

Projet Actuariat

Solène Corre, Florentin Dehooghe, François Delhaye

16 avril 2020

Table des matières

1	Présentation du projet	2
2	Exploration des jeux de données freMPL1 et freMPL2	2
2.1	Première visualisation des jeux de données	2
2.1.1	Nettoyage de données	4
2.1.2	Statistiques descriptives	4
2.1.3	Représentations graphiques des données	5
2.1.4	ACP	5
2.1.4.1	Exécution sur nos données freMPL2	5
2.1.5	AFC	8
3	GLM	12
3.1	Fréquence des sinistres	12
3.1.1	Présentation des lois utilisables	12
3.1.1.1	la loi de Poisson	12
3.1.1.2	la loi binomiale négative	13
3.2	Sévérité des sinistres	22
3.2.1	Présentation des lois utilisables	22
3.2.1.1	loi de gamma (ou d'Euler)	22
3.2.2	Inverse gauss	23
3.2.3	Calcul de la prime pure	25
3.2.4	Calcul de l'espérance $E(N)$	25
4	GAM	25
5	Bibliographie	30

6	Annexes	30
6.1	Affichage de l'implementation de la fonction <code>nettoyage_dataframe</code> :	30
6.2	Affichage de l'ensemble des représentations graphiques	32

1 Présentation du projet

L'assurance est un contrat par lequel, moyennant le versement d'une prime dont le montant est fixé a priori (en début de période de couverture), l'assureur s'engage à indemniser l'assuré pendant toute la période de couverture (généralement un an). Cette prime doit refléter le risque associé au contrat. 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, nous utiliserons cette dernière approche car on ne dispose pas des montants individuels de sinistre. Le but de ce projet est de proposer un tarificateur en se basant deux méthodes : 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. Un second objectif sera, en plus de calculer une prime pure par police, de déterminer une commerciale intégrant une marge pour risque. Une approche par simulation sera réalisée pour juger de l'adéquation du chargement par rapport à la charge sinistre totale portefeuille.

2 Exploration des jeux de données `freMPL1` et `freMPL2`

Un peu à la manière du machine learning, les données contenues dans `freMPL2` serviront de données d'entraînement de notre modèle et les données de `freMPL1` serviront pour tester notre modèle final.

2.1 Première visualisation des jeux de données

Les dimensions du jeu de données **`freMPL1`** sont (30595, 22). Ainsi, notre jeu contient 30595 données différentes, toutes définies par 22 caractéristiques différentes.

De même, les dimensions du jeu de données **`freMPL2`** sont (48295, 22). Ainsi, notre jeu contient 48295 données différentes, toutes définies par 22 caractéristiques différentes.

Les noms des caractéristiques des jeux de données sont les mêmes. Les différentes caractéristiques sont :

- **Exposure** : il s'agit d'une donnée de type numérique qui correspond à la fréquence d'exposition aux risques d'un individu sur une année. Par exemple, si l'individu a été exposé 100 jours, le chiffre affiché est 0,27 (= 100/365,25).
- **LicAge** : c'est un nombre entier de mois correspondant à l'âge de la licence de la personne concernée.
- **RecordBeg** : cela correspond à la date de début d'exposition aux risques.
- **RecordEnd** : c'est la date de fin d'exposition au risque. Si elle n'est pas renseignée, c'est que la personne est toujours exposée.
- **VehAge** : Il correspond à l'âge du véhicule en année(s). Il est composé en 9 catégories distinctes : "0", "1", "2", "3", "4", "5", "6-7", "8-9" et "10+".
- **Gender** : c'est le sexe de l'individu.
- **MariStat** : il s'agit du statut marital de la personne. Elle est soit célibataire ("Alone") soit autre chose ("Other").
- **SocioCateg** : Cela correspond à la catégorie socioprofessionnelle de l'individu. Les valeurs, comprises entre "CSP1" et "CSP99", correspondent à la classification française (voir lien suivant : https://fr.wikipedia.org/wiki/Professions_et_cat%C3%A9gories_socioprofessionnelles_en_France).

- **VehUsage** : Cela correspond à l'utilisation du véhicule par le propriétaire. Il est soit privée ("Private"), soit professionnel ("Professional"), ...
- **DrivAge** : C'est l'âge du conducteur (en années). Pour rappel, en France, la conduite est possible à partir de 18 ans.
- **HasKmLimit** : il s'agit d'une valeur numérique spécifiant si oui ("1") ou non ("0") l'assurance comporte une limite kilométrique.
- **BonusMalus** : c'est un variable de type numérique, dont la valeur est comprise entre 50 et 350, précisant si la personne possède des bonus ou des malus. Si la valeur est inférieure à 100, l'individu a droit à des bonus. Sinon, la personne a des malus.
- **VehBody** : il s'agit du type de modèle concerné par l'assurance de l'individu.
- **VehPrice** : c'est un indicateur correspondant au prix du véhicule.
- **VehEngine** : cela correspond au type de moteur que possède le véhicule.
- **VehEnergy** : cela correspond au type d'énergie consommé par le véhicule que possède le véhicule
- **VehMaxSpeed** : c'est la vitesse maximum que peut atteindre le véhicule. Les différentes catégories sont: "1-130 km/h", "130-140 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".
- **VehClass** : il s'agit de la classe du véhicule.
- **RiskVar** : Nombre compris entre 1 et 20 correspondant au risque inconnu probable.
- **ClaimAmount** : c'est le montant total de la garantie) laquelle peut prétendre l'assuré.
- **Garage** : il s'agit du type de garage auquel se rend l'assuré.
- **ClaimInd** : c'est un indicateur précisant si oui ou non l'assuré peut prétendre à une garantie.

Regardons maintenant les premiers éléments composant le jeu de données **freMPL1**:

	1	2	3
Exposure	0.583	0.200	0.083
LicAge	366	187	169
RecordBeg	2004-06-01	2004-10-19	2004-07-16
RecordEnd	NA	NA	2004-08-16
VehAge	2	0	1
Gender	Female	Male	Female
MariStat	Other	Alone	Other
SocioCateg	CSP1	CSP55	CSP1
VehUsage	Professional	Private+trip to office	Professional
DrivAge	55	34	33
HasKmLimit	0	0	0
BonusMalus	72	80	63
VehBody	sedan	microvan	other microvan
VehPrice	D	K	L
VehEngine	injection	direct injection overpowered	direct injection overpowered
VehEnergy	regular	diesel	diesel
VehMaxSpeed	160-170 km/h	170-180 km/h	170-180 km/h
VehClass	B	M1	M1
ClaimAmount	0	0	0
RiskVar	15	20	17
Garage	None	None	None
ClaimInd	0	0	0

et aussi les premiers éléments composants **freMPL2**:

	1	2	3
Exposure	0.583	0.416	0.583
LicAge	579	361	366

	1	2	3
RecordBeg	2004-06-01	2004-01-01	2004-06-01
RecordEnd	NA	2004-06-01	NA
VehAge	10+	1	2
Gender	Male	Female	Female
MariStat	Other	Other	Other
SocioCateg	CSP60	CSP1	CSP1
VehUsage	Private	Professional	Professional
DrivAge	83	55	55
HasKmLimit	0	0	0
BonusMalus	50	58	72
VehBody	sedan	sedan	sedan
VehPrice	N	D	D
VehEngine	injection	injection	injection
VehEnergy	regular	regular	regular
VehMaxSpeed	190-200 km/h	160-170 km/h	160-170 km/h
VehClass	H	B	B
RiskVar	14	15	15
ClaimAmount	0	0	0
Garage	None	None	None
ClaimInd	0	0	0

2.1.1 Nettoyage de données

Remarquons qu'il serait intéressant de faire un peu de nettoyage de données avant d'effectuer quelconques travaux sur celles-ci. Pour cela, nous allons créer une fonction qui servira à nettoyer les 2 dataframes.

Cette fonction (appelée `nettoyage_dataframe`) prend l'un des deux dataframes en paramètres et effectue les opérations suivantes :

- Suppression des données des individus assurés moins d'un jour (Exposure)
- Modification des données des individus ayant un ClaimAmount négatif
- Suppression de la colonne associée au sexe de la personne
- Réduction du nombre de catégories socioprofessionnels
- Traduction des données (VehBody, MariStat, VehUsage, VehEngine, VehEnergy, Garage)

2.1.2 Statistiques descriptives

Regardons maintenant plus précisément les valeurs particulières de ces colonnes (valeurs minimum et maximum, moyenne, médiane, quantiles, ...)

On remarquera qu'il existe des données manquantes dans la colonne RecEnd, ce qui signifie que les individus concernés sont toujours assurés.

On peut aussi utiliser la méthode `describe` du package `Hmisc` pour avoir un aperçu de la dispersion des données.

Mais cela ne vaut pas une représentation graphique.

2.1.3 Représentations graphiques des données

2.1.4 ACP

L'ACP permet d'analyser et de visualiser un jeu de données contenant des individus décrits par plusieurs variables quantitatives. C'est une méthode statistique qui permet d'explorer des données dites multivariées (données avec plusieurs variables). Chaque variable pourrait être considérée comme une dimension différente. L'analyse en composantes principales est utilisée pour extraire et de visualiser les informations importantes contenues dans une table de données multivariées. L'ACP synthétise cette information en seulement quelques nouvelles variables appelées composantes principales. Ces nouvelles variables correspondent à une combinaison linéaire des variables originels. Le nombre de composantes principales est inférieur ou égal au nombre de variables d'origine.

2.1.4.1 Exécution sur nos données freMPL2

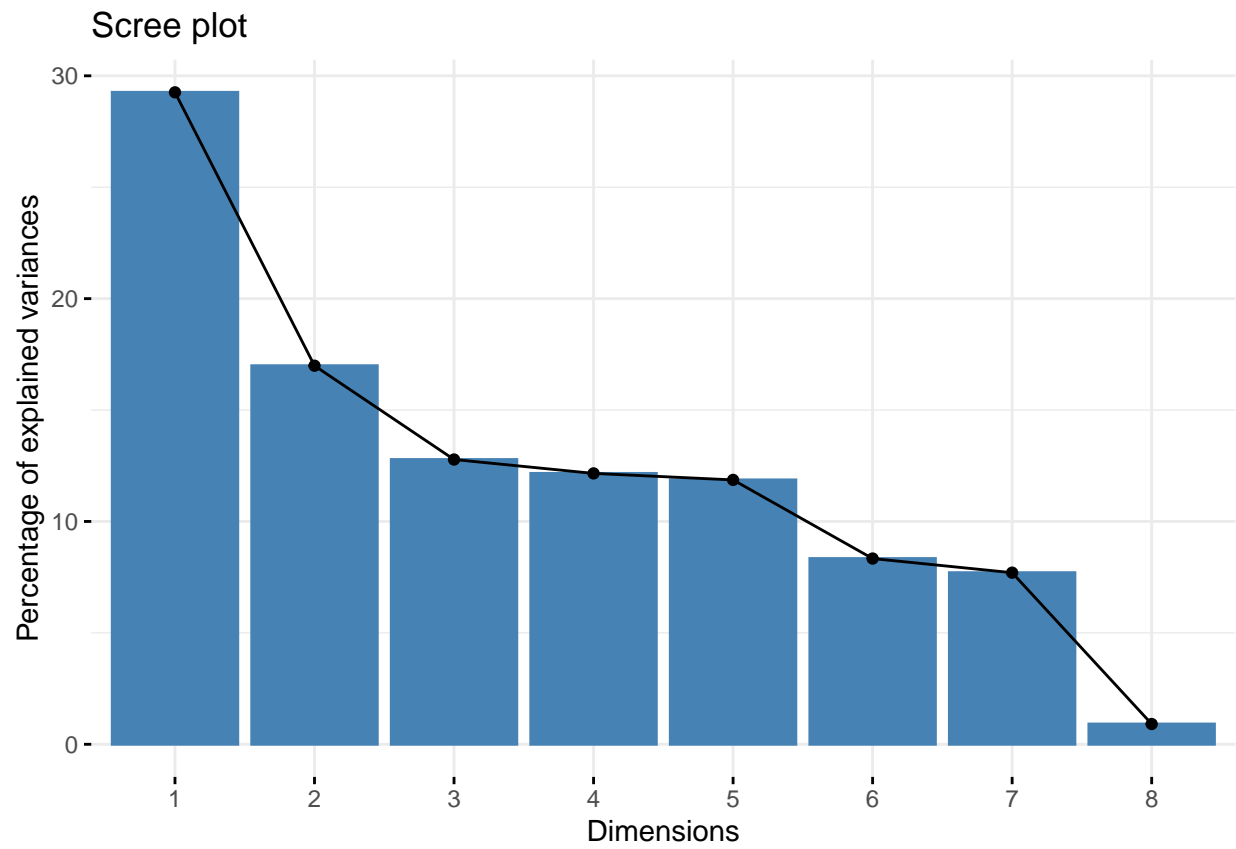
Attention, les valeurs doivent être numériques.

On va donc convertir nos valeurs en numérique :

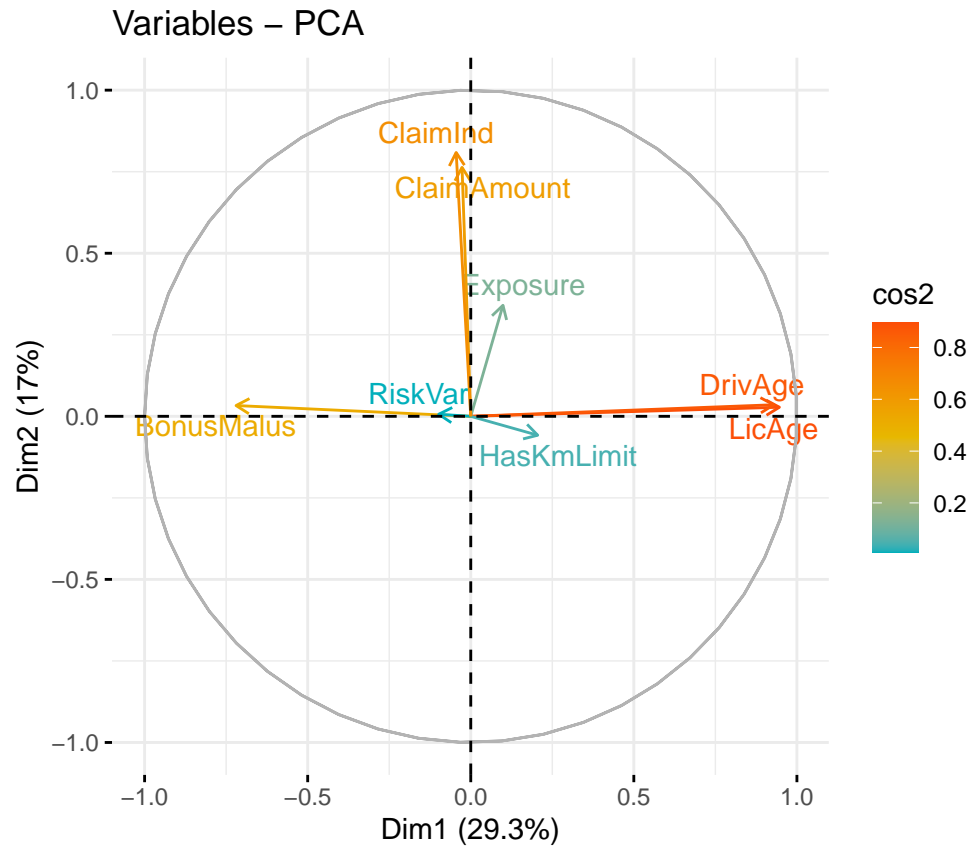
Certaines colonnes sont catégorisées et pourraient nous être utiles pour exécuter notre ACP. Il n'est cependant pas judicieux d'appliquer une conversion numérique à ces colonnes puisqu'on leur attribue une valeur arbitraire nous faisant penser à une classification des différents facteurs possibles. Pour éviter cela, on va donc utiliser la méthode `model.matrix()` qui crée une matrice binaire spécifiant à quel facteur correspond une ligne du dataframe.

