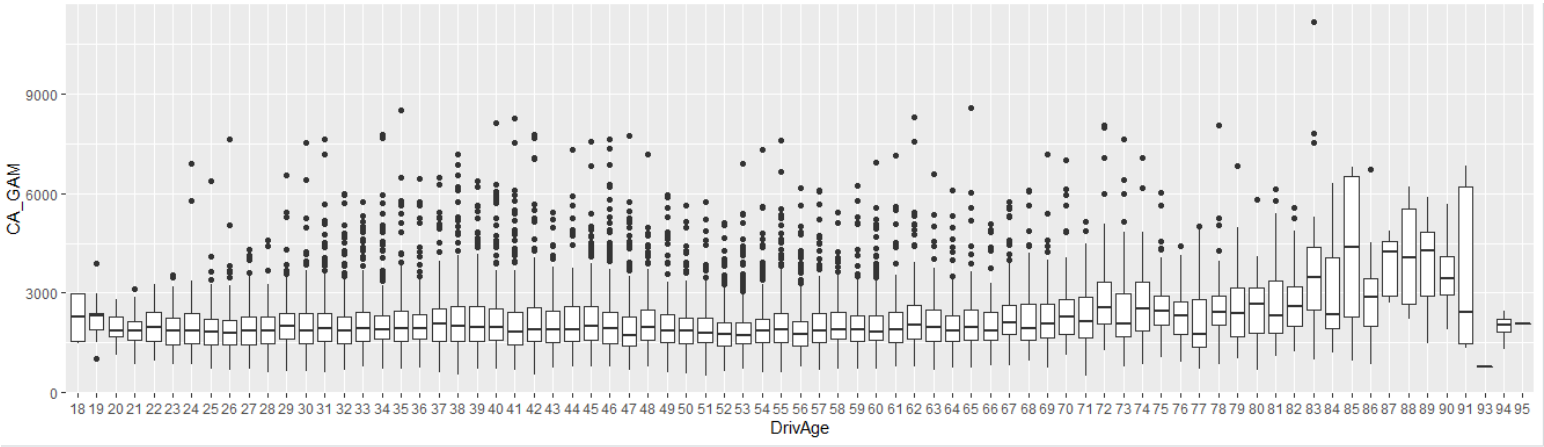

FIGURE 3.13: Boxplot de la prédiction des sinistres déclarés en fonction du nombre d'années de permis de l'assuré.

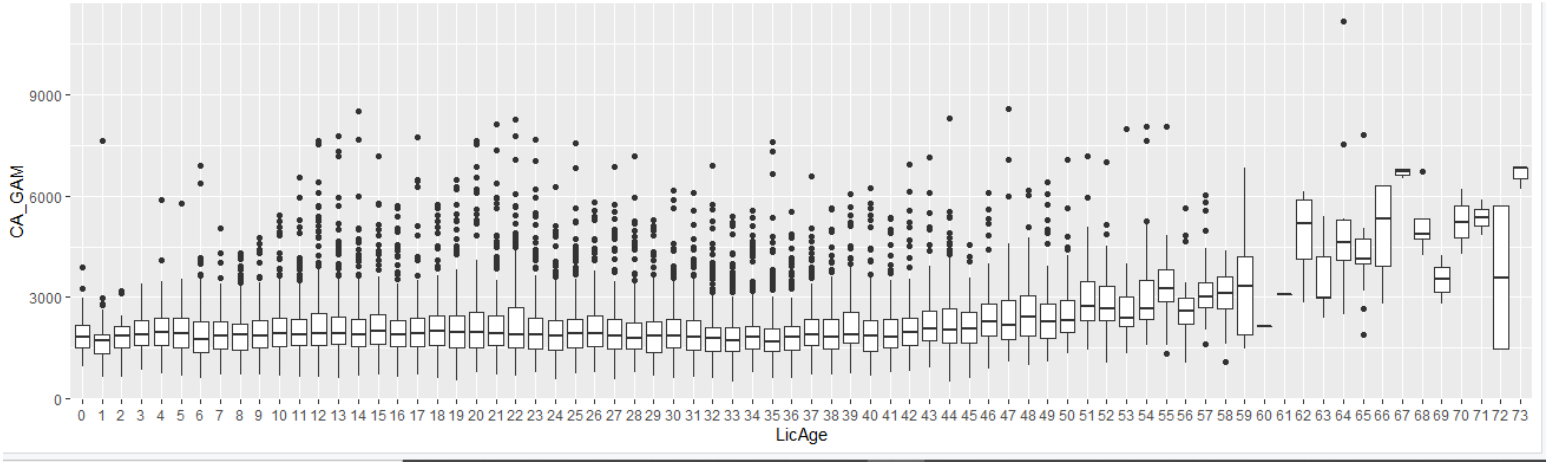

FIGURE 3.14: Boxplot de la prédiction des sinistres déclarés en fonction de l'âge de l'assuré.

On observe que les sinistres déclarés prédits ont tendance à être plus élevés lorsque les individus sont assez âgés.

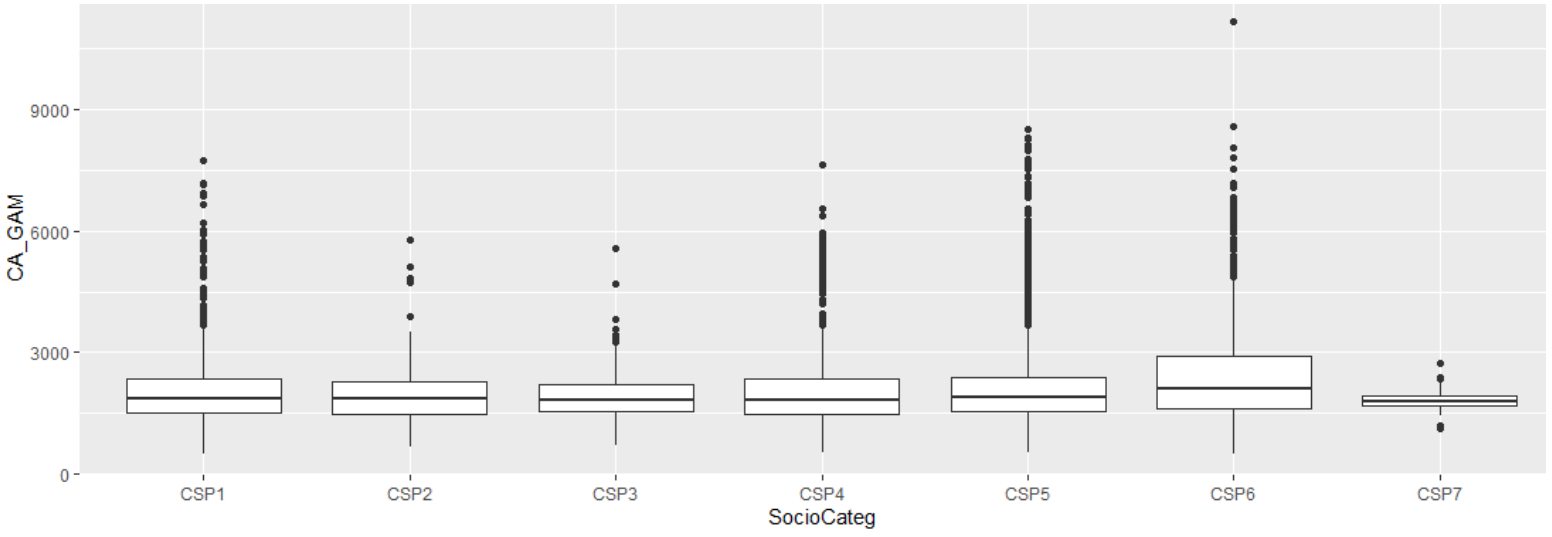

FIGURE 3.15: Boxplot de la prédiction des sinistres déclarés en fonction de la CSP de l'assuré.

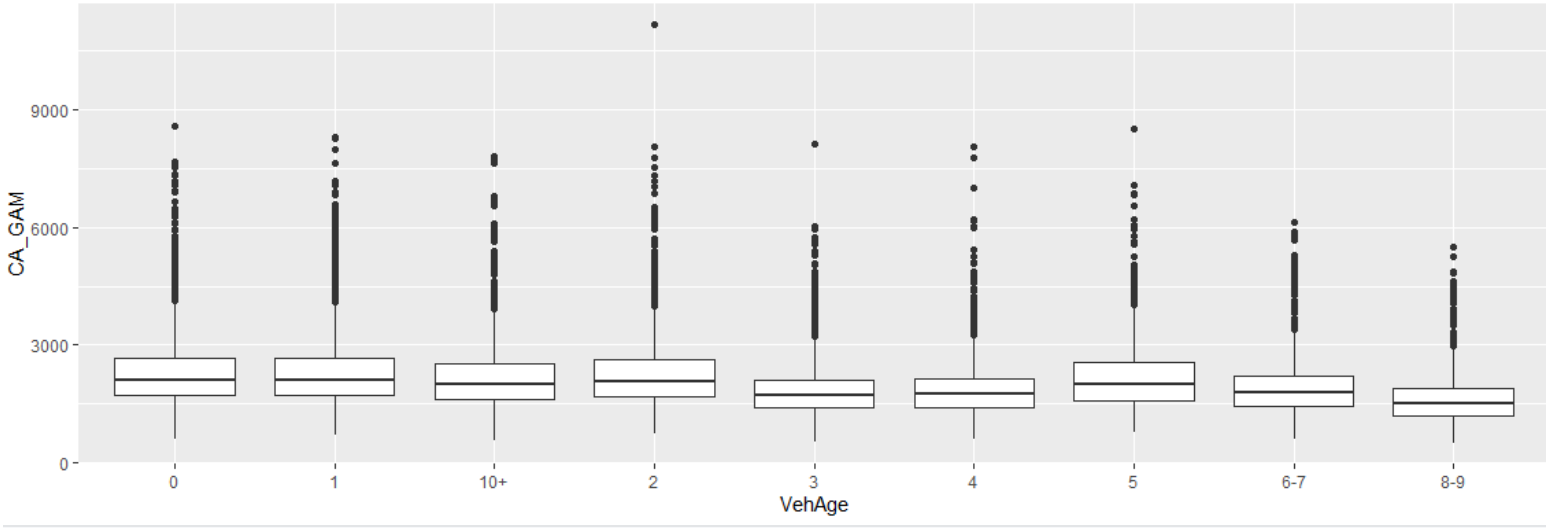

FIGURE 3.16: Boxplot de la prédiction des sinistres déclarés en fonction de l'ancienneté du véhicule de l'assuré.

On voit peu de différences sur la prédiction des sinistres en fonction de la CSP des assurés. La CSP7 semble plus écrasée mais cela est certainement dû au nombre limité de CSP7 dans notre base de donnnées test.

On remarque que les sinistres prédits ont tendance à être un peu plus importants lorsque les véhicules sont récents plutôt que lorsqu'ils sont anciens. Cependant la différence n'est pas flagrante.

3.11 La prime pure

Dans le domaine de l'assurance, la prime pure correspond au montant de sinistres moyen auquel devra faire face l'assureur. En faisant payer une telle prime à ses assurés, il aura en moyenne un bénéfice nul.

Dans un contexte d'assurance non vie, la prime pure peut être modélisée par une approche fréquence-sévérité, qui repose sur les hypothèses suivantes :
• Les charges des sinistres individuels sont des variables aléatoires indépendantes et identiquement distribuées.
• Les variables représentant le fait d'avoir un sinistre ou non, sont des variables aléatoires indépendantes et identiquement distribuées.
• La fréquence et la sévérité sont supposées indépendantes.

La charge totale des sinistres de l'assuré i est modélisée, sur une période d'assurance donnée, par le produit suivant :

$$X_i = I_i B_i.$$

où B_i désigne le montant du sinistre individuel de l'assuré i et I_i est le fait que l'assuré est un sinistre ($I_i = 1$) ou non ($I_i = 0$). Sous l'hypothèse d'indépendance entre la fréquence et la sévérité, nous obtenons :

$$E[X_i] = E[I_i].E[B_i].$$

Ainsi, la prime actuarielle peut être estimée :
• En calibrant séparément un modèle pour la fréquence "ClaimInd" :

$$E[I_i|X_i = x_i] = (predict.GAM.claimInd)_i = E[y_i|x_i] = \frac{e^{x_i^{DISCRETE}\beta + s(x_i^{CONTINUE})}}{1+e^{x_i^{DISCRETE}\beta + s(x_i^{CONTINUE})}}.$$

• En calibrant séparément un modèle pour la sévérité "ClaimAmount" :

$$E[B_i|X_i = x_i] = (predict.GAM.claimAmount)_i = E[y_i|x_i] = \exp(x_i^{DISCRETE}\beta + s(x_i^{CONTINUE})).$$

Le calcul de la prime pure est donc donnée par :

$$\pi_i = E[X_i] = E[I_i|X_i = x_i].E[B_i|X_i = x_i] = (predict.ALM.claimInd)_i.(predict.GAM.claimAmount)_i.$$

Puisque les "*ClaimInd*" ont été prédit via une loi de Bernouilli :

$$E[I_i] = P(I_i = 1) = (predict.GAM.claimInd)_i.$$

Ainsi, nous avons pu établir un montant de prime pure aux assurés de la base de données test (freMLP34.test).

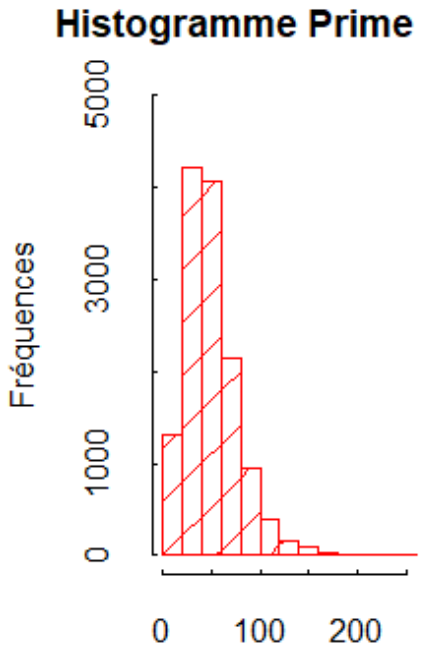

FIGURE 3.17: Histogramme de la prime pure calculée par les GAMs.

Dans le cas de notre base test, la prime pure est répartie entre **4.02** et **395.98**, avec un premier quartile de **30.45**, une moyenne de **49.88** et un troisième quartile de **62.93**.

Tout comme pour la prime pure des GLMs, nous allons observer l'évolution de la prime pure en fonction des caractéristiques des assurés. Nous allons donc de nouveau réaliser des boxplots.

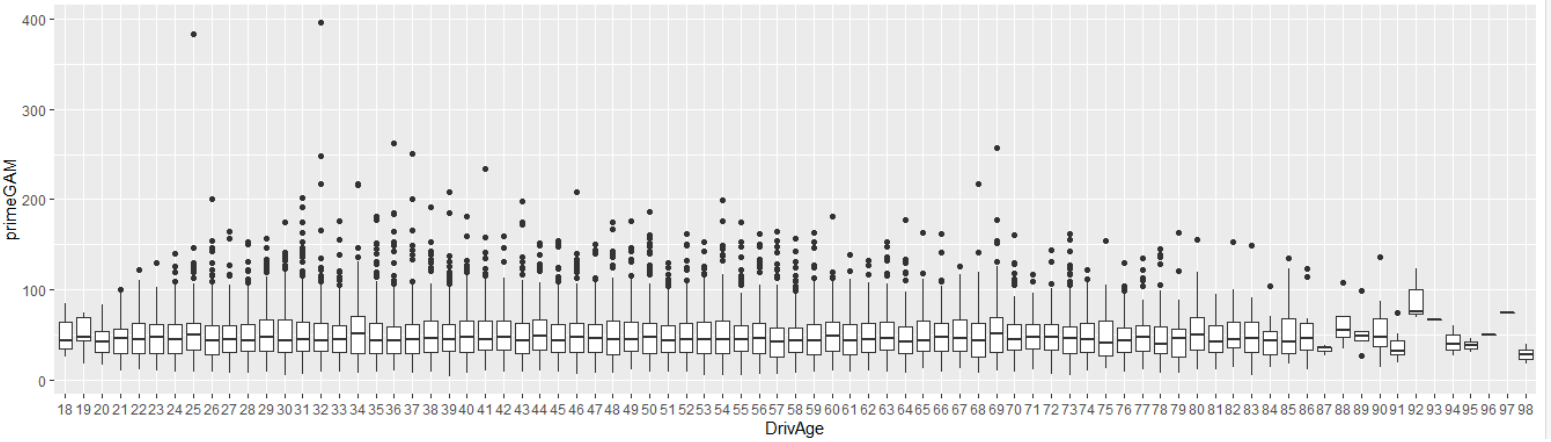

FIGURE 3.18: Boxplot de la prime pure en fonction de l'âge de l'assuré.

