Université Paris Dauphine

Mémoire M1 - Mathématiques Appliquées

Tarification en assurance IARD avec les GLM en intégrant les données issues de la sécurité routière

Victoire BELVEZE
Hadrien LEFLOCH
Elodie LIU

3 mai 2021

Table des matières

Introduction

Une prime d'assurance est la somme que paie le souscripteur d'un contrat à un assureur en échange de garanties définies. Il est nécessaire que cette prime reflète le risque associé au contrat. En effet, pour chaque police d'assurance, la prime est une fonction de variables de tarification qui permet alors de segmenter la population en fonction de son risque d'avoir un accident ou non.

La réalisation d'un tarif en assurance IARD s'appuie sur l'analyse de la prime pure dans le cadre d'un modèle fréquence/sévérité dans lequel l'effet des variables explicatives sur le niveau du risque est modélisé par des modèles de régression de type modèles linéaires généralisés (GLM).

On utilisera ici une approche fréquence/sévérité afin d'estimer le coût annuel d'une police d'assurance.

Nous tenterons d'estimer la prime pure avec des GLM à l'aide de la méthode Forward-Backward puis ajouterons les données de la sécurité routière afin de tester la pertinence de cet apport.

La prime pure est définie par :

$$\mathbb{E}[X] = \mathbb{E}[N] \times \mathbb{E}[B_k] \tag{1}$$

où X est la variable aléatoire représentant les coûts monétaires au risques, N la fréquence des sinistres (le nombre de sinistre pour une période donnée) et Bk la sévérité des sinistres (les montants des sinistres).

Il va alors falloir estimer la loi de la fréquence et de la sévérité à l'aide des données `pg17trainpol` et `pg17trainclaim`.

1 Exploration des données et corrélations

Tout d'abord, commençons par analyser les jeux de données que nous avons : `pg17trainpol` et `pg17trainclaim` afin de sélectionner les variables intéressantes pour la création du modèle et de déterminer le meilleur possible.

1.1 Analyse descriptive

L'analyse descriptive permet, entre autres, de déterminer les caractéristiques d'un individu moyen afin de connaître la population assurée et de vérifier la pertinence des variables tout en étudiant de façon plus ou moins succincte la corrélation entre les variables, notion primordiale lors de la modélisation.

(Mettre un histogramme de l'âge de notre population peut-être)

1.2 ACP/AFC

Pour la sélection des variables, nous effectuerons une ACP/AFC.
Nous utiliserons également un tableau de corrélation pour repérer les variables redondantes et les retirer du modèle.

On remarque plusieurs groupes de variables. Un premier groupe qui concerne les caractéristiques techniques du véhicule de l'assuré ; un second qui regroupe les valeurs liées au conducteur assuré et enfin un troisième qui regroupe les variables qui concernent l'ancienneté du véhicule.

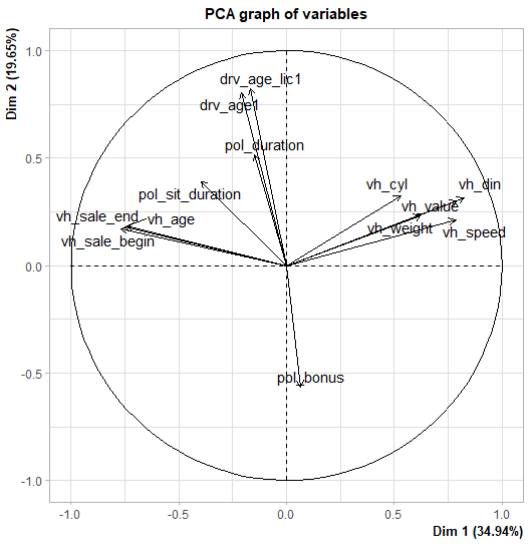


FIGURE 1 – ACP des variables

1.3 Graphique de corrélation

Nous pouvons observer les groupes de données ayant des corrélations fortes :

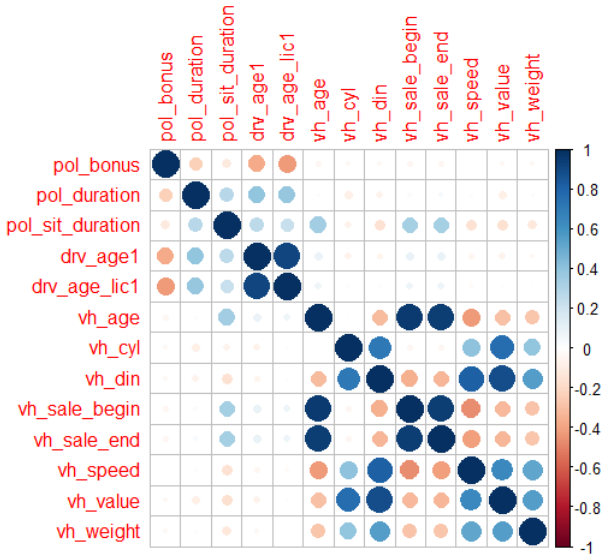


FIGURE 2 – Graphique de corrélation

Nous avons ensuite regroupé les deux jeux de données principales basées sur notre sujet pour s'intéresser à la sévérité et à la fréquence des indemnités.
Pour calculer des modèles plus rapidement, nous avons décidé de former des groupes représentatifs de population. Par exemple nous avons formé des groupes sur l'âge des conducteurs et sur des catégories de polices d'assurances qui avaient beaucoup de facteurs.

Ces aspects constituant une première approche et un préliminaire à la réalisation de la modélisation du risque automobile que nous allons à présent décrire, nous pouvons alors commencer à s'interroger sur les lois suivies par ces données.

2 GLM : fréquence et sévérité

2.1 Modélisation de la fréquence des sinistres

À l'aide de nos données, nous allons alors pouvoir proposer des modèles linéaires généralisés. Nous cherchons ici à modéliser la fréquence des sinistres. On définit alors N la variable aléatoire représentant le nombre de sinistres, à valeurs dans les entiers naturels.

2.1.1 Choix de la loi de probabilité de la fréquence

Pour écrire un GLM, nous devons d'abord choisir une loi de probabilité pour N au sein de la famille exponentielle naturelle.
Pour rappel, la variable N appartient à la famille exponentielle naturelle si sa densité de probabilité s'écrit sous la forme :

$$f_N(x) = \exp(\frac{1}{\gamma(\phi)}(x\theta - b(\theta) + c(x, \phi))$$

où c est une fonction dérivable, b une fonction trois fois dérivable et la dérivé première de b est inversible.

En effet, la loi de Poisson et la loi Binomiale Négative appartiennent toutes les deux à la famille exponentielle.

Ainsi, d'après le cours d'Actuariat 1 et de part cette approche, nous avons le choix entre entre une loi Binomiale Négative et une loi de Poisson pour modéliser N.

Comparaison de l'espérance avec la variance Nous pouvons tout d'abord utiliser un critère basé sur les moments de la fréquence.
Dans le cas d'une loi de Poisson :

$$\mathbb{V}[N] = \mathbb{E}[N]$$

Tandis que dans le cas d'une loi Binomiale Négative :

$$\mathbb{V}[N] > \mathbb{E}[N]$$

En définissant une fonction `dispersion_test()` qui permet de calculer les moments de la fréquence des sinistres, nous pouvons de tester si la variable suit une distribution de la loi de Poisson :

```
dispersion_test(TabNA$freq)
```

```
## Mean: 0.2751861
## Variance: 0.6430191
## Probability of being drawn from Poisson distribution: 0
```

Avec ce critère, on devrait alors choisir la loi Binomiale Négative pour modéliser la fréquence.

Nous pourrons alors conclure que la fréquence suit ici une loi Binomiale Négative bien qu'usuellement en assurance non-vie, la fréquence se modélise par une loi de Poisson.

2.1.2 Choix de la fonction de lien

À présent, nous modélisons le lien entre l'espérance des N_i et les variables explicatives au travers d'une fonction g inversible :

$$g(\mathbb{E}[x_i]) = x_i\beta$$

Par défaut, nous choisissons la fonction de lien canonique qui est identique pour la loi de Poisson et la loi Binomiale Négative : $g(\mu) = \log(\mu)$ et obtenons alors :

$$\mathbb{E}[N_i] = g^{-1}(x_i\beta) = \exp(x_i\beta)$$

2.1.3 Estimation des coefficients de la fréquence

Nous allons estimer les coefficients β_j par maximum de vraisemblance.