Affichage du résultat :

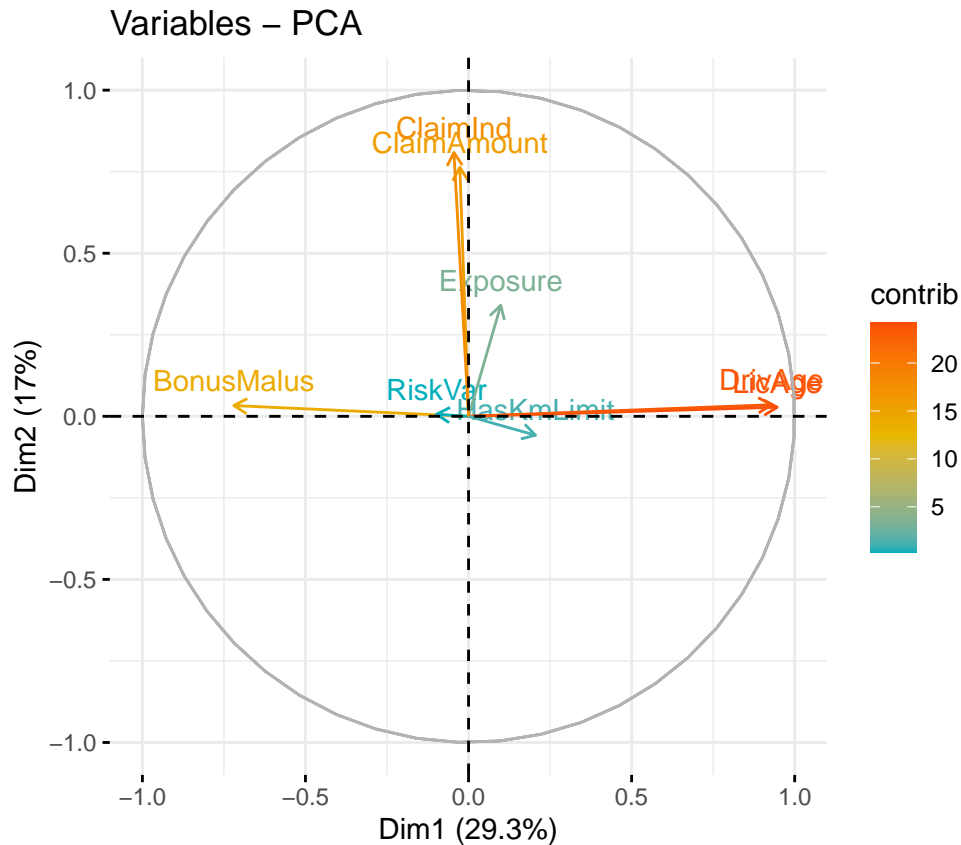
##	eigenvalue	variance.percent	cumulative.variance.percent
## Dim.1	2.34068251	29.2585314	29.25853
## Dim.2	1.35919145	16.9898931	46.24842
## Dim.3	1.02241078	12.7801347	59.02856
## Dim.4	0.97257721	12.1572151	71.18577
## Dim.5	0.94923929	11.8654911	83.05127
## Dim.6	0.66687171	8.3358964	91.38716
## Dim.7	0.61613751	7.7017189	99.08888
## Dim.8	0.07288954	0.9111193	100.00000



utilisation de \cos^2 pour juger de la qualité de la représentation :



contribution des colonnes aux dimensions :



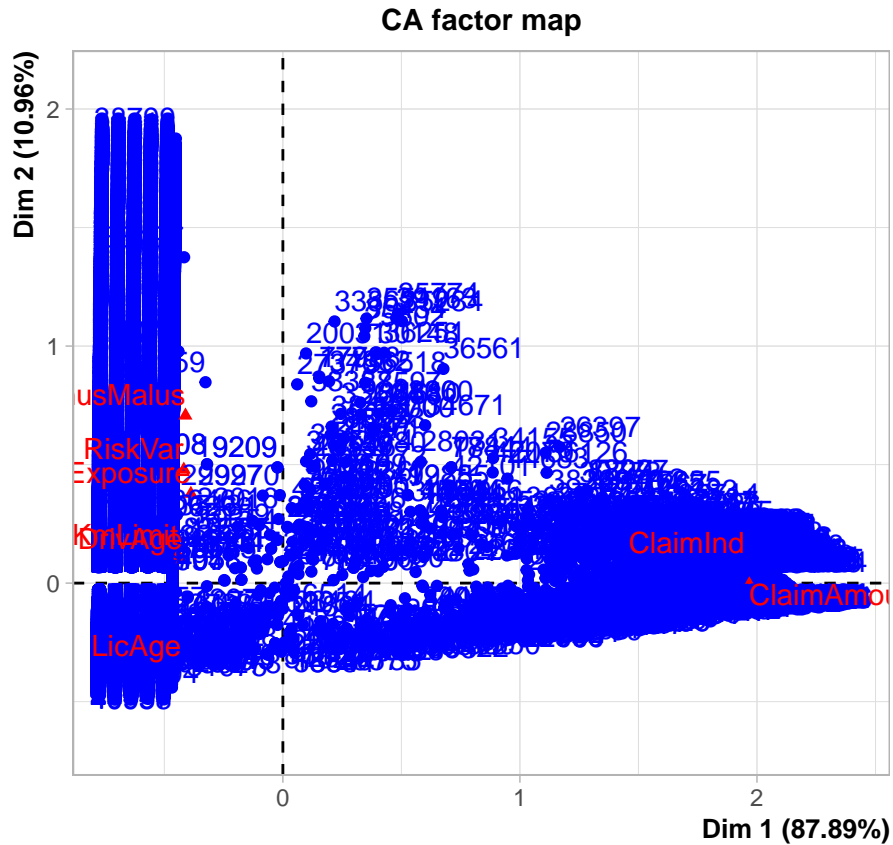
Description des dimensions

Dans les sections précédentes, nous avons décrit comment mettre en évidence les variables en fonction de leurs contributions aux composantes principales.

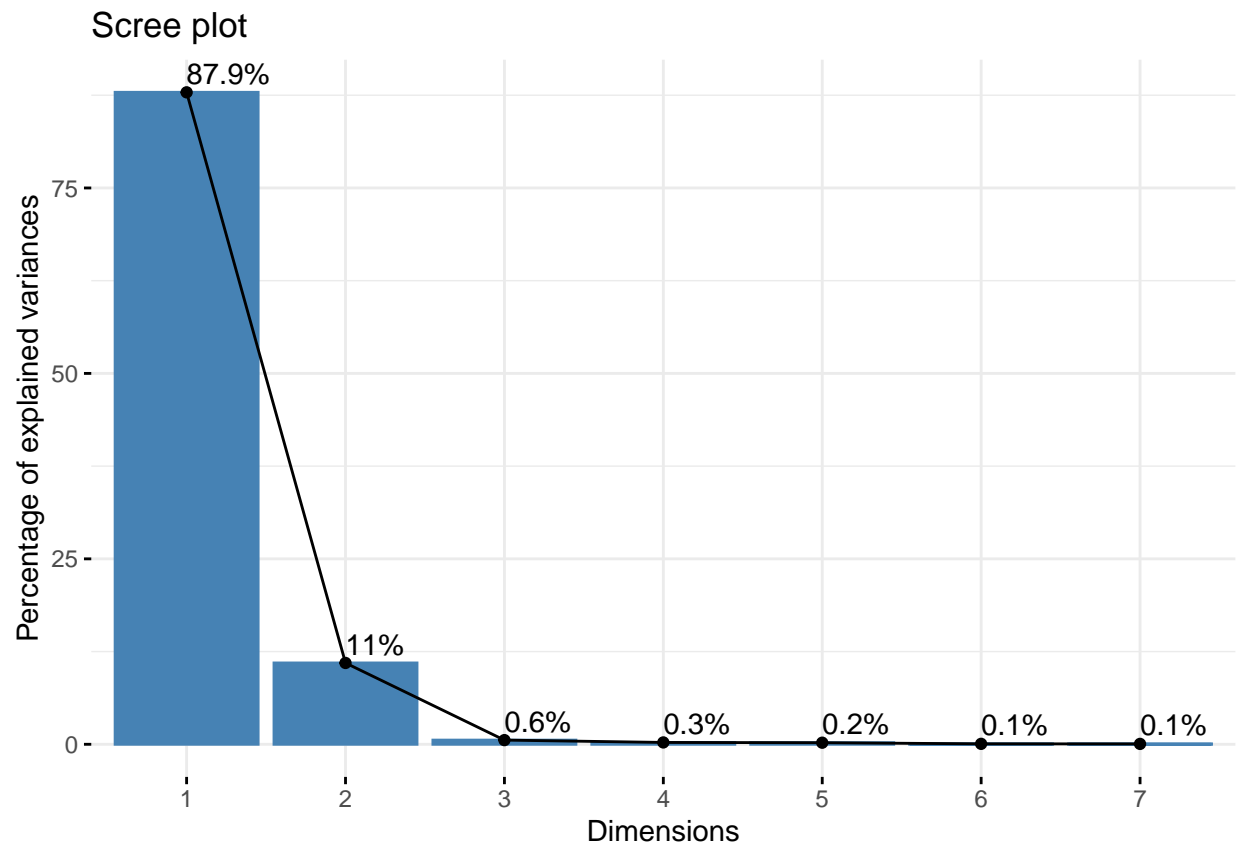
Notez également que la fonction `dimdesc()` [dans `FactoMineR`], pour dimension description (en anglais), peut être utilisée pour identifier les variables les plus significativement associées avec une composante principale donnée . Elle peut être utilisée comme suit:

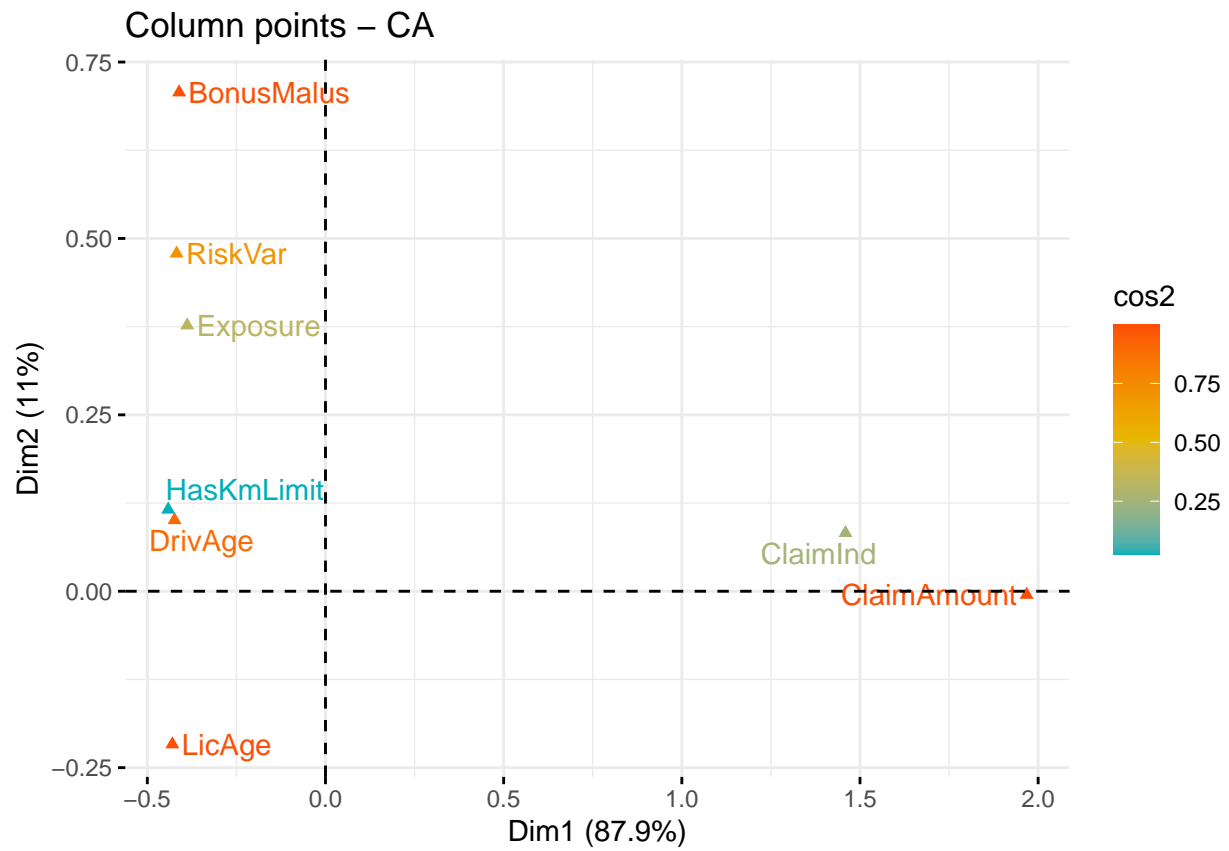
2.1.5 AFC

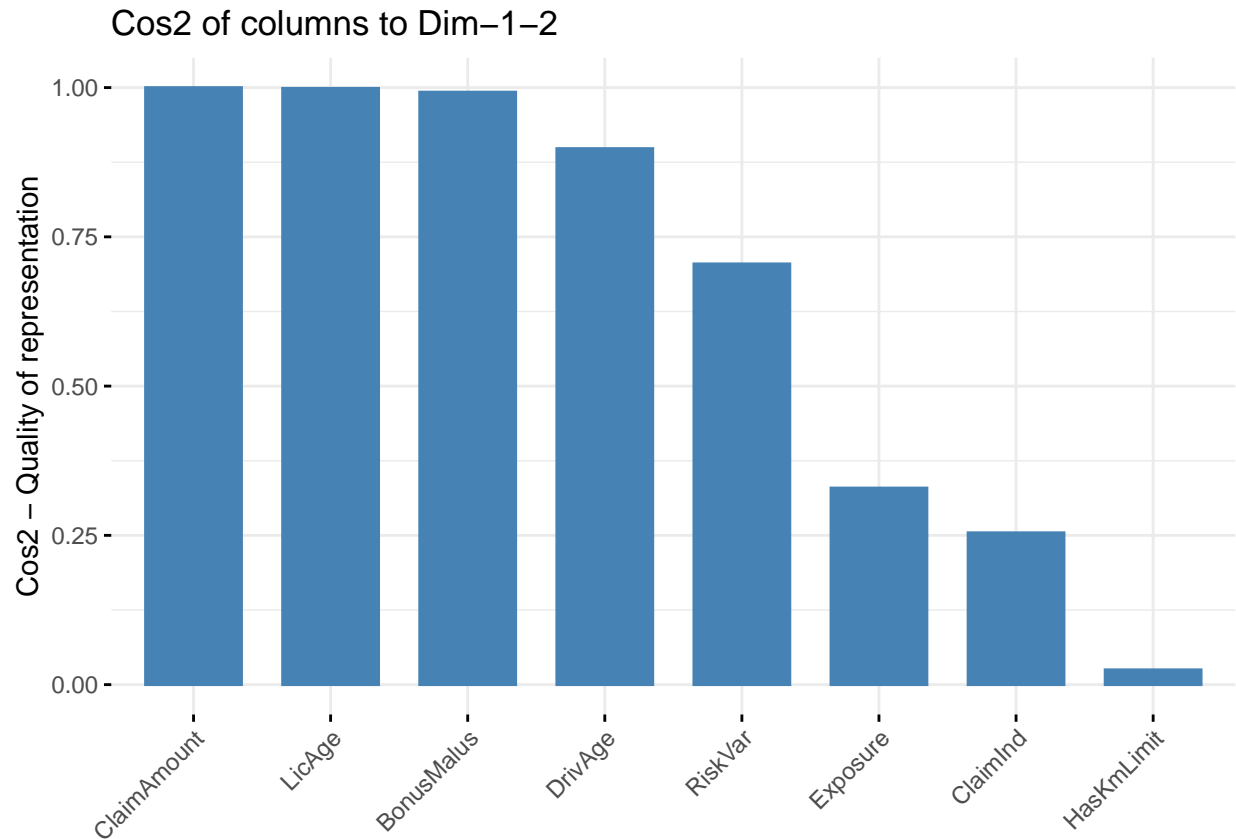
L'analyse factorielle des correspondances est une extension de l'analyse en composantes principales pour analyser l'association entre deux variables qualitatives (ou catégorielles). L'AFC permet de résumer et de visualiser l'information contenue dans le tableau de contingence formé par les deux variables catégorielles. Le tableau de contingence contient les fréquences formées par les deux variables.



##	eigenvalue	variance.percent	cumulative.variance.percent
## Dim.1	0.8368752926	87.89057231	87.89057
## Dim.2	0.1043926431	10.96355601	98.85413
## Dim.3	0.0053764426	0.56464639	99.41877
## Dim.4	0.0023878835	0.25078103	99.66956
## Dim.5	0.0020457193	0.21484615	99.88440
## Dim.6	0.0005902286	0.06198717	99.94639
## Dim.7	0.0005104719	0.05361094	100.00000







```
##          coord
## HasKmLimit -0.4413484
## LicAge     -0.4297221
## DrivAge    -0.4235596
## RiskVar    -0.4178123
## BonusMalus -0.4108171
## Exposure   -0.3880547
## ClaimInd    1.4592963
## ClaimAmount 1.9675944
```

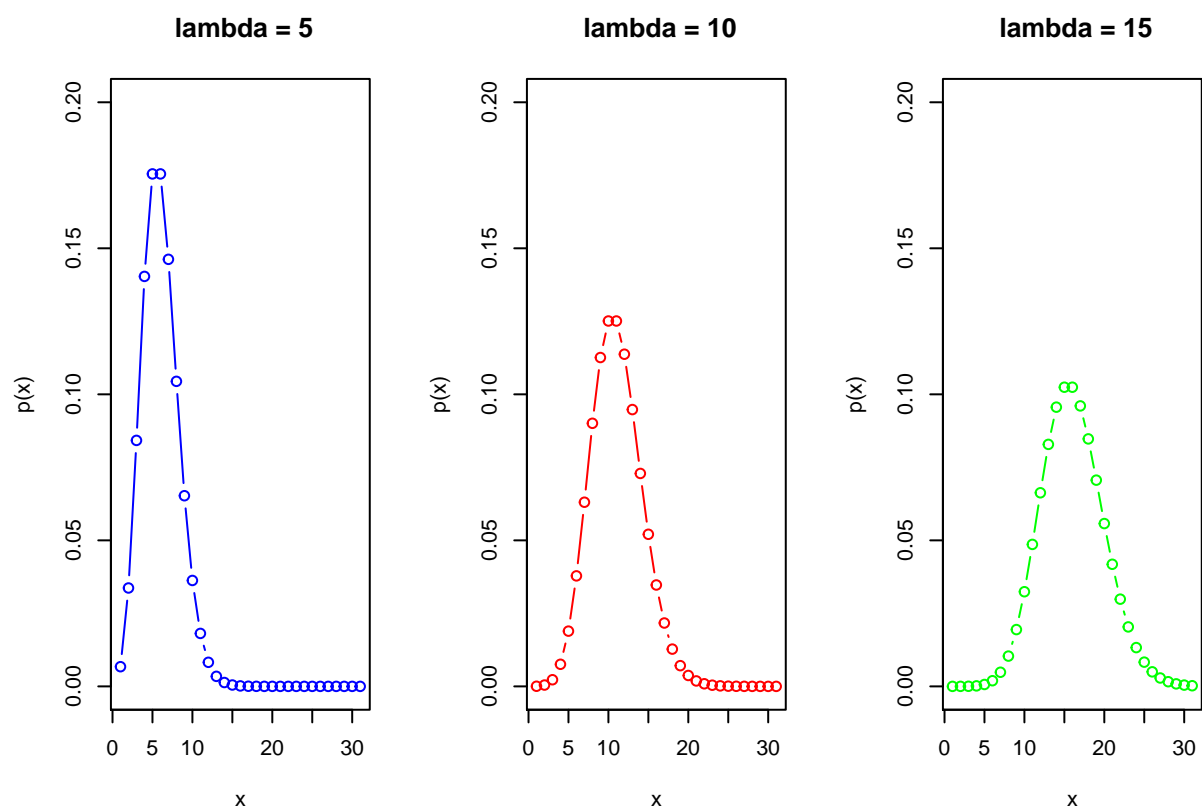
3 GLM

Les lois les plus utilisées sont : - Poisson ou binomiale négative pour les fréquences des sinistres - Gamma et Inverse gauss pour la sévérité des sinistres

3.1 Fréquence des sinistres

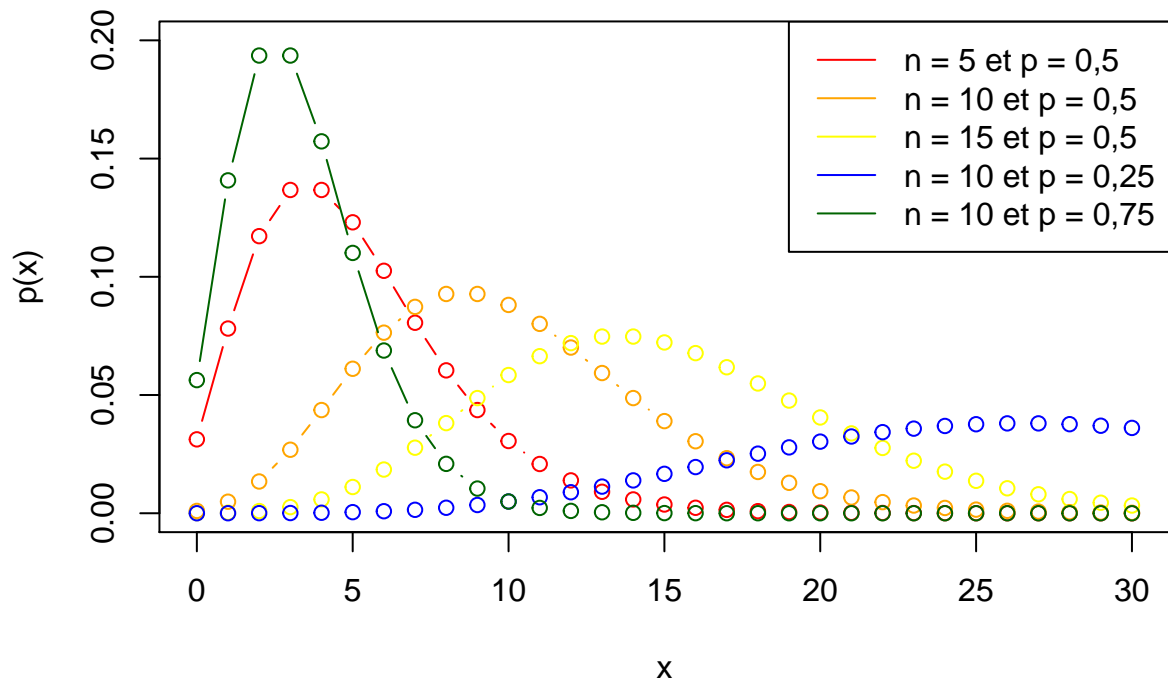
3.1.1 Présentation des lois utilisables

3.1.1.1 la loi de Poisson



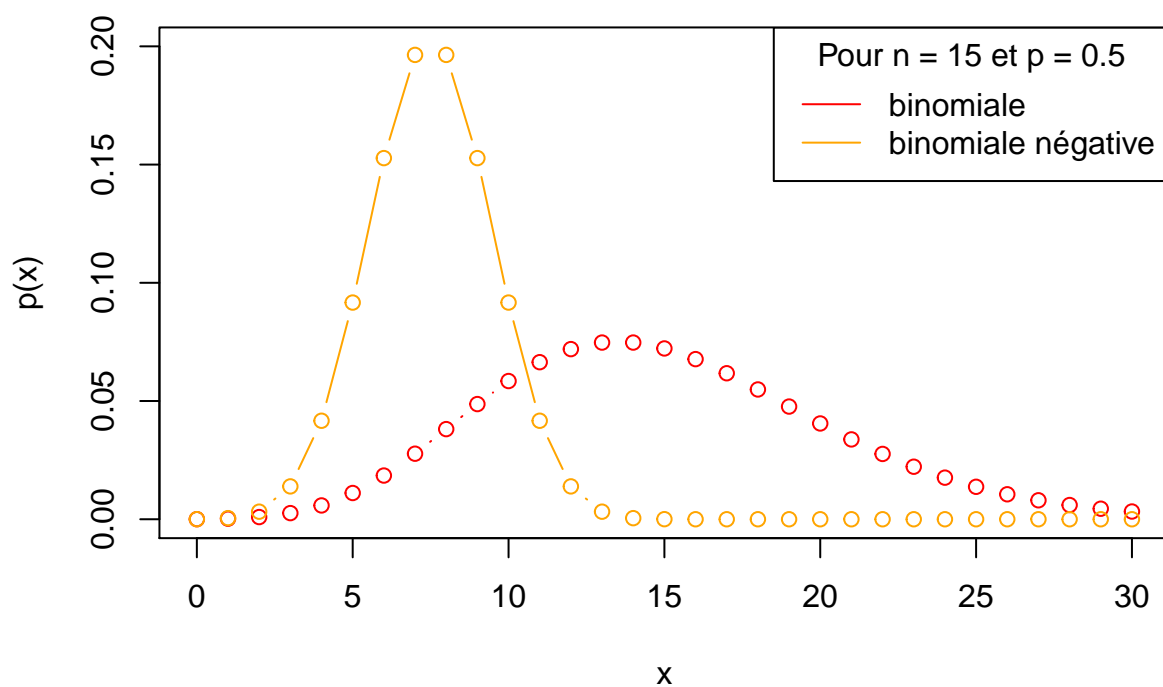
3.1.1.2 la loi binomiale négative

Quelques exemples de loi binomiale négative



comparaison binomiale et binomiale négative :