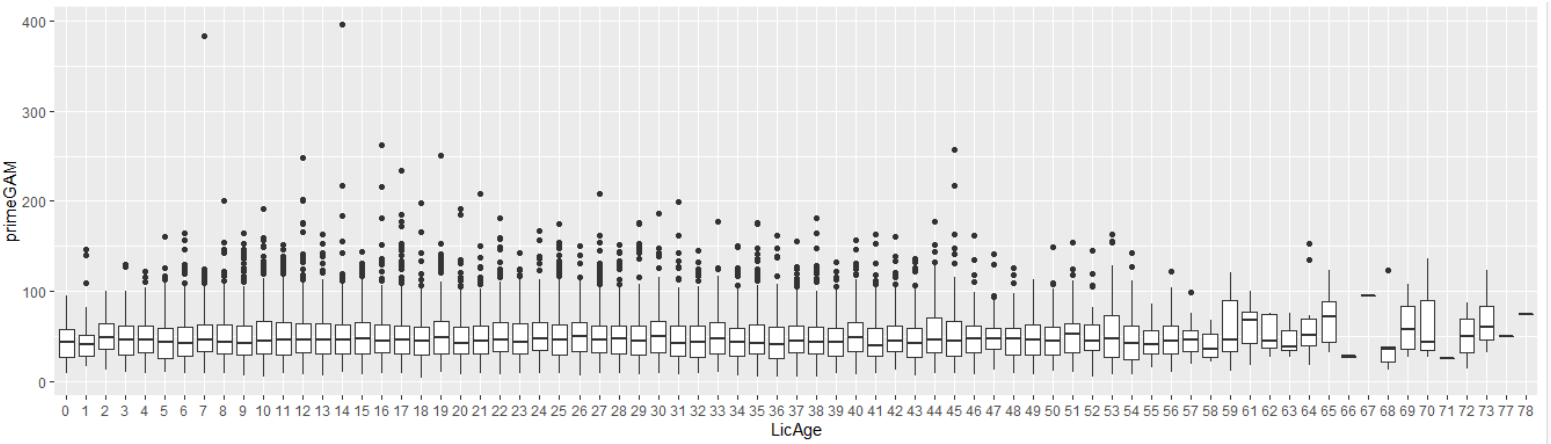

FIGURE 3.19: Boxplot de la prime pure en fonction du nombre d'années de permis de l'assuré.

Les figures 3.14 et 3.15 montrent qu'il y a moins de disparités dans les primes pures, lorsque les assurés sont très jeunes avec peu d'années de permis de conduire. Au contraire, il y en a plus lorsque les assurés sont plus âgés.

Il est logique de ne pas avoir beaucoup d'assurés avec plus de 70 années de permis.

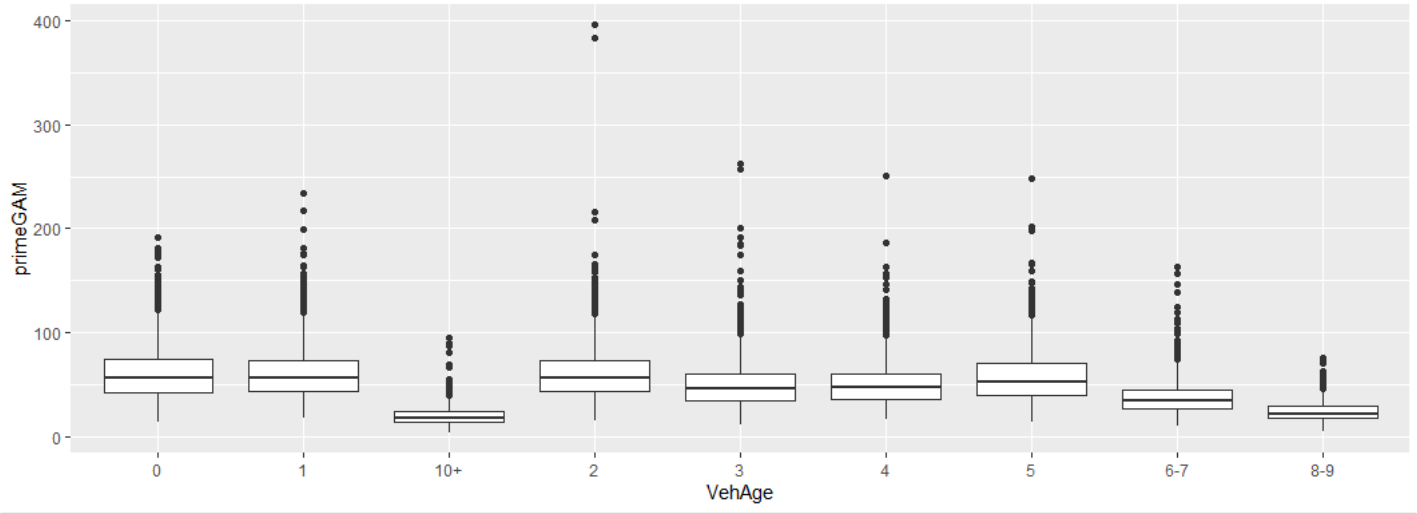

FIGURE 3.20: Boxplot de la prime pure en fonction de l'âge du véhicule de l'assuré.

Le boxplot de la prime pure en fonction de l'âge du véhicule de l'assuré montre des écarts de primes visibles en fonction du nombre d'années du véhicule.

Plus le véhicule est ancien, plus la prime pure a tendance à diminuer.

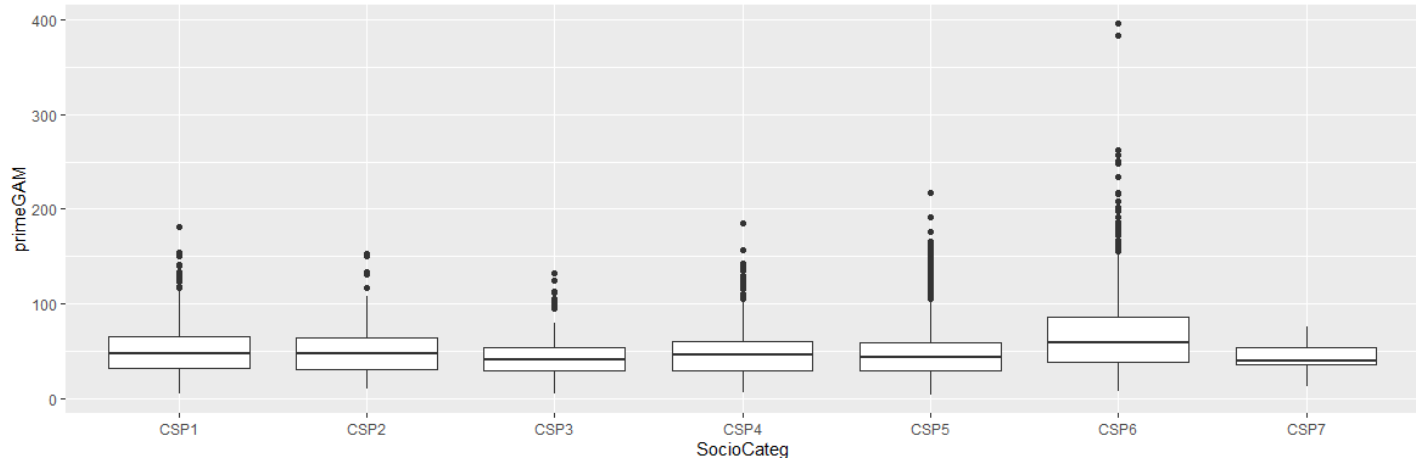

FIGURE 3.21: Boxplot de la prime pure en fonction de la catégorie socio-professionnel de l'assuré.

On observe peu de variations de la prime pure en fonction de la CSP. La CSP7 pourrait avoir en moyenne une prime pure plus élevée, mais cela n'est pas observable ici car elle est trop peu représentée dans notre base de données test. On peut remarquer la prime pure a d'assez grands écarts dans la CSP 6 correspondant aux assurés ouvriers.

Cela est peut être en partie dû aux deux points extrêmes que nous voyons en haut du graphique.

Chapitre 4

Choix de la prime

4.1 Objectif de solvabilité

4.1.1 Mutualisation et Segmentation

La mutualisation et la segmentation sont deux éléments fondamentaux du problème assurantiel.

Le premier, la mutualisation est l'essence même de l'assurance. Elle correspond au fait de considérer un très grand nombre de risques (ici de potentiels sinistres), afin de réduire le risque moyen.
Basé sur la loi des grands nombres, la mutualisation considère que les individus sont des variables aléatoires indépendantes et identiquement distribuées. Elle doit donc s'appliquer sur un ensemble de risques homogènes, ce qui requiert au prélable une segmentation.

Le deuxième principe, la segmentation, permet d'obtenir des risques homogènes. Ce point est essentiel car en général les risques ne sont pas homogènes. Il est donc nécessaire de segmenter les groupes d'assurés selon leurs profils et leurs risques.
La segmentation permet d'attribuer une prime adaptée au l'assuré et du risque qui peut être couvert par l'assurance.

4.1.2 Prime avec chargement

La problématique majeure de l'assureur est de savoir si il sera solvable. C'est-à-dire, en cas de sinistres, si il pourra indemniser tous ses assurés. Au cas contraire, l'assurance fera faillite.
Ainsi, si il fait payer à ses assurés une prime pure $E[X_i] = E[I_i].E[B_i]$, l'assureur sera garantit de ne pas perdre d'argent en moyenne.

Cependant, le montant des sinistres n'étant jamais égal à sa moyenne car aléatoire, il serait très risqué de faire payer une telle prime aux assurés.
Il sera alors essentiel de calculer une nouvelle prime de la forme $\Pi(X_i) = (1+\eta)E[X_i]$ où le coefficient η (strictement positif) correspond au chargement.

On peut introduire une nouvelle fonction, celle de la probabilité de ruine :

$$\psi_n = P(X_1 + X_2 + ... + X_n > \Pi_1 + \Pi_2 + ... + \Pi_n).$$

où $X_1 + X_2 + ... + X_n$ réprésente la somme des sinistres des assurés 1 à n et $\Pi_1 + \Pi_2 + ... + \Pi_n$, la somme des primes payées par ces mêmes assurés. Cette quantité correspond à la probabilité pour l'assureur de ne pas être en mesure d'indemniser ses assurés.

En supposant les X_i indépendants et identiquement distribués, on observe aisément ces propriétés sur la prime $\Pi(X_i) = (1+\eta)E[X_i]$ via la loi des grands nombres :

- Si $\eta < 0$(prime sans chargement), alors $\Pi(X_i) < E[X_i]$ et $\psi_n \underset{n\to+\infty}{\longrightarrow} 1$.
- Si $\eta > 0$ (prime avec chargement), alors $\Pi(X_i) > E[X_i]$ et $\psi_n \underset{n\to+\infty}{\longrightarrow} 0$.

La prime avec chargement nous permet de faire tendre la probabilité de ruine vers 0 lorsque le nombre d'assurés tend vers l'infini, c'est le principe de mutualisation. Il sera donc essentiel de déterminer un chargement sur notre prime.

4.2 Les primes et leurs propriétés

4.2.1 Prime d'assurance (tarification)

Il existe 3 niveaux de primes (étapes de tarification) :

- **Prime pure :** $\Pi(X) = E[X]$ l'espérance de la simistralité. Cette prime permet de compenser la simistralité en moyenne et pas de se prémunir contre les fluctrations.
- **Prime technique :** $\Pi_{technique}(X) = E[X]+CT$ avec CT représente le chargement technique.En général, CT est proportionnel à la prime pure ($CT = \eta.E[X]$), on écrit :
  $\Pi_{technique}(X) = (1+\eta)E[X]$.
- **Prime commerciale :** $\Pi_{commerciale}(X) = E[X] + CC$ avec CC représente le chargement commercial qui est proportionnel à la prime commerciale ($CC = \gamma.\Pi_{commerciale}(X)$), alors la prime commerciale s'écrit : $\Pi_{commerciale}(X) = \frac{E[X]}{1-\gamma}$ avec γ est le taux de chargement commercial. CC n'est pas lié à l'activité de la simistralité.

4.2.2 Les propriétés désirées

Dans cette partie est énoncée les différentes propriétés désirables pour une prime. Pour une variable aléatoire X, représentant le sinistre, d'espérance finie, on distingue les propriétés désirables suivantes pour la prime $\Pi(X)$:

- Marge de sécurité, $\Pi(X) > E[X]$.
- Exclusion de marge injustifiée, si $X = a$ p.s. alors $\Pi(X) = a$ p.s.
- Additivité,si X_1 et X_2 sont des risques indépendants alors $\Pi(X_1 + X_2) = \Pi(X_1) + \Pi(X_2)$.
- Sous-Additivité, si X_1 et X_2 sont des risques quelconques alors $\Pi(X_1 + X_2) \leq \Pi(X_1) + \Pi(X_2)$.
- Invariance d'échelle, pour $a > 0$ $\Pi(aX) = a\Pi(X)$.
- Invariance de translation, pour $a > 0$ $\Pi(a + X) = a + \Pi(X)$.
- Maximum, pour $l > 0$ et $X \leq l$ p.s. alors $\Pi(X) \leq l$.

Dans notre cas, nous avons commencé par déterminer la prime pure. Puis, comme il sera expliqué dans la partie suivante, nous avons calibré une prime avec chargement (principe de la valeur espérée) via une analyse par simulation.
Une telle prime est de la forme $\Pi(X_i) = (1+\eta)E[X_i]$ Le principe de la valeur espérée respecte plusieurs des propriétés enoncées ci-dessus (Marge de sécurité, Additivité, Sous Additivité et Invariance d'échelle).

4.3 Simulation d'une prime avec chargement

La prime avec chargement est de la forme $\Pi(X_i) = (1+\eta)E[X_i]$. Nous avons donc utilisé une méthode par simulation afin de déterminer $\eta > 0$.
Le but de cette simulation était de déterminer une prime avec chargement de sorte que dans 99 % des cas, la somme des primes soit supérieure à la somme des montants des sinistres.

Pour cela, nous avons tiré de façon aléatoire avec remise les sinistres (ClaimAmount) de notre base de données test.
Via une méthode dite de boostraping, nous avons pu obtenir 1000 simulations de la charge totale des sinistres (la somme des ClaimAmount).

Par la suite, nous avons établis des primes avec chargement $\Pi(X_i) = (1+\eta)E[X_i]$, avec 200 η différents, allant de 0.01 à 0.200 .
Ensuite, nous avons déterminer les quantiles à 99%, 95% et 90 % en utilisant les primes pures des modèles GLM puis avec celles des modèles GAM. Un quantile à x % réprésente le montant de chargement nécessaire afin que l'assurance ait une probabilité de solvabilité (donc de non ruine) de x%.

4.3.1 Simulation avec les GLM

Dans un premier temps, nous allons appliquer cette analyse par simulation aux modèles GLM.

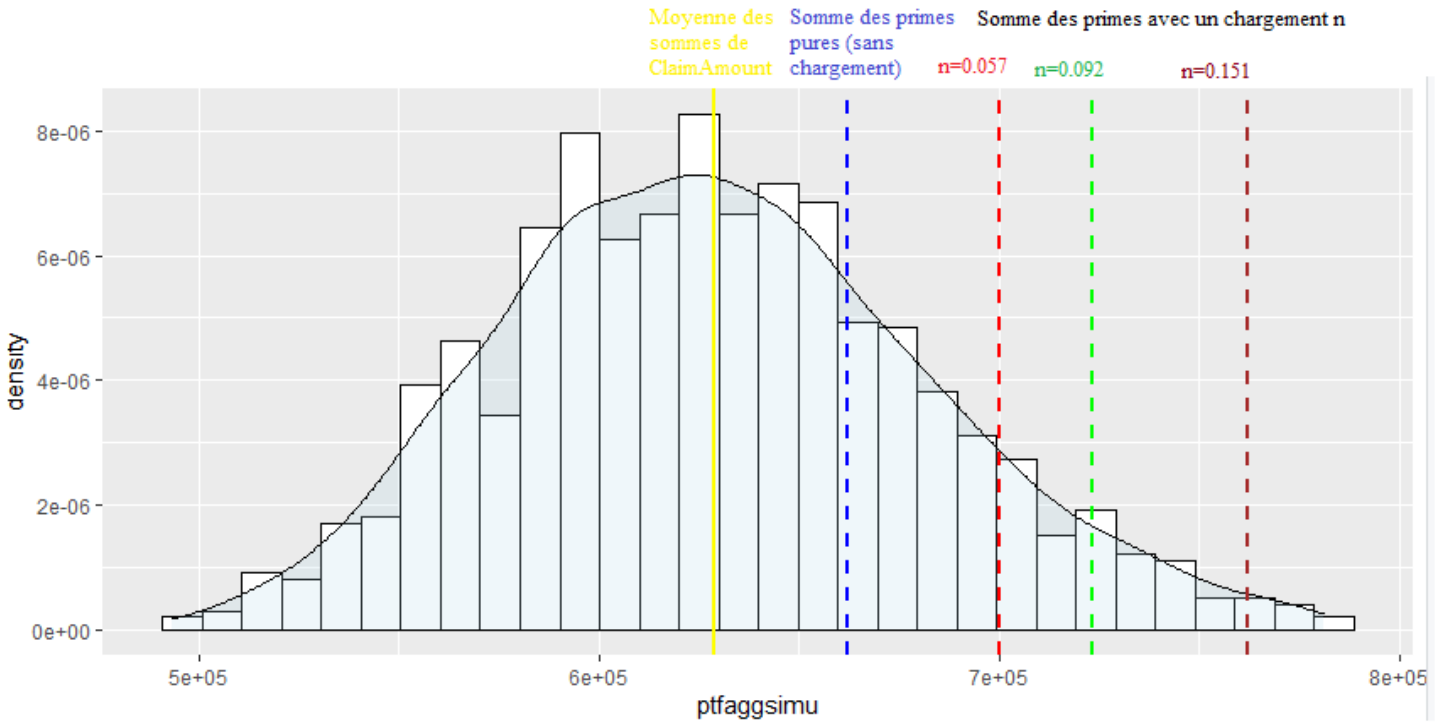

FIGURE 4.1: Histogramme des sommes de sinistres (ClaimAmount) prédits par GLM.

Dans ce graphique, on observe que l'utilisation de la prime pure n'est pas viable, car elle permet la solvabilité de l'assurance qu'avec une probabilité de 74.80%.
Elle causerait donc la ruine d'une assurance avec presque une chance sur quatre. On se tourne donc du côté des primes avec chargement. Nous avons tracé trois primes avec un chargement η. La première avec $\eta = 0.057$, permet à l'assurance d'être solvable dans 90 % des cas. La seconde $\eta = 0.092$ permet d'être solvable avec une probabilité de 95 %. Et enfin avec un chargement de $\eta = 0.151$, l'assurance a une probabilité de solvabilité de 99%, soit une probabilité de ruine égalle à 1%.