Une solution pour approcher l'estimateur du maximum de vraisemblance est d'utiliser des procédures itératives d'optimisation. mais ici les estimations seront réalisées par la fonction `glm()` pour la loi de Poisson et la fonction `glm.nb()` pour l'autre dans R. Nous allons tout d'abord mettre toutes les variables dans les modèles, nous obtenons ainsi pour la loi du Poisson :

- le modèle qu'on va utiliser pour estimer les coefficients suivant

```
summary(modelfullp)
```

```
##
## Call:
## glm(formula = freq ~ pol_bonus + pol_coverage + pol_pay_freq +
##     vh_sale_end + vh_value + vh_age + vh_cyl + vh_din + vh_fuel +
##     vh_sale_begin + vh_speed + vh_value + vh_weight + drv_sex1 +
##     drv_drv2 + drv_age1 + drv_age_lic1 + pol_sit_duration + pol_duration,
##     family = poisson(), data = Train)
```

- l'estimation du coefficient β pour chaque variable sur la colonne `Estimate`

```
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -6.607e-01  1.083e-01  -6.098 1.07e-09 ***
## pol_bonus         5.110e-02  1.050e-02   4.867 1.14e-06 ***
## pol_coverage     -1.581e-01  9.567e-03 -16.522  < 2e-16 ***
## pol_pay_freq     -8.093e-03  5.385e-03  -1.503  0.13291
## vh_sale_end      -1.266e-02  4.003e-03  -3.162  0.00157 **
## vh_value          1.160e-05  1.857e-06   6.247 4.18e-10 ***
## vh_age            9.888e-03  4.407e-03   2.244  0.02486 *
## vh_cyl            2.941e-05  3.149e-05   0.934  0.35034
## vh_din            1.523e-03  6.734e-04   2.261  0.02374 *
## vh_fuel          -1.952e-01  1.910e-02 -10.220  < 2e-16 ***
## vh_sale_begin    -1.126e-02  3.982e-03  -2.827  0.00469 **
## vh_speed         -1.891e-03  5.780e-04  -3.271  0.00107 **
## vh_weight        -1.399e-04  2.272e-05  -6.157 7.43e-10 ***
## drv_sex1         -5.049e-03  1.454e-02  -0.347  0.72845
## drv_drv2          1.700e-01  1.402e-02  12.122  < 2e-16 ***
## drv_age1         -6.348e-02  1.525e-02  -4.164 3.13e-05 ***
## drv_age_lic1      5.896e-02  8.863e-03   6.652 2.89e-11 ***
## pol_sit_duration -4.084e-02  6.517e-03  -6.267 3.69e-10 ***
## pol_duration     -1.556e-02  2.480e-03  -6.275 3.49e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

et pour la loi Binomiale Négative :

- nous avons le modèle

```
summary(modelfullnb)
```

```
##
## Call:
## glm.nb(formula = freq ~ pol_bonus + pol_coverage + pol_pay_freq +
##     vh_sale_end + vh_value + vh_age + vh_cyl + vh_din + vh_fuel +
##     vh_sale_begin + vh_speed + vh_value + vh_weight + drv_sex1 +
##     drv_drv2 + drv_age1 + drv_age_lic1 + pol_sit_duration + pol_duration,
##     data = Train, init.theta = 0.23244923, link = log)
##
```

- l'estimation du coefficient β qui est également sur la colonne `Estimate`

```
## Coefficients:
##                   Estimate Std. Error z value Pr(>|z|)
## (Intercept)      -6.144e-01  1.633e-01  -3.762 0.000169 ***
## pol_bonus         3.899e-02  1.599e-02   2.439 0.014720 *
## pol_coverage     -1.551e-01  1.340e-02 -11.574  < 2e-16 ***
## pol_pay_freq     -5.228e-03  8.048e-03  -0.650 0.515943
## vh_sale_end      -1.396e-02  5.722e-03  -2.439 0.014710 *
## vh_value          1.154e-05  2.950e-06   3.910 9.23e-05 ***
## vh_age            1.247e-02  6.252e-03   1.995 0.046083 *
## vh_cyl            3.214e-05  4.685e-05   0.686 0.492691
## vh_din            1.896e-03  1.046e-03   1.813 0.069897 .
## vh_fuel          -1.837e-01  2.880e-02  -6.377 1.81e-10 ***
## vh_sale_begin    -1.185e-02  5.628e-03  -2.105 0.035291 *
## vh_speed         -2.267e-03  8.832e-04  -2.567 0.010265 *
## vh_weight        -1.247e-04  3.488e-05  -3.576 0.000348 ***
## drv_sex1         -9.734e-03  2.174e-02  -0.448 0.654295
## drv_drv2          1.862e-01  2.128e-02   8.750  < 2e-16 ***
## drv_age1         -9.019e-02  2.250e-02  -4.009 6.09e-05 ***
## drv_age_lic1      6.894e-02  1.315e-02   5.241 1.60e-07 ***
## pol_sit_duration -4.572e-02  9.175e-03  -4.984 6.24e-07 ***
## pol_duration     -1.667e-02  3.734e-03  -4.465 7.99e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

2.1.4 Test et sélection des variables du modèle

La sélection de modèle peut être vue comme la recherche du modèle optimal parmi toutes les possibilités.
Pour sélectionner le meilleur modèle, nous allons nous appuyer sur un critère qui permet de comparer les modèles entre eux comme par exemple le critère AIC.

Nous utilisons alors une méthode pas-à-pas. Trois méthodes sont souvent utilisées :

- **Méthode Forward** : cette méthode part du modèle réduit à l'intercept et on le compare utilisant le critère AIC à les modèles contenant une variable explicatives. Nous choisissons alors le meilleur modèle ayant la plus petite valeur d'AIC. On ajoute ensuite une variable parmi les autres covariables et on choisit de nouveau le meilleur modèle selon le critère. On s'arrête quand l'ajout d'une variable n'améliore pas la valeur AIC.
- **Méthode Backward** : cette méthode a une stratégie inverse de la précédente, elle consiste à partir du modèle complet et on enlève une à une les variables en comparant les modèles deux à deux avec le critère AIC.
- **Méthode Both (Forward-Backward)** : cette méthode est un mélange des deux autres. À chaque étape, on ajoute ou enlève une variable et on choisit le meilleur modèle, ensuite on recommence.

Ici, nous décidons d'appliquer la méthode Forward-Backward en utilisant la commande `step`. Cependant, nous avons aussi besoin de réaliser des analyses de la variance(anova). En effet, la

fonction `anova()` avec `test='Chisq'` sur R permet d'étudier si chaque covariable a un effet significatif pour expliquer la variable réponse.

Ainsi le résultat de plusieurs sélections des variables pour le modèle avec la loi de Poisson est :

```
summary(modelfinalp)

##
## Call:
## glm(formula = freq ~ pol_bonus + pol_coverage + vh_sale_end +
##     vh_value + vh_din + vh_fuel + vh_speed + vh_weight + drv_drv2 +
##     drv_age_lic1 + pol_sit_duration + pol_duration, family = poisson(),
##     data = TabNA)
##
```

Tandis que pour le modèle avec la loi Binomiale Négative nous obtenons :

```
summary(modelfinalnb)

##
## Call:
## glm.nb(formula = freq ~ vh_sale_end + vh_value + pol_coverage +
##     vh_fuel + drv_drv2 + pol_sit_duration + pol_duration + vh_age +
##     vh_weight + drv_age_lic1 + drv_age1 + pol_bonus, data = TabNA,
##     init.theta = 0.2303273661, link = log)
##
```

Phrase de conclusion ?

2.1.5 Sélection du meilleur modèle GLM

Après avoir obtenu deux modèles GLM pour la fréquence, nous allons modéliser alors la fréquence des sinistres grâce au meilleur modèle GLM chosisi selon, soit le critère de la déviance, soit le critère AIC, soit le critère BIC.