Comparaison entre loi binomiale et loi binomiale négative



Test du modèle GLM avec la loi de Poisson :

```
r #calibration d'une loi de Poisson glm1 <-  
glm(ClaimInd~DrivAge+VehAge+VehUsage+BonusMalus+VehBody+VehMaxSpeed, offset =  
log(Exposure), family=poisson("log"), data=freMPL2) summary(glm1)
```

```

## ## Call: ## glm(formula = ClaimInd ~ DrivAge + VehAge + VehUsage + BonusMalus + ##
VehBody + VehMaxSpeed, family = poisson("log"), data = freMPL2, ##      offset =
log(Exposure)) ## ## Deviance Residuals: ##      Min      1Q    Median      3Q
Max ## -0.7406  -0.3540  -0.2683  -0.1758   3.5785 ## ## Coefficients: ##
Estimate Std. Error z value Pr(>|z|) ## (Intercept)                                -1.451769
0.299832  -4.842 1.29e-06 *** ## DrivAge(20,25]                                -0.572739
0.239860  -2.388 0.016949 * ## DrivAge(25,30]                                -0.869588
0.232667  -3.737 0.000186 *** ## DrivAge(30,35]                                -0.819072
0.231959  -3.531 0.000414 *** ## DrivAge(35,40]                                -0.745800
0.232851  -3.203 0.001360 ** ## DrivAge(40,45]                                -0.842333
0.235696  -3.574 0.000352 *** ## DrivAge(45,50]                                -0.936159
0.236941  -3.951 7.78e-05 *** ## DrivAge(50,55]                                -0.920040
0.236138  -3.896 9.77e-05 *** ## DrivAge(55,60]                                -0.847122
0.237082  -3.573 0.000353 *** ## DrivAge(60,65]                                -0.786225
0.244622  -3.214 0.001309 ** ## DrivAge(65,70]                                -0.775962
0.254946  -3.044 0.002337 ** ## DrivAge(70,75]                                -1.011553
0.270779  -3.736 0.000187 *** ## DrivAge(75,80]                                -1.418121
0.316862  -4.476 7.62e-06 *** ## DrivAge(80,104]                             -0.677142
0.293429  -2.308 0.021017 * ## VehAge1                                      -0.098937
0.099875  -0.991 0.321874 ## VehAge10+                                     -0.230448   0.085593
-2.692 0.007095 ** ## VehAge2                                           0.023017   0.095496
0.241 0.809536 ## VehAge3                                           0.005242   0.100133   0.052
0.958247 ## VehAge4                                           -0.055284   0.102154  -0.541 0.588384
## VehAge5                                           -0.040324   0.105113  -0.384 0.701252 ##
VehAge6-7                                           -0.140546   0.099284  -1.416 0.156894 ##
VehAge8-9                                           -0.100558   0.099237  -1.013 0.310909 ##
VehUsageprivée et trajet vers bureau 0.138089   0.054904   2.515 0.011899 * ##
VehUsageprofessionnel 0.190700   0.068017   2.804 0.005051 ** ##
VehUsagetrajet professionnel 0.161592   0.176992   0.913 0.361246 ##
BonusMalus(100,351] 0.504666   0.093149   5.418 6.03e-08 *** ##
VehBodymicrovan 0.107266   0.165941   0.646 0.518013 ##
VehBodyautobus -0.189304   0.374354  -0.506 0.613080 ##
VehBodycoupé -0.011214   0.158709  -0.071 0.943670 ##
VehBodyautre microvan -0.003269   0.159403  -0.021 0.983636 ##
VehBodyberline -0.112103   0.120714  -0.929 0.353061 ##
VehBodySUV 0.134478   0.149639   0.899 0.368820 ##
VehBodybreak -0.445587   0.167322  -2.663 0.007744 ** ##
VehBodycamionnette 0.209691   0.155274   1.350 0.176868 ##
VehMaxSpeed130-140 km/h -0.052937   0.167987  -0.315 0.752667 ##
VehMaxSpeed140-150 km/h -0.017114   0.160254  -0.107 0.914955 ##
VehMaxSpeed150-160 km/h 0.109537   0.153027   0.716 0.474113 ##
VehMaxSpeed160-170 km/h 0.006457   0.154812   0.042 0.966729 ##
VehMaxSpeed170-180 km/h 0.116861   0.155896   0.750 0.453492 ##
VehMaxSpeed180-190 km/h -0.079147   0.162400  -0.487 0.626004 ##
VehMaxSpeed190-200 km/h 0.097111   0.163630   0.593 0.552859 ##
VehMaxSpeed200-220 km/h 0.063538   0.166487   0.382 0.702729 ##
VehMaxSpeed220+ km/h 0.277671   0.171568   1.618 0.105571 ## --- ##
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1 ## ## (Dispersion
parameter for poisson family taken to be 1) ## ##      Null deviance: 12564  on 47496
degrees of freedom ## Residual deviance: 12410  on 47454  degrees of freedom ## AIC:
16764 ## ## Number of Fisher Scoring iterations: 6

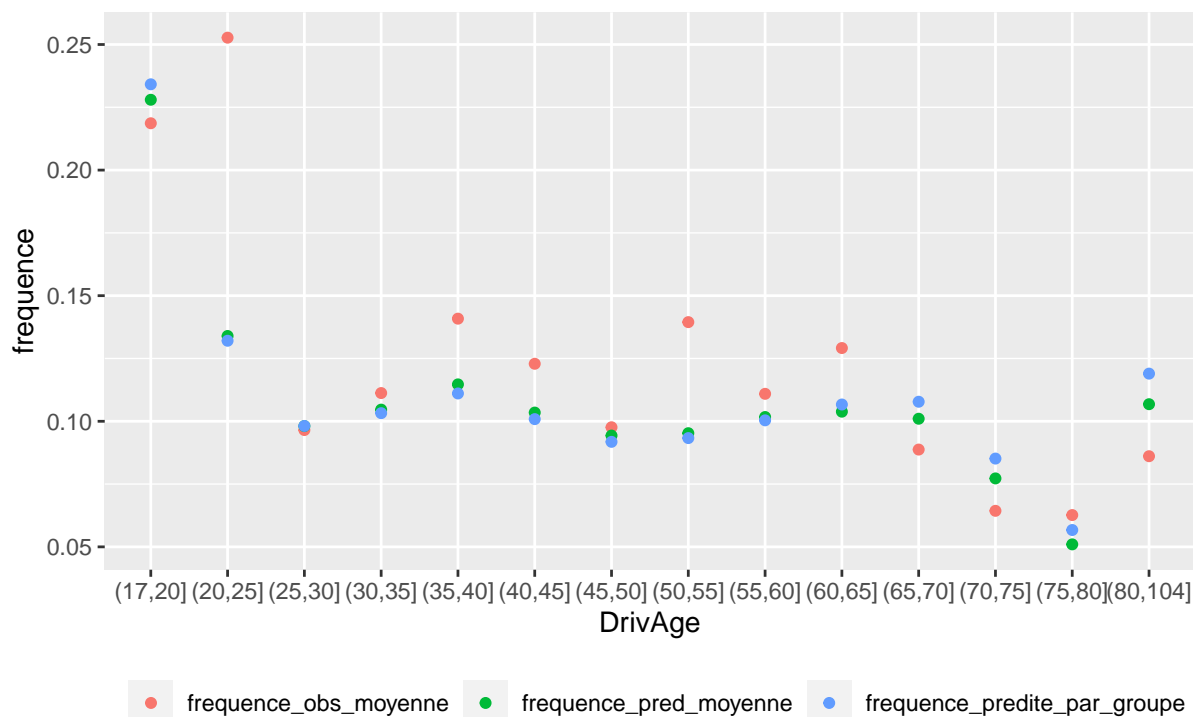
```

```

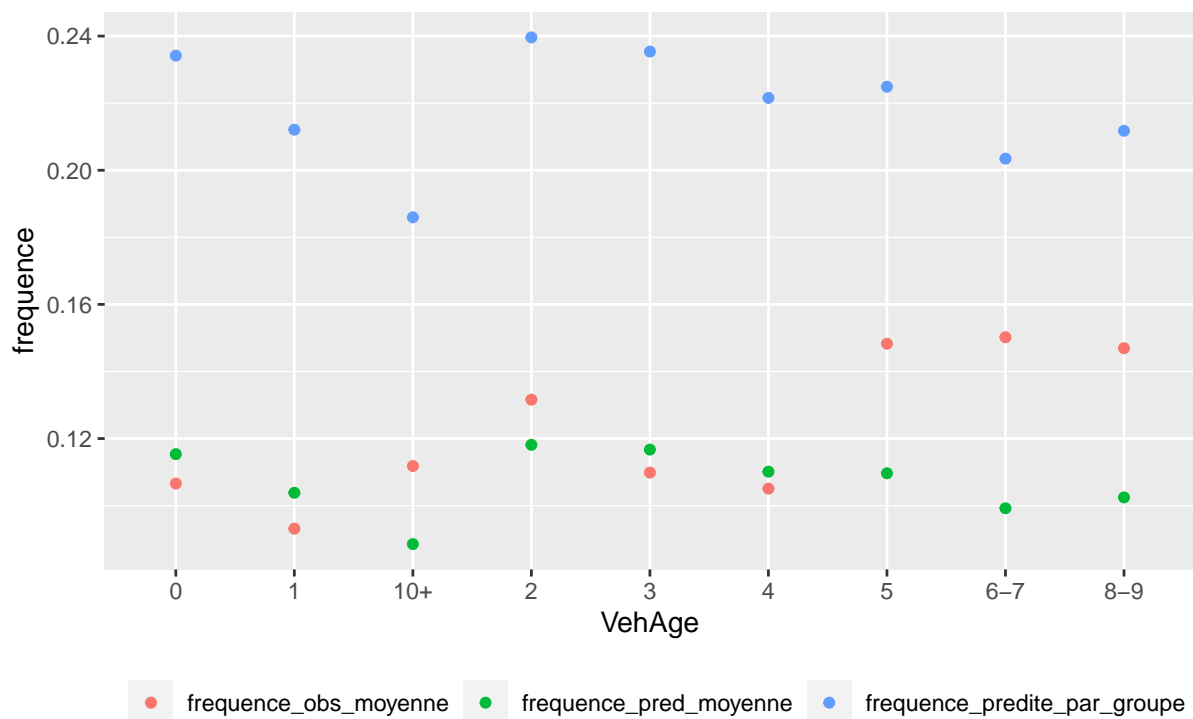
r #glm1$aic #glm1$null.deviance #glm1$deviance #glm1$iter predict1 <- predict(glm1,
freMPL2, type = 'response') recap1 <- data.frame(ClaimInd = freMPL2$ClaimInd,
Predict_ClaimInd = predict1, difference = abs(freMPL2$ClaimInd- predict1))
head(recap1,10)
##      ClaimInd Predict_ClaimInd difference ## 1          0          0.05426268 0.05426268
## 2          0          0.03828256 0.03828256 ## 3          0          0.06060942 0.06060942
## 4          0          0.02965685 0.02965685 ## 5          0          0.01052081 0.01052081
## 6          0          0.04753376 0.04753376 ## 7          1          0.06880799 0.93119201
## 8          0          0.06867037 0.06867037 ## 9          0          0.02019737 0.02019737
## 10         0          0.05467572 0.05467572
r recap1[recap1$difference<0.5 & recap1$ClaimInd == 1,]
## [1] ClaimInd      Predict_ClaimInd difference ## <0 rows> (or 0-length
row.names)
r # Ce modèle ne fonctionne pas pour prédire les ClaimInd égaux à 1
### Calcul de l'espérance E(N)
FORMULE :  $E(N/x) = g^{(-1)}(aX_i + e_i)$  avec g la fonction de lien et e l'offset
Ainsi, pour une loi de Poisson, on a :  $E(N/x) = \exp(aX_i + \ln(\text{Exposure})) = \text{Exposure} * \exp(aX_i)$ 

```

Nombre de sinistres et leur prédiction selon l'âge du conducteur



Nombre de sinistres et leur prédiction selon l'âge du véhicule



Quasipoisson:

SurDispersion????

```
##
## Call:
## glm(formula = ClaimInd ~ DrivAge + VehAge + VehUsage + BonusMalus +
##      VehBody + VehMaxSpeed, family = quasipoisson("log"), data = freMPL2,
##      offset = log(Exposure))
##
## Deviance Residuals:
##      Min        1Q    Median        3Q        Max
## -0.7406   -0.3540   -0.2683   -0.1758    3.5785
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      -1.451769    0.321813  -4.511 6.46e-06 ***
## DrivAge(20,25]     -0.572739    0.257445  -2.225 0.026106 *
## DrivAge(25,30]     -0.869588    0.249724  -3.482 0.000498 ***
## DrivAge(30,35]     -0.819072    0.248965  -3.290 0.001003 **
## DrivAge(35,40]     -0.745800    0.249922  -2.984 0.002845 **
## DrivAge(40,45]     -0.842333    0.252975  -3.330 0.000870 ***
## DrivAge(45,50]     -0.936159    0.254312  -3.681 0.000232 ***
## DrivAge(50,55]     -0.920040    0.253450  -3.630 0.000284 ***
## DrivAge(55,60]     -0.847122    0.254464  -3.329 0.000872 ***
## DrivAge(60,65]     -0.786225    0.262556  -2.995 0.002750 **
## DrivAge(65,70]     -0.775962    0.273637  -2.836 0.004574 **
## DrivAge(70,75]     -1.011553    0.290630  -3.481 0.000501 ***
## DrivAge(75,80]     -1.418121    0.340092  -4.170 3.05e-05 ***
## DrivAge(80,104]    -0.677142    0.314941  -2.150 0.031556 *
## VehAge1            -0.098937    0.107197  -0.923 0.356038
## VehAge10+         -0.230448    0.091868  -2.508 0.012129 *
## VehAge2            0.023017    0.102497   0.225 0.822322
## VehAge3            0.005242    0.107475   0.049 0.961097
## VehAge4           -0.055284    0.109643  -0.504 0.614113
## VehAge5           -0.040324    0.112819  -0.357 0.720774
## VehAge6-7         -0.140546    0.106563  -1.319 0.187208
## VehAge8-9         -0.100558    0.106512  -0.944 0.345123
## VehUsageprivée et trajet vers bureau 0.138089    0.058929   2.343 0.019117 *
## VehUsageprofessionnel 0.190700    0.073003   2.612 0.008998 **
## VehUsagetrajet professionnel 0.161592    0.189968   0.851 0.394979
## BonusMalus(100,351] 0.504666    0.099978   5.048 4.49e-07 ***
## VehBodymicrovan    0.107266    0.178106   0.602 0.547005
## VehBodyautobus    -0.189304    0.401799  -0.471 0.637542
## VehBodycoupé      -0.011214    0.170345  -0.066 0.947513
## VehBodyautre microvan -0.003269    0.171089  -0.019 0.984754
## VehBodyberline    -0.112103    0.129564  -0.865 0.386914
## VehBodySUV         0.134478    0.160609   0.837 0.402428
## VehBodybreak      -0.445587    0.179589  -2.481 0.013100 *
## VehBodycamionnette 0.209691    0.166658   1.258 0.208320
## VehMaxSpeed130-140 km/h -0.052937    0.180303  -0.294 0.769065
## VehMaxSpeed140-150 km/h -0.017114    0.172003  -0.099 0.920745
## VehMaxSpeed150-160 km/h 0.109537    0.164246   0.667 0.504833
## VehMaxSpeed160-170 km/h 0.006457    0.166162   0.039 0.969001
## VehMaxSpeed170-180 km/h 0.116861    0.167325   0.698 0.484928
```

```

## VehMaxSpeed180-190 km/h          -0.079147   0.174306  -0.454 0.649781
## VehMaxSpeed190-200 km/h          0.097111   0.175626   0.553 0.580305
## VehMaxSpeed200-220 km/h          0.063538   0.178693   0.356 0.722163
## VehMaxSpeed220+ km/h             0.277671   0.184147   1.508 0.131592
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for quasipoisson family taken to be 1.152001)
##
##      Null deviance: 12564  on 47496  degrees of freedom
## Residual deviance: 12410  on 47454  degrees of freedom
## AIC: NA
##
## Number of Fisher Scoring iterations: 6

##      ClaimInd Predict_ClaimInd difference
## 1          0          0.05426268 0.05426268
## 2          0          0.03828256 0.03828256
## 3          0          0.06060942 0.06060942
## 4          0          0.02965685 0.02965685
## 5          0          0.01052081 0.01052081
## 6          0          0.04753376 0.04753376
## 7          1          0.06880799 0.93119201
## 8          0          0.06867037 0.06867037
## 9          0          0.02019737 0.02019737
## 10         0          0.05467572 0.05467572

##
## Call:
## glm(formula = ClaimInd ~ DrivAge + VehAge + VehUsage + BonusMalus +
##      VehBody + VehMaxSpeed, family = negative.binomial(phi), data = freMPL2,
##      offset = log(Exposure))
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -0.6977  -0.3492  -0.2664  -0.1755   3.4666
##
## Coefficients:
##
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      -1.462299   0.332912  -4.392 1.12e-05 ***
## DrivAge(20,25]    -0.569682   0.268934  -2.118 0.034155 *
## DrivAge(25,30]    -0.868870   0.261126  -3.327 0.000877 ***
## DrivAge(30,35]    -0.813323   0.260373  -3.124 0.001787 **
## DrivAge(35,40]    -0.738177   0.261351  -2.824 0.004738 **
## DrivAge(40,45]    -0.837677   0.264308  -3.169 0.001529 **
## DrivAge(45,50]    -0.935331   0.265617  -3.521 0.000430 ***
## DrivAge(50,55]    -0.912650   0.264719  -3.448 0.000566 ***
## DrivAge(55,60]    -0.844737   0.265813  -3.178 0.001484 **
## DrivAge(60,65]    -0.779618   0.273718  -2.848 0.004398 **
## DrivAge(65,70]    -0.775822   0.284699  -2.725 0.006431 **
## DrivAge(70,75]    -1.014987   0.301088  -3.371 0.000749 ***
## DrivAge(75,80]    -1.420910   0.348470  -4.078 4.56e-05 ***
## DrivAge(80,104]   -0.679998   0.325771  -2.087 0.036862 *
## VehAge1           -0.103531   0.108830  -0.951 0.341452

