Projet Actuariat

Solène Corre, Florentin Dehooghe, François Delhaye

13 avril 2020

Table des matières

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

De même, les dimmensions 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. es 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 renseigner, 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_catégories_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 kilométique.

— **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
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

	1	2	3
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 précisement 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

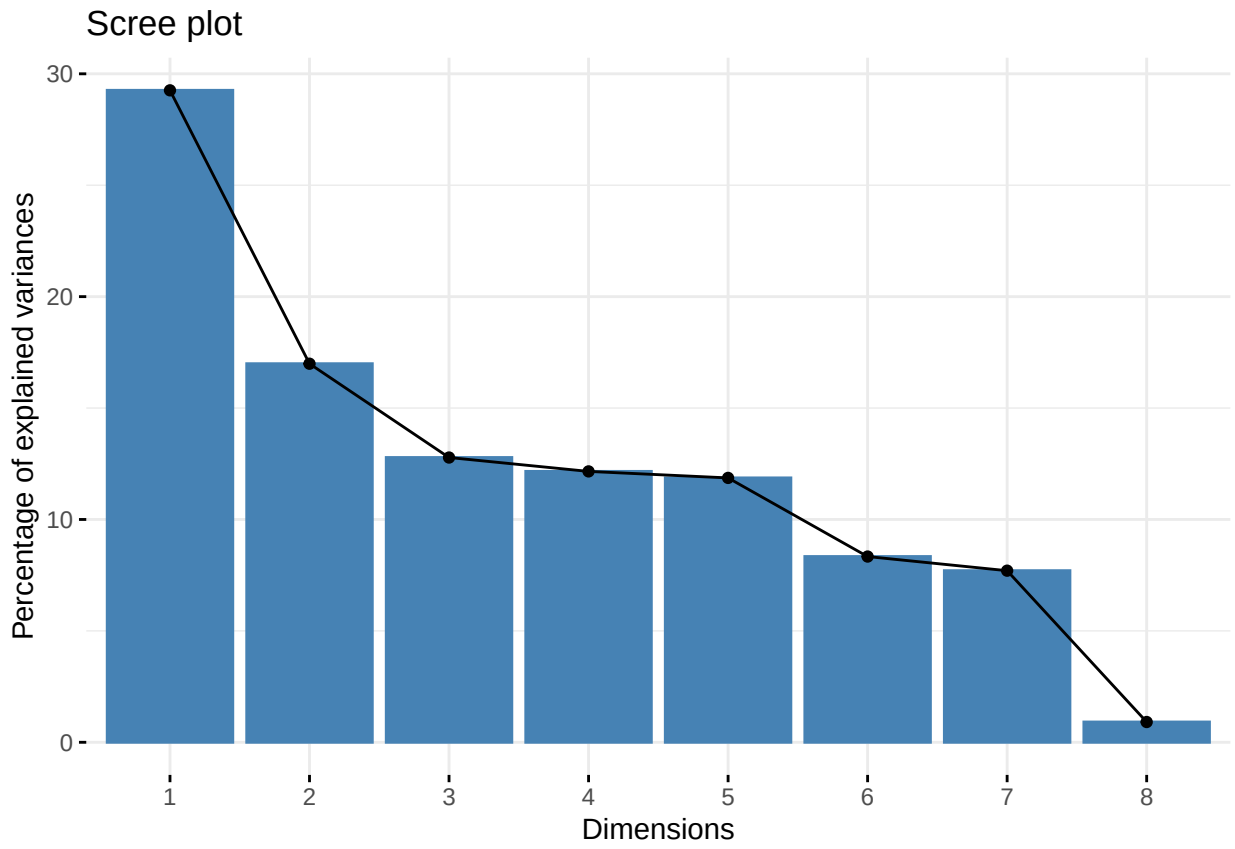
Attention, les valeurs doivent être numériques.

On va donc convertir nos valeurs en numérique :

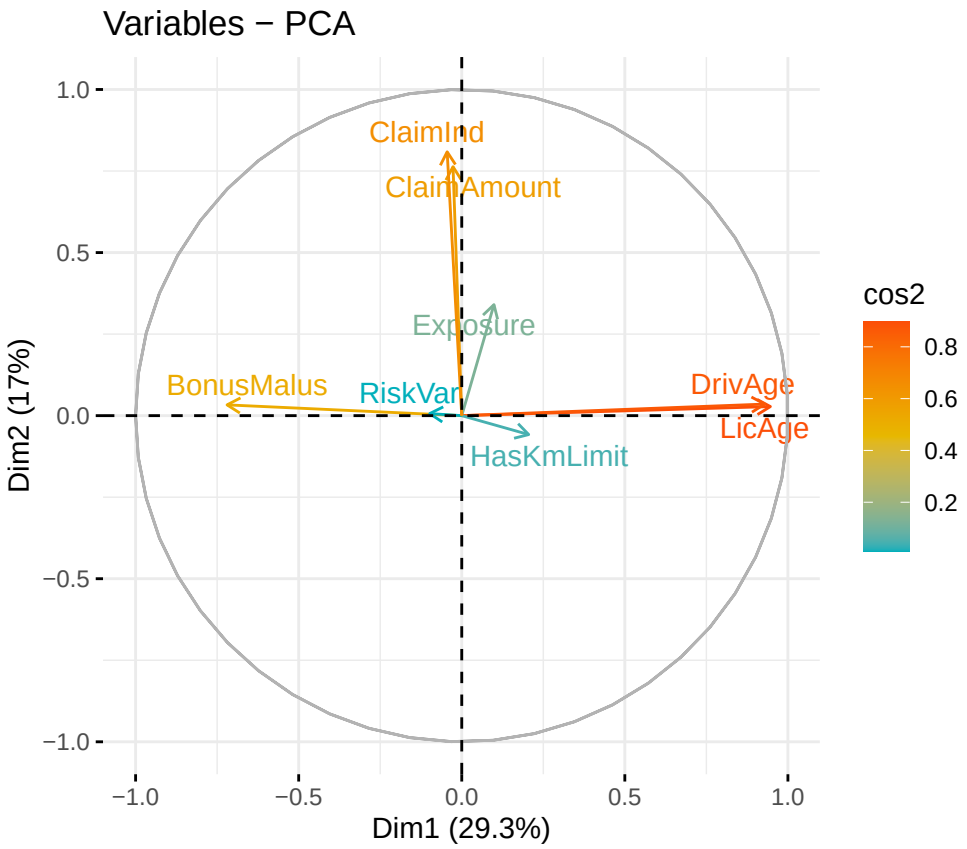
Certaines colonnes sont catégorisés et pourraient nous être utiles pour éxécuter notre ACP. Il n'est cependant pas judicieux d'appliquer une conversion numérique à ces colonnes puisqu'on leur attribue une valeur arbitraire nous faisant penser à une classification des différents facteurs possibles. Pour 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 :