Critère AIC et BIC Ce critère consiste à calculer et comparer les valeurs d'AIC et BIC de ces deux modèle en notant L la vraisemblance maximisée, k le nombre de paramètre du modèle et n le nombre d'individus :

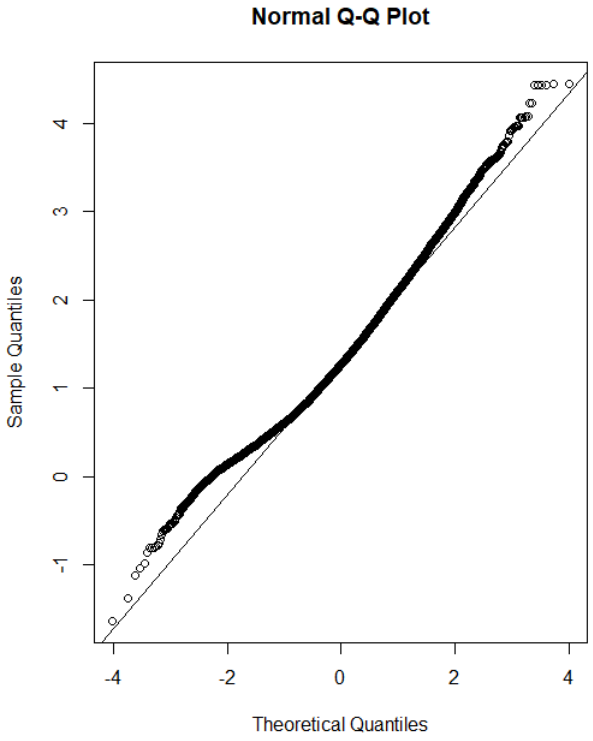
- $AIC = -2log(L) + 2k$
- $BIC = -2log(L) + klog(n)$

Celui ayant des valeurs plus petite est le meilleur modèle pour modéliser la fréquence.

	AIC	BIC
Poisson	148421	148544.9
Binomiale négative	126776.1	126909.6

Nous observons que le modèle avec la loi Binomiale négative admet le plus petit AIC et BIC, il est donc le meilleur modèle.

Comparaison des erreurs Nous allons donner les anomalies que nous allons repérer sur le modèle fait et les limites du tarificateur que l'on a créée.
On peut observer la pertinence du modèle créé avec un diagramme quantile-quantile en regardant les log-résidus :



Les log-résidus semblent donc assez bien repartis normalement.

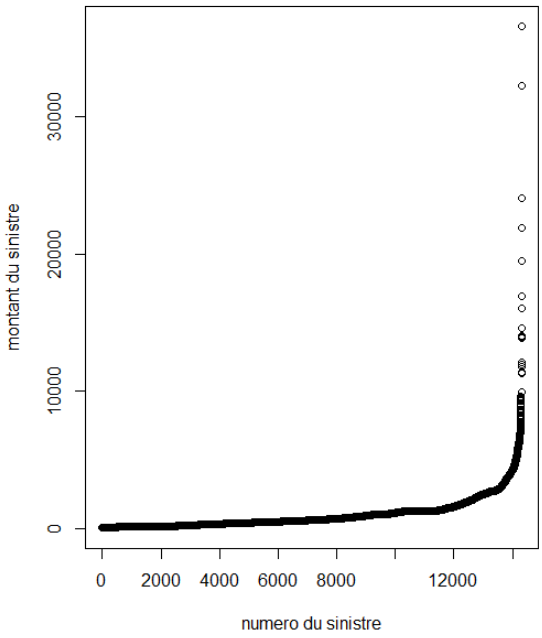
2.2 Modélisation de la sévérité des sinistres

Nous allons emprunter des références et nous allons choisir la modélisation de la sévérité en fonction de ce que nous allons y trouver.

La sévérité étant une variable continue, on a le choix entre la loi Gamma et la loi inverse-gaussienne comme loi du coût de sinistres. En effet, ces deux lois appartiennent à la famille

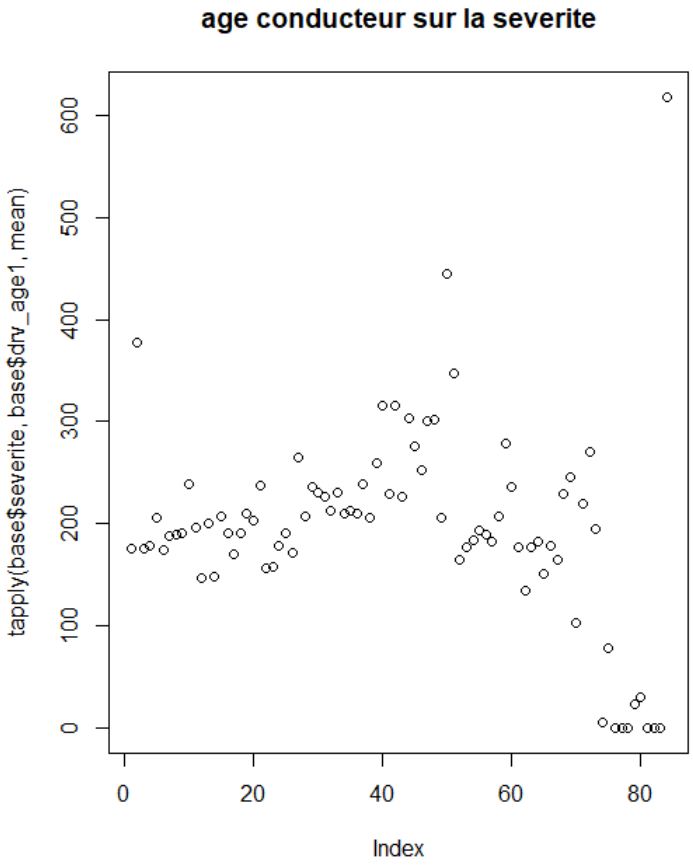
exponentielle. De plus, la loi Gamma et la loi inverse-gaussienne sont à support dans $]0, \infty[$, nous décidons d'extraire les polices ayant un coût de sinistre strictement positif.

Avant de commencer la modélisation, nous observons des sinistres ayant un montant exceptionnellement élevé, ceci pouvant affecter le résultat. Nous les appelons **les sinistres graves**. Nous décidons donc de traiter ces sinistres graves par la méthode par écrêtement.
Nous regardons la forme de nos données de sévérité à l'aide du graphique suivant :



2.2.1 Étude de corrélation entre les coûts de sinistres et les variables explicatives

Nous nous intéressons à présent aux corrélations de certaines variables avec la sévérité.
Avec ce premier graphique ci-dessous, nous voyons que l'âge du conducteur est une variable qui a sûrement un rôle à jouer dans l'explication de la sévérité d'un accident. Ainsi, nous pouvons déduire que les variables corrélées à l'âge du conducteur seront indispensables dans le modèle que nous allons proposer.



2.2.2 Estimation des coefficients du coût de sinistres

Nous choisissons la fonction logarithme comme fonction de lien pour la loi Gamma et la loi inverse-gaussienne. Bien que la fonction de lien canonique soit la fonction inverse pour la loi Gamma et $x \longmapsto \frac{1}{x^2}$ pour l'autre loi, la fonction logarithme est utilisée plus fréquemment.
Les coefficients de la régression β_j sont inconnus et doivent être estimés. Ceci sera effectué de la même façon que pour la fréquence.
Nous allons tout d'abord mettre toutes les variables dans le modèle, nous obtenons ainsi le résultat pour la loi Gamma suivant :

- le modèle qu'on va utiliser pour estimer les coefficients

```
summary(modelfullg)
```

```
##
## Call:
## glm(formula = claim_amount ~ pol_coverage + pol_pay_freq + pol_bonus +
##     pol_duration + pol_sit_duration + drv_drv2 + drv_sex1 + drv_age_lic1 +
##     drv_age_lic2 + vh_fuel + vh_type + vh_din + vh_sale_end +
##     vh_value + vh_age, family = Gamma(link = log), data = Trainsev)
##
```

- l'estimation du coefficient β pour chaque variable sur la colonne `Estimate`

```
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       6.623e+00  1.180e-01  56.103  < 2e-16 ***
## pol_coverage     -6.619e-02  1.657e-02  -3.995 6.52e-05 ***
## pol_pay_freq      4.100e-03  8.927e-03   0.459  0.64608
## pol_bonus         1.606e-02  1.679e-02   0.957  0.33873
## pol_duration     -2.673e-03  4.122e-03  -0.648  0.51677
## pol_sit_duration  1.221e-02  1.086e-02   1.124  0.26089
## drv_drv2         -2.111e-02  3.468e-02  -0.609  0.54280
## drv_sex1         -2.264e-02  2.416e-02  -0.937  0.34873
## drv_age_lic1      2.259e-02  9.369e-03   2.411  0.01591 *
## drv_age_lic2      3.887e-04  9.524e-04   0.408  0.68318
## vh_fuel           7.459e-02  2.707e-02   2.756  0.00586 **
## vh_type           9.455e-02  4.165e-02   2.270  0.02321 *
## vh_din           -5.882e-04  7.617e-04  -0.772  0.43999
## vh_sale_end      -7.821e-03  6.198e-03  -1.262  0.20707
## vh_value          9.082e-06  3.029e-06   2.999  0.00272 **
## vh_age           -3.788e-03  6.036e-03  -0.628  0.53032
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Gamma family taken to be 1.396093)
##
##     Null deviance: 11469  on 11451  degrees of freedom
## Residual deviance: 11243  on 11436  degrees of freedom
## AIC: 180765
```

ainsi pour la loi inverse-gaussienne :