```

```

## VehAge10+                -0.226093    0.093002   -2.431  0.015058 *
## VehAge2                   0.023953    0.104193    0.230  0.818176
## VehAge3                   -0.001491    0.109334   -0.014  0.989120
## VehAge4                   -0.053792    0.111236   -0.484  0.628684
## VehAge5                   -0.035584    0.114326   -0.311  0.755614
## VehAge6-7                 -0.142977    0.108012   -1.324  0.185603
## VehAge8-9                 -0.097269    0.107854   -0.902  0.367136
## VehUsageprivée et trajet vers bureau  0.138644    0.059444    2.332  0.019686 *
## VehUsageprofessionnel      0.190946    0.073923    2.583  0.009796 **
## VehUsagetrajet professionnel  0.160772    0.193005    0.833  0.404852
## BonusMalus(100,351]       0.511375    0.102572    4.986  6.20e-07 ***
## VehBodymicrovan           0.116024    0.180936    0.641  0.521366
## VehBodyautobus            -0.193217    0.406113   -0.476  0.634239
## VehBodycoupé              -0.002137    0.172842   -0.012  0.990134
## VehBodyautre microvan      0.004896    0.173727    0.028  0.977518
## VehBodyberline            -0.108410    0.131727   -0.823  0.410516
## VehBodySUV                0.138170    0.163601    0.845  0.398363
## VehBodybreak              -0.444081    0.181310   -2.449  0.014317 *
## VehBodycamionnette        0.205938    0.169712    1.213  0.224962
## VehMaxSpeed130-140 km/h    -0.045603    0.183023   -0.249  0.803235
## VehMaxSpeed140-150 km/h    -0.010764    0.174559   -0.062  0.950832
## VehMaxSpeed150-160 km/h     0.115458    0.166865    0.692  0.488990
## VehMaxSpeed160-170 km/h     0.010421    0.168746    0.062  0.950758
## VehMaxSpeed170-180 km/h     0.119944    0.169896    0.706  0.480202
## VehMaxSpeed180-190 km/h    -0.074942    0.176796   -0.424  0.671646
## VehMaxSpeed190-200 km/h     0.099375    0.178290    0.557  0.577270
## VehMaxSpeed200-220 km/h     0.066605    0.181338    0.367  0.713397
## VehMaxSpeed220+ km/h       0.283654    0.187145    1.516  0.129603
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for Negative Binomial(1) family taken to be 1.108235)
##
## Null deviance: 11037 on 47496 degrees of freedom
## Residual deviance: 10892 on 47454 degrees of freedom
## AIC: 16895
##
## Number of Fisher Scoring iterations: 6

## ClaimInd Predict_ClaimInd difference
## 1          0          0.05409608 0.05409608
## 2          0          0.03828893 0.03828893
## 3          0          0.06095570 0.06095570
## 4          0          0.02988352 0.02988352
## 5          0          0.01054311 0.01054311
## 6          0          0.04763453 0.04763453
## 7          1          0.06832583 0.93167417
## 8          0          0.06818918 0.06818918
## 9          0          0.02021753 0.02021753
## 10         0          0.05503726 0.05503726

## [1] ClaimInd Predict_ClaimInd difference
## <0 rows> (or 0-length row.names)

```

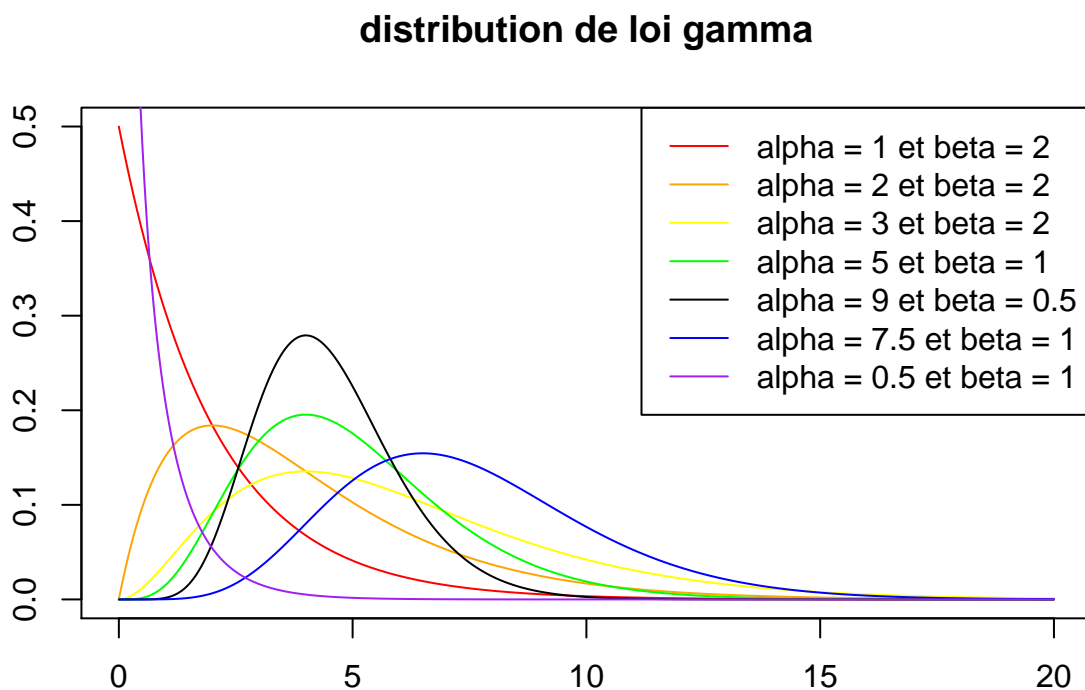
Ce modèle binomiale négative semble intéressant. Reste à savoir comment déterminer le coefficient ϕ . Et si nos paramètres sont les bons.

Le résumé de notre modèle révèle des informations intéressantes. La performance d'une régression logistique est évaluée avec des métriques clés spécifiques : - AIC (Critère d'information d'Akaike): Il mesure l'ajustement lorsqu'une pénalité est appliquée au nombre de paramètres. Des valeurs AIC plus petites indiquent que le modèle est plus proche de la vérité. - Null deviance : Il s'agit de la déviance du modèle nul, c'est-à-dire qu'il est caractérisé par aucun facteur. - Residual deviance : Il s'agit de la déviance du modèle avec toutes les variables. - Number of Fisher Scoring iterations : Il s'agit du nombre d'itérations avant la convergence.

3.2 Sévérité des sinistres

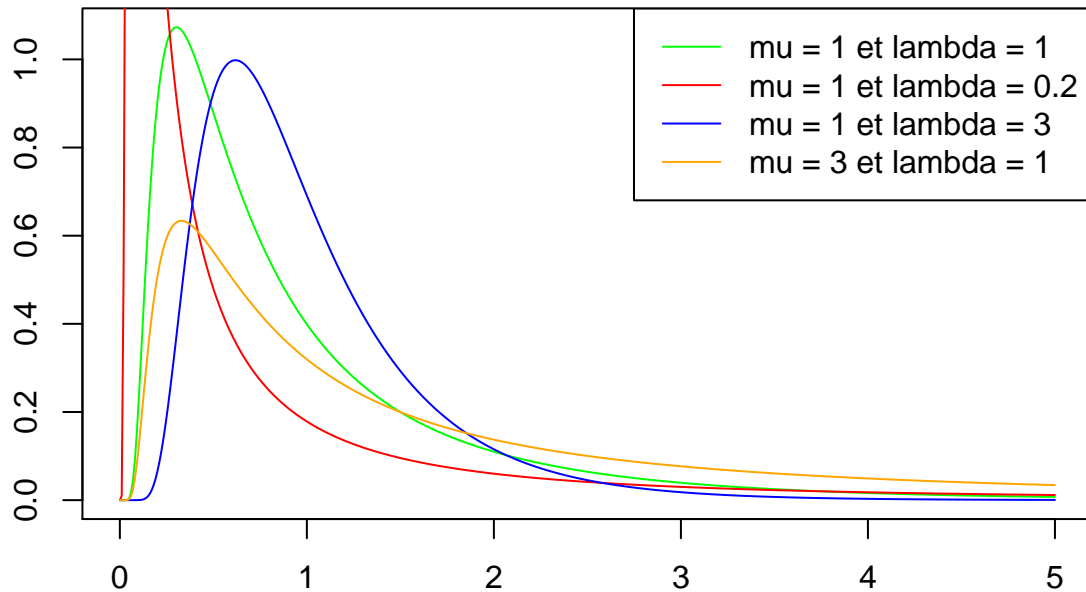
3.2.1 Présentation des lois utilisables

3.2.1.1 loi de gamma (ou d'Euler)



3.2.2 Inverse gauss

distribution de loi inverse gaussienne



```
##
## Call:
## glm(formula = ClaimAmount ~ DrivAge + VehAge + VehUsage + BonusMalus +
##      VehBody + VehMaxSpeed, family = poisson(link = "log"), data = freMPL2,
##      offset = ClaimInd)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -76.73  -13.41  -10.78   -8.93   875.26
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)      5.456643   0.007017   777.652  <2e-16 ***
## DrivAge(20,25]    -0.424181   0.005738  -73.923  <2e-16 ***
## DrivAge(25,30]    -0.973242   0.005656 -172.068  <2e-16 ***
## DrivAge(30,35]    -1.064342   0.005651 -188.360  <2e-16 ***
## DrivAge(35,40]    -0.303377   0.005570  -54.468  <2e-16 ***
## DrivAge(40,45]    -0.809725   0.005685 -142.422  <2e-16 ***
## DrivAge(45,50]    -1.177359   0.005798 -203.072  <2e-16 ***
## DrivAge(50,55]    -1.004243   0.005742 -174.880  <2e-16 ***
## DrivAge(55,60]    -0.609482   0.005697 -106.984  <2e-16 ***
## DrivAge(60,65]    -0.576607   0.005887  -97.942  <2e-16 ***
## DrivAge(65,70]    -0.467455   0.006060  -77.134  <2e-16 ***
## DrivAge(70,75]    -0.787713   0.006473 -121.685  <2e-16 ***
```

```

## DrivAge(75,80] -1.756620 0.008901 -197.342 <2e-16 ***
## DrivAge(80,104] 0.092281 0.006397 14.426 <2e-16 ***
## VehAge1 0.329440 0.002300 143.255 <2e-16 ***
## VehAge10+ -0.147633 0.002128 -69.363 <2e-16 ***
## VehAge2 0.420784 0.002251 186.928 <2e-16 ***
## VehAge3 0.208768 0.002397 87.085 <2e-16 ***
## VehAge4 0.130719 0.002472 52.884 <2e-16 ***
## VehAge5 0.805306 0.002168 371.458 <2e-16 ***
## VehAge6-7 -0.032855 0.002438 -13.478 <2e-16 ***
## VehAge8-9 -0.205910 0.002566 -80.249 <2e-16 ***
## VehUsageprivée et trajet vers bureau 0.173977 0.001266 137.379 <2e-16 ***
## VehUsageprofessionnel 0.249854 0.001522 164.168 <2e-16 ***
## VehUsagetrajet professionnel -0.096946 0.004608 -21.037 <2e-16 ***
## BonusMalus(100,351] 0.996682 0.001638 608.318 <2e-16 ***
## VehBodymicrovan 1.077846 0.003330 323.654 <2e-16 ***
## VehBodyautobus -0.479404 0.009816 -48.841 <2e-16 ***
## VehBodycoupé -0.136407 0.003936 -34.655 <2e-16 ***
## VehBodyautre microvan 0.238155 0.003566 66.777 <2e-16 ***
## VehBodyberline -0.198333 0.002929 -67.716 <2e-16 ***
## VehBodySUV -0.003896 0.003591 -1.085 0.278
## VehBodybreak -0.257014 0.003834 -67.030 <2e-16 ***
## VehBodycamionnette 0.212400 0.003678 57.745 <2e-16 ***
## VehMaxSpeed130-140 km/h -0.910847 0.003628 -251.047 <2e-16 ***
## VehMaxSpeed140-150 km/h -0.547519 0.003201 -171.020 <2e-16 ***
## VehMaxSpeed150-160 km/h -0.639480 0.003051 -209.598 <2e-16 ***
## VehMaxSpeed160-170 km/h -0.614308 0.003054 -201.139 <2e-16 ***
## VehMaxSpeed170-180 km/h -0.295551 0.003043 -97.128 <2e-16 ***
## VehMaxSpeed180-190 km/h -1.092236 0.003359 -325.157 <2e-16 ***
## VehMaxSpeed190-200 km/h -0.624298 0.003331 -187.439 <2e-16 ***
## VehMaxSpeed200-220 km/h -0.726016 0.003448 -210.556 <2e-16 ***
## VehMaxSpeed220+ km/h -0.501455 0.003596 -139.461 <2e-16 ***
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
## Null deviance: 25385264 on 47496 degrees of freedom
## Residual deviance: 23249752 on 47454 degrees of freedom
## AIC: Inf
##
## Number of Fisher Scoring iterations: 8

## ClaimAmount Predict_ClaimAmount difference
## 1 0 97.38378 97.38378
## 2 0 67.96897 67.96897
## 3 0 74.46994 74.46994
## 4 0 210.30779 210.30779
## 5 0 136.20994 136.20994
## 6 0 136.20994 136.20994
## 7 1204 263.98761 940.01239
## 8 0 97.11561 97.11561
## 9 0 41.29664 41.29664
## 10 0 102.21140 102.21140

```


3.2.3 Calcul de la prime pure

Soit X le coût monétaire au risque

Selon le modèle général, $X = \text{SOMME de } 1 \text{ à } N \text{ des } B_k$

où N correspond au nombre de sinistres et B_k correspond au montant de sinistres

Autrement dit, N représente la fréquence (variable discrète) et B_k la sévérité (variable continue positive)

En admettant que la fréquence n'a pas d'influence sur la sévérité et que les montants des sinistres ont le même comportement aléatoire, on a : $E(X) = E(N).E(B)$ (prime pure).

Comment calculer $E(N)$? Comment calculer $E(B)$?