4.3.2 Simulation avec les GAM

Nous allons désormais réaliser cette analyse par simulation sur les GAM. Nous avons essayé plusieurs taux de chargements comme pour les GLM puis nous avons tracé la répartion des prédictions de sommes des sinistres.

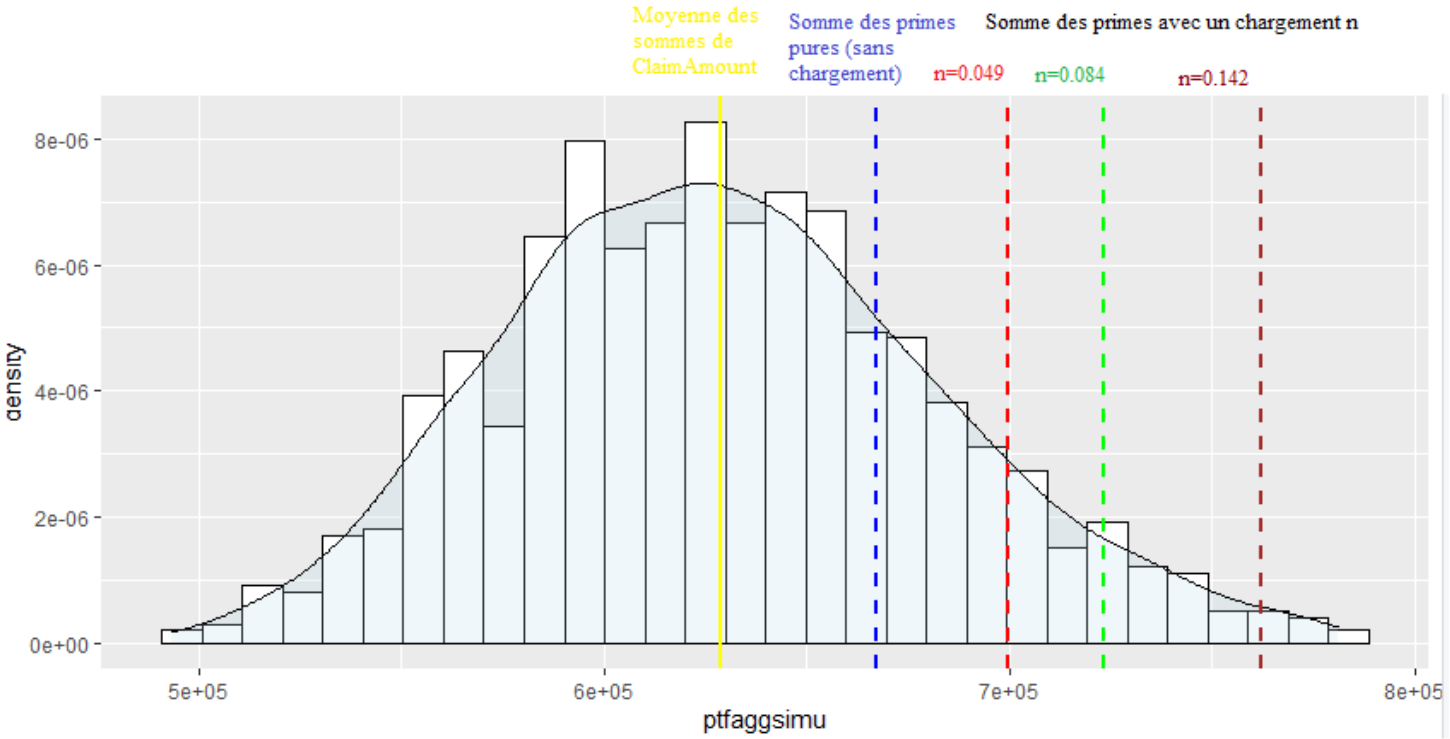

FIGURE 4.2: Histogramme des 10 000 Somme de sinistres (ClaimAmount) prédits par GAM.

Dans ce graphique, on observe que l'utilisation de la prime pure engendre une probabilité de non ruine de 76.90 %, Nous observerons donc plutôt les primes avec chargement. Nous avons tracé trois primes avec un chargement η. La première avec $\eta = 0.049$, permet à l'assurance d'être solvable dans 90 % des cas. La seconde $\eta = 0.084$ permet d'être solvable avec une probabilité de 95 %. Et enfin avec un chargement de $\eta = 0.142$, l'assurance a une probabilité de solvabilité de 99%, soit une probabilité de ruine égale à 1%.

4.3.3 Prime avec chargement finale

L'analyse par simulation nous donné un résultat assez similaire entre le modèle GLM et le modèle GAM, avec un chargement de 0.151 pour les GLM et 0.142 pour les GAM.

Mais on remarque tout de même que les modèles GLM préconisent un chargement plus important de 0.009, ce qui peut devenir conséquent lorsqu'on prend en compte l'ensemble des primes.

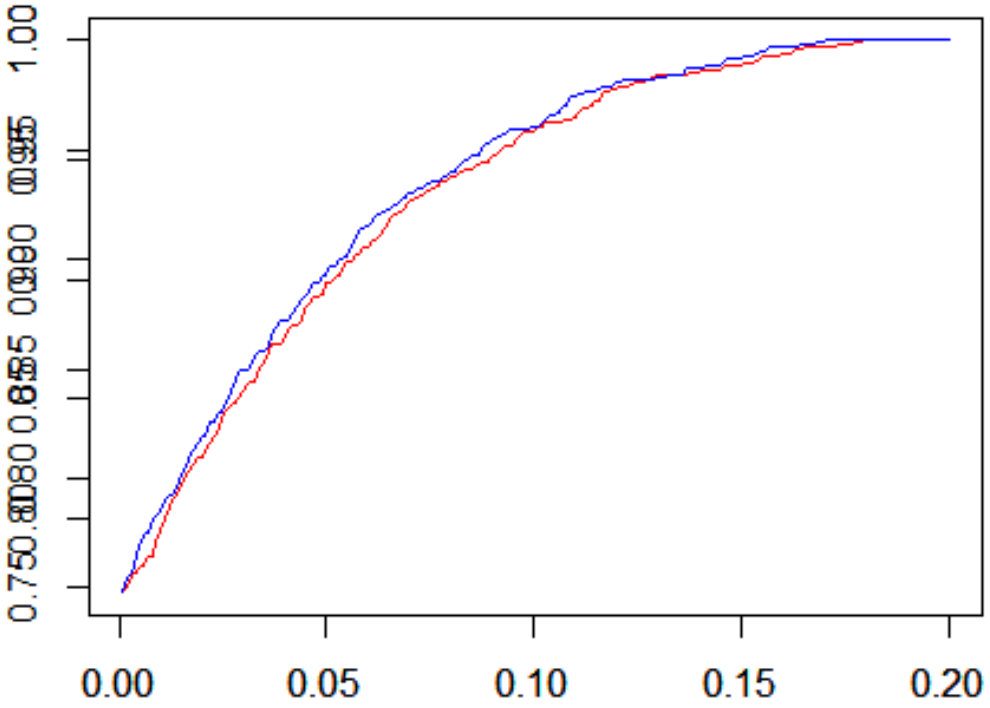

FIGURE 4.3: Evolution de la probabilité de solvabilité de l'assurance en fonction du taux de chargement (GAM en bleu, GLM en rouge).

Dans ce graphique, on observe que lorsque le taux de chargement augmente, la probabilité de solvabilité de l'assurance tend vers 1 (donc sa probabilité de ruine tend vers 0). On remarque aussi que la probabilité de solvabilité a tendance à croître un peu moins vite pour les GLM que pour les GAM.

Ce résultat est cohérent car nous avons dû choisir un taux de chargement un peu plus important pour les modèles GLM que les GAM.

Donc, la prime commerciale (avec chargement) est dans notre cas $\Pi(X_i) = (1+\eta)E[X_i] = 1.151E[X_i]$ pour les GLM et $\Pi(X_i) = (1+\eta)E[X_i] = 1.142E[X_i]$ pour les GAM.

Ainsi, avec une telle prime, la somme des primes est supérieure à la charge totale des sinistres avec une probabilité supérieure à 99%.

4.4 Comparaison des primes avec chargement GLM et GAM

4.4.1 Répartition des primes

L'analyse par simulation a permis de déterminer un taux de chargement pour les modèles GLM et les modèles GAM. Comme expliqué dans la partie précédente nous obtenons pour les modèles GLM $\Pi(X_i) = (1+\eta)E[X_i] = 1.151E[X_i]$ et pour les GAM $\Pi(X_i) = (1+\eta)E[X_i] = 1.142E[X_i]$.
Ces deux primes avec chargement sont assez proches au niveau du montant général. La prime GLM (respectivement GAM) admet un premier quartile de 35.104 (34.78 pour GAM) une moyenne de 56.983 (56.96 pour GAM) et un troisième quartile de 72.029 (71.86 pour GAM).

Pour mieux observer la répartition de ces deux primes avec chargement, nous avons tracé deux histogrammes de la prime avec chargement (un pour chaque méthode) sur un même graphique.

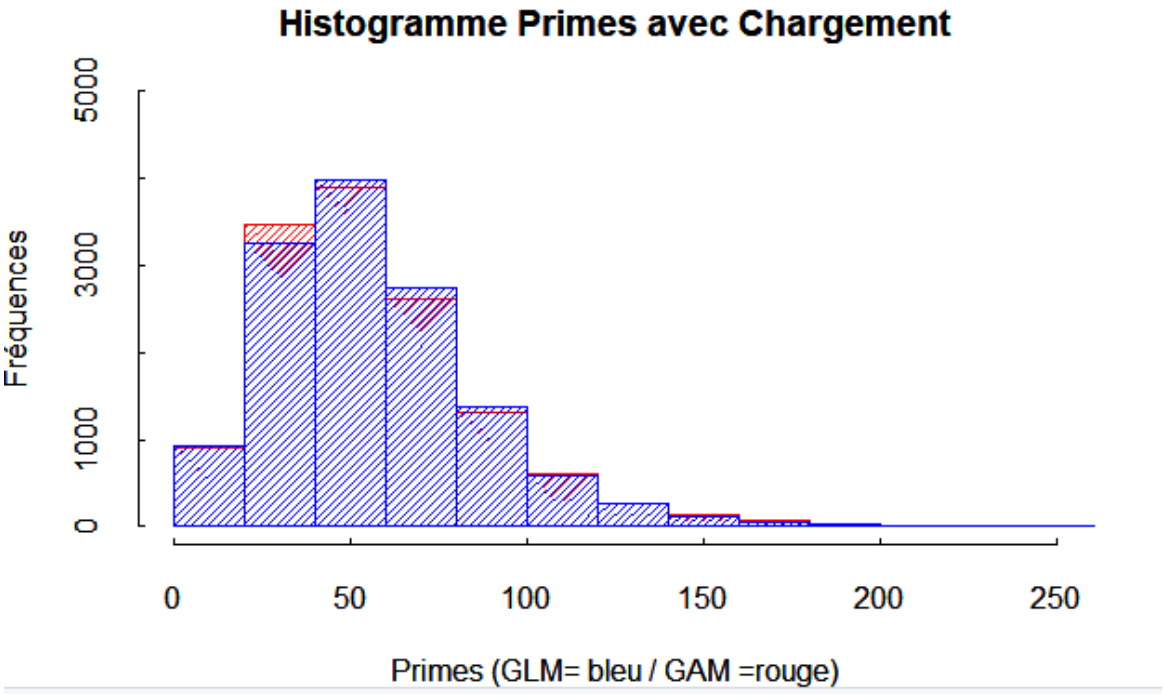

FIGURE 4.4: Histogrammes des primes avec chargement, selon la méthode GLM ou GAM.

On remarque que les petites primes (inférieures à 50) sont en plus grandes proportions pour le modèle GAM que celui GLM (6201 pour GLM contre 6370 pour GAM), ce qui est logique en vu des quartiles observés.

Cependant les primes du modèle GAM sont aussi en plus grandes proportions chez les grandes primes (supérieures à 150), avec 191 primes pour le GAM contre 159 pour le GLM.

Les primes avec chargement sont donc plus dispersées pour le modèle GAM que le modèle GLM.

Mais, on observe des valeurs extrêmes plus importantes chez le modèle GLM (jusqu'à 965.69) que chez le modèle GAM (jusqu'à 452.20).

4.4.2 Analyse des écarts des primes

Il peut être intéréssant d'obsever l'écart de prime avec chargement entre les deux modèles.
On introduit donc une nouvelle variable pour chaque individu :

$$Ecart_i = |PrimeGAM_i - PrimeGLM_i|.$$

L'écart entre les deux primes est compris entre 0.0001 et 833.83, pour une médiane de 4.06 et une moyenne de 6.44 . Cependant, le deuxième plus grand écart diminue de façon conséquente en tombant à 171.04 .
Ce plus gros écart de 833.83 correspond à une assurée atypique, car il s'agit d'une femme de 92 ans qui a assuré un véhicule pouvant atteindre 180-190 km/h. Le nombre très réduit d'assurés de ce type complique l'estimation du risque et explique cette différence très grande de primes entre les deux méthodes. On se demande alors si les grands écarts concernent des catégories d'assurés en particulier ou si ils apparaissent assez aléatoirement.
Pour répondre à cette question, nous allons utiliser des boxplots avec plusieurs spécificités d'assurés.

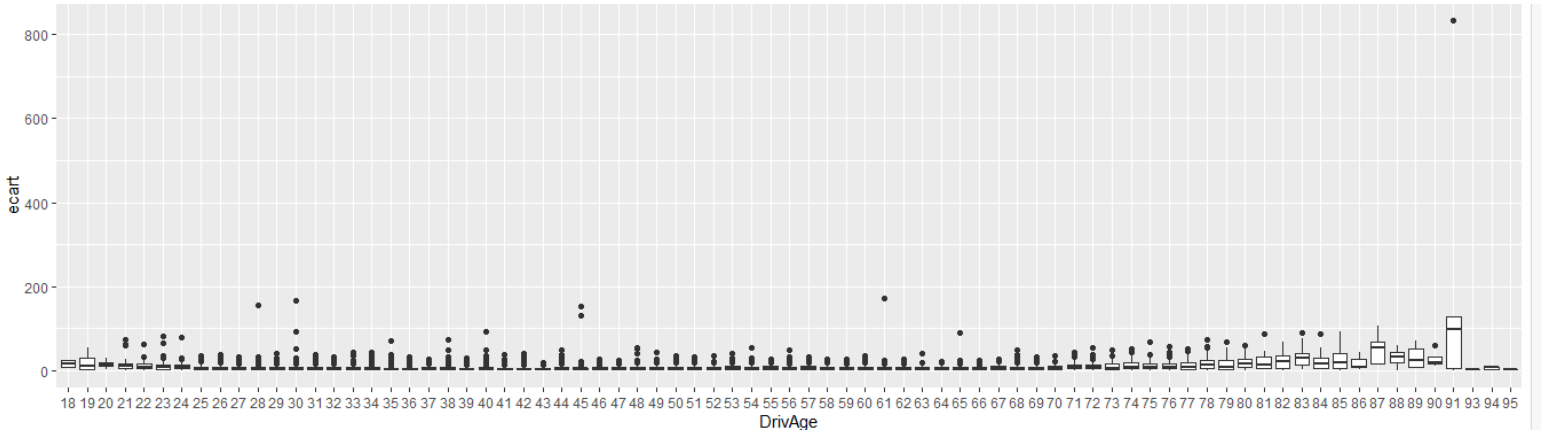

FIGURE 4.5: Ecart entre les deux méthodes de primes selon l'âge de l'assuré

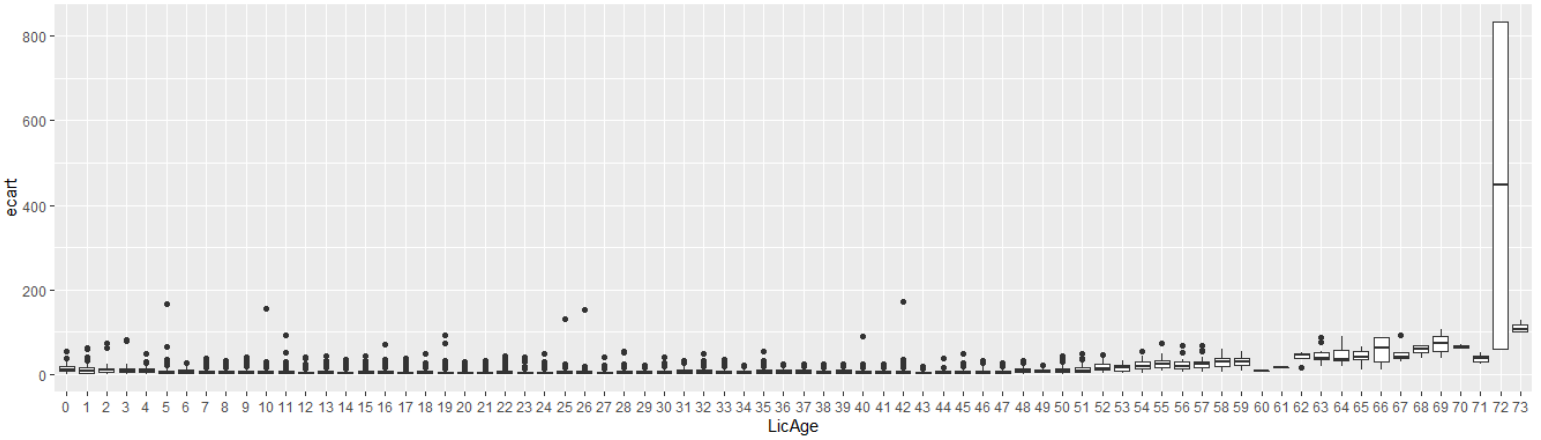

FIGURE 4.6: Ecart entre les deux méthodes de primes selon le nombre d'années de permis de l'assuré

Ces deux premiers graphiques nous indiquent que les plus gros écarts de prédictions sont présents pour des assurés très jeunes ou assez âgés.