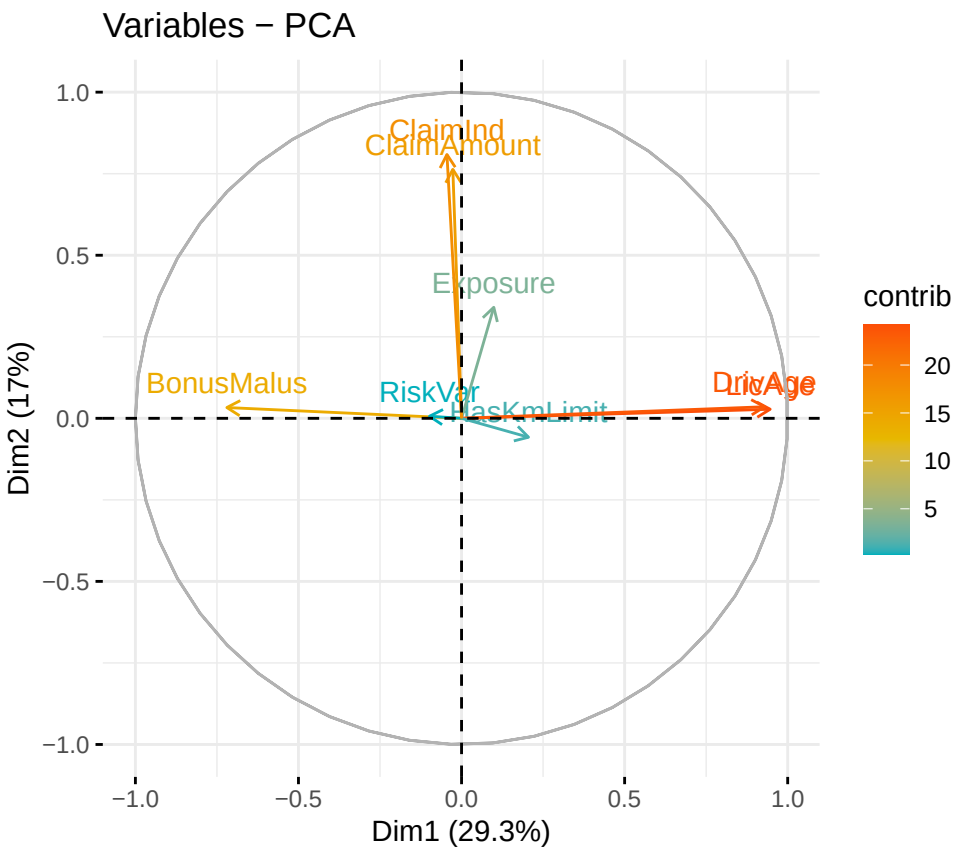
```
##       eigenvalue variance.percent cumulative.variance.percent
## Dim.1 2.34068251       29.2585314                    29.25853
## Dim.2 1.35919145       16.9898931                    46.24842
## Dim.3 1.02241078       12.7801347                    59.02856
## Dim.4 0.97257721       12.1572151                    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 cos2 pour juger de la qualité de la représentation :



contribution des colonnes aux dimensions :



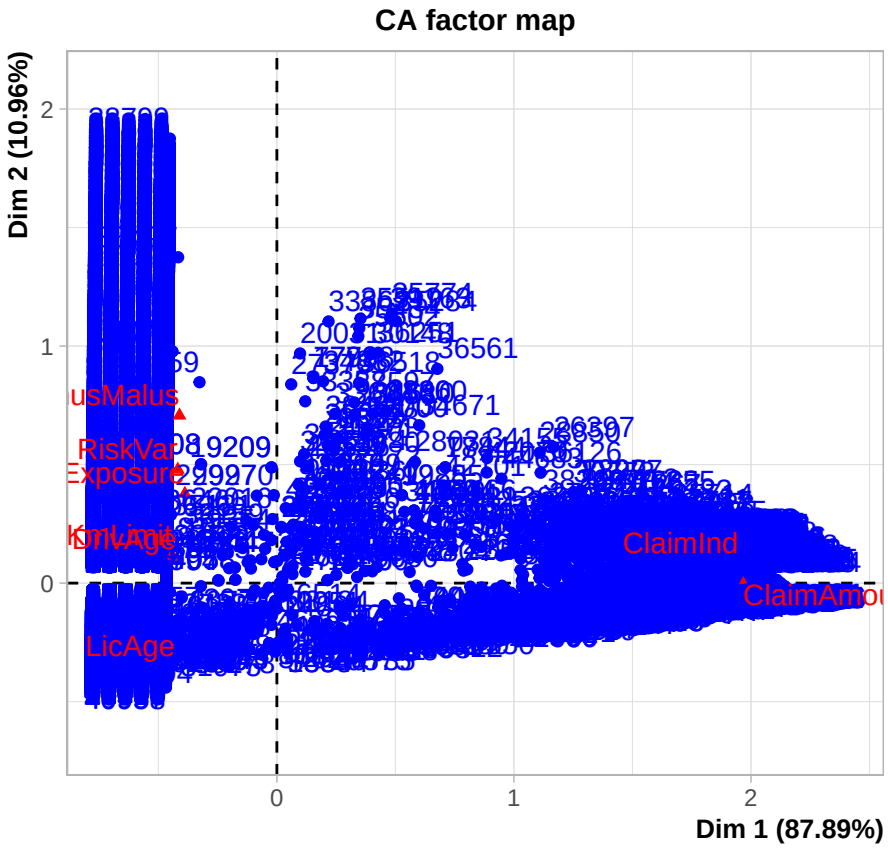
Description des dimensions

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

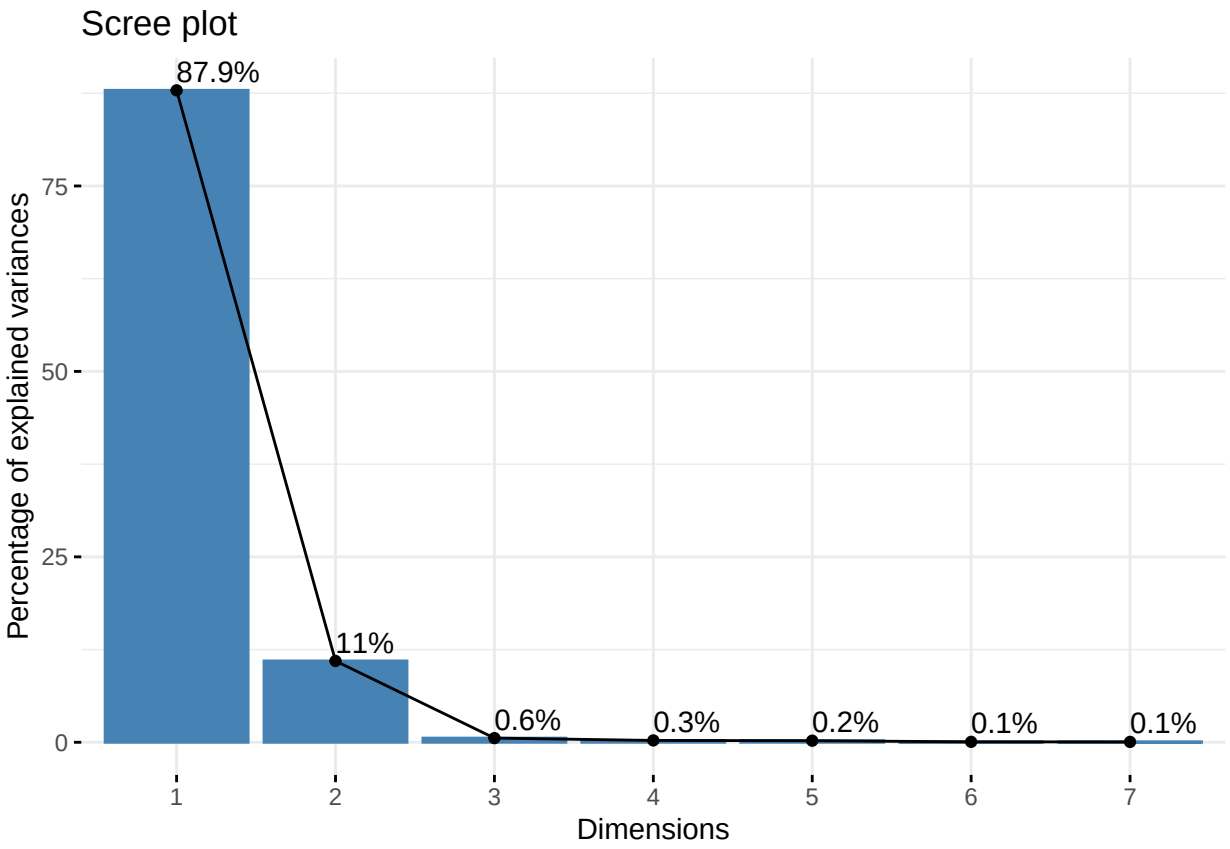
Notez également que la fonction dimdesc() [dans FactoMineR], pour dimension description (en anglais), peut être utilisée pour identifier les variables les plus significativement associées avec une composante principale donnée . Elle peut être utilisée comme suit:

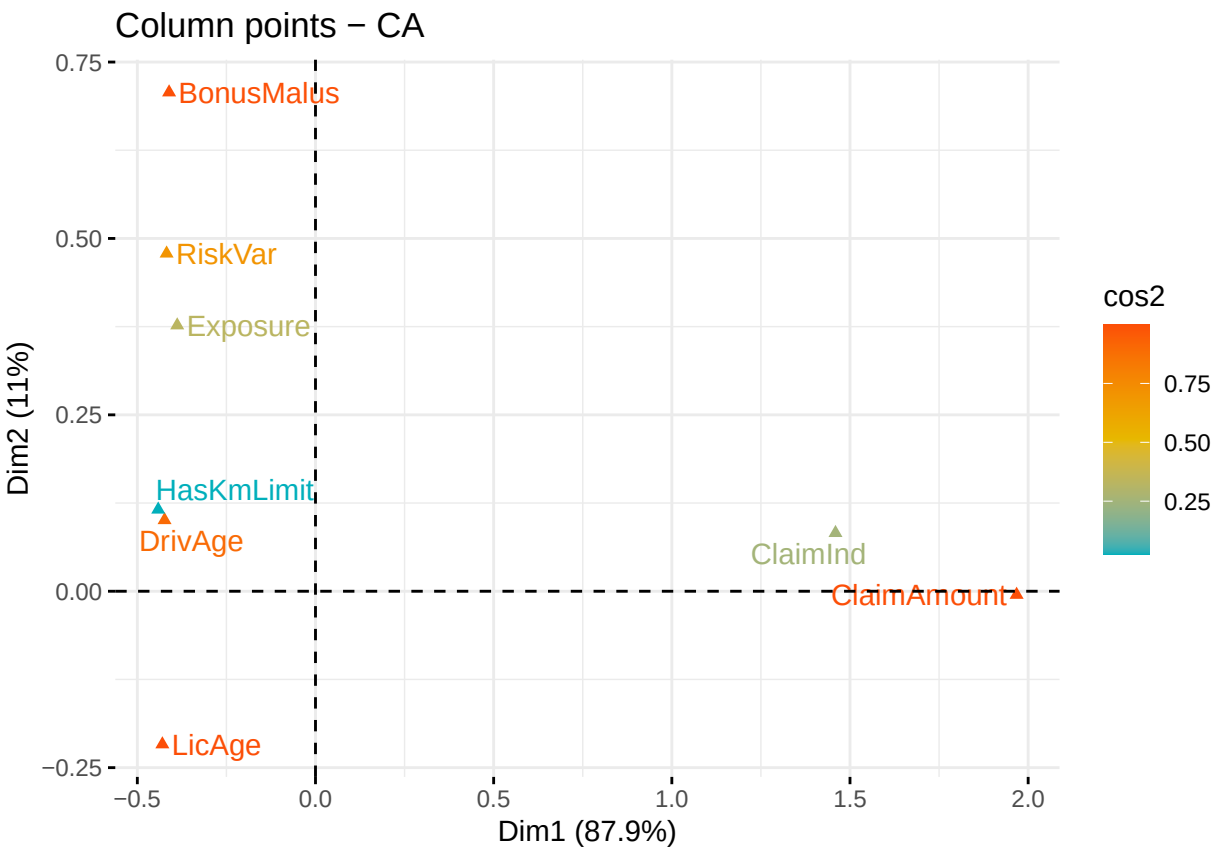
2.1.5 AFC

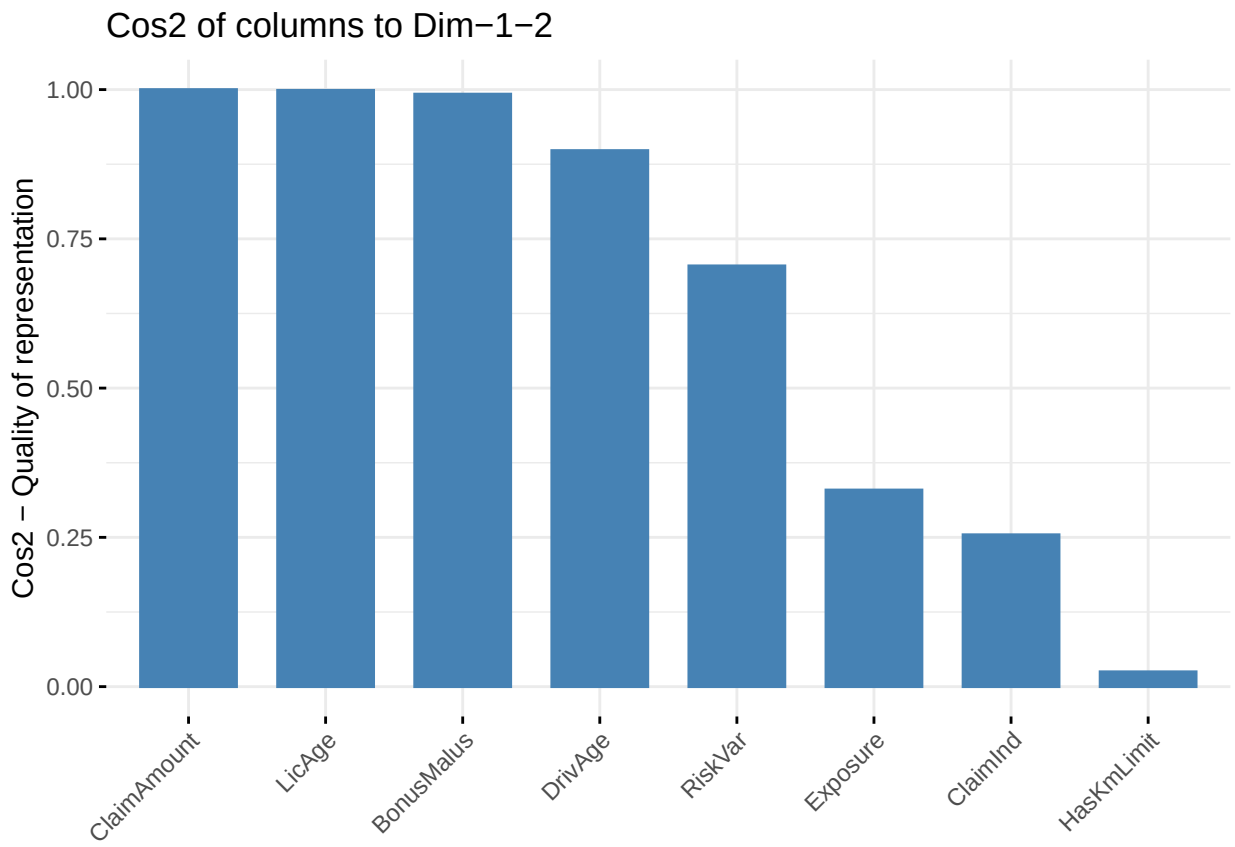
L' analyse factorielle des correspondances est une extension de l'analyse en composantes principales pour analyser l'association entre deux variables qualitatives (ou catégorielles). L'AFC permet de résumer et de visualiser l'information contenue dans le tableau de contingence formé par les deux variables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


Scree plot
87.9%
11%
0.6%
0.3%
0.2%
0.1%
0.1%
Percentage of explained variances
0
25
50
75
1
2
3
4
5
6
7
Dimensions


Column points − CA
BonusMalus
RiskVar
Exposure
HasKmLimit
DrivAge
ClaimInd
ClaimAmount
LicAge
0.75
0.50
0.25
0.00
−0.25
−0.5
0.0
0.5
1.0
1.5
2.0
Dim2 (11%)
Dim1 (87.9%)
cos2
0.75
0.50
0.25