- nous avons le modèle

```
summary(modelfullig)
```

```
##
## Call:
## glm(formula = claim_amount ~ pol_coverage + pol_pay_freq + pol_bonus +
##     pol_duration + pol_sit_duration + drv_drv2 + drv_sex1 + drv_age_lic1 +
##     drv_age_lic2 + vh_fuel + vh_type + vh_din + vh_sale_end +
##     vh_value + vh_age, family = inverse.gaussian(link = "log"),
##     data = Trainsev)
```

- l'estimation du coefficient β qui est également sur la colonne `Estimate`

```
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)       6.615e+00  1.161e-01  56.955  < 2e-16 ***
## pol_coverage     -6.174e-02  1.570e-02  -3.932 8.47e-05 ***
## pol_pay_freq      6.685e-03  8.999e-03   0.743  0.45761
## pol_bonus         1.502e-02  1.692e-02   0.888  0.37479
## pol_duration     -2.825e-03  4.156e-03  -0.680  0.49668
## pol_sit_duration  1.025e-02  1.072e-02   0.957  0.33861
## drv_drv2         -1.646e-02  3.472e-02  -0.474  0.63554
## drv_sex1         -2.448e-02  2.430e-02  -1.008  0.31371
## drv_age_lic1      2.417e-02  9.443e-03   2.559  0.01050 *
## drv_age_lic2      3.054e-04  9.688e-04   0.315  0.75262
## vh_fuel           7.610e-02  2.760e-02   2.757  0.00584 **
## vh_type           8.423e-02  3.951e-02   2.132  0.03304 *
## vh_din           -5.264e-04  7.853e-04  -0.670  0.50267
## vh_sale_end      -8.569e-03  6.048e-03  -1.417  0.15658
## vh_value          8.792e-06  3.191e-06   2.755  0.00588 **
## vh_age           -1.743e-03  5.849e-03  -0.298  0.76568
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for inverse.gaussian family taken to be 0.00144102)
##
##     Null deviance: 19.915  on 11451  degrees of freedom
## Residual deviance: 19.692  on 11436  degrees of freedom
## AIC: 179561
```

2.2.3 Sélection des variables

Après une première sélection de variables réalisée précédemment, il est possible qu'il y ait encore des variables qui ne soient pas significatives dans le modèle. Nous décidons alors d'effectuer encore les mêmes procédures de sélection des variables pour le coût des sinistres que nous avons pratiqué pour modéliser la fréquence.

Le résultat est obtenu grâce à la commande `step`, donnant alors les variables explicatives pour la loi Gamma suivantes :

- `pol_coverage`
- `vh_fuel`
- `vh_sale_end`
- `pol_bonus`
- `drv_age_lic1`
- `vh_value`
- `vh_type`
- `drv_drv2`
- `drv_age_lic2`

Tandis que pour l'autre loi nous avons les covariables suivantes :

- `pol_coverage`
- `vh_fuel`
- `vh_sale_end`
- `pol_bonus`
- `drv_age_lic1`
- `vh_value`

En appliquant ensuite la commande `anova`, nous pouvons retirer des variables qui ne sont pas significatives dans les modèles. Nous obtenons ainsi le modèle avec la loi Gammma :

```
summary(modelfinalg)
```

```
##
## Call:
## glm(formula = claim_amount ~ pol_coverage + vh_sale_end + vh_type +
##     vh_value, family = Gamma(link = log), data = sev_totale)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.2093  -0.9879  -0.4533   0.2609   8.7882
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   6.792e+00  8.111e-02  83.737  < 2e-16 ***
## pol_coverage -7.929e-02  1.500e-02  -5.287 1.26e-07 ***
## vh_sale_end  -8.525e-03  2.378e-03  -3.585 0.000338 ***
## vh_type       9.631e-02  3.667e-02   2.626 0.008640 **
## vh_value      4.729e-06  1.167e-06   4.051 5.14e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Gamma family taken to be 1.525207)
##
##     Null deviance: 14467  on 14314  degrees of freedom
## Residual deviance: 14239  on 14310  degrees of freedom
## AIC: 226084
##
## Number of Fisher Scoring iterations: 6
```

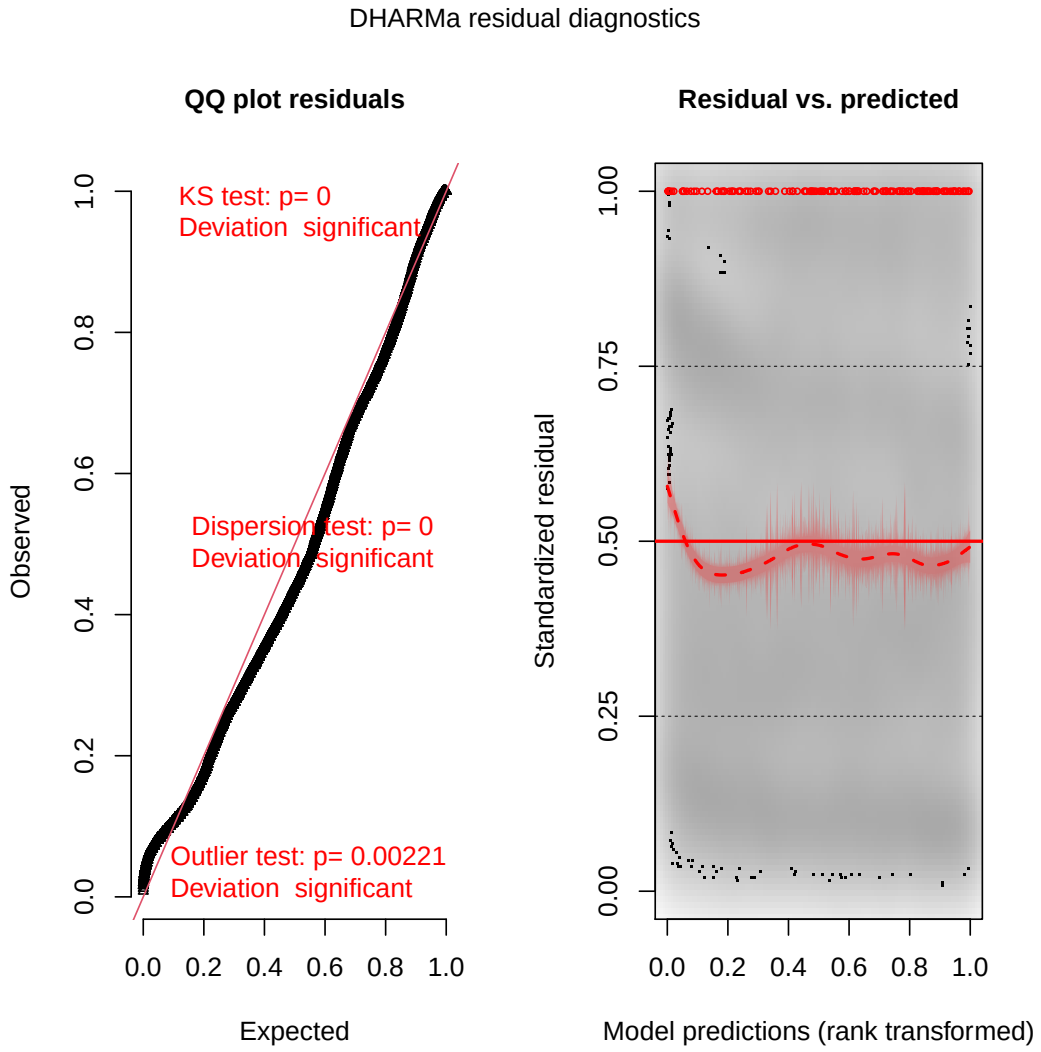
Tandis que pour la loi inverse-gaussienne :

```r
summary(modelfinalig)
```