Ce sont des catégories d'assurés sur lesquels nous avons peu de données. Notre base test compte 38 individus de 20 ans ou moins, et 66 individus de 85 ans ou plus. Il est donc cohérent d'observer des différences sur ces classes d'assurés.

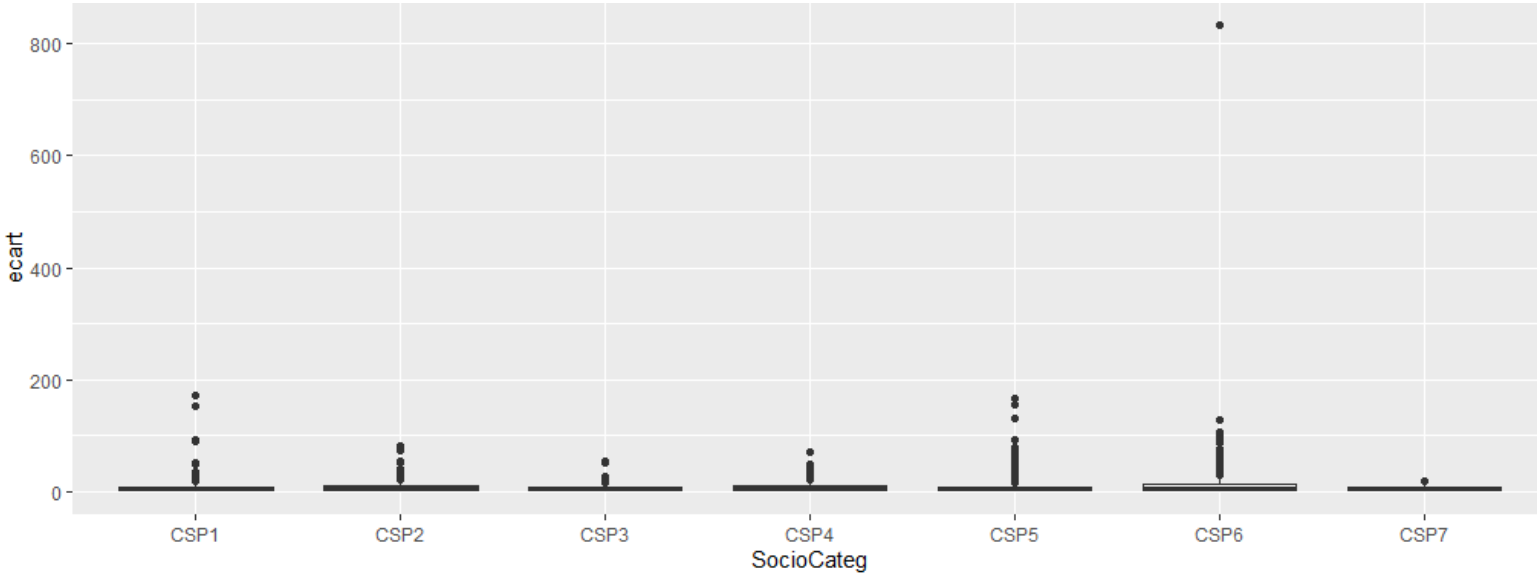

FIGURE 4.7: Ecart entre les deux méthodes de primes selon la CSP de l'assuré

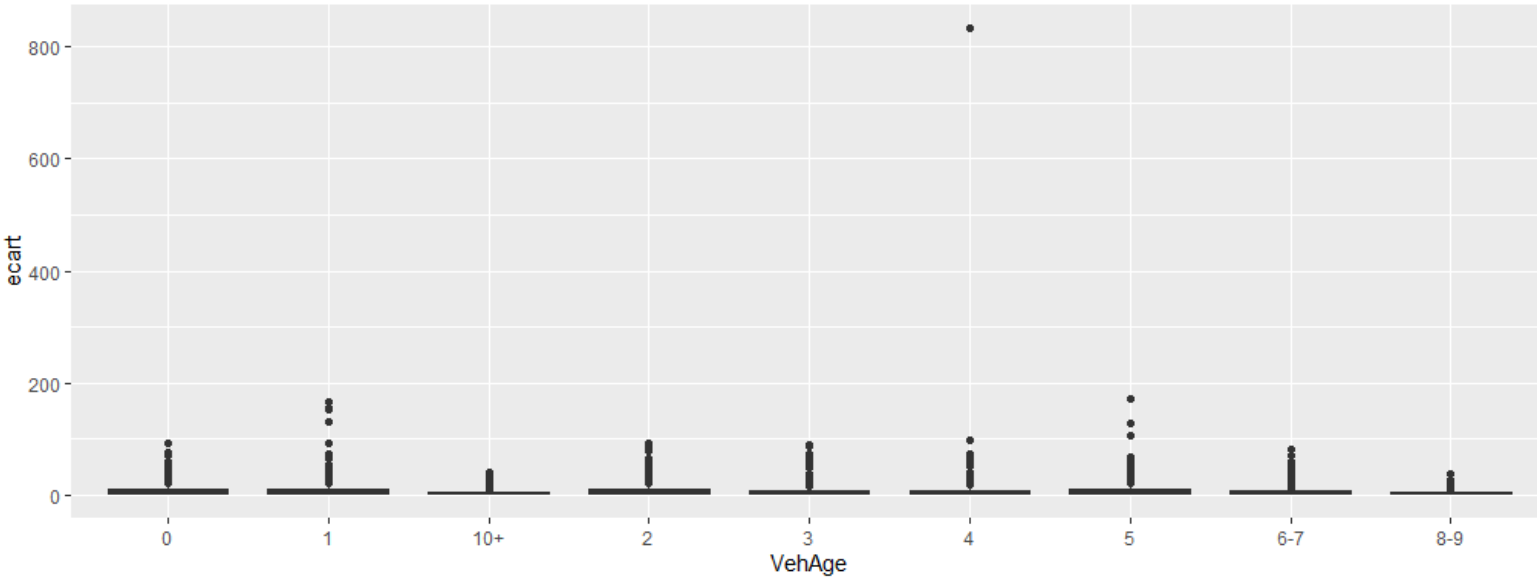

FIGURE 4.8: Ecart entre les deux méthodes de primes selon le nombre d'années du véhicule de l'assuré

Sur ces deux catégories, nous avons un nombre conséquent de données pour chaque catégorie. Nous observons pas d'impact conséquent de la CSP ou du nombre d'années du véhicule de l'individu sur l'écart de primes entre les deux méthodes.

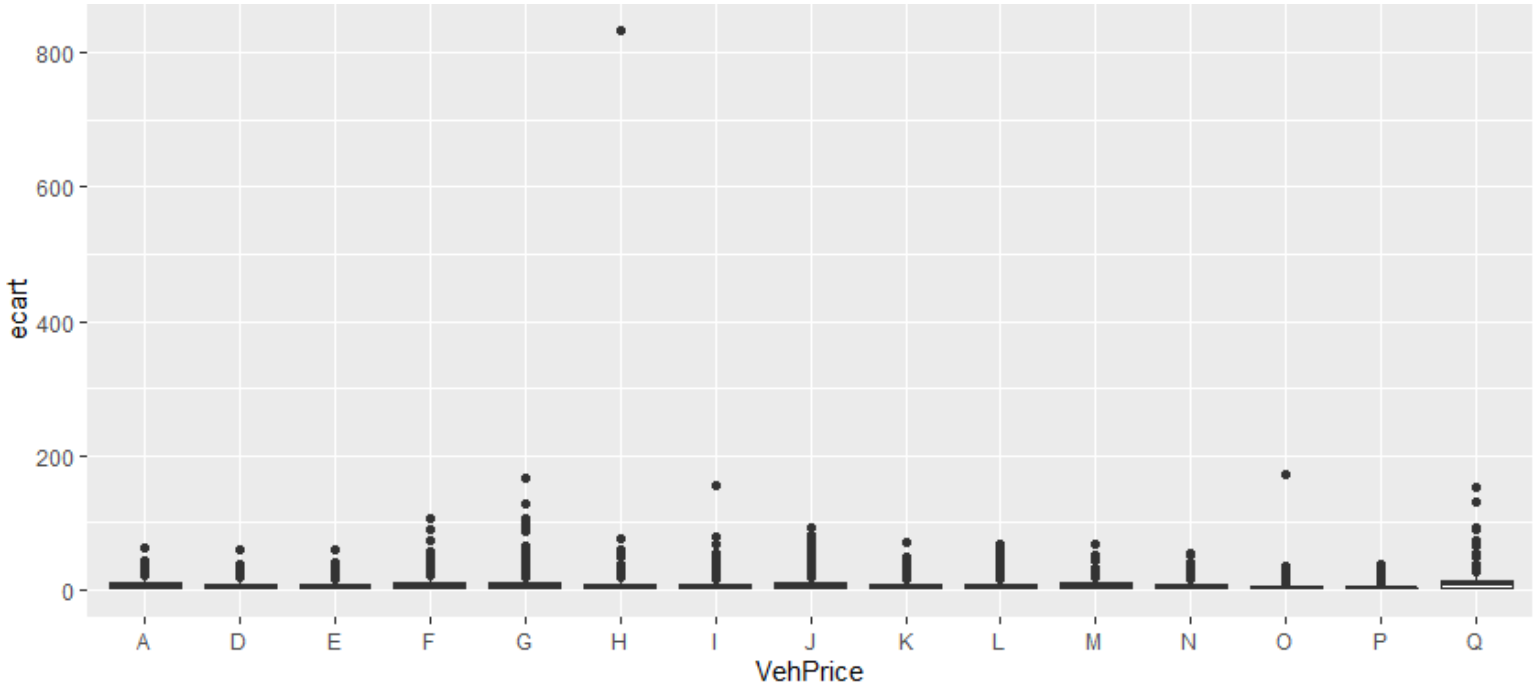

FIGURE 4.9: Ecart entre les deux méthodes de primes selon le prix du véhicule de l'assuré

On remarque que l'écart de primes a tendance à être plus élevé dans la classe de prix Q.
Cette classe de prix regroupait plusieurs classes de prix importants comprenant peu d'individus, cependant la classe Q contient désormais 996 individus après ce regroupement.

Ce qui pourrait expliquer cet écart est donc que cette classe de prix est à très grand risque, car les sinistres issus de cette classe Q engendre des coûts importants en raison de la valeur de ces véhicules.

Conclusion

Notre travail nous a permis d'étudier les GLM et les GAM et de les appliquer à un cas de tarification en assurance. Il y a une principale différence entre ces deux modèles, le modèle additif généralisé permet d'utiliser un lien non linéaire alors que les modèles linéaires généralisés ne font que des liens linéaires.

Afin de déterminer une prime pure pour chaque individu de notre base de données, nous avons dû prédire le montant des sinistres déclarés (ClaimAmount) Ensuite, nous avons calculé la probabilité pour un assuré de déclarer un sinistre ($PClaimInd = 1$). Ainsi, nous avons prédit ces variables avec un modèle GLM puis un GAM en faisant à chaque fois une sélection de variable à l'aide du critère AIC.

Nous avons remarqué que l'AIC du GAM est moins élevé que le AIC du GLM, on peut donc supposer que le modèle GAM prédit mieux ces deux variables.

Ensuite, nous avons calculé la prime pure de chaque individu dans une approche fréquence/sévérité. La prime pure correspond alors au produit du sinistre prédit pour l'assuré et la probabilité pour ce même individu de déclarer un sinistre. Cela correspond à une prime :
$\Pi(X_i) = E[ClaimAmount].E[ClaimInd] = (predict.ClaimAmount).(P(ClaimInd = 1))$
. Les primes pures des deux méthodes sont assez similaires, bien que celles du GAM soient en moyenne un peu plus élevées sur la base de données test.

L'enjeux principal de la tarification est de couvrir l'ensemble des coûts des sinistres déclarés avec les primes colléctées. Ainsi, afin de diminuer la probabilité de ruine de l'assurance, il est courant d'appliquer un taux de chargement.
Nous avons calculé ce taux de chargement via une analyse par simulation afin que dans 99% des cas la somme des primes pures collectées soient supérieures au coût total des sinistres déclarés.
Nous avons donc trouvé un chargement de **0.151** pour le modèle GLM et de **0.142** pour le GAM.

Dans notre cas, les deux méthodes nous donne donc des résultats assez simillaires. Cependant, malgré un risque de sur-apprentissage, le GAM aura tendance a faire de meilleurs prévisions car il a la possibilité de créer des liens non linéaires entre les variables.

Bibliographie

[**1**] Charpentier, A. et Denuit, M ; (2004A). Mathématiques de l'assurance non vie - Tome 1 Principes fondamentaux de théorie du risque .Economica

[**2**] Charpentier, A. et Denuit, M ; (2004b) Mathématiques de l'assurance non vie - Tome 2 Tarification et provisionnement .Economica.

[**3**] Charpentier, A. ; édition (2014). Computional Actuarial Science with R. Chapman et Hall-CRC.

[**4**] R CORE TEAM (2019). R : A Language and Environment for Statistical Computing. R Foundation for Statistical Computing. Vienna, Austria. URL : https ://www.R-project.org/.

[**5**] McCullagh, P. et Nelder, J. A. (1989). Generalized Linear Models. 2nd. Chapman et Hall.

[**6**] Hastie, T. J. et Tibshirani, R. J. (1990). Generalized Additive Models. Chapman et Hall.- (1995). Generalized Additive Models.to appear in Encyclopedia of Statistical Sciences.

[**7**] Nelder, J. A. et Wedderburn, R. W. M. (1972). Generalized Linear Models.Journal of the Royal Statistical Society.

[**8**] Turner, H. (2008). Introduction to Generalized linear models. Rapp. tech. Vienna University of Economics et Business.

[**9**] Wood, S. N. (2001). mgcv : GAMs and Generalized Ridge Regression for R.R News 1, p. 20-25.

[**10**] Simon N. Wood (2006). Generalized Additive Models An Introduction with R. 2nd. Chapman et Hall.

[**11**] Christophe DUTANG (2020). Cours Actuariat 1

[**12**] Katia MEZIANI (2020). Cours Modèles Linéaires Généralisés

Annexes

Annexe A

Code R des études statistiques

```
#-------------------------------------------------
#Statistiques descriptives Variables quantitatives
#-------------------------------------------------

# Certaines analyses statistiques de type summary ou autres
# ne sont pas toujours indiquees dans ce programme car ce
# sont des elements triviaux

Variables_quantitatives_freMPL3 <- freMPL3[,+c(1,2,10,11,13,20,21)]


#Representation des correlations : plus l'ellipse ressemble a un cercle et moins
#les variables sont correlees. Plus l'ellipse ressemble a une droite et plus les
#variables sont correlees.

corrplot(cor(Variables_quantitatives_freMPL3),method = "ellipse")

plot(Variables_quantitatives_freMPL3)

# => Une forte correclation entre DriveAge et LicAge.

Variables_quantitatives_freMPL4 <- freMPL4[,+c(1,2,10,11,13,20,21)]


#Representation des correlations : plus l'ellipse ressemble a un cercle et moins
#les variables sont correlees. Plus l'ellipse ressemble a une droite et plus les
#variables sont correlees.

corrplot(cor(Variables_quantitatives_freMPL4),method = "ellipse")

plot(Variables_quantitatives_freMPL4)

# => Une forte correclation entre DriveAge et LicAge.


#-------------------------------------------------
#                          ACP
#-------------------------------------------------

TabACP <- freMPL4[,+c(1,2,10,13,20,21)]
```

```
summary(TabACP)

res.pca <- PCA(TabACP, scale.unit = TRUE, ncp=2, graph = F)

res.pca

round(res.pca$eig,2)

res.pca <- PCA(TabACP, scale.unit = TRUE, ncp=2, graph = T)

#-----------------------------------------------------
#                          AFC
#-----------------------------------------------------

tableau <- housetasks[c(1,2,3,4,5,6),c(1,2)]
colnames(tableau) = c("ClaimInd=0","ClaimInd=1")
rownames(tableau) <- c("Male","Female","Alone","Other","HasKmLimit=0",
                       "HasKmLimit=1")

tableau[1,1] <- 18263
tableau[2,1] <- 11085
tableau[3,1] <- 7113
tableau[4,1] <- 22235
tableau[5,1] <- 26111
tableau[6,1] <- 3237

tableau[1,2] <- 762
tableau[2,2] <- 483
tableau[3,2] <- 311
tableau[4,2] <- 936
tableau[5,2] <- 1138
tableau[6,2] <- 109


View(tableau)

chisq <- chisq.test (tableau)
chisq

# 1. convertir les donnees en tant que table
dt <- as.table(as.matrix (tableau))
# 2. Graphique
balloonplot(t (dt), main = "tableau", xlab = "", ylab = "", label = FALSE,
            show.margins = FALSE)


res.ca <- CA (tableau, graph = FALSE)

res.ca <- CA (tableau, graph = FALSE)

eig.val <- get_eigenvalue (res.ca)
```