```
##                 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
```

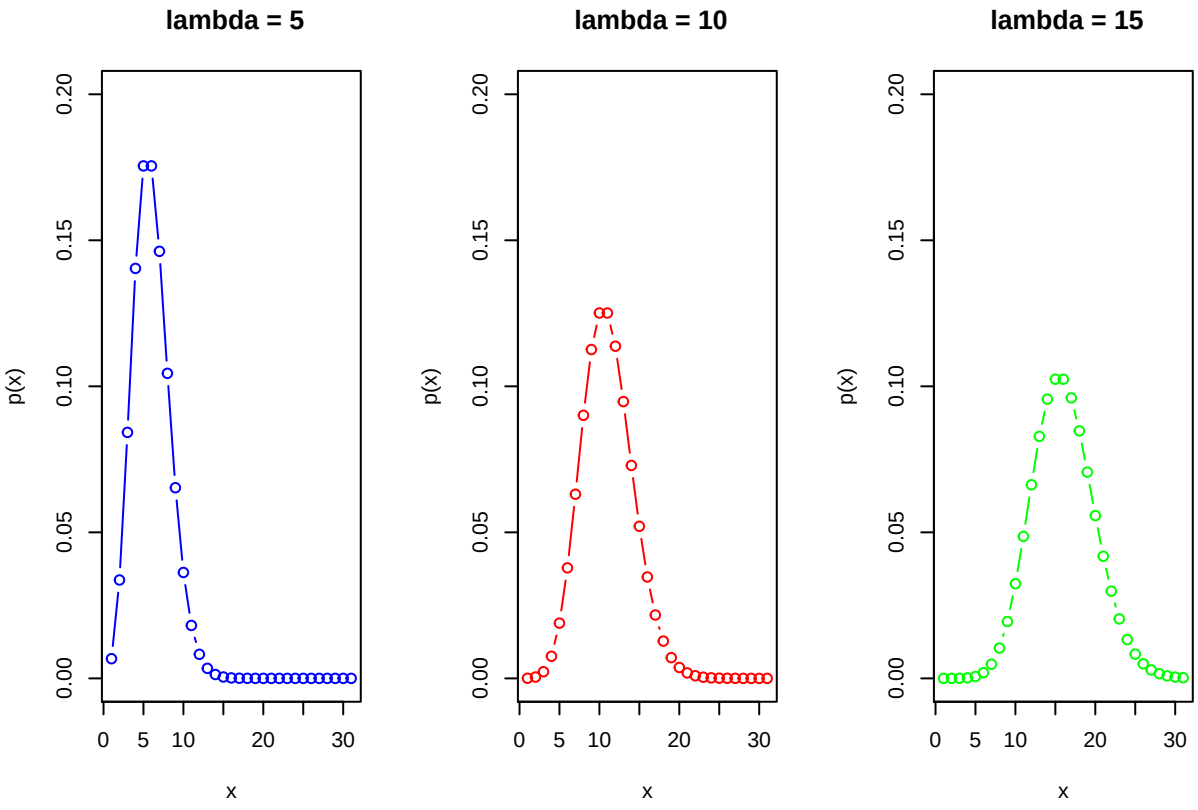
3 GLM

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

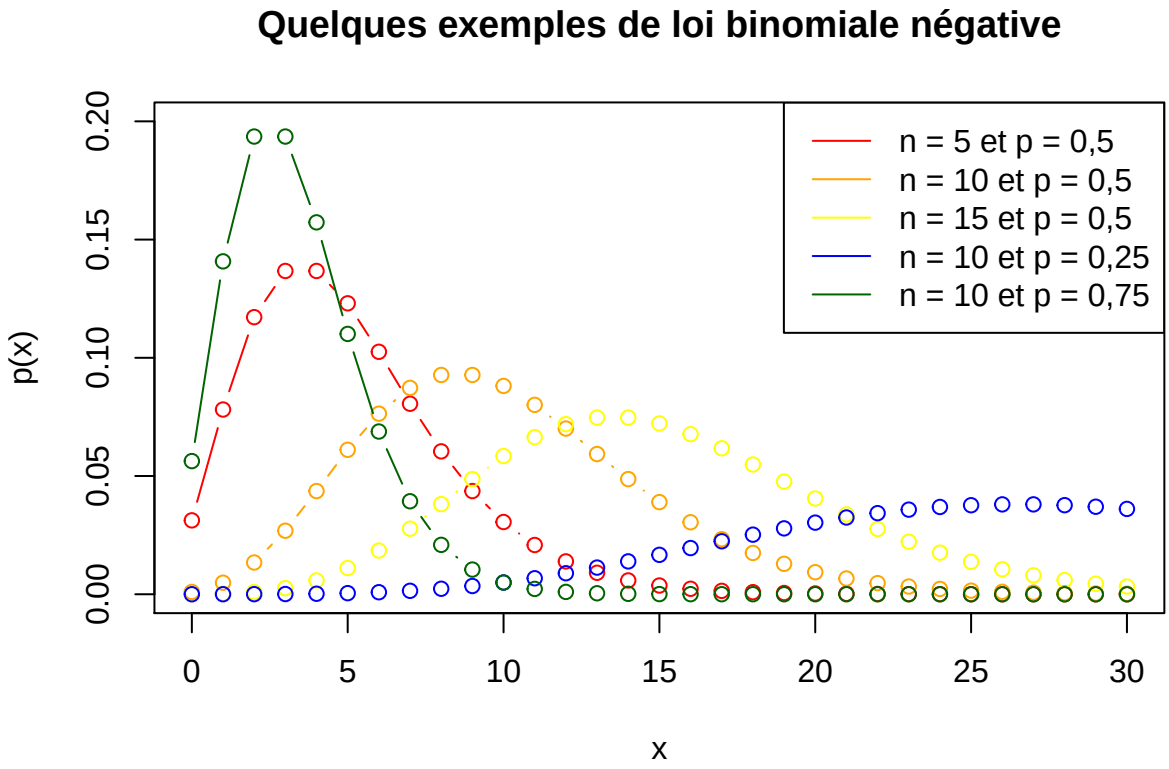
3.1 Fréquence des sinistres

3.1.1 Présentation des lois utilisables

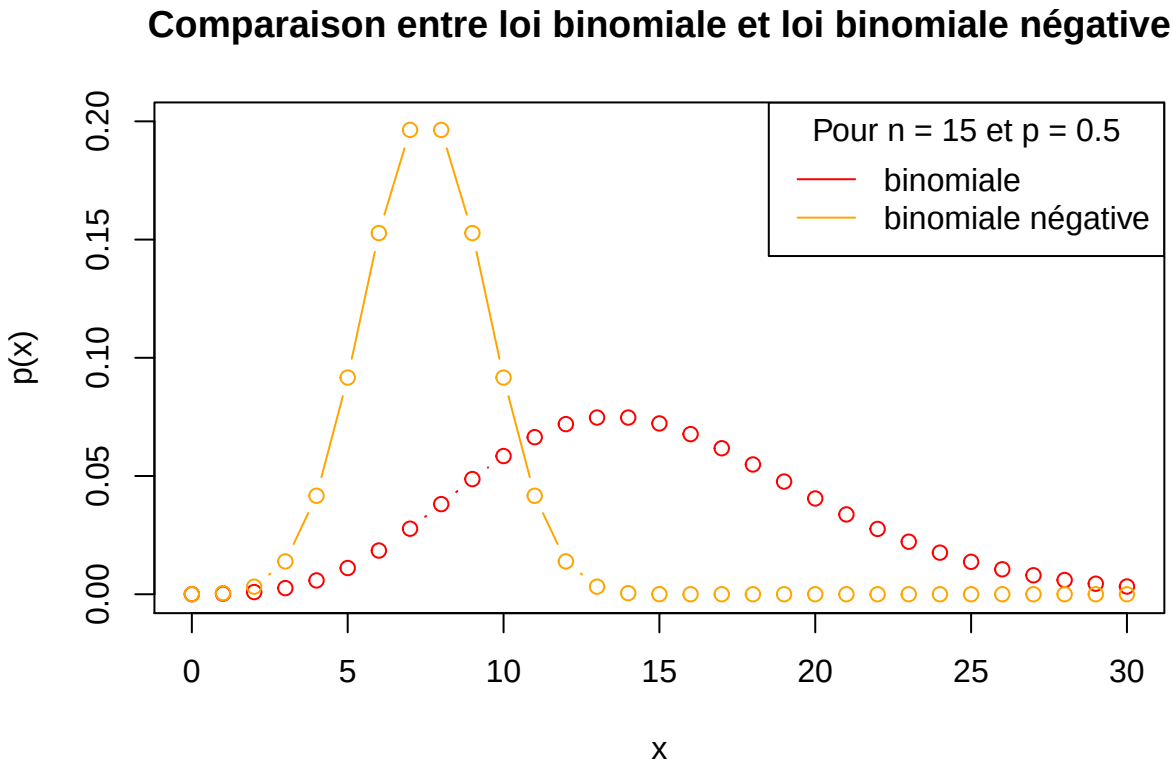
3.1.1.1 la loi de Poisson



3.1.1.2 la loi binomiale négative



comparaison binomiale et binomiale négative :



Test du modèle GLM avec la loi de Poisson :

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