3.2.4 Calcul de l'espérance $E(N)$

FORMULE : $E(N/x) = g^{-1}(aX_i + e_i)$ avec g la fonction de lien et e l'offset

Ainsi, pour une loi de Poisson, on a : $E(N/x) = \exp(aX_i + \ln(\text{Exposure})) = \text{Exposure} \cdot \exp(aX_i)$

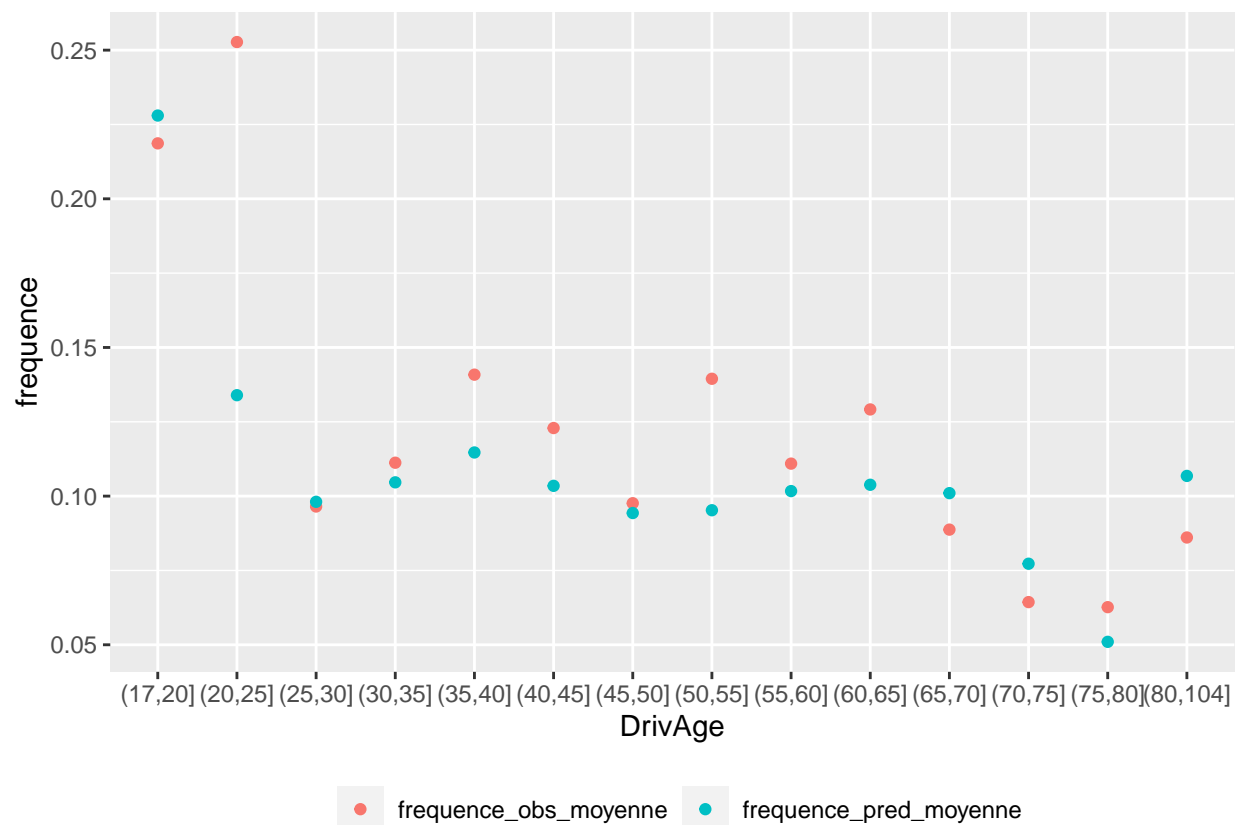
4 GAM

Poisson

```
##
## Call: gam(formula = ClaimInd ~ DrivAge + VehAge + VehUsage + BonusMalus +
##      VehBody + VehMaxSpeed + ClaimAmount + offset(log(Exposure)),
##      family = poisson("log"), data = freMPL2)
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -4.0092 -0.3517 -0.2669 -0.1754  3.5271
##
## (Dispersion Parameter for poisson family taken to be 1)
##
##      Null Deviance: 13242.19 on 47496 degrees of freedom
## Residual Deviance: 12133.83 on 47453 degrees of freedom
## AIC: 16489.83
##
## Number of Local Scoring Iterations: 13
##
## Anova for Parametric Effects
##           Df Sum Sq Mean Sq F value    Pr(>F)
## DrivAge     13      119      9.17   8.4573 < 2.2e-16 ***
## VehAge       8       115     14.42  13.2945 < 2.2e-16 ***
## VehUsage     3        44     14.66  13.5159 8.223e-09 ***
## BonusMalus   1       136    135.59 125.0455 < 2.2e-16 ***
## VehBody      8        96     12.05  11.1088 8.175e-16 ***
## VehMaxSpeed  9        33      3.62   3.3372 0.0004341 ***
## ClaimAmount  1       567    566.78 522.6860 < 2.2e-16 ***
## Residuals  47453    51456      1.08
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##      ClaimInd Predict_ClaimInd difference
## 1         0      0.055373732 0.055373732
## 2         0      0.037528548 0.037528548
## 3         0      0.060909897 0.060909897
## 4         0      0.021647032 0.021647032
## 5         0      0.009361829 0.009361829
## 6         0      0.042297419 0.042297419
## 7         1      0.072709703 0.927290297
## 8         0      0.068518516 0.068518516
## 9         0      0.021225067 0.021225067
## 10        0      0.049482260 0.049482260
```

```
##      ClaimInd Predict_ClaimInd difference
## 30734         1      0.5663232 0.4336768
```



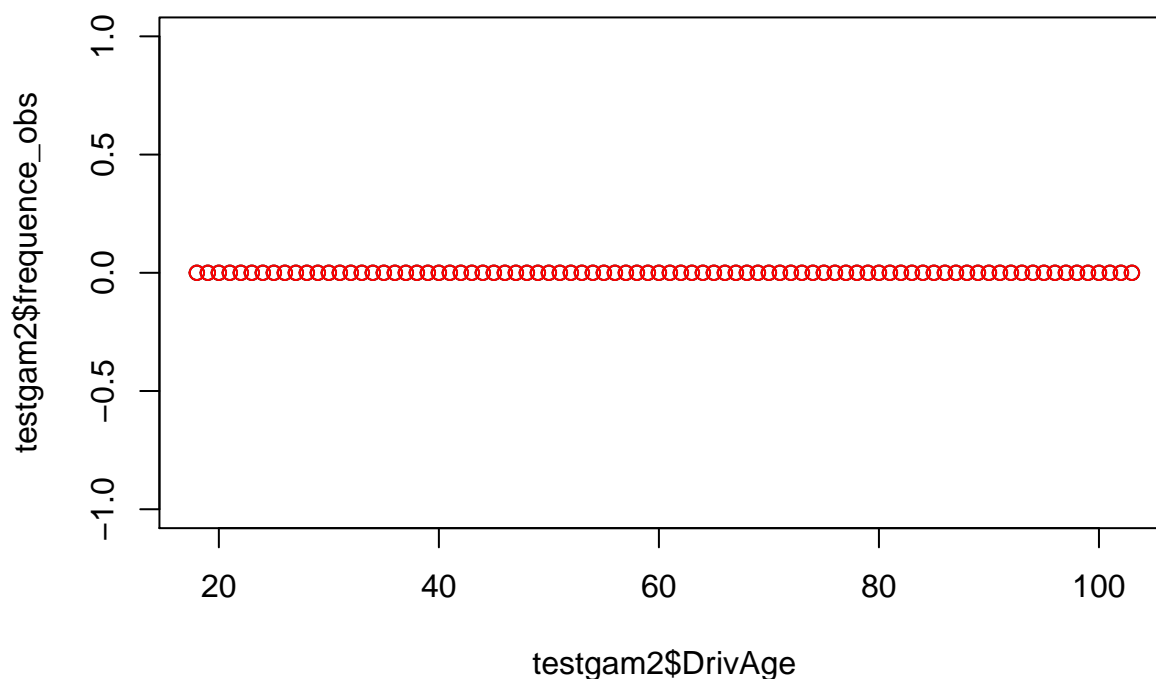
quasipoisson

```
##
## Call: gam(formula = ClaimInd ~ DrivAge + VehAge + VehUsage + BonusMalus +
##      VehBody + VehMaxSpeed + ClaimAmount + offset(log(Exposure)),
##      family = quasipoisson("log"), data = freMPL2)
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -4.0092 -0.3517 -0.2669 -0.1754  3.5271
##
## (Dispersion Parameter for quasipoisson family taken to be 1.0844)
```

```
##
##      Null Deviance: 13242.19 on 47496 degrees of freedom
## Residual Deviance: 12133.83 on 47453 degrees of freedom
## AIC: NA
##
## Number of Local Scoring Iterations: 13
##
## Anova for Parametric Effects
##      Df Sum Sq Mean Sq F value    Pr(>F)
## DrivAge      13      119      9.17    8.4573 < 2.2e-16 ***
## VehAge        8      115     14.42   13.2945 < 2.2e-16 ***
## VehUsage       3       44     14.66   13.5159 8.223e-09 ***
## BonusMalus     1      136   135.59 125.0455 < 2.2e-16 ***
## VehBody        8       96     12.05   11.1088 8.175e-16 ***
## VehMaxSpeed     9       33      3.62    3.3372 0.0004341 ***
## ClaimAmount     1      567   566.78 522.6860 < 2.2e-16 ***
## Residuals  47453   51456      1.08
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##      ClaimInd Predict_ClaimInd difference
## 1           0      0.055373732 0.055373732
## 2           0      0.037528548 0.037528548
## 3           0      0.060909897 0.060909897
## 4           0      0.021647032 0.021647032
## 5           0      0.009361829 0.009361829
## 6           0      0.042297419 0.042297419
## 7           1      0.072709703 0.927290297
## 8           0      0.068518516 0.068518516
## 9           0      0.021225067 0.021225067
## 10          0      0.049482260 0.049482260
```

```
##      ClaimInd Predict_ClaimInd difference
## 30734          1      0.5663232 0.4336768
```



binomiale negative

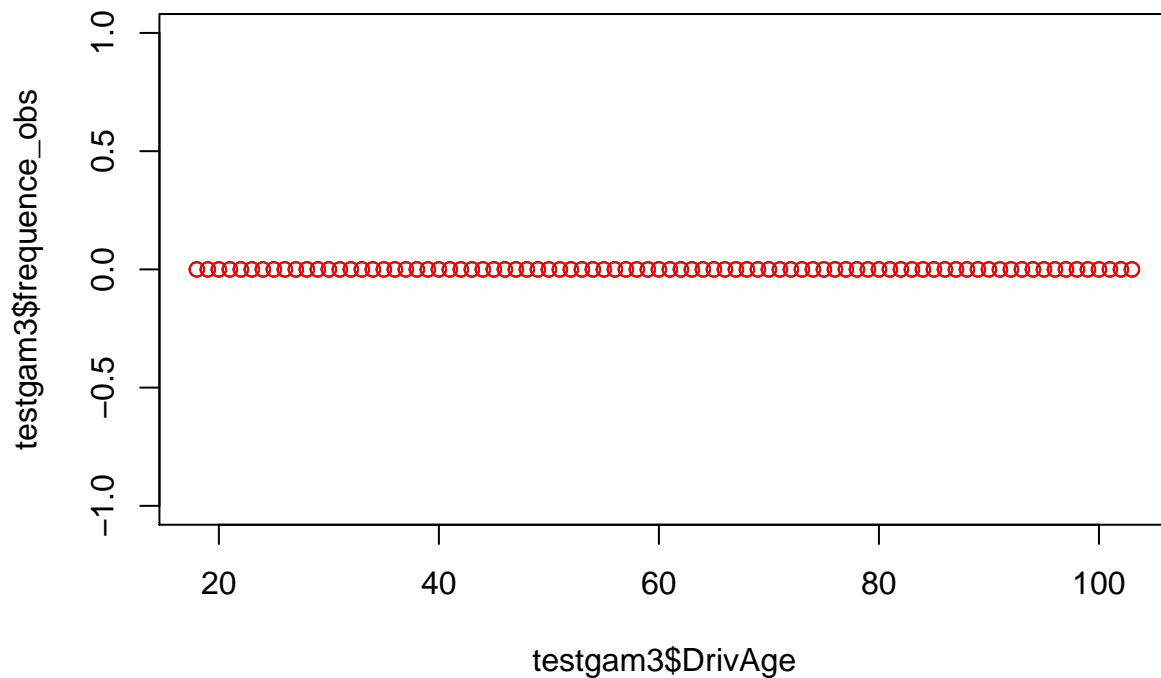
```
##
## Call: gam(formula = ClaimInd ~ DrivAge + VehAge + VehUsage + BonusMalus +
##       VehBody + VehMaxSpeed + ClaimAmount + offset(log(Exposure)),
##       family = negative.binomial(phi), data = freMPL2)
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -10.7020  -0.3191  -0.2446  -0.1614   3.3291
##
## (Dispersion Parameter for Negative Binomial(1) family taken to be 0.7152)
##
##      Null Deviance: 11687.96 on 47496 degrees of freedom
## Residual Deviance: 9354.674 on 47453 degrees of freedom
## AIC: 15359.38
##
## Number of Local Scoring Iterations: 30
##
## Anova for Parametric Effects
##
```

	Df	Sum Sq	Mean Sq	F value	Pr(>F)
DrivAge	13	272	20.9	29.222	< 2.2e-16 ***
VehAge	8	212	26.4	36.980	< 2.2e-16 ***
VehUsage	3	28	9.3	13.014	1.715e-08 ***
BonusMalus	1	202	201.6	281.894	< 2.2e-16 ***
VehBody	8	366	45.7	63.964	< 2.2e-16 ***
VehMaxSpeed	9	112	12.4	17.387	< 2.2e-16 ***

```
## ClaimAmount      1  16387 16387.4 22912.767 < 2.2e-16 ***
## Residuals    47453  33939      0.7
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##      ClaimInd Predict_ClaimInd  difference
## 1          0      0.035458371 0.035458371
## 2          0      0.032945147 0.032945147
## 3          0      0.048772592 0.048772592
## 4          0      0.018672176 0.018672176
## 5          0      0.008746301 0.008746301
## 6          0      0.039516421 0.039516421
## 7          1      0.101983460 0.898016540
## 8          0      0.054841158 0.054841158
## 9          0      0.018109026 0.018109026
## 10         0      0.041347986 0.041347986
```

```
##      ClaimInd Predict_ClaimInd  difference
## 21987         1      1.0174463 0.01744631
## 21995         1      1.0717149 0.07171490
## 45295         1      0.9161833 0.08381670
```