Annexe B

Code R des Modèles linéaires généralisés (GLMs)

```
#########################    GLM ClaimInd    #################

levels(freMPL34$SocioCateg) <- factor(c("CSP1","CSP1","CSP1","CSP1","CSP2",
                                       "CSP2","CSP2","CSP2","CSP2","CSP2",
                                       "CSP3","CSP3","CSP3","CSP4","CSP4",
                                       "CSP4","CSP4","CSP4","CSP4","CSP4",
                                       "CSP5","CSP5","CSP5","CSP5","CSP5",
                                       "CSP5","CSP6","CSP6","CSP6","CSP6",
                                       "CSP6","CSP6","CSP7","CSP7","CSP7",
                                       "CSP7","CSP7","CSP9"))
levels(freMPL34$VehPrice) <- factor(c("A","A","A","D","E","F","G","H","I","J",
                                     "K","L","M","N","O","P","Q","Q","Q","Q",
                                     "Q","Q","Q","Q","Q","Q","Q"))
levels(freMPL34$VehMaxSpeed) <- factor(c("1-140 km/h","1-140 km/h",
                                        "140-150 km/h","150-160 km/h",
                                        "160-170 km/h","170-180 km/h",
                                        "180-190 km/h","190-200 km/h",
                                        "200-220 km/h","220+ km/h "))

freMPL34 <- freMPL34[(freMPL34$VehEnergy != "eletric") &
                       (freMPL34$VehEnergy != "GPL"),]
freMPL34$VehEnergy <- droplevels(freMPL34$VehEnergy)
freMPL34 <- freMPL34[(freMPL34$VehEnergy != "eletric") &
                       (freMPL34$VehEnergy != "GPL"),]
freMPL34$VehEnergy <- droplevels(freMPL34$VehEnergy)
freMPL34$VehEngine <- droplevels(freMPL34$VehEngine)
freMPL34$LicAge <- floor(freMPL34$LicAge/12)
freMPL34$LicAge3 <- (freMPL34$LicAge-23)^2

n <- NROW(freMPL34) ; p <- round(0.8*n)
index.app <- sample(1:n, p)
freMPL34.app <- freMPL34[index.app, ]
freMPL34.test <- freMPL34[-index.app, ]


freMPL34.app.reg <- model.matrix(ClaimInd ~ LicAge+LicAge3 + VehAge +
                                   Gender + MariStat + SocioCateg + VehUsage +
```

```
                                    DrivAge + HasKmLimit + DeducType +
                                    BonusMalus + VehBody + VehPrice + VehEngine +
                                    VehEnergy + VehMaxSpeed + VehClass +
                                    RiskVar + Garage, data=freMPL34.app)

freMPL34.test.reg <- model.matrix(ClaimInd ~ LicAge+LicAge3 + VehAge +
                                    Gender + MariStat + SocioCateg + VehUsage +
                                    DrivAge + HasKmLimit + DeducType +
                                    BonusMalus + VehBody + VehPrice +
                                    VehEngine + VehEnergy + VehMaxSpeed +
                                    VehClass + RiskVar + Garage,
                                    data=freMPL34.test)

colnames(freMPL34.app.reg)[ colnames(freMPL34.app.reg)
=="VehAge10+"] <- "VehAge10"
colnames(freMPL34.app.reg)[ colnames(freMPL34.app.reg)
=="VehAge6-7"] <- "VehAge6"
colnames(freMPL34.app.reg)[ colnames(freMPL34.app.reg)
=="VehAge8-9"] <- "VehAge8"
colnames(freMPL34.app.reg)[ colnames(freMPL34.app.reg)
=="VehUsagePrivate+trip to office"] <- "VehUsagePrivate_trip_to_office"
colnames(freMPL34.app.reg)[ colnames(freMPL34.app.reg)
=="VehUsageProfessional run"] <- "VehUsageProfessional_run"
colnames(freMPL34.app.reg)[ colnames(freMPL34.app.reg)
=="DeducTypePartially refunded"] <- "DeducTypePartially_refunded"
colnames(freMPL34.app.reg)[ colnames(freMPL34.app.reg)
=="VehBodyother microvan"] <- "VehBodyother_microvan"
colnames(freMPL34.app.reg)[ colnames(freMPL34.app.reg)
=="VehBodysport utility vehicle"] <- "VehBodysport_utility_vehicle"
colnames(freMPL34.app.reg)[ colnames(freMPL34.app.reg)
=="VehBodystation wagon"] <- "VehBodystation_wagon"
colnames(freMPL34.app.reg)[ colnames(freMPL34.app.reg)
=="VehEnginedirect injection overpowered"] <- "VehEnginedirect_injection_overpowered"
colnames(freMPL34.app.reg)[ colnames(freMPL34.app.reg)
=="VehEngineinjection overpowered"] <- "VehEngineinjection_overpowered"
colnames(freMPL34.app.reg)[ colnames(freMPL34.app.reg)
=="VehMaxSpeed140-150 km/h"] <- "VehMaxSpeed140_150_km_h"
colnames(freMPL34.app.reg)[ colnames(freMPL34.app.reg)
=="VehMaxSpeed150-160 km/h"] <- "VehMaxSpeed150_160_km_h"
colnames(freMPL34.app.reg)[ colnames(freMPL34.app.reg)
=="VehMaxSpeed160-170 km/h"] <- "VehMaxSpeed160_170_km_h"
colnames(freMPL34.app.reg)[ colnames(freMPL34.app.reg)
=="VehMaxSpeed170-180 km/h"] <- "VehMaxSpeed170_180_km_h"
colnames(freMPL34.app.reg)[ colnames(freMPL34.app.reg)
=="VehMaxSpeed180-190 km/h"] <- "VehMaxSpeed180_190_km_h"
colnames(freMPL34.app.reg)[ colnames(freMPL34.app.reg)
=="VehMaxSpeed190-200 km/h"] <- "VehMaxSpeed190_200_km_h"
colnames(freMPL34.app.reg)[ colnames(freMPL34.app.reg)
=="VehMaxSpeed200-220 km/h"] <- "VehMaxSpeed200_220_km_h"
colnames(freMPL34.app.reg)[ colnames(freMPL34.app.reg)
=="VehMaxSpeed220+ km/h "] <- "VehMaxSpeed220_km_h"
colnames(freMPL34.app.reg)[ colnames(freMPL34.app.reg)
=="GaragePrivate garage"] <- "GaragePrivate_garage"


colnames(freMPL34.test.reg)[ colnames(freMPL34.test.reg)
```

```
=="VehAge10+"] <- "VehAge10"
colnames(freMPL34.test.reg)[ colnames(freMPL34.test.reg)
=="VehAge6-7"] <- "VehAge6"
colnames(freMPL34.test.reg)[ colnames(freMPL34.test.reg)
=="VehAge8-9"] <- "VehAge8"
colnames(freMPL34.test.reg)[ colnames(freMPL34.test.reg)
=="VehUsagePrivate+trip to office"] <- "VehUsagePrivate_trip_to_office"
colnames(freMPL34.test.reg)[ colnames(freMPL34.test.reg)
=="VehUsageProfessional run"] <- "VehUsageProfessional_run"
colnames(freMPL34.test.reg)[ colnames(freMPL34.test.reg)
=="DeducTypePartially refunded"] <- "DeducTypePartially_refunded"
colnames(freMPL34.test.reg)[ colnames(freMPL34.test.reg)
=="VehBodyother microvan"] <- "VehBodyother_microvan"
colnames(freMPL34.test.reg)[ colnames(freMPL34.test.reg)
=="VehBodysport utility vehicle"] <- "VehBodysport_utility_vehicle"
colnames(freMPL34.test.reg)[ colnames(freMPL34.test.reg)
=="VehBodystation wagon"] <- "VehBodystation_wagon"
colnames(freMPL34.test.reg)[ colnames(freMPL34.test.reg)
=="VehEnginedirect injection overpowered"] <- "VehEnginedirect_injection_overpowered"
colnames(freMPL34.test.reg)[ colnames(freMPL34.test.reg)
=="VehEngineinjection overpowered"] <- "VehEngineinjection_overpowered"
colnames(freMPL34.test.reg)[ colnames(freMPL34.test.reg)
=="VehMaxSpeed140-150 km/h"] <- "VehMaxSpeed140_150_km_h"
colnames(freMPL34.test.reg)[ colnames(freMPL34.test.reg)
=="VehMaxSpeed150-160 km/h"] <- "VehMaxSpeed150_160_km_h"
colnames(freMPL34.test.reg)[ colnames(freMPL34.test.reg)
=="VehMaxSpeed160-170 km/h"] <- "VehMaxSpeed160_170_km_h"
colnames(freMPL34.test.reg)[ colnames(freMPL34.test.reg)
=="VehMaxSpeed170-180 km/h"] <- "VehMaxSpeed170_180_km_h"
colnames(freMPL34.test.reg)[ colnames(freMPL34.test.reg)
=="VehMaxSpeed180-190 km/h"] <- "VehMaxSpeed180_190_km_h"
colnames(freMPL34.test.reg)[ colnames(freMPL34.test.reg)
=="VehMaxSpeed190-200 km/h"] <- "VehMaxSpeed190_200_km_h"
colnames(freMPL34.test.reg)[ colnames(freMPL34.test.reg)
=="VehMaxSpeed200-220 km/h"] <- "VehMaxSpeed200_220_km_h"
colnames(freMPL34.test.reg)[ colnames(freMPL34.test.reg)
=="VehMaxSpeed220+ km/h "] <- "VehMaxSpeed220_km_h"
colnames(freMPL34.test.reg)[ colnames(freMPL34.test.reg)
=="GaragePrivate garage"] <- "GaragePrivate_garage"


# GLM ClaimInd avec le modele Binomiale

GLM.claimInd <- glm(ClaimInd ~ LicAge3 + VehAge1 + VehAge10 + VehAge2 + VehAge3 +
                      VehAge4 + VehAge5 + VehAge6 + VehAge8 + GenderMale +
                      MariStatOther + SocioCategCSP2 + SocioCategCSP3 +
                      SocioCategCSP4 + SocioCategCSP5 + SocioCategCSP6 +
                      SocioCategCSP7 + SocioCategCSP9 +
                      VehUsagePrivate_trip_to_office + VehUsageProfessional +
                      VehUsageProfessional_run + DrivAge + HasKmLimit +
                      DeducTypeNormal + DeducTypePartially_refunded +
                      DeducTypeProportional + DeducTypeRefunded + BonusMalus +
                      VehBodycabriolet + VehBodycoupe + VehBodymicrovan +
                      VehBodyother_microvan + VehBodysedan +
                      VehBodysport_utility_vehicle + VehBodystation_wagon +
                      VehBodyvan + VehPriceD + VehPriceE + VehPriceF + VehPriceG +
```

```
                        VehPriceH + VehPriceI + VehPriceJ + VehPriceK + VehPriceL +
                        VehPriceM + VehPriceN + VehPriceO + VehPriceP + VehPriceQ +
                        VehEnginedirect_injection_overpowered + VehEngineinjection +
                        VehEngineinjection_overpowered + VehEnergyregular +
                        VehMaxSpeed140_150_km_h + VehMaxSpeed150_160_km_h +
                        VehMaxSpeed160_170_km_h + VehMaxSpeed170_180_km_h +
                        VehMaxSpeed180_190_km_h + VehMaxSpeed190_200_km_h +
                        VehMaxSpeed200_220_km_h + VehMaxSpeed220_km_h  + VehClassA +
                        VehClassB + VehClassH + VehClassM1 + VehClassM2 + RiskVar +
                        GarageNone + GaragePrivate_garage,
                      family=binomial(link = 'logit'),
                      data=cbind.data.frame(ClaimInd=freMPL34.app$ClaimInd,
                      freMPL34.app.reg))

#Selection par AIC (Forward, Backward, Stepwise)

GLM.claimInd <- stepAIC(GLM.claimInd, trace=TRUE, direction=c("both"))
AIC(GLM.claimInd)

GLM.claimInd <- stepAIC(GLM.claimInd, trace=TRUE, direction=c("backward"))
AIC(GLM.claimInd)

GLM.claimInd <- stepAIC(GLM.claimInd, trace=TRUE, direction=c("forward"))
AIC(GLM.claimInd)

#le GLM ClaimInd selectionne


GLM.ClaimInd <- glm(ClaimInd ~ LicAge3 + BonusMalus + VehAge10 + VehAge6 +
                        VehAge8 + SocioCategCSP6 + VehUsageProfessional +
                        VehUsageProfessional_run + DeducTypePartially_refunded +
                        DeducTypeRefunded + VehBodystation_wagon + VehPriceO +
                        VehPriceP + VehPriceQ + VehMaxSpeed140_150_km_h +
                        VehMaxSpeed150_160_km_h + VehMaxSpeed160_170_km_h +
                        VehMaxSpeed170_180_km_h + VehMaxSpeed180_190_km_h +
                        VehMaxSpeed190_200_km_h + VehMaxSpeed200_220_km_h +
                        VehMaxSpeed220_km_h + VehClassA + GaragePrivate_garage,
                      family = binomial(link = 'logit'),
                      data = cbind.data.frame(ClaimInd = freMPL34.app$ClaimInd,
                      freMPL34.app.reg))


termplot(GLM.claimInd, partial.resid = TRUE, se = TRUE)

plot(GLM.claimInd)

summary(GLM.claimInd)

#Proba d'avoir ClaimInd=1

p1 <- length(freMPL34[freMPL34$ClaimInd==1,]$ClaimInd)
p2 <- length(freMPL34$ClaimInd)

proba_ind_1 <- p1 / p2

#Prediction du modele
```

```
predict.GLM.claimInd.test <- predict(GLM.claimInd, newdata
                          = as.data.frame(freMPL34.test.reg), type = "response")

predict.GLM.claimInd.test.bis=as.factor(ifelse(predict.GLM.claimInd.test
                                              > proba_ind_1,1,0))

predict.GLM.claimInd.app <- predict(GLM.claimInd, newdata
                           = as.data.frame(freMPL34.app.reg), type = "response")

predict.GLM.claimInd.app.bis=as.factor(ifelse(predict.GLM.claimInd.app
                                             > proba_ind_1,1,0))

#Matrice de confusion sur l'echantillon test

confusionMatrix(data = predict.GLM.claimInd.test.bis, reference
                =as.factor(freMPL34.test$ClaimInd))

#Matrice de confusion sur l'echantillon d'apprentissage

confusionMatrix(data = predict.GLM.claimInd.app.bis, reference
                =as.factor(freMPL34.app$ClaimInd))

#Erreur sur l'echantillon d'apprentissage

mean((predict.GLM.claimInd.app - freMPL34.app[,"ClaimInd"])**2)

#Erreur sur l'echantillon test

mean((predict.GLM.claimInd.test - freMPL34.test[,"ClaimInd"])**2)

# Conclusion:

# => L'erreur est petit, Accuracy acceptable donc on peut accepter le modele


########################   GLM ClaimAmount    #################

freMPL34.app.sinistre <- subset(freMPL34.app, ClaimAmount > 0)
freMPL34.test.sinistre <- subset(freMPL34.test, ClaimAmount > 0)

freMPL34.app.sinistre.reg <- model.matrix(ClaimAmount ~ LicAge + VehAge +
                                          Gender + MariStat + SocioCateg +
                                          VehUsage + DrivAge + HasKmLimit +
                                          DeducType + BonusMalus + VehBody +
                                          VehPrice + VehEngine + VehEnergy +
                                          VehMaxSpeed + VehClass + RiskVar +
                                          Garage, data=freMPL34.app.sinistre)

freMPL34.test.sinistre.reg <- model.matrix(ClaimAmount ~ LicAge + VehAge +
                                           Gender + MariStat + SocioCateg +
                                           VehUsage + DrivAge + HasKmLimit +
                                           DeducType + BonusMalus + VehBody +
                                           VehPrice + VehEngine + VehEnergy +
                                           VehMaxSpeed + VehClass + RiskVar +
                                           Garage, data=freMPL34.test.sinistre)
```

```
colnames(freMPL34.app.sinistre.reg)[ colnames(freMPL34.app.sinistre.reg)
=="VehAge10+"] <- "VehAge10"
colnames(freMPL34.app.sinistre.reg)[ colnames(freMPL34.app.sinistre.reg)
=="VehAge6-7"] <- "VehAge6"
colnames(freMPL34.app.sinistre.reg)[ colnames(freMPL34.app.sinistre.reg)
=="VehAge8-9"] <- "VehAge8"
colnames(freMPL34.app.sinistre.reg)[ colnames(freMPL34.app.sinistre.reg)
=="VehUsagePrivate+trip to office"] <- "VehUsagePrivate_trip_to_office"
colnames(freMPL34.app.sinistre.reg)[ colnames(freMPL34.app.sinistre.reg)
=="VehUsageProfessional run"] <- "VehUsageProfessional_run"
colnames(freMPL34.app.sinistre.reg)[ colnames(freMPL34.app.sinistre.reg)
=="DeducTypePartially refunded"] <- "DeducTypePartially_refunded"
colnames(freMPL34.app.sinistre.reg)[ colnames(freMPL34.app.sinistre.reg)
=="VehBodyother microvan"] <- "VehBodyother_microvan"
colnames(freMPL34.app.sinistre.reg)[ colnames(freMPL34.app.sinistre.reg)
=="VehBodysport utility vehicle"] <- "VehBodysport_utility_vehicle"
colnames(freMPL34.app.sinistre.reg)[ colnames(freMPL34.app.sinistre.reg)
=="VehBodystation wagon"] <- "VehBodystation_wagon"
colnames(freMPL34.app.sinistre.reg)[ colnames(freMPL34.app.sinistre.reg)
=="VehEnginedirect injection overpowered"] <- "VehEnginedirect_injection_overpowered"
colnames(freMPL34.app.sinistre.reg)[ colnames(freMPL34.app.sinistre.reg)
=="VehEngineinjection overpowered"] <- "VehEngineinjection_overpowered"
colnames(freMPL34.app.sinistre.reg)[ colnames(freMPL34.app.sinistre.reg)
=="VehMaxSpeed140-150 km/h"] <- "VehMaxSpeed140_150_km_h"
colnames(freMPL34.app.sinistre.reg)[ colnames(freMPL34.app.sinistre.reg)
=="VehMaxSpeed150-160 km/h"] <- "VehMaxSpeed150_160_km_h"
colnames(freMPL34.app.sinistre.reg)[ colnames(freMPL34.app.sinistre.reg)
=="VehMaxSpeed160-170 km/h"] <- "VehMaxSpeed160_170_km_h"
colnames(freMPL34.app.sinistre.reg)[ colnames(freMPL34.app.sinistre.reg)
=="VehMaxSpeed170-180 km/h"] <- "VehMaxSpeed170_180_km_h"
colnames(freMPL34.app.sinistre.reg)[ colnames(freMPL34.app.sinistre.reg)
=="VehMaxSpeed180-190 km/h"] <- "VehMaxSpeed180_190_km_h"
colnames(freMPL34.app.sinistre.reg)[ colnames(freMPL34.app.sinistre.reg)
=="VehMaxSpeed190-200 km/h"] <- "VehMaxSpeed190_200_km_h"
colnames(freMPL34.app.sinistre.reg)[ colnames(freMPL34.app.sinistre.reg)
=="VehMaxSpeed200-220 km/h"] <- "VehMaxSpeed200_220_km_h"
colnames(freMPL34.app.sinistre.reg)[ colnames(freMPL34.app.sinistre.reg)
=="VehMaxSpeed220+ km/h "] <- "VehMaxSpeed220_km_h"
colnames(freMPL34.app.sinistre.reg)[ colnames(freMPL34.app.sinistre.reg)
=="GaragePrivate garage"] <- "GaragePrivate_garage"

colnames(freMPL34.test.sinistre.reg)[ colnames(freMPL34.test.sinistre.reg)
=="VehAge10+"] <- "VehAge10"
colnames(freMPL34.test.sinistre.reg)[ colnames(freMPL34.test.sinistre.reg)
=="VehAge6-7"] <- "VehAge6"
colnames(freMPL34.test.sinistre.reg)[ colnames(freMPL34.test.sinistre.reg)
=="VehAge8-9"] <- "VehAge8"
colnames(freMPL34.test.sinistre.reg)[ colnames(freMPL34.test.sinistre.reg)
=="VehUsagePrivate+trip to office"] <- "VehUsagePrivate_trip_to_office"
colnames(freMPL34.test.sinistre.reg)[ colnames(freMPL34.test.sinistre.reg)
=="VehUsageProfessional run"] <- "VehUsageProfessional_run"
colnames(freMPL34.test.sinistre.reg)[ colnames(freMPL34.test.sinistre.reg)
=="DeducTypePartially refunded"] <- "DeducTypePartially_refunded"
colnames(freMPL34.test.sinistre.reg)[ colnames(freMPL34.test.sinistre.reg)
=="VehBodyother microvan"] <- "VehBodyother_microvan"
```

```
colnames(freMPL34.test.sinistre.reg)[ colnames(freMPL34.test.sinistre.reg)
=="VehBodysport utility vehicle"] <- "VehBodysport_utility_vehicle"
colnames(freMPL34.test.sinistre.reg)[ colnames(freMPL34.test.sinistre.reg)
=="VehBodystation wagon"] <- "VehBodystation_wagon"
colnames(freMPL34.test.sinistre.reg)[ colnames(freMPL34.test.sinistre.reg)
=="VehEnginedirect injection overpowered"] <- "VehEnginedirect_injection_overpowered"
colnames(freMPL34.test.sinistre.reg)[ colnames(freMPL34.test.sinistre.reg)
=="VehEngineinjection overpowered"] <- "VehEngineinjection_overpowered"
colnames(freMPL34.test.sinistre.reg)[ colnames(freMPL34.test.sinistre.reg)
=="VehMaxSpeed140-150 km/h"] <- "VehMaxSpeed140_150_km_h"
colnames(freMPL34.test.sinistre.reg)[ colnames(freMPL34.test.sinistre.reg)
=="VehMaxSpeed150-160 km/h"] <- "VehMaxSpeed150_160_km_h"
colnames(freMPL34.test.sinistre.reg)[ colnames(freMPL34.test.sinistre.reg)
=="VehMaxSpeed160-170 km/h"] <- "VehMaxSpeed160_170_km_h"
colnames(freMPL34.test.sinistre.reg)[ colnames(freMPL34.test.sinistre.reg)
=="VehMaxSpeed170-180 km/h"] <- "VehMaxSpeed170_180_km_h"
colnames(freMPL34.test.sinistre.reg)[ colnames(freMPL34.test.sinistre.reg)
=="VehMaxSpeed180-190 km/h"] <- "VehMaxSpeed180_190_km_h"
colnames(freMPL34.test.sinistre.reg)[ colnames(freMPL34.test.sinistre.reg)
=="VehMaxSpeed190-200 km/h"] <- "VehMaxSpeed190_200_km_h"
colnames(freMPL34.test.sinistre.reg)[ colnames(freMPL34.test.sinistre.reg)
=="VehMaxSpeed200-220 km/h"] <- "VehMaxSpeed200_220_km_h"
colnames(freMPL34.test.sinistre.reg)[ colnames(freMPL34.test.sinistre.reg)
=="VehMaxSpeed220+ km/h "] <- "VehMaxSpeed220_km_h"
colnames(freMPL34.test.sinistre.reg)[ colnames(freMPL34.test.sinistre.reg)
=="GaragePrivate garage"] <- "GaragePrivate_garage"



GLM.claimAmount <- glm(ClaimAmount ~  LicAge + DrivAge + BonusMalus +
                         HasKmLimit + RiskVar+VehAge1 + VehAge10 + VehAge2 +
                         VehAge3 + VehAge4 + VehAge5 + VehAge6 + VehAge8 +
                         DeducTypeNormal + DeducTypePartially_refunded +
                         DeducTypeProportional + DeducTypeRefunded + VehPriceD +
                         VehPriceE + VehPriceF + VehPriceG + VehPriceH +
                         VehPriceI + VehPriceJ + VehPriceK + VehPriceL +
                         VehPriceM + VehPriceN + VehPriceO + VehPriceP +
                         VehPriceQ + + VehEnginedirect_injection_overpowered +
                         VehEngineinjection + VehEngineinjection_overpowered +
                         VehEnergyregular + VehMaxSpeed140_150_km_h +
                         VehMaxSpeed150_160_km_h + VehMaxSpeed160_170_km_h +
                         VehMaxSpeed170_180_km_h + VehMaxSpeed180_190_km_h +
                         VehMaxSpeed190_200_km_h + VehMaxSpeed200_220_km_h +
                         VehMaxSpeed220_km_h + VehClassA + VehClassB +
                         VehClassH + VehClassM1 + VehClassM2+ GarageNone +
                         GaragePrivate_garage + VehUsagePrivate_trip_to_office +
                         VehUsageProfessional_run + VehBodycabriolet +
                         VehBodymicrovan + VehBodycoupe + SocioCategCSP2 +
                         SocioCategCSP4 + SocioCategCSP5 + SocioCategCSP7 +
                         SocioCategCSP9 ,
        family=Gamma(link = 'log') ,
        data= cbind.data.frame(ClaimAmount = freMPL34.app.sinistre$ClaimAmount,
        freMPL34.app.sinistre.reg) )


#Selection par AIC (Forward, Backward, Stepwise)
```

```
GLM.claimAmount <- stepAIC(GLM.claimAmount, trace=TRUE, direction=c("both"))
AIC(GLM.claimAmount)

GLM.claimAmount <- stepAIC(GLM.claimAmount, trace=TRUE, direction=c("backward"))
AIC(GLM.claimAmount)

GLM.claimAmount <- stepAIC(GLM.claimAmount, trace=TRUE, direction=c("forward"))
AIC(GLM.claimAmount)

#le GLM selectionne

GLM.claimAmount <- glm(ClaimAmount ~ LicAge + BonusMalus + HasKmLimit +
                         RiskVar + VehAge3 + VehAge4 + VehAge6 + VehAge8 +
                         DeducTypePartially_refunded + VehPriceE +
                         VehPriceJ + VehPriceM + VehPriceN + VehPriceQ +
                         VehEnergyregular + VehMaxSpeed140_150_km_h +
                         VehMaxSpeed160_170_km_h + VehMaxSpeed170_180_km_h +
                         VehMaxSpeed180_190_km_h + VehMaxSpeed190_200_km_h +
                         VehMaxSpeed200_220_km_h + VehClassA + VehClassM1 +
                         GarageNone + VehUsagePrivate_trip_to_office
                       + VehBodycabriolet,
                       family = Gamma(link = "log"),
        data = cbind.data.frame(ClaimAmount = freMPL34.app.sinistre$ClaimAmount,
        freMPL34.app.sinistre.reg),
                       weights = VehBodymicrovan + VehBodycoupe +
                         SocioCategCSP2 + SocioCategCSP4 +
                         SocioCategCSP5 + SocioCategCSP7 +
                         SocioCategCSP9)



summary(GLM.claimAmount)


#Prediction du modele

predict.GLM.claimAmount.app <- predict(GLM.claimAmount, newdata
                  = as.data.frame(freMPL34.app.sinistre.reg), type = "response")

predict.GLM.claimAmount.test <- predict(GLM.claimAmount, newdata
                 = as.data.frame(freMPL34.test.sinistre.reg), type = "response")

#Graphe de prediction

freMPL34.app.sinistre$prediction <- predict.GLM.claimAmount.app

freMPL34.test.sinistre$prediction <- predict.GLM.claimAmount.test

ggplot(freMPL34.test.sinistre, aes(x = prediction, y=ClaimAmount))
+ geom_point(color = "darkgreen", size = 3, alpha = 0.3) + geom_abline(color="blue")

ggplot(freMPL34.app.sinistre, aes(x = prediction, y=ClaimAmount))
+ geom_point(color = "darkgreen", size = 3, alpha = 0.3) + geom_abline(color="blue")
```

```
freMPL34.test.sinistre <- subset(freMPL34.test,ClaimAmount > 250 & ClaimAmount < 6000)

plot.new()
par(mar=c(4,4,3,5),main = "")
plot(freMPL34.test.sinistre$ClaimAmount,col = "blue",,axes=F,xlab="",ylab="")
axis(2, col="blue",col.axis="blue")
mtext("ClaimAmount",side=2,line=2.5,col="blue")
par(new = T)
plot(predict.GLM.claimAmount.test,col = "red",,axes=F,xlab="",ylab="")
axis( 4 ,col="red",col.axis="red")
mtext("Pr diction",side=4,line=2.5,col="red")
axis( 1 ,col="black",col.axis="black")
mtext("",side=1,line=2.5,col="black")


hist(predict.GLM.claimAmount.test)
densite <- density(predict.GLM.claimAmount.test)
plot(density(predict.GLM.claimAmount.test))


# Conclusion:

# =>On peut accepter le modele (modele assez bien)

#Calcul Prime Pure

predict.GLM.claimAmount.app <- predict(GLM.claimAmount, newdata
                            = as.data.frame(freMPL34.app.reg), type = "response")
PrimeP_Individus_GLM_app <- predict.GLM.claimAmount.app*predict.GLM.claimInd.app

PrimeP_Unique_GLM_app <- mean(PrimeP_Individus_GLM_app)

boxplot(PrimeP_Individus_GLM_app,
      main="Boxplot de la pr diction de la prime pur pour la base freMPL34.app")

hist(PrimeP_Individus_GLM_app,breaks=60,col="blue",density=5,xlab="Prime Pure",
ylab="Fr quences",main="Calcul e par les GLM",ylim=c(0,10000),xlim=c(0,250),tck=0.01

densite.predict.PrimePur.app <- density(PrimeP_Individus_GLM_app)
plot(densite.predict.PrimePur.app,xlim=c(0,250))


predict.GLM.claimAmount.test <- predict(GLM.claimAmount, newdata
                            = as.data.frame(freMPL34.test.reg), type = "response")
PrimeP_Individus_GLM_test <- predict.GLM.claimAmount.test*predict.GLM.claimInd.test

PrimeP_Unique_GLM_test <- mean(PrimeP_Individus_GLM_test)

boxplot(PrimeP_Individus_GLM_test,
    main="Boxplot de la pr diction de la prime pur pour la base freMPL34.test")

hist(PrimeP_Individus_GLM_test,breaks=60,col="red",density=5,xlab="Prime Pure",
ylab="Fr quences",main="Calcul e par les GLM",ylim=c(0,1800),xlim=c(0,250),tck=0.01)

densite.predict.PrimePur.test <- density(PrimeP_Individus_GLM_test)
plot(densite.predict.PrimePur.test,xlim=c(0,250))
```