```
##
## Call:
## glm(formula = ClaimInd ~ DrivAge + HasKmLimit + offset(log(Exposure)),
##     family = poisson("log"), data = freMPL2)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.4899  -0.3603  -0.2758  -0.1832   3.5914
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -2.025495   0.069449 -29.165  < 2e-16 ***
## DrivAge     -0.004983   0.001515  -3.289 0.001004 **
## HasKmLimit  -0.234207   0.070492  -3.322 0.000892 ***
## ---
## 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
```

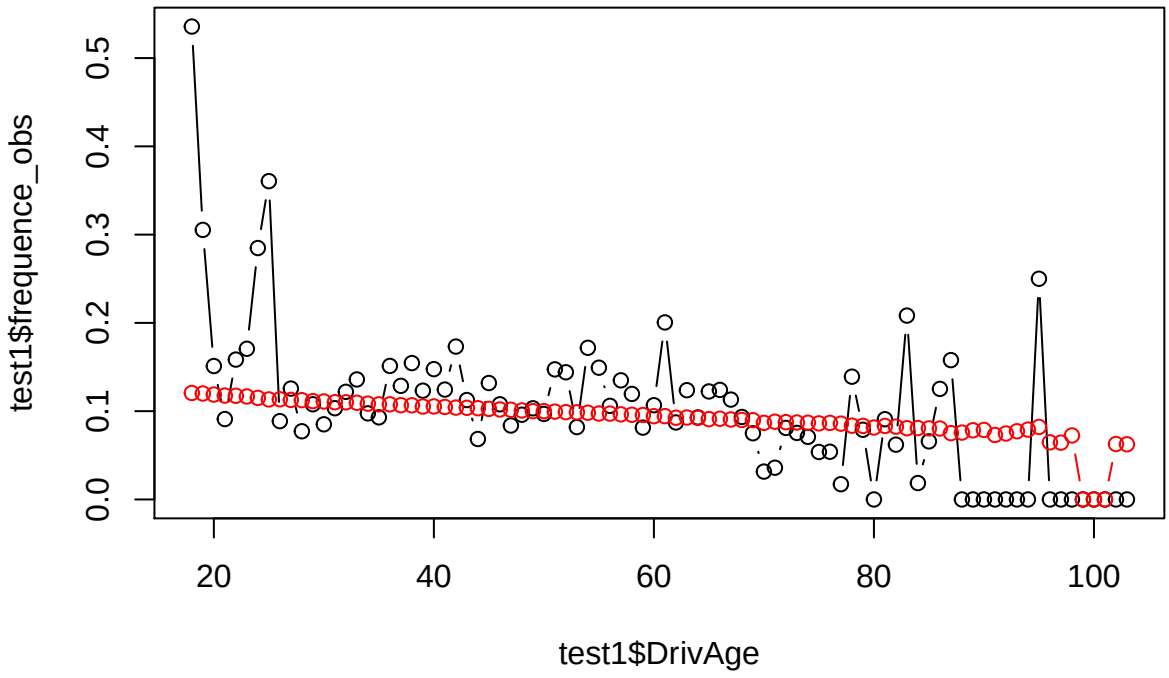
```
## Residual deviance: 12538  on 47494  degrees of freedom
## AIC: 16812
##
## Number of Fisher Scoring iterations: 6
```