5 Bibliographie

6 Annexes

6.1 Affichage de l'implementation de la fonction nettoyage_dataframe :

```
nettoyage_dataframe <- function(dt){  
  
  # Suppression des données des individus assurés moins d'un jour (Exposure)  
  dt <- subset(dt,dt$Exposure>1/365.25)  
  
  # Modification des données des individus ayant un ClaimAmount négatif  
  dt <- subset(dt,dt$ClaimAmount>=0)  
  
  # Suppression de la colonne associée au sexe de la personne et de ClaimInd  
  dt <- dt[,-6]  
  dt <- dt[,-21]  
  
  # Réduction du nombre de catégories socioprofessionnels  
  levels(dt$SocioCateg) <- c(levels(dt$SocioCateg), "CSP4", "CSP6",  
                             "CSP9")  
  
  for (i in 1:dim(dt)[1]){  
    if (dt$SocioCateg[i]%in%c("CSP1","CSP16","CSP18","CSP19")){  
      dt$SocioCateg[i]<-"CSP1"  
    }  
    if (dt$SocioCateg[i]%in%c("CSP2", "CSP20", "CSP21", "CSP22", "CSP23",  
                              "CSP25", "CSP26","CSP27", "CSP28")){  
      dt$SocioCateg[i]<-"CSP2"  
    }  
    if (dt$SocioCateg[i]%in%c("CSP3", "CSP30", "CSP31", "CSP32", "CSP33",  
                              "CSP35", "CSP36","CSP37", "CSP38", "CSP39")){  
      dt$SocioCateg[i]<-"CSP3"  
    }  
    if (dt$SocioCateg[i]%in%c("CSP40", "CSP41", "CSP42", "CSP43", "CSP46",  
                              "CSP47", "CSP48","CSP49")){  
      dt$SocioCateg[i]<-"CSP4"  
    }  
    if (dt$SocioCateg[i]%in%c("CSP5", "CSP50", "CSP51", "CSP55", "CSP56",  
                              "CSP57", "CSP59")){  
      dt$SocioCateg[i]<-"CSP5"  
    }  
    if (dt$SocioCateg[i]%in%c("CSP6", "CSP60", "CSP61", "CSP62", "CSP63",  
                              "CSP65", "CSP66")){  
      dt$SocioCateg[i]<-"CSP6"  
    }  
    if (dt$SocioCateg[i]%in%c("CSP7", "CSP70", "CSP73", "CSP74", "CSP77")){  
      dt$SocioCateg[i]<-"CSP7"  
    }  
    if (dt$SocioCateg[i]%in%c("CSP9", "CSP91")){  
      dt$SocioCateg[i]<-"CSP9"  
    }  
  }  
  dt$SocioCateg <- droplevels(dt$SocioCateg)
```

```

# Traduction des données (VehBody, MariStat, VehUsage, VehEngine, VehEnergy, Garage)
for (i in 1:dim(dt)[2]){
  # Type de véhicules
  if (colnames(dt)[i]=="VehBody"){
    levels(dt$VehBody) <- c(levels(dt$VehBody), "autobus", "coupé",
                           "autre microvan", "berline", "SUV", "break",
                           "camionnette")

    dt$VehBody[dt$VehBody == "bus"]<-"autobus"
    dt$VehBody[dt$VehBody == "coupe"]<-"coupé"
    dt$VehBody[dt$VehBody == "other microvan"]<-"autre microvan"
    dt$VehBody[dt$VehBody == "sedan"]<-"berline"
    dt$VehBody[dt$VehBody == "sport utility vehicle"]<-"SUV"
    dt$VehBody[dt$VehBody == "station wagon"]<-"break"
    dt$VehBody[dt$VehBody == "van"]<-"camionnette"
    dt$VehBody <- droplevels(dt$VehBody)
  }

  # Statut marital
  if (colnames(dt)[i]=="MariStat"){
    levels(dt$MariStat) <- c(levels(dt$MariStat), "célibataire", "autre")
    dt$MariStat[dt$MariStat == "Alone"]<-"célibataire"
    dt$MariStat[dt$MariStat == "Other"]<-"autre"
    dt$MariStat <- droplevels(dt$MariStat)
  }

  # Utilisation du véhicule
  if (colnames(dt)[i]=="VehUsage"){
    levels(dt$VehUsage) <- c(levels(dt$VehUsage), "privée",
                           "privée et trajet vers bureau", "professionnel",
                           "trajet professionnel" )

    dt$VehUsage[dt$VehUsage == "Private"]<-"privée"
    dt$VehUsage[dt$VehUsage == "Private+trip to office"]<-"
    "privée et trajet vers bureau"
    dt$VehUsage[dt$VehUsage == "Professional"]<-"professionnel"
    dt$VehUsage[dt$VehUsage == "Professional run"]<-"
    "trajet professionnel"
    dt$VehUsage <- droplevels(dt$VehUsage)
  }

  # Moteur du véhicule
  if (colnames(dt)[i]=="VehEngine"){
    levels(dt$VehEngine) <- c(levels(dt$VehEngine),
                           "injection directe surpuissante",
                           "électrique", "injection surpuissante")

    dt$VehEngine[dt$VehEngine == "direct injection overpowered"]<-"
    "injection directe surpuissante"
    dt$VehEngine[dt$VehEngine == "electric"]<-"électrique"
    dt$VehEngine[dt$VehEngine == "injection overpowered"]<-"
    "injection surpuissante"
    dt$VehEngine <- droplevels(dt$VehEngine)
  }

  # Energie utilisée par le véhicule
  if (colnames(dt)[i]=="VehEnergy"){
    levels(dt$VehEnergy) <- c(levels(dt$VehEnergy), "électrique", "essence")
    dt$VehEnergy[dt$VehEnergy == "regular"]<-"essence"
    dt$VehEnergy[dt$VehEnergy == "eletric"]<-"électrique"
  }
}

```

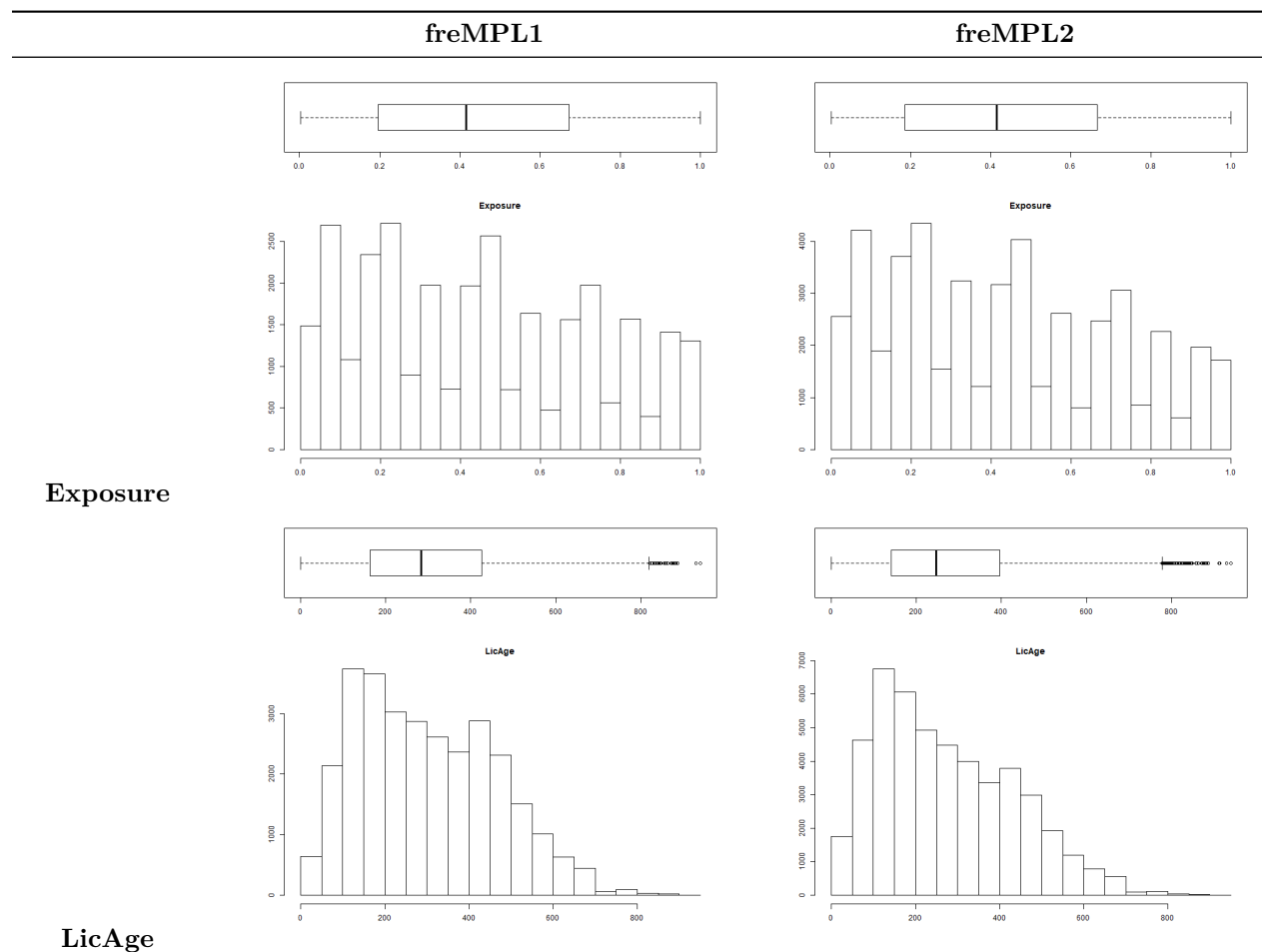
```

    dt$VehEnergy <- droplevels(dt$VehEnergy)
  }
  # Garage
  if (colnames(dt)[i]=="Garage"){
    levels(dt$Garage) <- c(levels(dt$Garage), "aucun", "garage indépendant",
                          "concessionnaire")

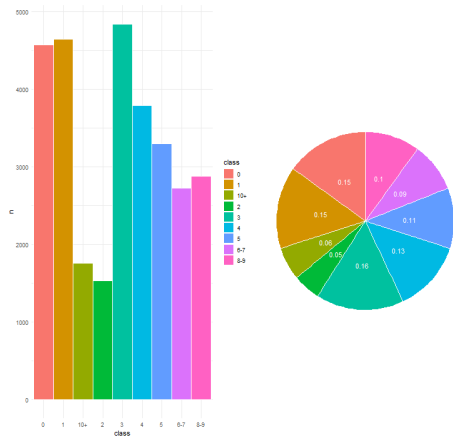
    dt$Garage[dt$Garage == "None"]<-"aucun"
    dt$Garage[dt$Garage == "Private garage"]<-"garage indépendant"
    dt$Garage[dt$Garage == "Collective garage"]<-"concessionnaire"
    dt$Garage <- droplevels(dt$Garage)
  }
}
return (dt)
}

```

6.2 Affichage de l'ensemble des représentations graphiques

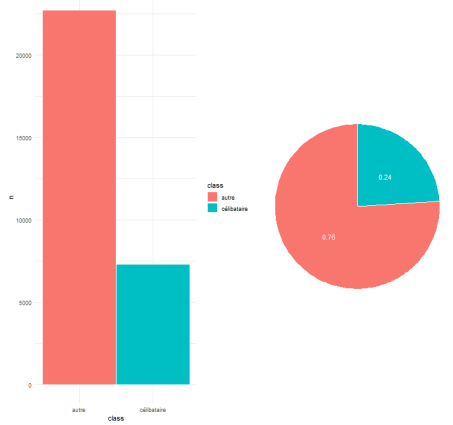
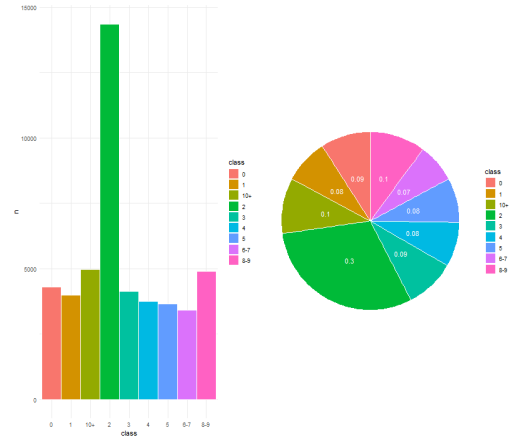


freMPL1

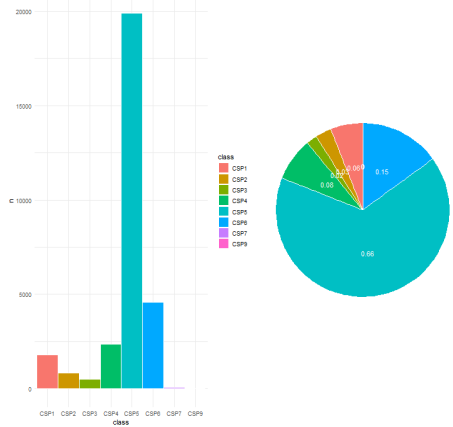
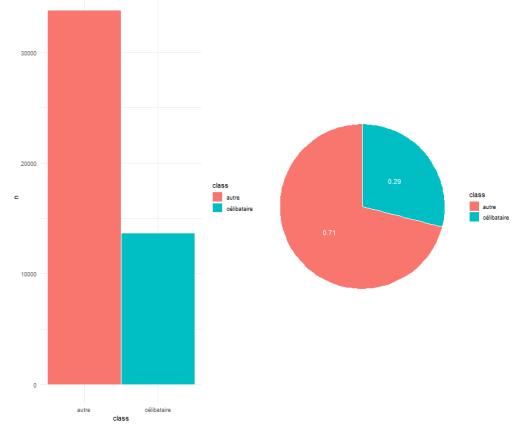