Annexe C

Code R des Modèles additifs généralisés (GAMs)

```
#########################    GAM ClaimInd    #################

levels(freMPL34$SocioCateg) <- factor(c("CSP1","CSP1","CSP1","CSP1","CSP2",
                                       "CSP2","CSP2","CSP2","CSP2","CSP2",
                                       "CSP3","CSP3","CSP3","CSP4","CSP4",
                                       "CSP4","CSP4","CSP4","CSP4","CSP4",
                                       "CSP5","CSP5","CSP5","CSP5","CSP5",
                                       "CSP5","CSP6","CSP6","CSP6","CSP6",
                                       "CSP6","CSP6","CSP7","CSP7","CSP7",
                                       "CSP7","CSP7","CSP9"))
levels(freMPL34$VehPrice) <- factor(c("A","A","A","D","E","F","G","H","I","J",
                                     "K","L","M","N","O","P","Q","Q","Q","Q",
                                     "Q","Q","Q","Q","Q","Q","Q","Q"))
levels(freMPL34$VehMaxSpeed) <- factor(c("1-140 km/h","1-140 km/h",
                                        "140-150 km/h","150-160 km/h",
                                        "160-170 km/h","170-180 km/h",
                                        "180-190 km/h","190-200 km/h",
                                        "200-220 km/h","220+ km/h "))

freMPL34 <- freMPL34[(freMPL34$VehEnergy != "eletric")
                     & (freMPL34$VehEnergy != "GPL"),]
freMPL34$VehEnergy <- droplevels(freMPL34$VehEnergy)
freMPL34 <- freMPL34[(freMPL34$VehEnergy != "eletric")
                     & (freMPL34$VehEnergy != "GPL"),]
freMPL34$VehEnergy <- droplevels(freMPL34$VehEnergy)
freMPL34$VehEngine <- droplevels(freMPL34$VehEngine)
freMPL34$LicAge <- floor(freMPL34$LicAge/12)

n <- NROW(freMPL34) ; p <- round(0.8*n)
index.app <- sample(1:n, p)
freMPL34.app <- freMPL34[index.app, ]
freMPL34.test <- freMPL34[-index.app, ]


freMPL34.app.reg <- model.matrix(ClaimInd ~ LicAge + VehAge +
                                   Gender + MariStat + SocioCateg + VehUsage +
                                   DrivAge + HasKmLimit + DeducType + BonusMalus +
                                   VehBody + VehPrice + VehEngine + VehEnergy +
```

```
                                    VehMaxSpeed + VehClass + RiskVar + Garage,
                                    data=freMPL34.app)

freMPL34.test.reg <- model.matrix(ClaimInd ~ LicAge + VehAge +
                                     Gender + MariStat + SocioCateg + VehUsage +
                                     DrivAge + HasKmLimit + DeducType + BonusMalus +
                                     VehBody + VehPrice + VehEngine + VehEnergy +
                                     VehMaxSpeed + VehClass + RiskVar + Garage,
                                     data=freMPL34.test)

colnames(freMPL34.app.reg)[ colnames(freMPL34.app.reg)
=="VehAge10+"] <- "VehAge10"
colnames(freMPL34.app.reg)[ colnames(freMPL34.app.reg)
 =="VehAge6-7"] <- "VehAge6"
colnames(freMPL34.app.reg)[ colnames(freMPL34.app.reg)
=="VehAge8-9"] <- "VehAge8"
colnames(freMPL34.app.reg)[ colnames(freMPL34.app.reg)
=="VehUsagePrivate+trip to office"] <- "VehUsagePrivate_trip_to_office"
colnames(freMPL34.app.reg)[ colnames(freMPL34.app.reg)
=="VehUsageProfessional run"] <- "VehUsageProfessional_run"
colnames(freMPL34.app.reg)[ colnames(freMPL34.app.reg)
=="DeducTypePartially refunded"] <- "DeducTypePartially_refunded"
colnames(freMPL34.app.reg)[ colnames(freMPL34.app.reg)
=="VehBodyother microvan"] <- "VehBodyother_microvan"
colnames(freMPL34.app.reg)[ colnames(freMPL34.app.reg)
=="VehBodysport utility vehicle"] <- "VehBodysport_utility_vehicle"
colnames(freMPL34.app.reg)[ colnames(freMPL34.app.reg)
=="VehBodystation wagon"] <- "VehBodystation_wagon"
colnames(freMPL34.app.reg)[ colnames(freMPL34.app.reg)
=="VehEnginedirect injection overpowered"] <- "VehEnginedirect_injection_overpowered"
colnames(freMPL34.app.reg)[ colnames(freMPL34.app.reg)
=="VehEngineinjection overpowered"] <- "VehEngineinjection_overpowered"
colnames(freMPL34.app.reg)[ colnames(freMPL34.app.reg)
=="VehMaxSpeed140-150 km/h"] <- "VehMaxSpeed140_150_km_h"
colnames(freMPL34.app.reg)[ colnames(freMPL34.app.reg)
=="VehMaxSpeed150-160 km/h"] <- "VehMaxSpeed150_160_km_h"
colnames(freMPL34.app.reg)[ colnames(freMPL34.app.reg)
=="VehMaxSpeed160-170 km/h"] <- "VehMaxSpeed160_170_km_h"
colnames(freMPL34.app.reg)[ colnames(freMPL34.app.reg)
=="VehMaxSpeed170-180 km/h"] <- "VehMaxSpeed170_180_km_h"
colnames(freMPL34.app.reg)[ colnames(freMPL34.app.reg)
=="VehMaxSpeed180-190 km/h"] <- "VehMaxSpeed180_190_km_h"
colnames(freMPL34.app.reg)[ colnames(freMPL34.app.reg)
=="VehMaxSpeed190-200 km/h"] <- "VehMaxSpeed190_200_km_h"
colnames(freMPL34.app.reg)[ colnames(freMPL34.app.reg)
=="VehMaxSpeed200-220 km/h"] <- "VehMaxSpeed200_220_km_h"
colnames(freMPL34.app.reg)[ colnames(freMPL34.app.reg)
=="VehMaxSpeed220+ km/h "] <- "VehMaxSpeed220_km_h"
colnames(freMPL34.app.reg)[ colnames(freMPL34.app.reg)
=="GaragePrivate garage"] <- "GaragePrivate_garage"


colnames(freMPL34.test.reg)[ colnames(freMPL34.test.reg)
=="VehAge10+"] <- "VehAge10"
colnames(freMPL34.test.reg)[ colnames(freMPL34.test.reg)
=="VehAge6-7"] <- "VehAge6"
```

```
colnames(freMPL34.test.reg)[ colnames(freMPL34.test.reg)
=="VehAge8-9"] <- "VehAge8"
colnames(freMPL34.test.reg)[ colnames(freMPL34.test.reg)
=="VehUsagePrivate+trip to office"] <- "VehUsagePrivate_trip_to_office"
colnames(freMPL34.test.reg)[ colnames(freMPL34.test.reg)
=="VehUsageProfessional run"] <- "VehUsageProfessional_run"
colnames(freMPL34.test.reg)[ colnames(freMPL34.test.reg)
=="DeducTypePartially refunded"] <- "DeducTypePartially_refunded"
colnames(freMPL34.test.reg)[ colnames(freMPL34.test.reg)
=="VehBodyother microvan"] <- "VehBodyother_microvan"
colnames(freMPL34.test.reg)[ colnames(freMPL34.test.reg)
=="VehBodysport utility vehicle"] <- "VehBodysport_utility_vehicle"
colnames(freMPL34.test.reg)[ colnames(freMPL34.test.reg)
=="VehBodystation wagon"] <- "VehBodystation_wagon"
colnames(freMPL34.test.reg)[ colnames(freMPL34.test.reg)
=="VehEnginedirect injection overpowered"] <- "VehEnginedirect_injection_overpowered"
colnames(freMPL34.test.reg)[ colnames(freMPL34.test.reg)
=="VehEngineinjection overpowered"] <- "VehEngineinjection_overpowered"
colnames(freMPL34.test.reg)[ colnames(freMPL34.test.reg)
=="VehMaxSpeed140-150 km/h"] <- "VehMaxSpeed140_150_km_h"
colnames(freMPL34.test.reg)[ colnames(freMPL34.test.reg)
=="VehMaxSpeed150-160 km/h"] <- "VehMaxSpeed150_160_km_h"
colnames(freMPL34.test.reg)[ colnames(freMPL34.test.reg)
=="VehMaxSpeed160-170 km/h"] <- "VehMaxSpeed160_170_km_h"
colnames(freMPL34.test.reg)[ colnames(freMPL34.test.reg)
=="VehMaxSpeed170-180 km/h"] <- "VehMaxSpeed170_180_km_h"
colnames(freMPL34.test.reg)[ colnames(freMPL34.test.reg)
=="VehMaxSpeed180-190 km/h"] <- "VehMaxSpeed180_190_km_h"
colnames(freMPL34.test.reg)[ colnames(freMPL34.test.reg)
=="VehMaxSpeed190-200 km/h"] <- "VehMaxSpeed190_200_km_h"
colnames(freMPL34.test.reg)[ colnames(freMPL34.test.reg)
=="VehMaxSpeed200-220 km/h"] <- "VehMaxSpeed200_220_km_h"
colnames(freMPL34.test.reg)[ colnames(freMPL34.test.reg)
=="VehMaxSpeed220+ km/h "] <- "VehMaxSpeed220_km_h"
colnames(freMPL34.test.reg)[ colnames(freMPL34.test.reg)
=="GaragePrivate garage"] <- "GaragePrivate_garage"

#modele GAM:

GLM.ClaimInd <- gam(ClaimInd ~ s(LicAge)+ s(RiskVar) + s(BonusMalus) + VehAge1 +
                      VehAge10 + VehAge2 + VehAge3 + VehAge4 + VehAge5 + VehAge6 +
                      VehAge8 + GenderMale + MariStatOther + SocioCategCSP2 +
                      SocioCategCSP3 + SocioCategCSP4 + SocioCategCSP5 +
                      SocioCategCSP6 + SocioCategCSP7 + SocioCategCSP9 +
                      VehUsagePrivate_trip_to_office + VehUsageProfessional +
                      VehUsageProfessional_run + DrivAge + HasKmLimit +
                      DeducTypeNormal + DeducTypePartially_refunded +
                      DeducTypeProportional + DeducTypeRefunded  +
                      VehBodycabriolet + VehBodycoupe + VehBodymicrovan +
                      VehBodyother_microvan + VehBodysedan +
                      VehBodysport_utility_vehicle + VehBodystation_wagon +
                      VehBodyvan + VehPriceD + VehPriceE + VehPriceF + VehPriceG +
                      VehPriceH + VehPriceI + VehPriceJ + VehPriceK + VehPriceL +
                      VehPriceM + VehPriceN + VehPriceO + VehPriceP + VehPriceQ +
                      VehEnginedirect_injection_overpowered + VehEngineinjection +
                      VehEngineinjection_overpowered + VehEnergyregular +
```

```
                        VehMaxSpeed140_150_km_h + VehMaxSpeed150_160_km_h +
                        VehMaxSpeed160_170_km_h + VehMaxSpeed170_180_km_h +
                        VehMaxSpeed180_190_km_h + VehMaxSpeed190_200_km_h +
                        VehMaxSpeed200_220_km_h + VehMaxSpeed220_km_h  + VehClassA +
                        VehClassB + VehClassH + VehClassM1 + VehClassM2 +
                        GarageNone + GaragePrivate_garage,
          family=binomial(link = 'logit'),
          data=cbind.data.frame(ClaimInd=freMPL34.app$ClaimInd, freMPL34.app.reg),
          method="REML")


# Le GAM selectionne :

GAM.claimInd <- gam(formula = ClaimInd ~ s(LicAge, k=34) + BonusMalus + VehAge10 +
                        VehAge6 + VehAge8 + SocioCategCSP6 + VehUsageProfessional +
                        VehUsageProfessional_run + DeducTypePartially_refunded +
                        DeducTypeRefunded + VehBodystation_wagon + VehPriceO +
                        VehPriceP + VehPriceQ + VehMaxSpeed140_150_km_h +
                        VehMaxSpeed150_160_km_h + VehMaxSpeed160_170_km_h +
                        VehMaxSpeed170_180_km_h + VehMaxSpeed190_200_km_h +
                        VehMaxSpeed200_220_km_h + VehMaxSpeed220_km_h + VehClassA +
                        GaragePrivate_garage,
        family = binomial("logit"),
        data = cbind.data.frame(ClaimInd = freMPL34.app$ClaimInd, freMPL34.app.reg),
        method="REML")

AIC(GAM.claimInd)

# Comparaison GLM et GAM:

#> AIC(GLM.claimInd) > AIC(GAM.claimInd)

# => on remarque le GAM est plus adapte au modele que le GLM.

plot(GAM.claimInd, pages=1)

# => Remarque: le modele fit bien.

# Verification de la qualite des modeles

summary(GAM.claimInd)

par(mfrow=c(2,2))
gam.check(GAM.claimInd)

#Verification des residus:

binnedplot(fitted(GAM.claimInd), residuals(GAM.claimInd, type="response"))

#Proba d'avoir ClaimInd=1

p1 <- length(freMPL34[freMPL34$ClaimInd==1,]$ClaimInd)
p2 <- length(freMPL34$ClaimInd)

proba_ind_1 <- p1 / p2
```

```
#Prediction du modele

predict.GAM.claimInd.test <- predict(GAM.claimInd, newdata
                        = as.data.frame(freMPL34.test.reg), type = "response")
predict.GAM.claimInd.test.bis=as.factor(ifelse(predict.GAM.claimInd.test
                                              > proba_ind_1,1,0))

predict.GAM.claimInd.app <- predict(GAM.claimInd, newdata
                         = as.data.frame(freMPL34.app.reg), type = "response")
predict.GAM.claimInd.app.bis=as.factor(ifelse(predict.GAM.claimInd.app
                                             > proba_ind_1,1,0))

#Matrice de confusion sur l'echantillon test

confusionMatrix(data = predict.GAM.claimInd.test.bis, reference
                =as.factor(freMPL34.test$ClaimInd))

#Matrice de confusion sur l'echantillon d'apprentissage

confusionMatrix(data = predict.GAM.claimInd.app.bis, reference
                =as.factor(freMPL34.app$ClaimInd))

#Erreur sur l'echantillon d'apprentissage

mean((predict.GAM.claimInd.app - freMPL34.app[,"ClaimInd"])**2)

#Erreur sur l'echantillon test

mean((predict.GAM.claimInd.test - freMPL34.test[,"ClaimInd"])**2)

# Conclusion:

# => L'erreur est petite, Accuracy acceptable donc on peut accepter le modele
#    (modele GAM est mieux adapte que GLM)

########################   GAM ClaimAmount    #################

freMPL34.app.sinistre <- subset

freMPL34.test.sinistre <- subset(freMPL34.test, ClaimAmount > 0)

freMPL34.app.sinistre.reg <- model.matrix(ClaimAmount ~ LicAge + VehAge +
                                          Gender + MariStat + SocioCateg +
                                          VehUsage + DrivAge + HasKmLimit +
                                          DeducType + BonusMalus + VehBody +
                                          VehPrice + VehEngine + VehEnergy +
                                          VehMaxSpeed + VehClass + RiskVar +