```
#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      0.050860088 0.050860088
## 2         0      0.041725357 0.041725357
## 3         0      0.058475681 0.058475681
## 4         0      0.022273341 0.022273341
## 5         0      0.009289614 0.009289614
## 6         0      0.041762515 0.041762515
## 7         1      0.050653001 0.949346999
## 8         0      0.050551695 0.050551695
## 9         0      0.024521120 0.024521120
## 10        0      0.082285626 0.082285626
```

```
# recap1[recap1$difference<0.5 & recap1$ClaimInd == 1,]
 # Ce modèle ne fonctionne pas pour prédire les ClaimInd égaux à 1


test1 <- data.frame(
  DrivAge = c(18:103),
  nb_individu = rep(0,times = 103-18+1),
  total_obs = rep(0,times = 103-18+1),
  total_pred = rep(0,times = 103-18+1),
  frequence_obs = rep(0,times = 103-18+1),
  frequence_pred = rep(0,times = 103-18+1)
)

for (i in 1:dim(freMPL2)[1]){
  age <- freMPL2$DrivAge[i]
  test1[test1$DrivAge==age,"nb_individu"] <- test1[test1$DrivAge==age,"nb_individu"]+1
  test1[test1$DrivAge==age,"total_obs"] <-
    test1[test1$DrivAge==age,"total_obs"]+(freMPL2$ClaimInd[i]/freMPL2$Exposure[i])
  test1[test1$DrivAge==age,"total_pred"] <-
    test1[test1$DrivAge==age,"total_pred"]+(predict1[i]/freMPL2$Exposure[i])
}


for (j in 1:dim(test1)[1]){
  if (test1$nb_individu[j]!=0){
    test1[j,"frequence_obs"] <- test1[j,"total_obs"]/ test1[j,"nb_individu"]
```

```
    test1[j,"frequence_pred"] <- test1[j,"total_pred"]/ test1[j,"nb_individu"]
  }
}

plot(test1$DrivAge,test1$frequence_obs, type = "b")
lines(test1$DrivAge,test1$frequence_pred, type = "b", col = "red")
```



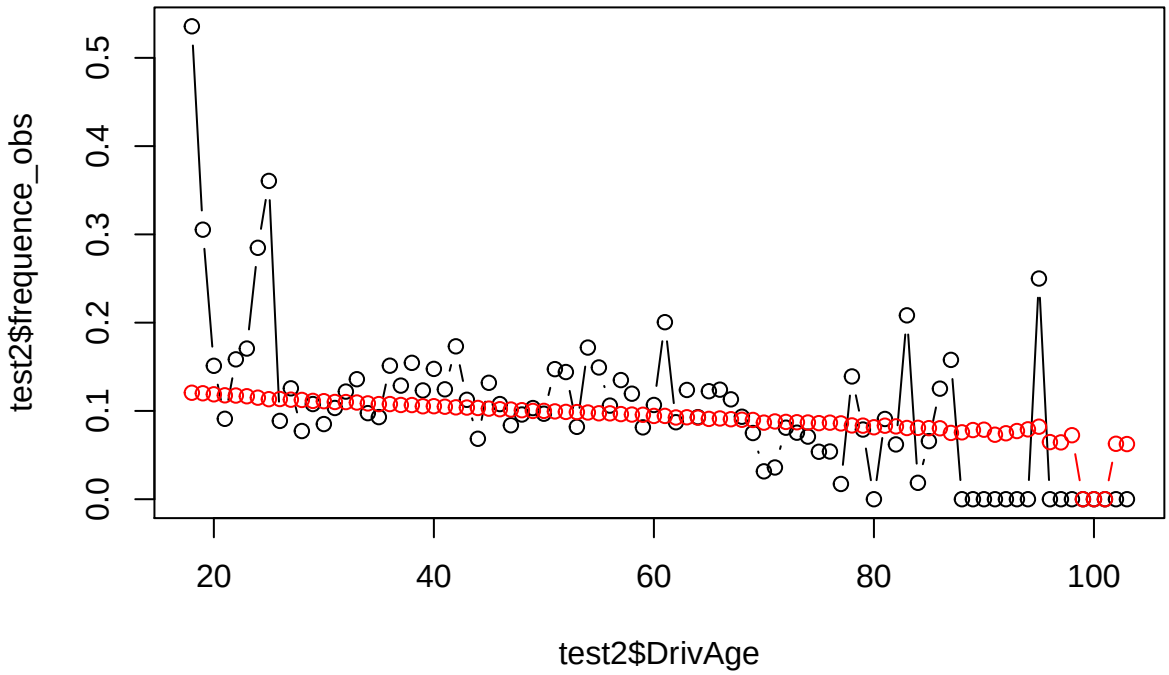
Quasipoisson:

Dispersion????

```
##
## Call:
## glm(formula = ClaimInd ~ DrivAge + HasKmLimit + offset(log(Exposure)),
##     family = quasipoisson("log"), data = freMPL2)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.4899  -0.3603  -0.2758  -0.1832   3.5914
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.025495   0.073661 -27.497  < 2e-16 ***
## DrivAge     -0.004983   0.001607  -3.101  0.00193 **
## HasKmLimit  -0.234207   0.074767  -3.132  0.00173 **
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for quasipoisson family taken to be 1.124963)
##
##     Null deviance: 12564  on 47496  degrees of freedom
## Residual deviance: 12538  on 47494  degrees of freedom
## AIC: NA
##
## Number of Fisher Scoring iterations: 6
```