VehAge

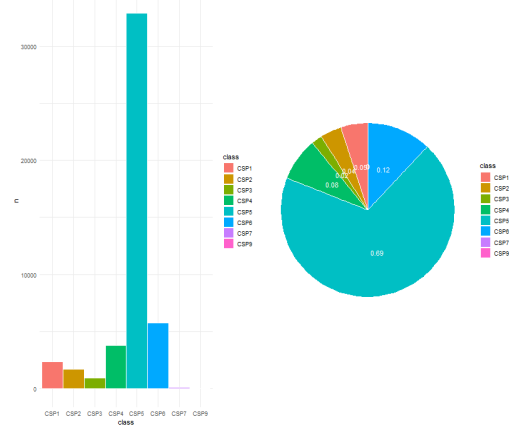
freMPL2



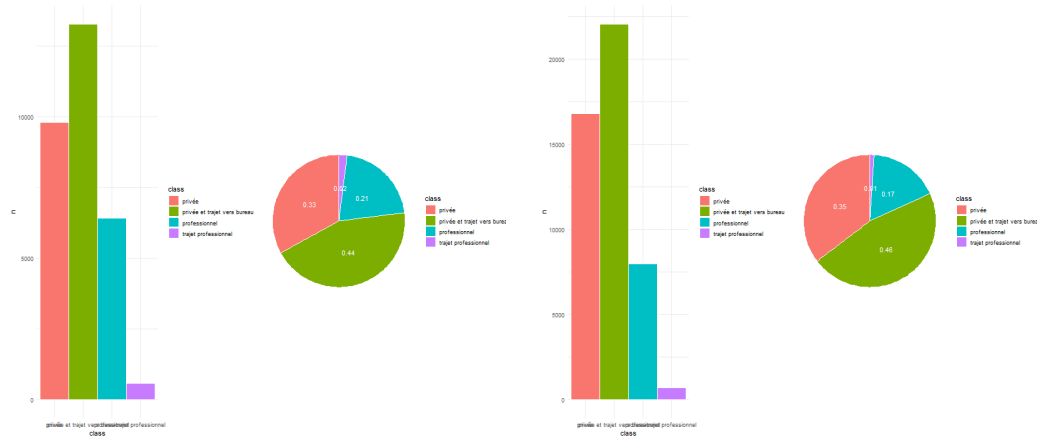
MariStat



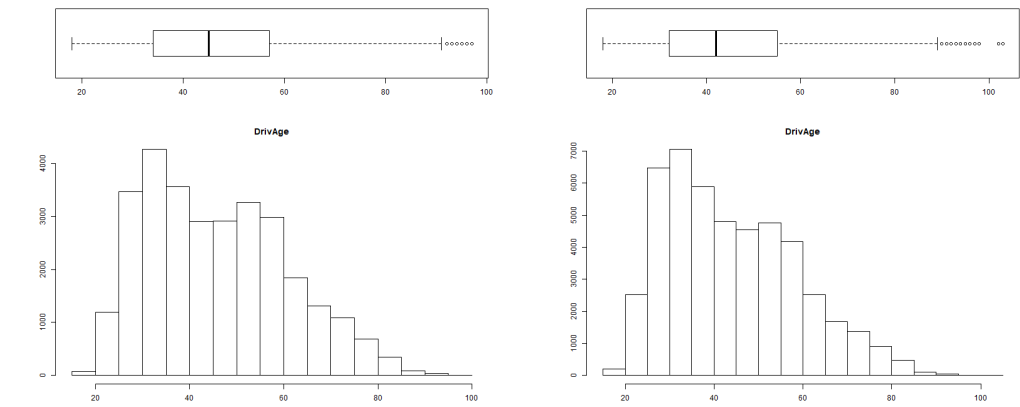
SocioCateg



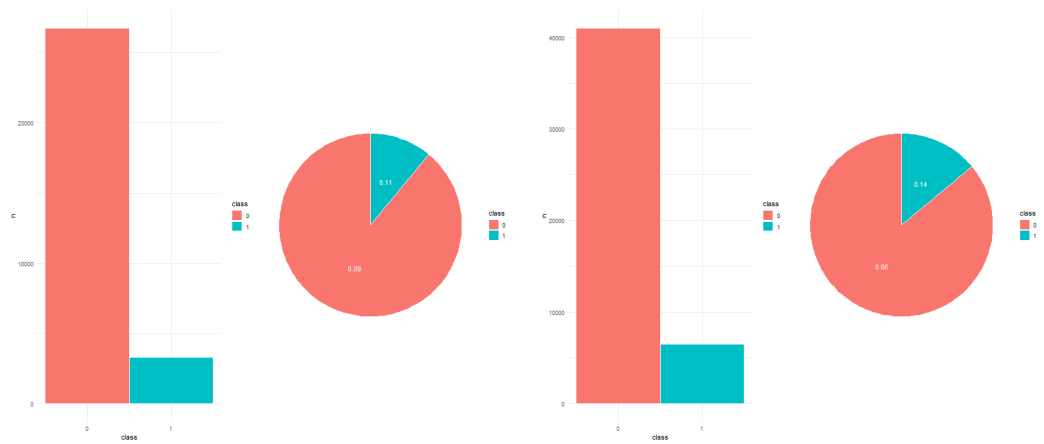
VehUsage



DrivAge

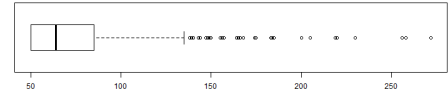
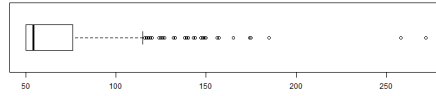


HasKmLimit



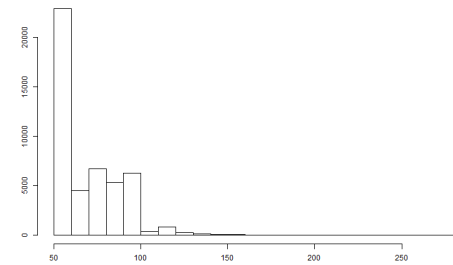
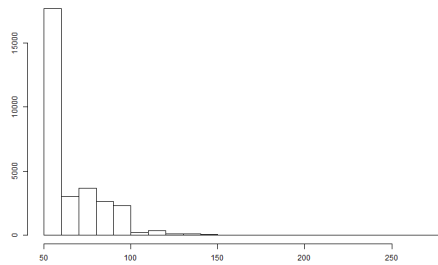
freMPL1

freMPL2

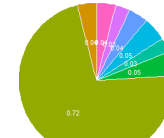
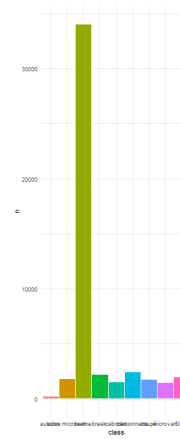
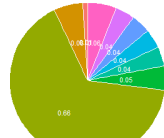
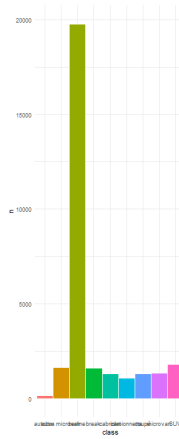


BonusMalus

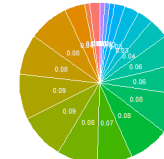
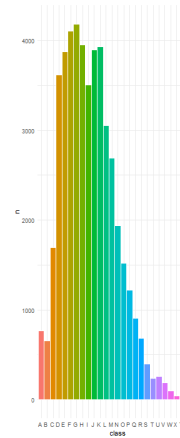
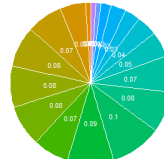
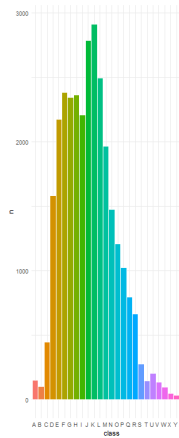
BonusMalus



BonusMalus

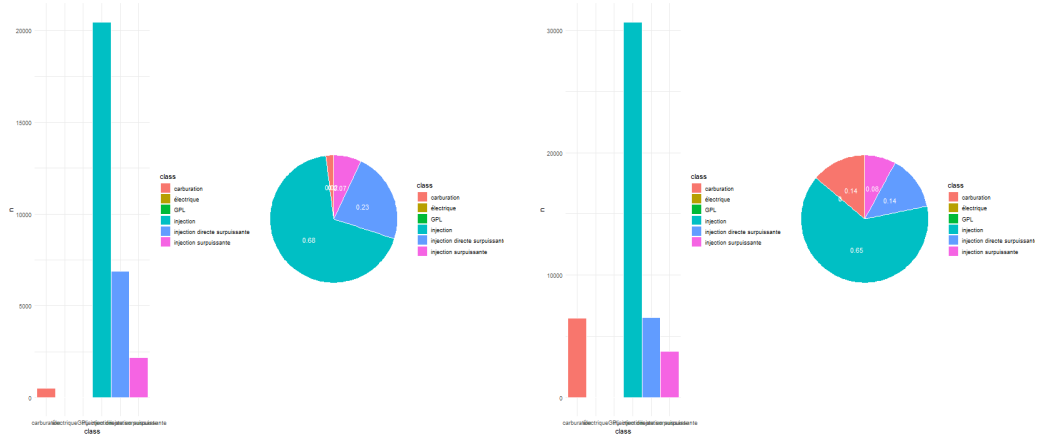


VehBody

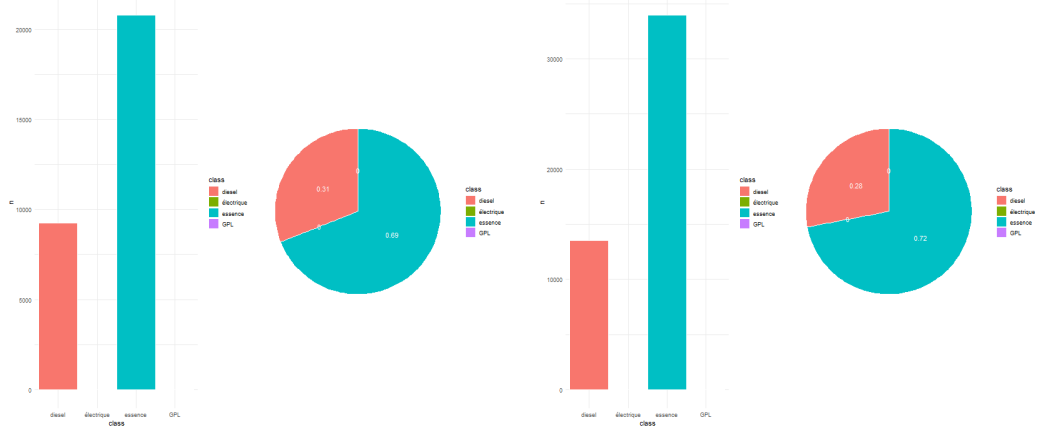


VehPrice

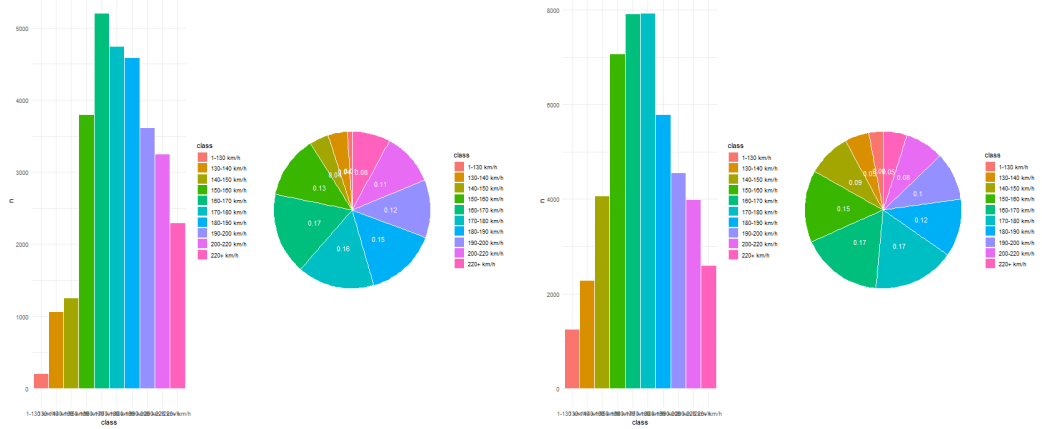
VehEngine



VehEnergy



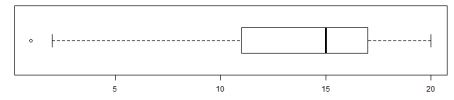
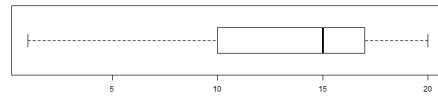
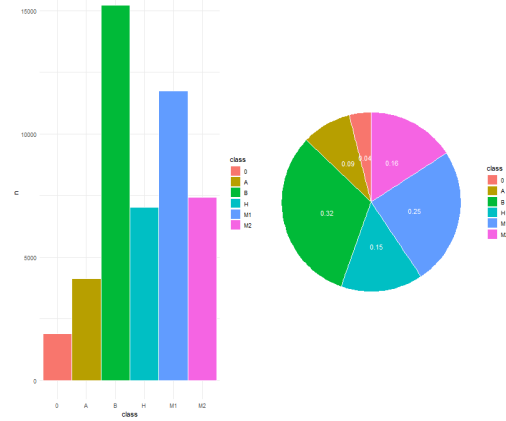
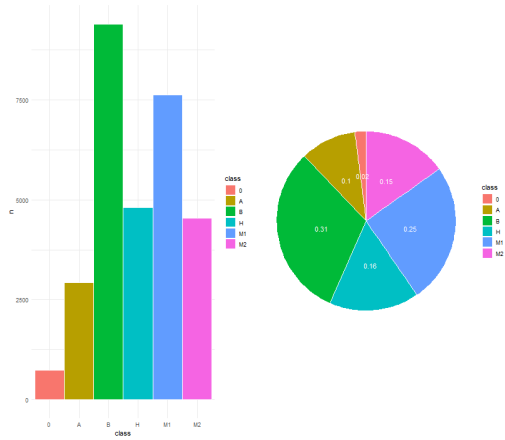
VehMaxSpeed



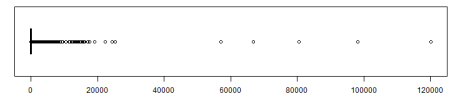
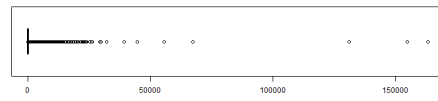
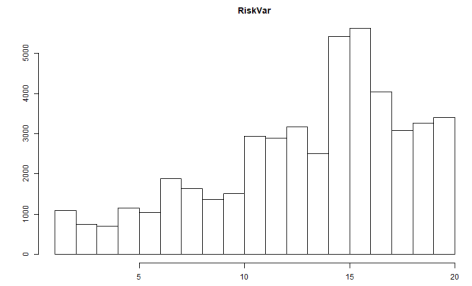
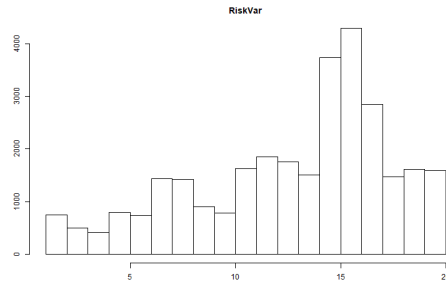
freMPL1

freMPL2

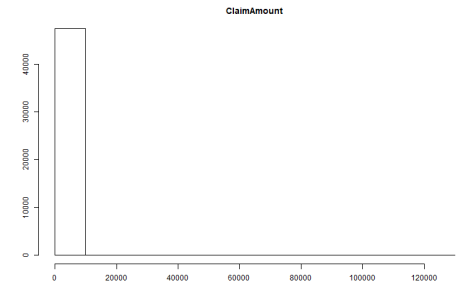
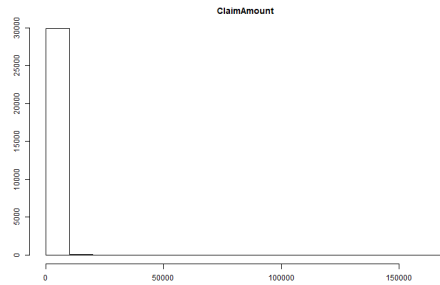
VehClass



RiskVar



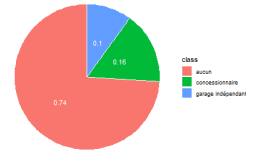
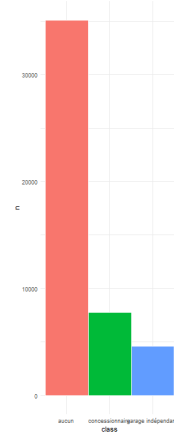
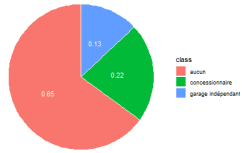
ClaimAmount



freMPL1

freMPL2

Garage



ClaimInd