                                          Garage, data=freMPL34.app.sinistre)

freMPL34.test.sinistre.reg <- model.matrix(ClaimAmount ~ LicAge + VehAge +
                                           Gender + MariStat + SocioCateg +
                                           VehUsage + DrivAge + HasKmLimit +
                                           DeducType + BonusMalus + VehBody +
                                           VehPrice + VehEngine + VehEnergy +
                                           VehMaxSpeed + VehClass + RiskVar +
                                           Garage, data=freMPL34.test.sinistre)
```

```
colnames(freMPL34.app.sinistre.reg)[ colnames(freMPL34.app.sinistre.reg)
=="VehAge10+"] <- "VehAge10"
colnames(freMPL34.app.sinistre.reg)[ colnames(freMPL34.app.sinistre.reg)
=="VehAge6-7"] <- "VehAge6"
colnames(freMPL34.app.sinistre.reg)[ colnames(freMPL34.app.sinistre.reg)
=="VehAge8-9"] <- "VehAge8"
colnames(freMPL34.app.sinistre.reg)[ colnames(freMPL34.app.sinistre.reg)
=="VehUsagePrivate+trip to office"] <- "VehUsagePrivate_trip_to_office"
colnames(freMPL34.app.sinistre.reg)[ colnames(freMPL34.app.sinistre.reg)
=="VehUsageProfessional run"] <- "VehUsageProfessional_run"
colnames(freMPL34.app.sinistre.reg)[ colnames(freMPL34.app.sinistre.reg)
=="DeducTypePartially refunded"] <- "DeducTypePartially_refunded"
colnames(freMPL34.app.sinistre.reg)[ colnames(freMPL34.app.sinistre.reg)
=="VehBodyother microvan"] <- "VehBodyother_microvan"
colnames(freMPL34.app.sinistre.reg)[ colnames(freMPL34.app.sinistre.reg)
=="VehBodysport utility vehicle"] <- "VehBodysport_utility_vehicle"
colnames(freMPL34.app.sinistre.reg)[ colnames(freMPL34.app.sinistre.reg)
=="VehBodystation wagon"] <- "VehBodystation_wagon"
colnames(freMPL34.app.sinistre.reg)[ colnames(freMPL34.app.sinistre.reg)
=="VehEnginedirect injection overpowered"] <- "VehEnginedirect_injection_overpowered"
colnames(freMPL34.app.sinistre.reg)[ colnames(freMPL34.app.sinistre.reg)
=="VehEngineinjection overpowered"] <- "VehEngineinjection_overpowered"
colnames(freMPL34.app.sinistre.reg)[ colnames(freMPL34.app.sinistre.reg)
=="VehMaxSpeed140-150 km/h"] <- "VehMaxSpeed140_150_km_h"
colnames(freMPL34.app.sinistre.reg)[ colnames(freMPL34.app.sinistre.reg)
=="VehMaxSpeed150-160 km/h"] <- "VehMaxSpeed150_160_km_h"
colnames(freMPL34.app.sinistre.reg)[ colnames(freMPL34.app.sinistre.reg)
=="VehMaxSpeed160-170 km/h"] <- "VehMaxSpeed160_170_km_h"
colnames(freMPL34.app.sinistre.reg)[ colnames(freMPL34.app.sinistre.reg)
=="VehMaxSpeed170-180 km/h"] <- "VehMaxSpeed170_180_km_h"
colnames(freMPL34.app.sinistre.reg)[ colnames(freMPL34.app.sinistre.reg)
=="VehMaxSpeed180-190 km/h"] <- "VehMaxSpeed180_190_km_h"
colnames(freMPL34.app.sinistre.reg)[ colnames(freMPL34.app.sinistre.reg)
=="VehMaxSpeed190-200 km/h"] <- "VehMaxSpeed190_200_km_h"
colnames(freMPL34.app.sinistre.reg)[ colnames(freMPL34.app.sinistre.reg)
=="VehMaxSpeed200-220 km/h"] <- "VehMaxSpeed200_220_km_h"
colnames(freMPL34.app.sinistre.reg)[ colnames(freMPL34.app.sinistre.reg)
=="VehMaxSpeed220+ km/h "] <- "VehMaxSpeed220_km_h"
colnames(freMPL34.app.sinistre.reg)[ colnames(freMPL34.app.sinistre.reg)
=="GaragePrivate garage"] <- "GaragePrivate_garage"

colnames(freMPL34.test.sinistre.reg)[ colnames(freMPL34.test.sinistre.reg)
=="VehAge10+"] <- "VehAge10"
colnames(freMPL34.test.sinistre.reg)[ colnames(freMPL34.test.sinistre.reg)
=="VehAge6-7"] <- "VehAge6"
colnames(freMPL34.test.sinistre.reg)[ colnames(freMPL34.test.sinistre.reg)
=="VehAge8-9"] <- "VehAge8"
colnames(freMPL34.test.sinistre.reg)[ colnames(freMPL34.test.sinistre.reg)
=="VehUsagePrivate+trip to office"] <- "VehUsagePrivate_trip_to_office"
colnames(freMPL34.test.sinistre.reg)[ colnames(freMPL34.test.sinistre.reg)
=="VehUsageProfessional run"] <- "VehUsageProfessional_run"
colnames(freMPL34.test.sinistre.reg)[ colnames(freMPL34.test.sinistre.reg)
=="DeducTypePartially refunded"] <- "DeducTypePartially_refunded"
colnames(freMPL34.test.sinistre.reg)[ colnames(freMPL34.test.sinistre.reg)
=="VehBodyother microvan"] <- "VehBodyother_microvan"
```

```
colnames(freMPL34.test.sinistre.reg)[ colnames(freMPL34.test.sinistre.reg)
=="VehBodysport utility vehicle"] <- "VehBodysport_utility_vehicle"
colnames(freMPL34.test.sinistre.reg)[ colnames(freMPL34.test.sinistre.reg)
=="VehBodystation wagon"] <- "VehBodystation_wagon"
colnames(freMPL34.test.sinistre.reg)[ colnames(freMPL34.test.sinistre.reg)
=="VehEnginedirect injection overpowered"] <- "VehEnginedirect_injection_overpowered"
colnames(freMPL34.test.sinistre.reg)[ colnames(freMPL34.test.sinistre.reg)
=="VehEngineinjection overpowered"] <- "VehEngineinjection_overpowered"
colnames(freMPL34.test.sinistre.reg)[ colnames(freMPL34.test.sinistre.reg)
=="VehMaxSpeed140-150 km/h"] <- "VehMaxSpeed140_150_km_h"
colnames(freMPL34.test.sinistre.reg)[ colnames(freMPL34.test.sinistre.reg)
=="VehMaxSpeed150-160 km/h"] <- "VehMaxSpeed150_160_km_h"
colnames(freMPL34.test.sinistre.reg)[ colnames(freMPL34.test.sinistre.reg)
=="VehMaxSpeed160-170 km/h"] <- "VehMaxSpeed160_170_km_h"
colnames(freMPL34.test.sinistre.reg)[ colnames(freMPL34.test.sinistre.reg)
=="VehMaxSpeed170-180 km/h"] <- "VehMaxSpeed170_180_km_h"
colnames(freMPL34.test.sinistre.reg)[ colnames(freMPL34.test.sinistre.reg)
=="VehMaxSpeed180-190 km/h"] <- "VehMaxSpeed180_190_km_h"
colnames(freMPL34.test.sinistre.reg)[ colnames(freMPL34.test.sinistre.reg)
=="VehMaxSpeed190-200 km/h"] <- "VehMaxSpeed190_200_km_h"
colnames(freMPL34.test.sinistre.reg)[ colnames(freMPL34.test.sinistre.reg)
=="VehMaxSpeed200-220 km/h"] <- "VehMaxSpeed200_220_km_h"
colnames(freMPL34.test.sinistre.reg)[ colnames(freMPL34.test.sinistre.reg)
=="VehMaxSpeed220+ km/h "] <- "VehMaxSpeed220_km_h"
colnames(freMPL34.test.sinistre.reg)[ colnames(freMPL34.test.sinistre.reg)
=="GaragePrivate garage"] <- "GaragePrivate_garage"

#Le modele ClaimAmount avec le modele gamma

GAM.claimAmount <- gam(ClaimAmount ~  s(LicAge) + s(BonusMalus) + HasKmLimit +
                         s(RiskVar) + VehAge1 + VehAge10 + VehAge2 + VehAge3 +
                         VehAge4 + VehAge5 + VehAge6 + VehAge8 + DeducTypeNormal +
                         DeducTypePartially_refunded+ DeducTypeProportional +
                         DeducTypeRefunded + VehPriceD + VehPriceE + VehPriceF +
                         VehPriceG + VehPriceH + VehPriceI + VehPriceJ +
                         VehPriceK + VehPriceL + VehPriceM + VehPriceN +
                         VehPriceO + VehPriceP + VehPriceQ +
                         VehEnginedirect_injection_overpowered + VehEngineinjection +
                         VehEngineinjection_overpowered + VehEnergyregular +
                         VehMaxSpeed140_150_km_h + VehMaxSpeed150_160_km_h +
                         VehMaxSpeed160_170_km_h + VehMaxSpeed170_180_km_h +
                         VehMaxSpeed180_190_km_h + VehMaxSpeed190_200_km_h +
                         VehMaxSpeed200_220_km_h + VehMaxSpeed220_km_h + VehClassA +
                         VehClassB + VehClassH + VehClassM1 + VehClassM2 +
                         GarageNone + GaragePrivate_garage +
                         VehUsagePrivate_trip_to_office + VehUsageProfessional_run +
                         VehBodycabriolet, VehBodymicrovan + VehBodycoupe +
                         SocioCategCSP2 + SocioCategCSP4 + SocioCategCSP5 +
                         SocioCategCSP7 + SocioCategCSP9 ,
                       family=Gamma(link = 'log') ,
          data= cbind.data.frame(ClaimAmount = freMPL34.app.sinistre$ClaimAmount,
         freMPL34.app.sinistre.reg), method = "REML" )

#Selection par AIC (Forward, Backward, Stepwise)

GAM.claimAmount <- stepAIC(GAM.claimAmount, trace=TRUE, direction=c("both"))
```

```
AIC(GAM.claimAmount)

GAM.claimAmount <- stepAIC(GAM.claimAmount, trace=TRUE, direction=c("backward"))
AIC(GAM.claimAmount)

GAM.claimAmount <- stepAIC(GAM.claimAmount, trace=TRUE, direction=c("forward"))
AIC(GAM.claimAmount)

#le GLM selectionne


GAM.claimAmount <- gam(ClaimAmount ~ s(LicAge, k=62)+ s(BonusMalus) + HasKmLimit +
                         s(RiskVar) + VehAge3 + VehAge4 + VehAge6 + VehAge8 +
                         DeducTypePartially_refunded +  VehPriceE + VehPriceJ +
                         VehPriceM + VehPriceN + VehPriceQ + VehEnergyregular +
                         VehMaxSpeed140_150_km_h + VehMaxSpeed160_170_km_h +
                         VehMaxSpeed170_180_km_h + VehMaxSpeed180_190_km_h +
                         VehMaxSpeed190_200_km_h + VehMaxSpeed200_220_km_h +
                         VehClassA + VehClassM1 + GarageNone +
                         VehUsagePrivate_trip_to_office + VehBodycabriolet,
        family = Gamma(link = "log"),
        data = cbind.data.frame(ClaimAmount = freMPL34.app.sinistre$ClaimAmount,
                                              freMPL34.app.sinistre.reg),
                      weights = VehBodymicrovan + VehBodycoupe +
                        SocioCategCSP2 + SocioCategCSP4 +SocioCategCSP5 +
                        SocioCategCSP7 + SocioCategCSP9,
        method = "REML")


# Comparaison GLM et GAM:

#> AIC(GLM.claiAmount) > AIC(GAM.claimAmount)

# => on remarque le GAM est plus adapte au modele que le GLM.

plot(GAM.claimAmount, pages=1)

# => Remarque: le modele fit bien.

# verification de la qualite des modeles


summary(GAM.claimAmount)

par(mfrow=c(2,2))
gam.check(GAM.claimAmount)

#Verification des residus:

binnedplot(fitted(GAM.claimAmount), residuals(GAM.claimAmount, type="response"))


#Prediction du modele

predict.GAM.claimAmount.app <- predict(GAM.claimAmount, newdata
                  = as.data.frame(freMPL34.app.sinistre.reg), type = "response")
```

```
predict.GAM.claimAmount.test <- predict(GAM.claimAmount, newdata
                = as.data.frame(freMPL34.test.sinistre.reg), type = "response")

#Graphe de prediction

freMPL34.app.sinistre$prediction <- predict.GAM.claimAmount.app

freMPL34.test.sinistre$prediction <- predict.GAM.claimAmount.test

ggplot(freMPL34.test.sinistre, aes(x = prediction, y=ClaimAmount))
+ geom_point(color = "darkgreen", size = 3, alpha = 0.3) + geom_abline(color="blue")

ggplot(freMPL34.app.sinistre, aes(x = prediction, y=ClaimAmount))
+ geom_point(color = "darkgreen", size = 3, alpha = 0.3) + geom_abline(color="blue")

freMPL34.test.sinistre <- subset(freMPL34.test,ClaimAmount > 700 & ClaimAmount < 6000)

plot.new()
par(mar=c(4,4,3,5),main = "")
plot(freMPL34.test.sinistre$ClaimAmount,col = "blue",,axes=F,xlab="",ylab="")
axis(2, col="blue",col.axis="blue")
mtext("ClaimAmount",side=2,line=2.5,col="blue")
par(new = T)
plot(predict.GAM.claimAmount.test,col = "red",,axes=F,xlab="",ylab="")
axis( 4 ,col="red",col.axis="red")
mtext("Pr diction",side=4,line=2.5,col="red")
axis( 1 ,col="black",col.axis="black")
mtext("",side=1,line=2.5,col="black")

# Valeur absolue de l'ecart de prediction entre GAM et GLM

freMPL34.test.sinistre$Abs_ecart_GAM_GLM_prediction <- abs(predict.GAM.claimAmount.tes
                                                    - predict.GLM.claimAmount.test

boxplot(freMPL34.test.sinistre$Abs_ecart_GAM_GLM_prediction)

hist(freMPL34.test.sinistre$Abs_ecart_GAM_GLM_prediction,breaks=20,col="black",
     density=5,xlab="abs(predict.GAM-predict.GLM)",ylab="Fr quences",
     main="Histogramme de l' cart  de pr diction entre GAM et GLM",ylim=c(0,60),
     xlim=c(0,485),tck=0.01)


# Conclusion:

# =>On peut accepter le modele (modele GAM est mieux adapte que GLM)

#Calcul Prime Pure

#On predit tous les sinistres

predict.GAM.claimAmount.test <- predict(GAM.claimAmount, newdata
                          = as.data.frame(freMPL34.test.reg), type = "response")

predict.GAM.claimInd.app <- predict(GAM.claimInd, newdata
                          = as.data.frame(freMPL34.app.reg), type = "response")
```

```
predict.GAM.claimAmount.app <- predict(GAM.claimAmount, newdata
                          = as.data.frame(freMPL34.app.reg), type = "response")
PrimeP_Individus_GAM_app<-predict.GAM.claimAmount.app*predict.GAM.claimInd.app
PrimeP_Unique_GAM_app<-mean(PrimeP_Individus_GAM_app)
PrimeP_Unique_GAM_test<-mean(PrimeP_Individus_GAM_test)
PrimeP_Individus_GAM_test<-predict.GAM.claimAmount.test*predict.GAM.claimInd.test

PrimeP_Unique_GAM_test
PrimeP_Unique_GLM_test

freMPL34.test$prime_GAM <- PrimeP_Individus_GAM_test
freMPL34.test$ClaimAmountPredit_GAM<-predict.GAM.claimAmount.test
freMPL34.test$ClaimIndPredit_GAM<-predict.GAM.claimInd.test
```