```
##    ClaimInd Predict_ClaimInd  difference
## 1         0      0.050860088 0.050860088
## 2         0      0.041725357 0.041725357
## 3         0      0.058475681 0.058475681
## 4         0      0.022273341 0.022273341
## 5         0      0.009289614 0.009289614
## 6         0      0.041762515 0.041762515
## 7         1      0.050653001 0.949346999
## 8         0      0.050551695 0.050551695
## 9         0      0.024521120 0.024521120
## 10        0      0.082285626 0.082285626
```



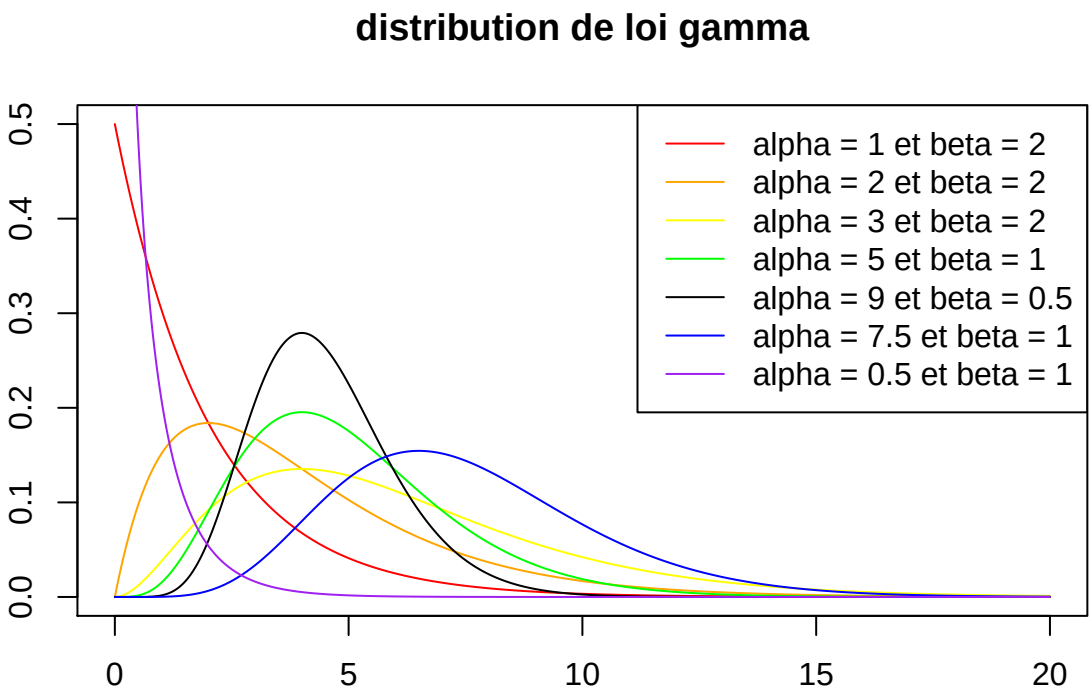
Le résumé de notre modèle révèle des informations intéressantes. La performance d'une régression logistique est évaluée avec des métriques clés spécifiques : - AIC (Critère d'information d'Akaike): Il mesure l'ajustement lorsqu'une pénalité est appliquée au nombre de paramètres. Des valeurs AIC plus petites indiquent que le modèle est plus proche de la vérité. Ici, il est de 213416.2 ce qui n'est pas bon du tout. - Null deviance : Il

s'agit de la déviance du modèle nul, c'est-à-dire qu'il 'est caractérisé par aucun facteur. Elle est de 297600 sur 25387 degrés de liberté. - Residual deviance : Il s'agit de la déviance du modèle avec toutes les variables. Elle est de 102549 sur 25304 degrés de liberté. - Number of Fisher Scoring iterations : Il s'agit du nombre d'itérations avant la convergence

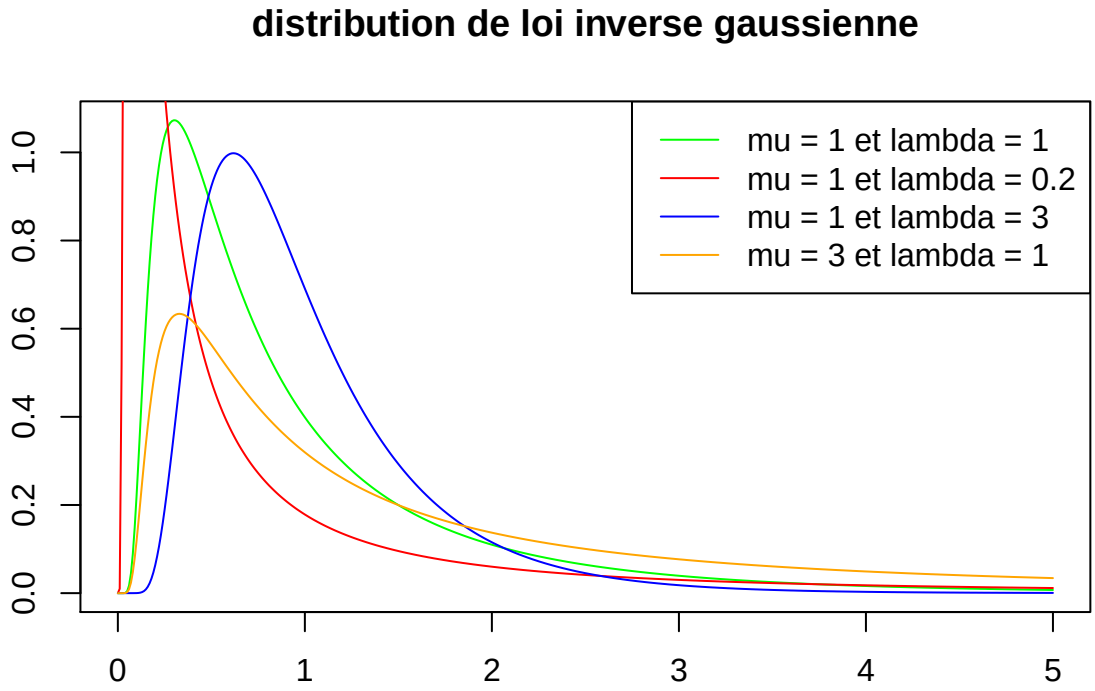
3.2 Sévérité des sinistres

3.2.1 Présentation des lois utilisables

3.2.1.1 loi de gamma (ou d'Euler)



3.2.2 Inverse gauss




```
##
## Call:
## glm(formula = ClaimAmount ~ DrivAge + VehAge + VehClass + VehBody +
##     VehEnergy + offset(log(Exposure)), family = poisson("log"),
##     data = freMPL2)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
##  -63.90   -14.66   -10.99    -7.14  1029.62
##
## Coefficients:
##                      Estimate Std. Error  z value Pr(>|z|)
## (Intercept)          6.198e+00  4.503e-03 1376.477   <2e-16 ***
## DrivAge             -3.924e-03  3.692e-05 -106.294   <2e-16 ***
## VehAge1              3.307e-01  2.300e-03  143.811   <2e-16 ***
## VehAge10+            3.885e-02  2.105e-03   18.458   <2e-16 ***
## VehAge2              4.465e-01  2.255e-03  197.977   <2e-16 ***
## VehAge3              3.281e-01  2.396e-03  136.955   <2e-16 ***
## VehAge4              2.530e-01  2.473e-03  102.292   <2e-16 ***
## VehAge5              9.985e-01  2.171e-03  459.957   <2e-16 ***
## VehAge6-7            1.381e-01  2.430e-03   56.837   <2e-16 ***
## VehAge8-9           -3.814e-02  2.547e-03  -14.973   <2e-16 ***
## VehClassA           -8.989e-01  3.485e-03 -257.949   <2e-16 ***
## VehClassB           -4.401e-01  2.767e-03 -159.044   <2e-16 ***
```

```
## VehClassH             -4.042e-01  2.910e-03 -138.895   <2e-16 ***
## VehClassM1            -7.188e-01  2.748e-03 -261.580   <2e-16 ***
## VehClassM2            -5.934e-01  2.953e-03 -200.943   <2e-16 ***
## VehBodymicrovan        9.728e-01  3.254e-03  298.930   <2e-16 ***
## VehBodyautobus        -9.103e-01  9.900e-03  -91.951   <2e-16 ***
## VehBodycoupé          -1.131e-01  3.910e-03  -28.931   <2e-16 ***
## VehBodyautre microvan  1.354e-01  3.481e-03   38.899   <2e-16 ***
## VehBodyberline        -2.833e-01  2.896e-03  -97.824   <2e-16 ***
## VehBodySUV            -1.479e-01  3.613e-03  -40.931   <2e-16 ***
## VehBodybreak          -4.943e-01  3.862e-03 -127.977   <2e-16 ***
## VehBodycamionnette    -2.289e-01  3.813e-03  -60.040   <2e-16 ***
## VehEnergyGPL          -1.644e+01  1.221e+02   -0.135    0.893
## VehEnergyélectrique    4.314e-01  2.886e-02   14.949   <2e-16 ***
## VehEnergyessence      -4.312e-01  1.160e-03 -371.797   <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: 32219893  on 47496  degrees of freedom
## Residual deviance: 30860875  on 47471  degrees of freedom
## AIC: Inf
##
## Number of Fisher Scoring iterations: 9