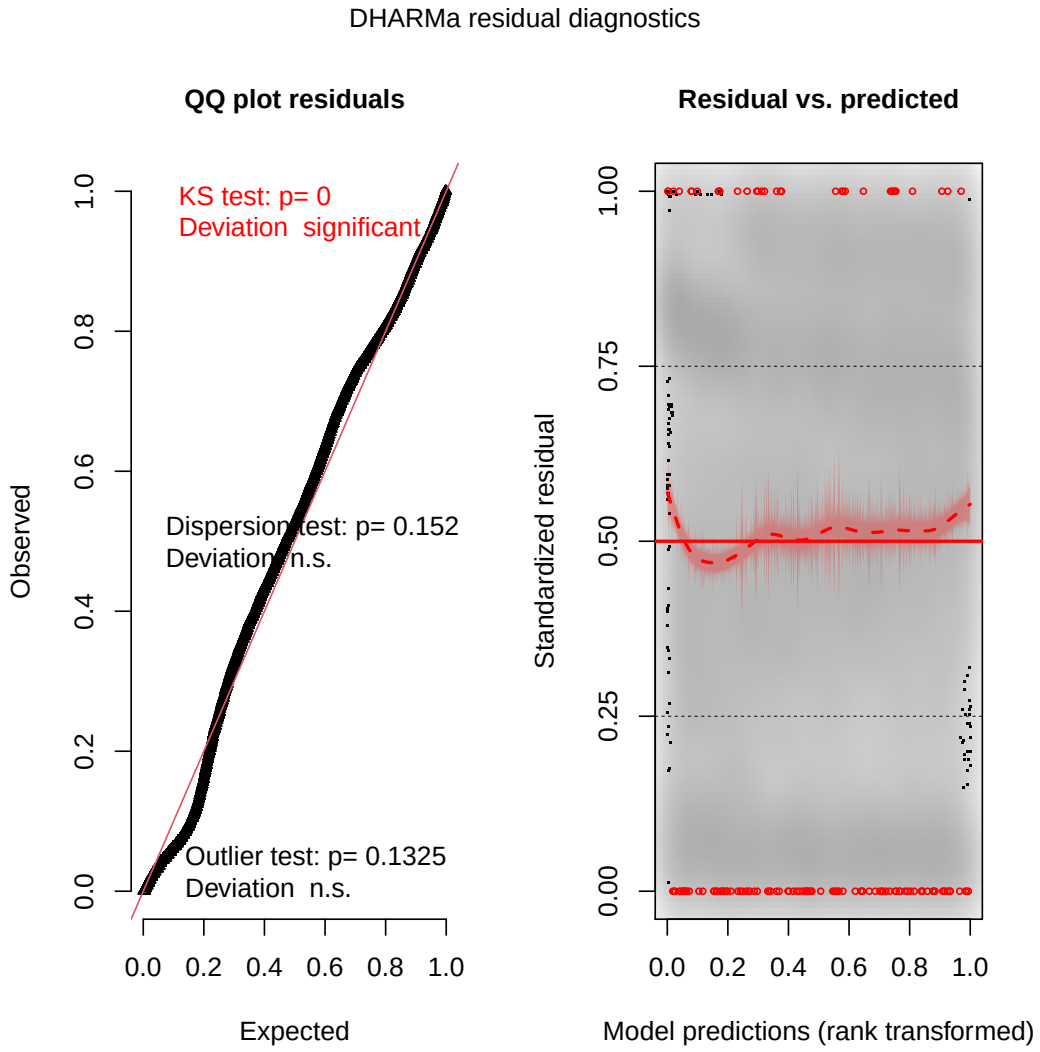
```
##
## Call:
## glm(formula = claim_amount ~ pol_coverage + vh_fuel + vh_sale_end +
##     vh_value, family = inverse.gaussian(link = "log"), data = sev_totale)
##
## Deviance Residuals:
##       Min         1Q     Median         3Q        Max
## -0.158765  -0.040267  -0.015549   0.008008   0.214329
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   6.798e+00  5.329e-02 127.574  < 2e-16 ***
## pol_coverage -7.591e-02  1.405e-02  -5.403 6.64e-08 ***
## vh_fuel       1.022e-01  2.337e-02   4.372 1.24e-05 ***
## vh_sale_end  -9.159e-03  2.289e-03  -4.001 6.34e-05 ***
## vh_value      6.579e-06  1.337e-06   4.920 8.75e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for inverse.gaussian family taken to be 0.001567052)
##
##     Null deviance: 24.932  on 14314  degrees of freedom
## Residual deviance: 24.686  on 14310  degrees of freedom
## AIC: 224374
##
## Number of Fisher Scoring iterations: 9
```

2.2.4 Sélection du meilleur modèle GLM

Nous obtenons un GLM pour la loi Gamma prenant en compte 4 variables. Pour le vérifier nous allons utiliser la fonction `simulateResiduals` de la librairie DHARMa qui permet d'obtenir :



Le premier graphique(à gauche) est un QQ-plot qui permet de présenter l'adéquation des résidus à une loi normale centrée réduite
Nous obtenons aussi pour le modèle avec la loi inverse-gaussienne :


DHARMa residual diagnostics
QQ plot residuals
KS test: p= 0
Deviation significant
Dispersion test: p= 0.152
Deviation n.s.
Outlier test: p= 0.1325
Deviation n.s.
Observed
Expected
0.0
0.2
0.4
0.6
0.8
1.0
Residual vs. predicted
Standardized residual
Model predictions (rank transformed)
0.00
0.25
0.50
0.75
1.00

3 Calcul de la prime pure pour les polices étudiées

Pour le calcul de la prime pure, nous allons utiliser les valeurs des espérances estimées de la fréquence des accidents (notée N) et de leur sévérité(notée B). Nous allons utiliser les coefficients de la fréquence et de la sévérité que nous avons obtenus avec les modèles construits précédemment.
Les variables explicatives des deux modèles sont :

i	x_i
1	vh_sale_end
2	vh_value
3	pol_coverage
4	vh_fuel
5	drv_drv2
6	pol_sit_duration
7	pol_duration
8	vh_age
9	vh_weight
10	drv_age_lic1
11	drv_age1
12	pol_bonus

TABLE 1 – Variables explicatives pour modéliser la fréquence

i	x_i
1	pol_coverage
2	vh_sale_end
3	vh_type
4	vh_value

TABLE 2 – Variables explicatives pour modéliser la sévérité

En utilisant la fontion logarithme qu'on a appliqué comme fonction de lien pour expliquer la variable N, on a la relation de son espérence suivante :

$$\mathbb{E}[N] = \exp\{\beta_0 + \sum_{i=1}^{12} \beta_i \times x_i\} \tag{2}$$

De même, on a choisi la même fonction de lien pour expliquer la variable B :

$$\mathbb{E}[B] = \exp\{\gamma_0 + \sum_{j=1}^{4} \gamma_j \times x_j\} \tag{3}$$

En appliquant les relations (2) et (3), nous avons la formule du calcul de la prime pure suivante :

$$\begin{aligned}\mathbb{E}[X] &= \mathbb{E}[N] \times \mathbb{E}[B] \\ &= \exp\{\beta_0 + \sum_{i=1}^{12} \beta_i \times x_i\} \times \exp\{\gamma_0 + \sum_{j=1}^{4} \gamma_j \times x_j\}\end{aligned} \tag{4}$$

Nous avons donc le résultat suivant :

```
summary(primepure$prime_pure)
```

```
##     Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    18.86  180.44  260.06  273.35  351.62 2242.97
```

FIGURE 3 – résultat de calcul de la prime pure

La moyenne des primes pures étant 273.35 nous parait logique.

```
hist(primepure$prime_pure,main='Prime pure',xlab='montant de la prime')
```