Annexe D

Code R Choix de la prime

```
# /         /           /           /
#
# Observation visuel des primes
# Et Analyse par simulation de la prime avec chargement
#
# /         /           /           /


#Histogramme Primes  pures
hist(PrimeP_Individus_GAM_test,breaks=15,col="red",density=5,xlab="Prime Pure",
ylab="Frequences",main="Histogramme Prime Pure",ylim=c(0,5000),xlim=c(0,250),
tck=0.01)
hist(PrimeP_Individus_GLM_test,breaks=50,col="blue",density=5,xlab="Prime Pure",
ylab="Frequences",main="Histogramme Prime Pure",ylim=c(0,5000),xlim=c(0,250),
tck=0.01)

#Simulation de la Prime avec Chargement

raggsum <- function(mydata)
{
  n <- NROW(mydata)
  i <- sample.int(n, replace=TRUE)
  sum(mydata[i, ]$ClaimAmount)
}
raggsum(freMPL34.test)

ptfaggsimu <- replicate(10^3, raggsum(freMPL34.test))

proba_GLM={}
for(i in 1:200) {
  proba_GLM[i]<-sum(ptfaggsimu<(1+i/1000)*sum(PrimeP_Individus_GLM_test))/1000
}
summary(proba_GLM)

summary(proba_GLM<0.99)
summary(proba_GLM<0.95)
summary(proba_GLM<0.90)

#Donne sur notre analyse les valeurs ci dessous
#Elles vont un peu changer a chaque simulation
#Le quantile de 0.99 est donc 1.151 pour les GLM
```

```
#Le quantile de 0.99 est donc 1.092 pour les GLM
#Le quantile de 0.99 est donc 1.057 pour les GLM

#Avec les GAM

proba_GAM={}
for(i in 1:200) {
  proba_GAM[i]<-sum(ptfaggsimu<(1+i/1000)*sum(PrimeP_Individus_GAM_test))/1000
}
summary(proba_GAM)

summary(proba_GAM<0.99)
summary(proba_GAM<0.95)
summary(proba_GAM<0.90)

#Le quantile de 0.99 est donc 1.142 pour les GAM
#Le quantile de 0.95 est donc 1.084 pour les GAM
#Le quantile de 0.90 est donc 1.049 pour les GAM

#plot des probas de non ruine GAM

plot((1:200)/1000,proba_GAM,type='l',ylab='probabilite',xlab='Taux de chargement')

#plot des probas de non ruine GLM

plot((1:200)/1000,proba_GLM,type='l',ylab='probabilite',xlab='Taux de chargement')


#Histogramme des  somme des ClaimAmount
#Avec en valeurs de gauche a droite (GLM):
#prime pure / moyenne somme claimamount / Prime commerciale avec chargement 0.057/
#Prime commerciale avec chargement 0.092 / Prime commerciale avec chargement 0.151
#Histogramme sur les GLM

base<-data.frame(1:1000)
base$ptfaggsimu<-ptfaggsimu
base$num<-1:length(ptfaggsimu)

p1<- ggplot(base, aes(x=ptfaggsimu))
p1+geom_histogram(aes(y=..density..),color="black", fill="white")
+geom_density(alpha=.2, fill="lightblue")
+geom_vline(aes(xintercept=sum(PrimeP_Individus_GLM_test)),
            color="blue", linetype="dashed", size=1)
+geom_vline(aes(xintercept=1.057*sum(PrimeP_Individus_GLM_test)),color="red",
            linetype="dashed", size=1)
+geom_vline(aes(xintercept=1.092*sum(PrimeP_Individus_GLM_test)),color="green",
      linetype="dashed", size=1)+geom_vline(aes(xintercept=mean(ptfaggsimu)),
            color="yellow", size=1)
+geom_vline(aes(xintercept=1.151*sum(freMPL34.test$prime_GLM)),color="brown",
            linetype="dashed", size=1)

#Histogramme sur les GAM
#on met les valeurs des quantiles de GAM

p2<- ggplot(base, aes(x=ptfaggsimu))
p2+geom_histogram(aes(y=..density..),color="black", fill="white")
```

```
+geom_density(alpha=.2, fill="lightblue")
+geom_vline(aes(xintercept=sum(PrimeP_Individus_GAM_test)),color="blue",
            linetype="dashed", size=1)
+geom_vline(aes(xintercept=1.049*sum(PrimeP_Individus_GAM_test)),color="red",
linetype="dashed", size=1)
+geom_vline(aes(xintercept=1.084*sum(PrimeP_Individus_GAM_test)),color="green",
linetype="dashed", size=1)+geom_vline(aes(xintercept=mean(ptfaggsimu)),
color="yellow", size=1)
+geom_vline(aes(xintercept=1.142*sum(freMPL34.test$prime_GAM)),color="brown",
            linetype="dashed", size=1)

#On peut observer les deux en m me temps

plot.new()
par(mar=c(4,4,3,5))
plot((1:200)/1000,proba_GLM,type='l',ylab='',xlab='',col='red')
par(new = T)
plot((1:200)/1000,proba_GAM,type='l',ylab='',xlab='',col='blue')


#On transforme en facteurs pour pouvoir faire des boxplots avec ggplot
#Boxplots avec ClaimAmount

freMPL34.test$CA_GAM<-predict.GAM.claimAmount.test
freMPL34.test$CA_GLM<-predict.GAM.claimAmount.test

freMPL34.test$prime<-PrimeP_Individus_GLM_test
freMPL34.test$primeGAM<-PrimeP_Individus_GAM_test
freMPL34.test$VehAge<-as.factor(freMPL34.test$VehAge)
freMPL34.test$DrivAge<-as.factor(freMPL34.test$DrivAge)
freMPL34.test$LicAge<-as.factor(freMPL34.test$LicAge)
freMPL34.test$SocioCateg<-as.factor(freMPL34.test$SocioCateg)
freMPL34.test$CA_GAM<-predict.GAM.claimAmount.test
freMPL34.test$CA_GLM<-predict.GLM.claimAmount.test

p <- ggplot(freMPL34.test, aes(x=SocioCateg, y=CA_GAM)) +
  geom_boxplot()
p2 <- ggplot(freMPL34.test, aes(x=VehAge, y=CA_GAM)) +
  geom_boxplot()
p3 <- ggplot(freMPL34.test, aes(x=DrivAge, y=CA_GAM)) +
  geom_boxplot()
p4 <- ggplot(freMPL34.test, aes(x=LicAge, y=CA_GAM)) +
  geom_boxplot()

p4


q <- ggplot(freMPL34.test, aes(x=SocioCateg, y=CA_GLM)) +
  geom_boxplot()
q2 <- ggplot(freMPL34.test, aes(x=VehAge, y=CA_GLM)) +
  geom_boxplot()
q3 <- ggplot(freMPL34.test, aes(x=DrivAge, y=CA_GLM)) +
  geom_boxplot()
q4 <- ggplot(freMPL34.test, aes(x=LicAge, y=CA_GLM)) +
  geom_boxplot()
```

```
#Boxplots avec la prime pure

p <- ggplot(freMPL34.test, aes(x=SocioCateg, y=prime)) +
  geom_boxplot()
p2 <- ggplot(freMPL34.test, aes(x=VehAge, y=prime)) +
  geom_boxplot()
p3 <- ggplot(freMPL34.test, aes(x=DrivAge, y=prime)) +
  geom_boxplot()
p4 <- ggplot(freMPL34.test, aes(x=LicAge, y=prime)) +
  geom_boxplot()

p <- ggplot(freMPL34.test, aes(x=SocioCateg, y=prime)) +
  geom_boxplot()
p2 <- ggplot(freMPL34.test, aes(x=VehAge, y=prime)) +
  geom_boxplot()
p3 <- ggplot(freMPL34.test, aes(x=DrivAge, y=prime)) +
  geom_boxplot()
p4 <- ggplot(freMPL34.test, aes(x=LicAge, y=prime)) +
  geom_boxplot()

q <- ggplot(freMPL34.test, aes(x=SocioCateg, y=primeGAM)) +
  geom_boxplot()
q2 <- ggplot(freMPL34.test, aes(x=VehAge, y=primeGAM)) +
  geom_boxplot()
q3 <- ggplot(freMPL34.test, aes(x=DrivAge, y=primeGAM)) +
  geom_boxplot()
q4 <- ggplot(freMPL34.test, aes(x=LicAge, y=primeGAM)) +
  geom_boxplot()
p
# On trace des histogrammes sur les primes avec chargement
# Et des boxplots de l'ecart de ces primes

freMPL34.test$pc_GAM <- 1.142*PrimeP_Individus_GAM_test
freMPL34.test$pc_GLM <- 1.151*PrimeP_Individus_GLM_test

freMPL34.test$ClaimAmountPredit_GAM<-predict.GAM.claimAmount.test
freMPL34.test$ClaimIndPredit_GAM<-predict.GAM.claimInd.test

plot.new()
par(mar=c(4,4,3,5))
hist(1.142*PrimeP_Individus_GAM_test,breaks=30,col="red",density=30,
     xlab="Primes (GLM= bleu / GAM =rouge)",
     ylab="Frequences",main="Histogramme Primes avec Chargement",ylim=c(0,5000),
     xlim=c(0,250),tck=0.01)
par(new=T)
hist(1.151*PrimeP_Individus_GLM_test,breaks=50,col="blue",density=30,
     xlab="Primes (GLM= bleu / GAM =rouge)",
     ylab="Frequences",main="Histogramme Primes avec Chargement",ylim=c(0,5000),
     xlim=c(0,250),tck=0.01)

sum(1.151*PrimeP_Individus_GLM_test)-sum(1.151*PrimeP_Individus_GAM_test)


summary(freMPL34.test$pc_GAM)
summary(freMPL34.test$pc_GLM)
```

```
summary(freMPL34.test$pc_GAM>150)
summary(freMPL34.test$pc_GLM>150)

summary(freMPL34.test$pc_GAM<50)
summary(freMPL34.test$pc_GLM<50)

freMPL34.test$ecart<-abs(freMPL34.test$pc_GLM-freMPL34.test$pc_GAM)

#dans notre base test on voulait observer un grand ecart superieur a 833
#summary(freMPL34.test[freMPL34.test$ecart>833,])

p <- ggplot(freMPL34.test, aes(x=SocioCateg, y=ecart)) +
  geom_boxplot()
p2 <- ggplot(freMPL34.test, aes(x=VehAge, y=ecart)) +
  geom_boxplot()
p3 <- ggplot(freMPL34.test, aes(x=DrivAge, y=ecart)) +
  geom_boxplot()
p4 <- ggplot(freMPL34.test, aes(x=LicAge, y=ecart)) +
  geom_boxplot()
p5 <- ggplot(freMPL34.test, aes(x=VehPrice, y=ecart)) +
  geom_boxplot()
p6 <- ggplot(freMPL34.test, aes(x=MariStat, y=ecart)) +
  geom_boxplot()
```