##    ClaimAmount Predict_ClaimAmount difference
## 1            0            70.33698   70.33698
## 2            0            72.35726   72.35726
## 3            0           113.85021  113.85021
## 4            0           111.00698  111.00698
## 5            0            27.86287   27.86287
## 6            0           125.39347  125.39347
## 7         1204           220.61219  983.38781
## 8            0           220.17097  220.17097
## 9            0            52.68245   52.68245
## 10           0           111.79994  111.79994
```

3.2.3 Calcul de la prime pure

Soit X le coût monétaire au risque

Selon le modèle général, X = SOMME de 1 à N des Bk

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

Autrement dit, N représente la fréquence (variable discrète) et Bk 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)?

4 Bibliographie

5 Annexes

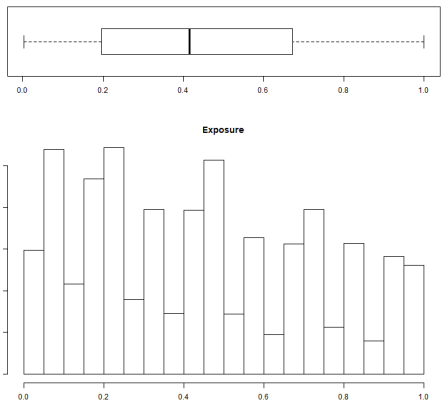
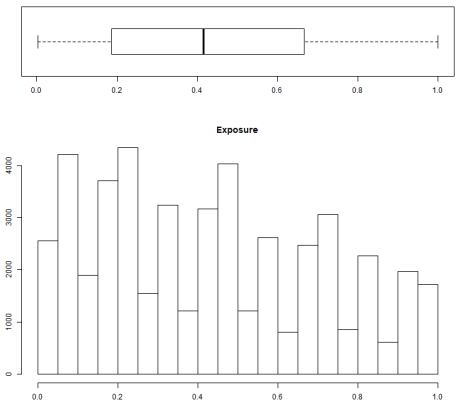
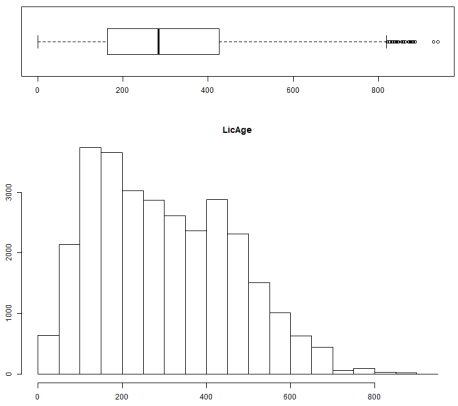
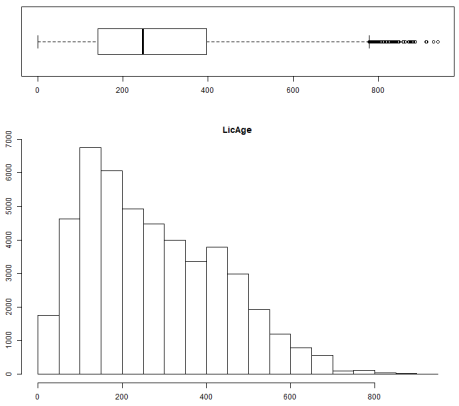
5.1 Affichage de l'implementation de la fonction nettoyage_dataframe :

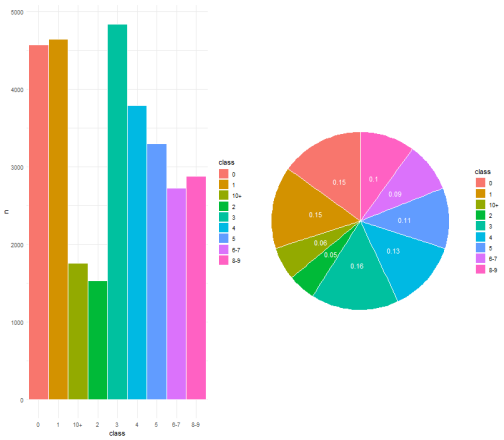
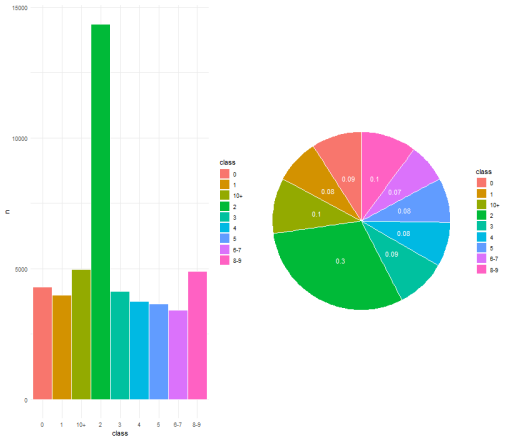
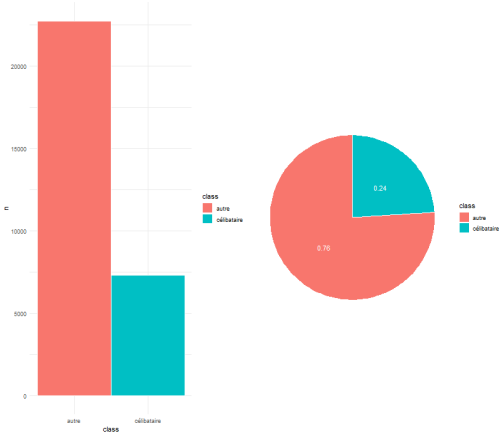
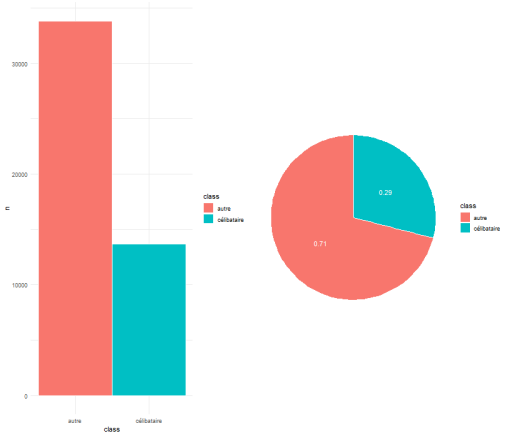
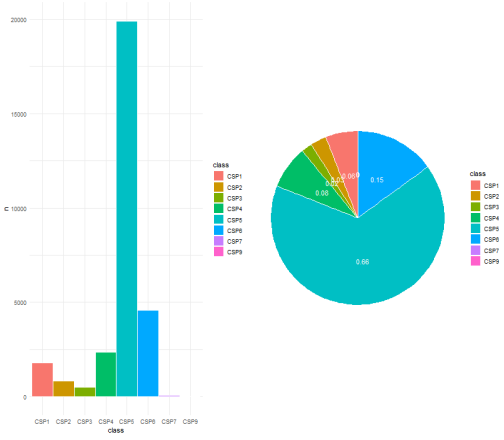
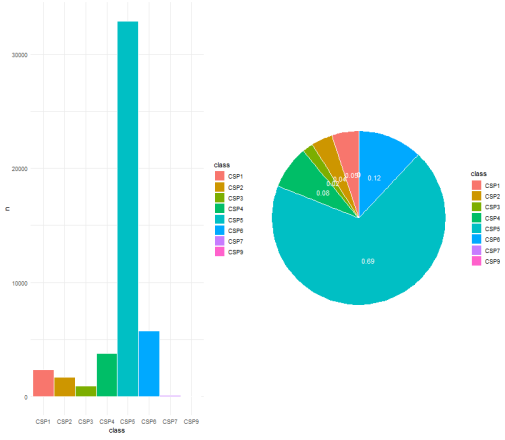
```
nettoyage_dataframe <- function(dt){