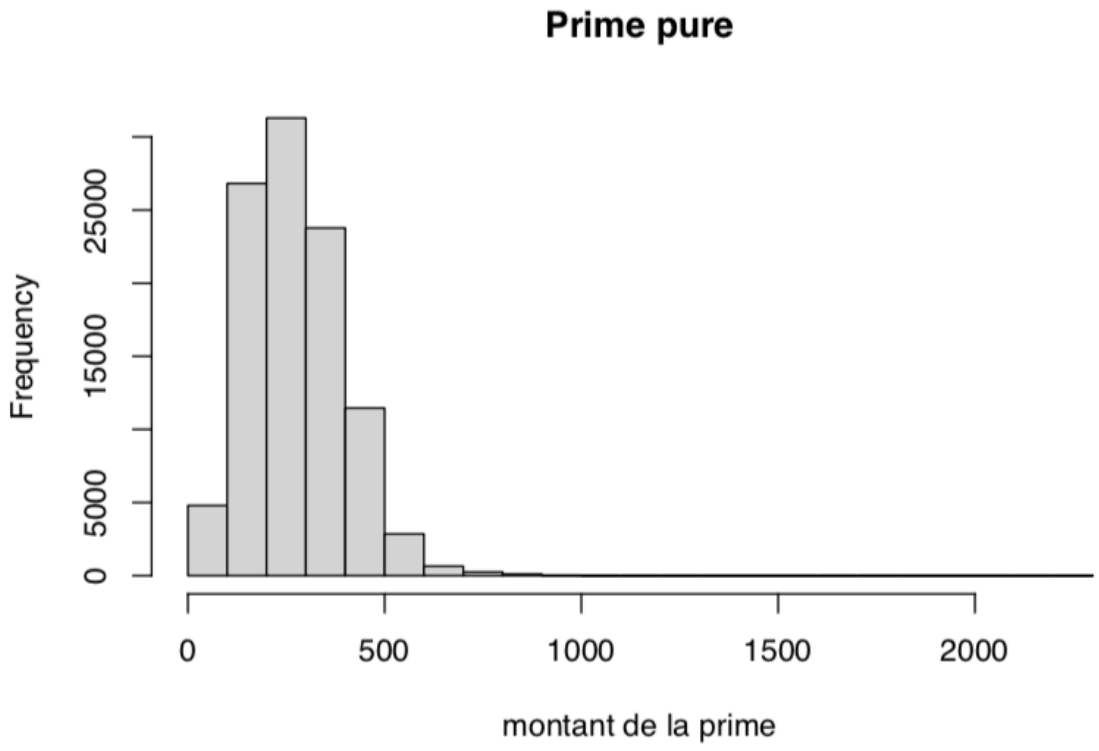


FIGURE 4 – Histogramme des montants de prime pure

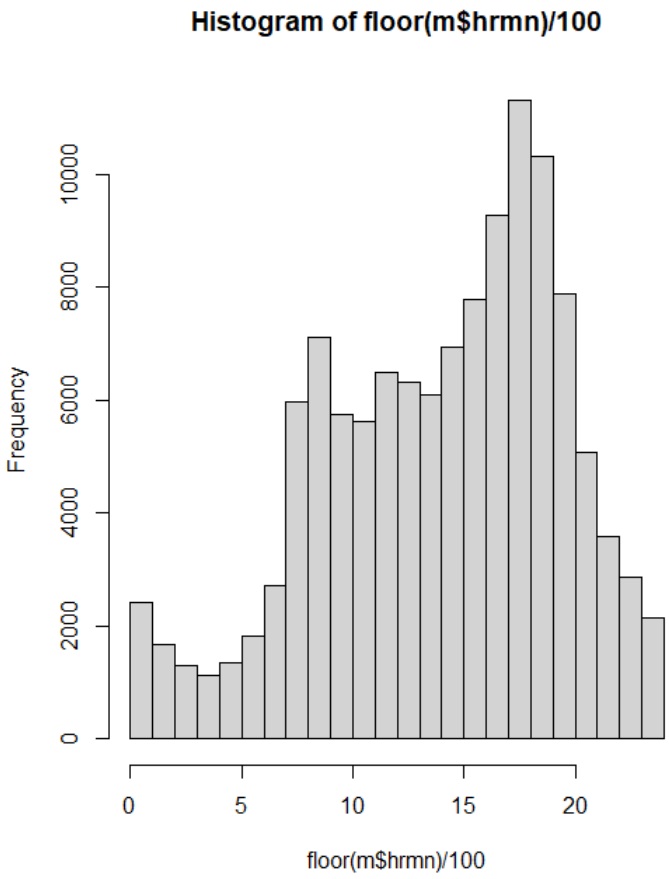
4 Ajout des données ONISR

Les données ONISR sont les données gouvernementales de l'Observatoire national interministériel de la sécurité routière, ONISR (2018) où tout accident corporel survenant en France entre 2005 et 2017 y est répertoriée. L'observatoire met à disposition 4 fichiers annuels sur les caractéristiques de l'accident, le lieux des accidents, les véhicules impliquées et les usagers impliqués.

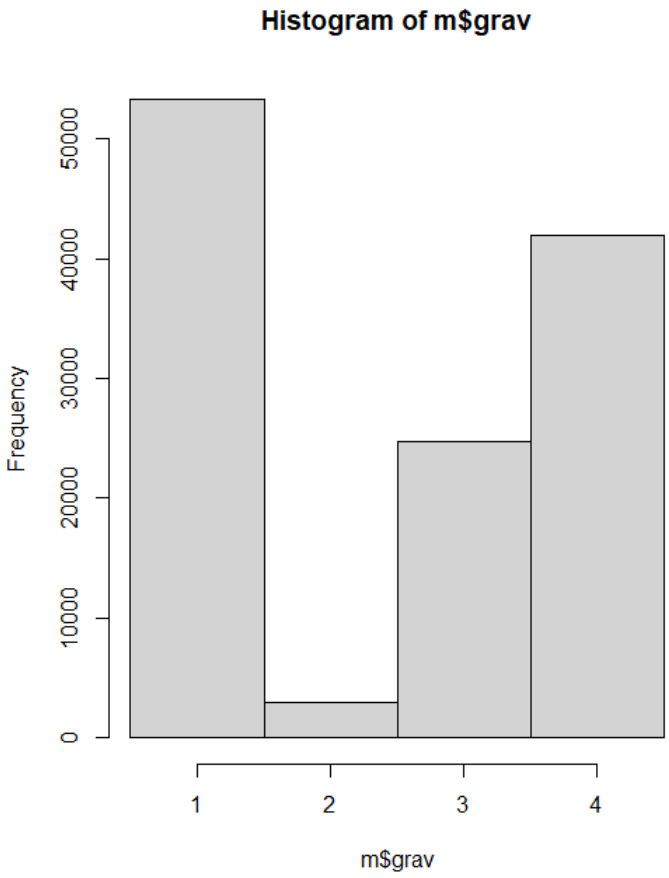
Nous testerons dans cette partie la pertinence de l'apport de ces données nationale à notre GLM.
Nous allons employer les mêmes méthodes que dans les sections précédentes puis comparerons, à la fin, les tarifs obtenus.

4.1 Exploration des données ONISR et premières remarques

Nombre de sinistres par heure dans une journée :



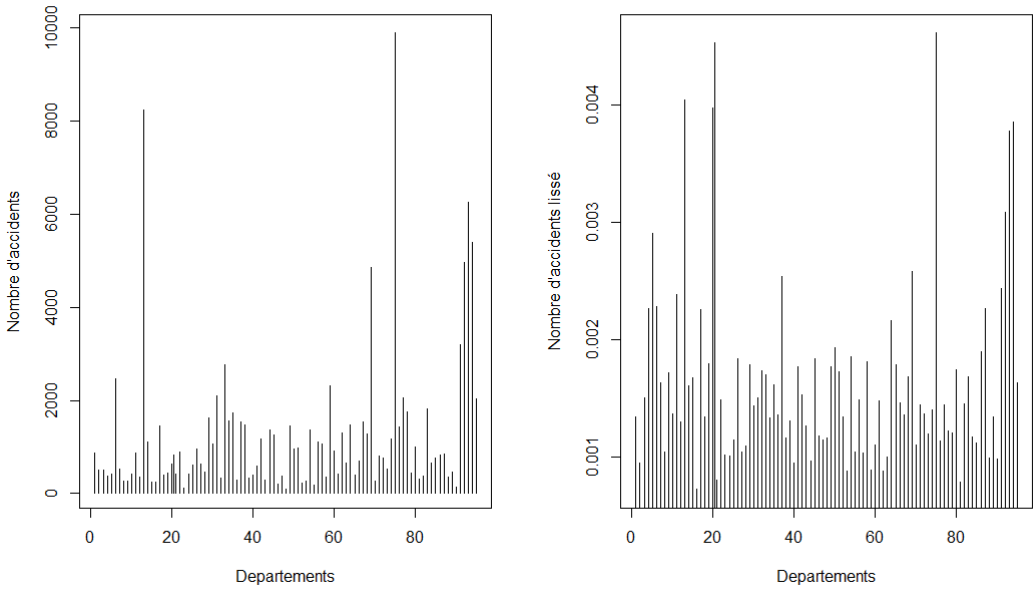
Nombre de sinistres par gravité :



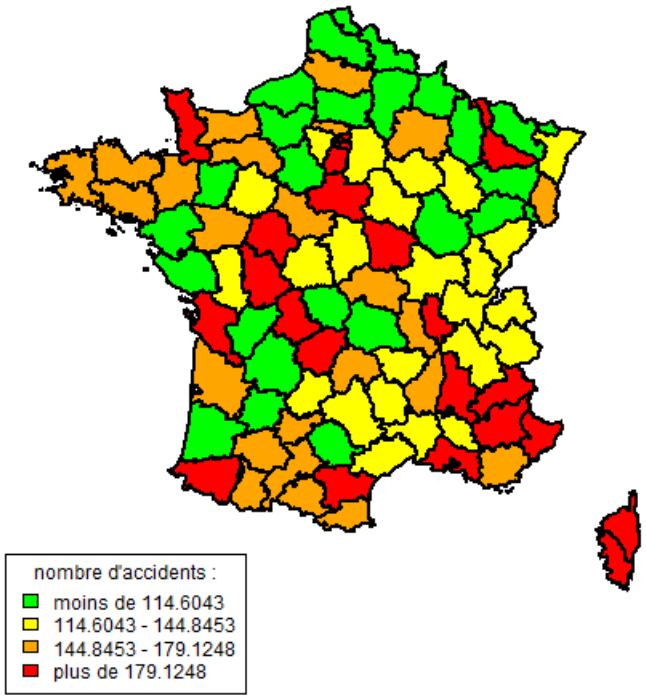
4.2 Modélisation de la sévérité et de la fréquence sur les données ONISR

L'idee que nous allons mettre en place pour incorporer les nouvelles données apportées par la base ONISR est le partitionnement des données. Il s'agit de définir des groupes homogènes à l'aide des informations dont on dispose sur ces données. Comme nous disposons des codes postaux des communes des assures dans la base de données `pg17testyear1`, nous allons organiser les départements métropolitains en groupes, aussi appelés 'clusters'. Nous auront donc de nouvelles variables qui affineront nos modèles linéaires généralisés exposes dans la première partie.
Le but va être de regrouper les départements en fonction de deux scores différents. Le premier

sera un score reflétant la fréquence de sinistres et le second reflétera la sévérité de ces sinistres. Pour créer ces scores, nous prenons différentes variables disponibles dans les données ONISR qui expliquent soit la fréquence de sinistres, soit leur gravite.
Pour la fréquence :
Pour lisser les différences entre les départements, notamment au niveau de la densité de population, nous avons importé les recensements de population par département afin d'observer de manière plus objective les départements où sont comptés le plus de sinistres. Nous avons donc importe une base de données comportant le nombre d'habitant par département.



On observe notamment le cas de la Corse : comparé à sa population, le nombre d'accidents y est très élevé. Aussi, on observe une nette corrélation entre les zones très urbaines et le nombre d'accidents, ce qui semble assez cohérent. D'une façon plus visuelle, nous avons :

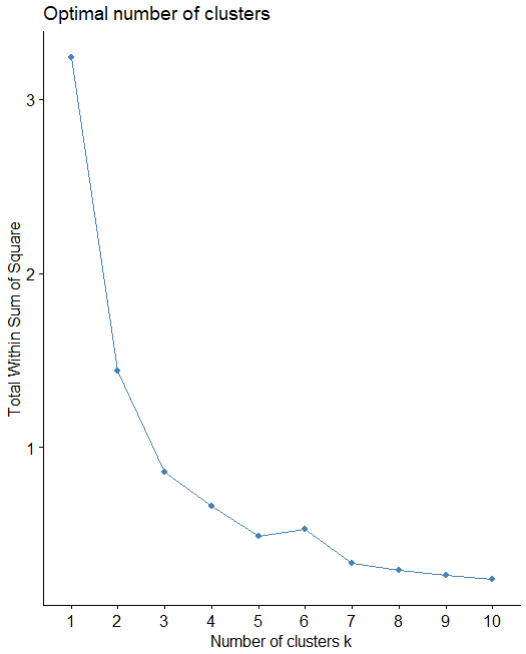


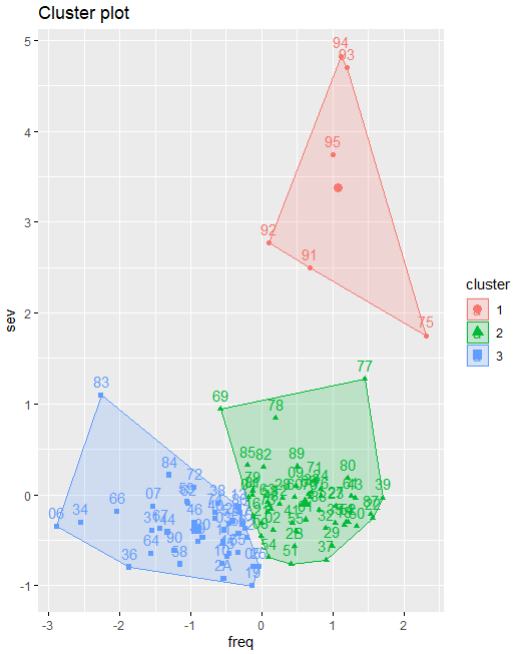
Comme le nombre d'habitant par département joue un rôle important dans le nombre d'accidents dénombrés, il est intéressant de créer un score de population nomme `scorepop` qui permet d'attribuer une valeur dans [0,1]. On remarque que les zones très urbaines ont tendance a avoir un nombre d'accidents plus élevé que les départements plus ruraux. On crée donc un score lie aux sinistres en agglomération que l'on a nomme `scoreagg`. Plus ce score est proche de 1, plus le nombre de sinistres arrivés en agglomération représente une grosse part dans le nombre de sinistres total. Cela va permettre d'avoir une certaine représentation de l'urbanisation des départements.
Nous prenons également en compte les conditions de surface de sol lorsque le sinistre est arrive. En effet, après avoir retire les données en condition 'normales', on s'intéresse aux conditions liées a la pluie et a la neige. On a donc un score de surface nomme `scoresurf`.
Nous utilisons les données sur les intersections. Le score lie aux intersections est nomme `scoreint`. Nous avons regroupe tous les types d'intersections en un seul groupe par soucis de simplicité. Les formes d'intersections ne semblaient pas importantes.

Nous normalisons les scores afin de ne pas avoir de problème de poids dans les scores. En sommant puis normalisant les scores, on obtient un score `scorefreq` qui représente la fréquence de sinistres par département. Plus un département est proche de 1 plus il a de facteurs pouvant augmenter le nombre de sinistres.

Pour la sévérité :
Dans les données ONISR, cinq variables expliquent la sévérité des sinistres : le type d'obstacle touche `obsm`, le type de collision `col`, le point de choc initial `choc`, la localisation de l'accident (en ou hors agglomération) `agg` et enfin la gravite de blessure de l'usager `grav`.
Nous obtenons le score de severite `scoresev`, par somme et normalisation des scores : `scorecoll`, `scoreobsm`, `scorechoc`, `scoreagg`, `scoregrav`.

Ainsi, pour chaque département nous avons un score de fréquence et de sévérité. Grâce a ces scores, nous pouvons créer des clusters de départements, on peut y distinguer 3 groupes clairs :





Pour inclure ces groupes dans les modèles linéaires généralisés on va créer des clusters de fréquence et des cluster de sévérité. Ça permettra d'avoir une meilleur précision dans les clusters.

Dep	clusterfreq$cluster	clustersev$cluster
01	1	3
02	1	3
03	2	3
04	2	3
05	3	2
06	3	2
07	2	3
08	1	2
09	1	3
10	2	2
11	2	3
12	1	3

FIGURE 5 – Exemple de clusters par departement

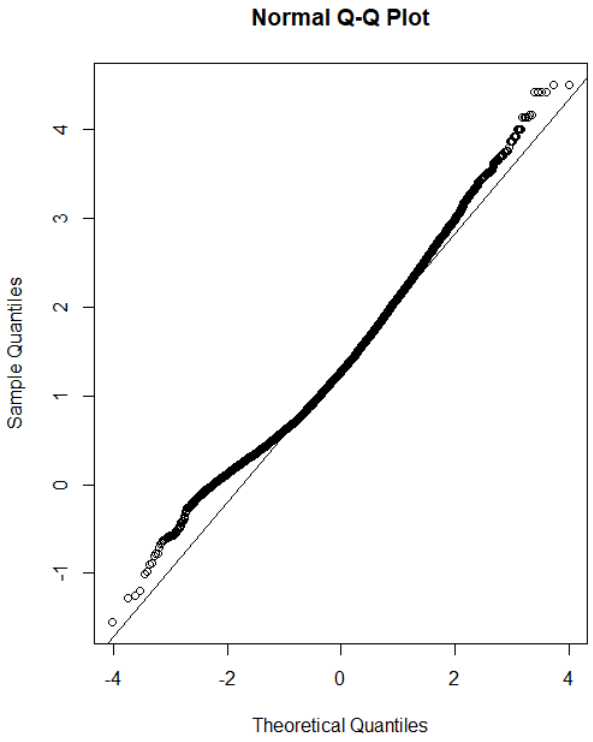
On a créé deux variables explicatives supplémentaires : `ClusterFreq` et `ClusterSev`. On intègre `ClusterFreq` dans le modèle de fréquence obtenu précédemment et on a donc les variables explicatives :

- `vh_sale_end`
- `vh_value`
- `vh_fuel`
- `drv_drv2`
- `pol_sit_duration`
- `pol_duration`
- `vh_age`
- `vh_weight`
- `drv_age_lic1`
- `drv_age_1`
- `pol_bonus`
- `ClusterFreq`

On verifie que le modele nouvellement cree est toujours correct :

	Df	Deviance	Resid. Df	Resid. Dev	Pr(>Chi)	
NULL			102079	48983		
vh_sale_end	1	721.58	102078	48261	< 2.2e-16	***
vh_value	1	274.23	102077	47987	< 2.2e-16	***
pol_coverage	1	120.78	102076	47866	< 2.2e-16	***
vh_fuel	1	128.16	102075	47738	< 2.2e-16	***
drv_drv2	1	91.89	102074	47646	< 2.2e-16	***
pol_sit_duration	1	50.52	102073	47595	1.178e-12	***
pol_duration	1	26.55	102072	47569	2.566e-07	***
vh_age	1	10.63	102071	47558	0.0011150	**
vh_weight	1	15.62	102070	47543	7.738e-05	***
drv_age_lic1	1	12.86	102069	47530	0.0003362	***
drv_age1	1	25.15	102068	47505	5.316e-07	***
pol_bonus	1	8.18	102067	47496	0.0042254	**
ClusterFreq	1	24.90	102066	47472	6.047e-07	***

FIGURE 6 – anova pour la fréquence



On integre `ClusterSev` dans le modèle de sévérité obtenu précédemment et on a donc les variables explicatives :

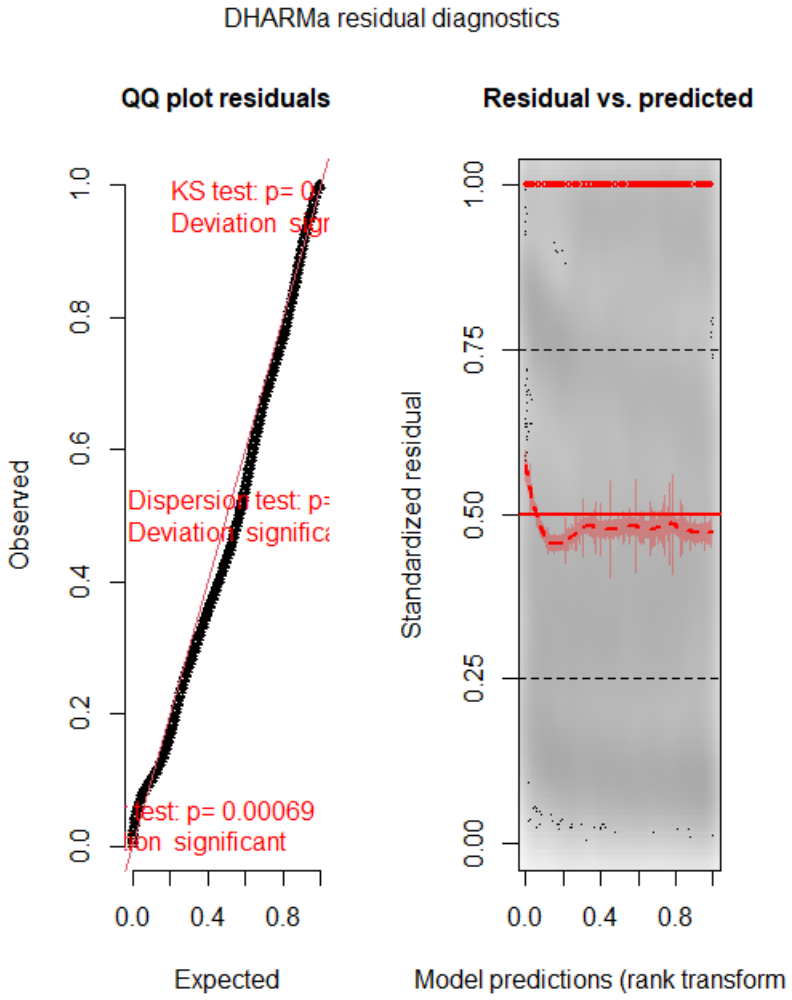
- `pol_coverage`
- `vh_sale_end`
- `vh_type`
- `vh_value`
- `ClusterSev`

Par des tests anova, on voit bien que le modèle est plus fin que le précèdent grâce aux nou-

	Df	Deviance	Resid. Df	Resid. Dev	Pr(>Chi)	
NULL			14314	14467		
pol_coverage	1	160.946	14313	14306	< 2.2e-16	***
vh_sale_end	1	30.790	14312	14275	6.792e-06	***
vh_type	1	9.901	14311	14265	0.01072	*
vh_value	1	25.932	14310	14239	3.629e-05	***
ClusterSev	1	29.791	14309	14209	9.577e-06	***

FIGURE 7 – anova pour la sévérité

velles variables créées a partir des données ONISR.
On verifie que le modele nouvellement cree est toujours correct :



On peut calculer la prime pure, on obtient la répartition suivante :

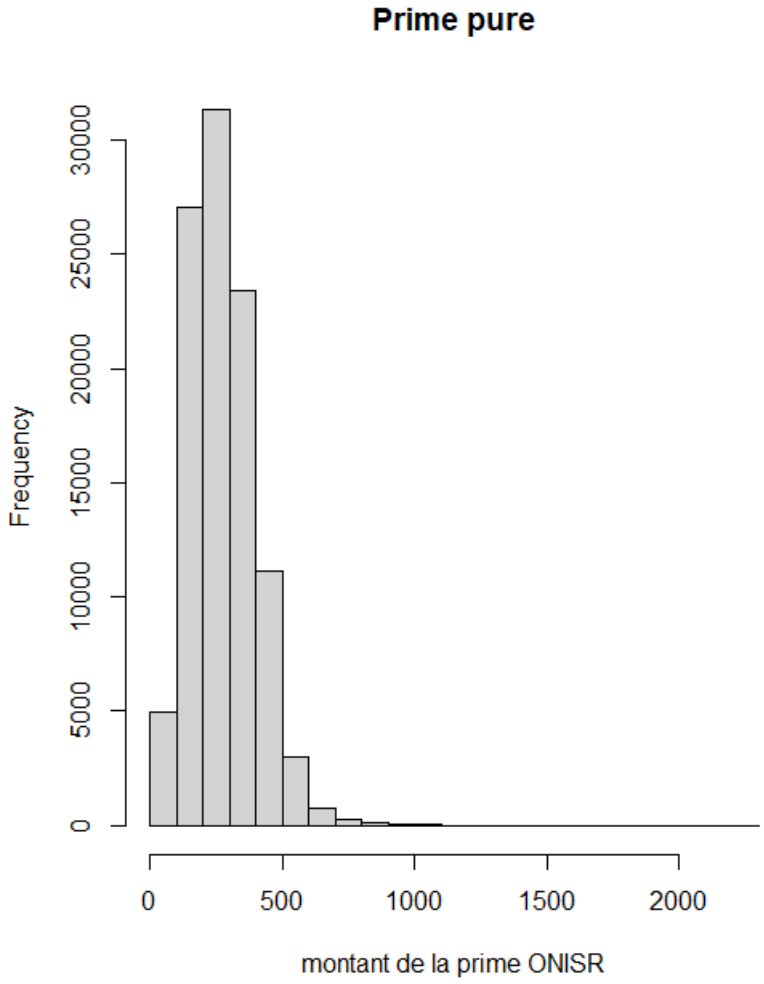


FIGURE 8 – Histogramme des montants de prime pure ONISR

4.3 Comparaison avec les tests `pg17testyear1`

Pour la prime pure des données `pg17testyear1` :

```
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 18.86  180.44  260.06  273.35  351.62 2242.97
```

Pour la prime pure des données ONISR :

```
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 18.72  179.23  259.08  272.98  350.44 2290.08
```

Comparaison a faire La moyenne de la soustraction des primes pures et des primes pure ONISR est de 0,37. La différence totale est de 37361,06.