  # 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)
```

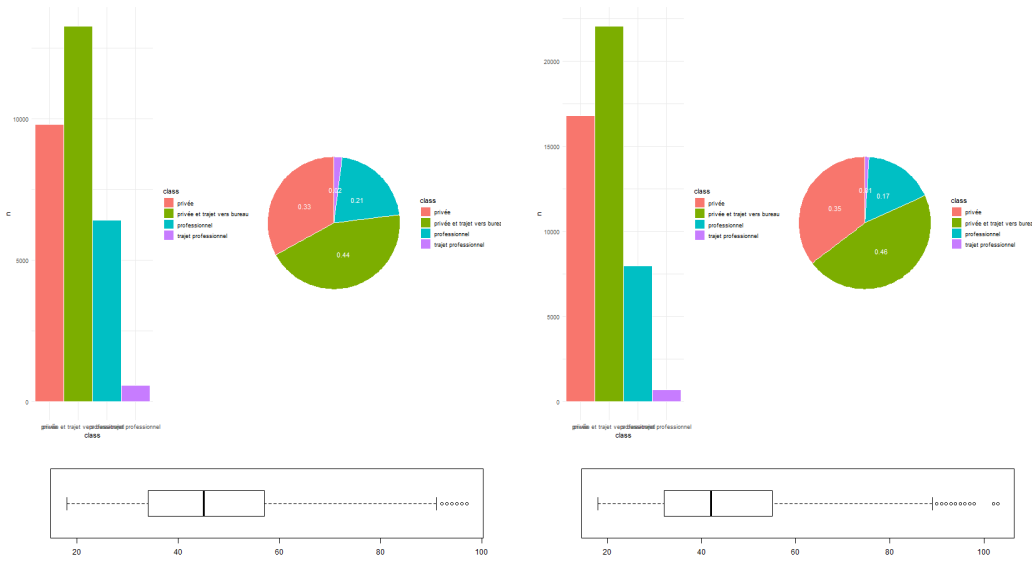
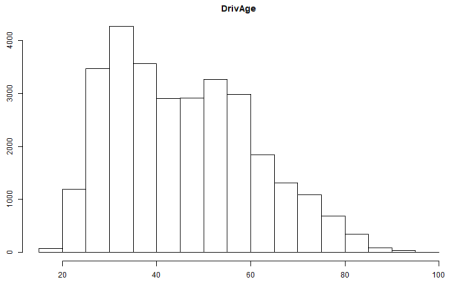
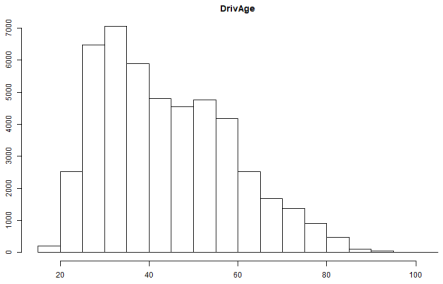
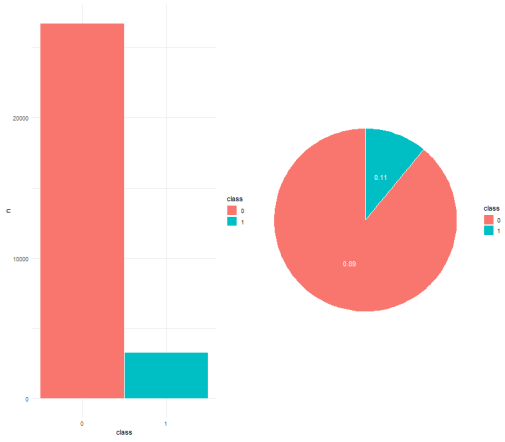
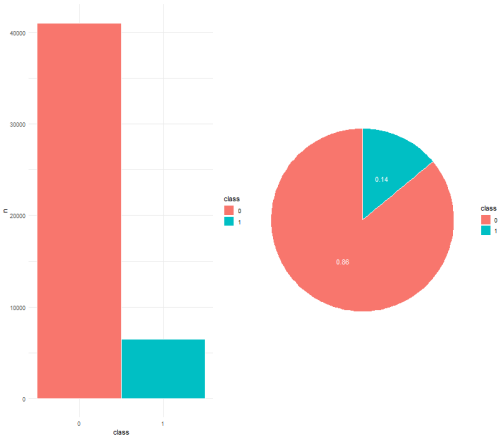
```
# Traduction des données (VehBody, MariStat, VehUsage, VehEngine, VehEnergy, Garage)
for (i in 1:dim(dt)[2]){
  # Type de véhicules
  if (colnames(dt)[i]=="VehBody"){
    levels(dt$VehBody) <- c(levels(dt$VehBody), "autobus", "coupé",
                                  "autre microvan", "berline","SUV",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

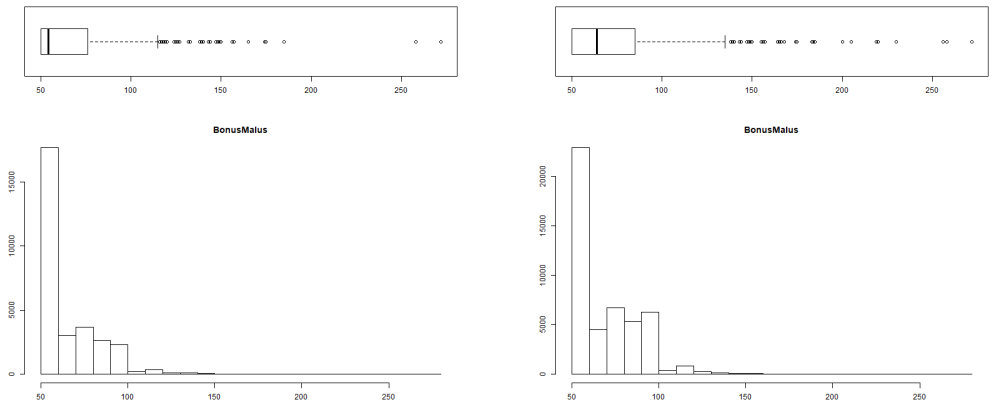
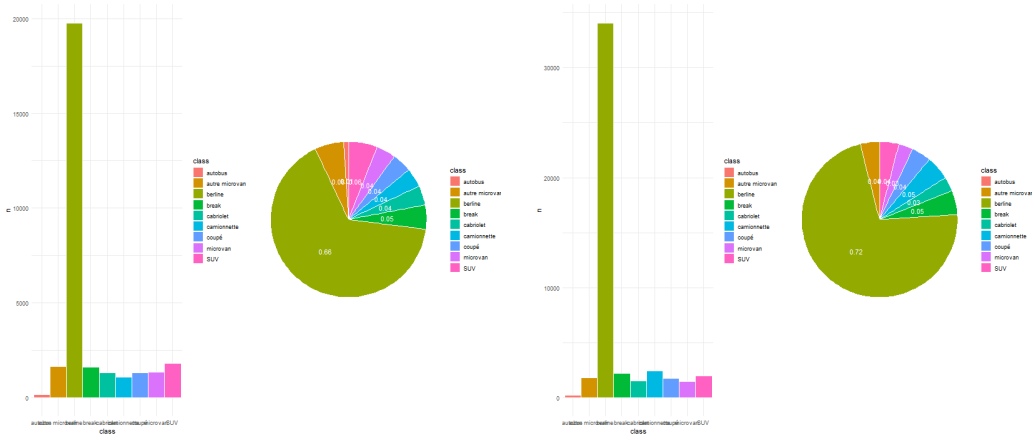
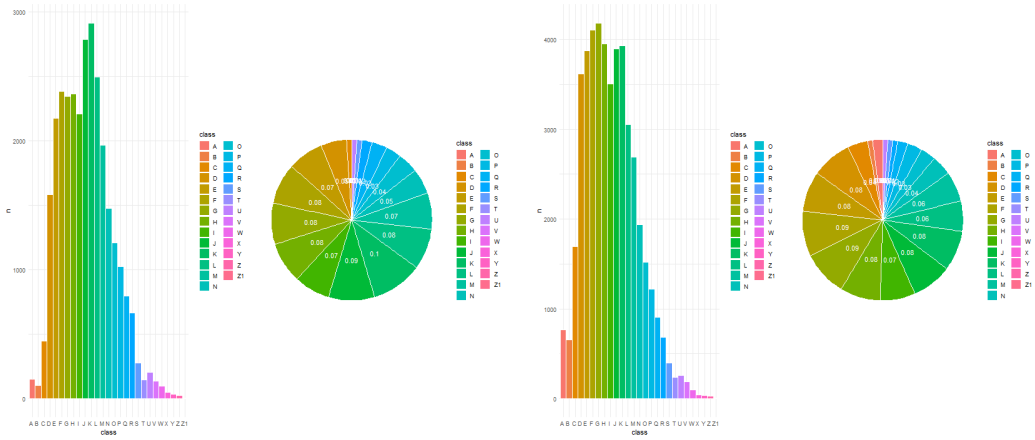
```
      dt$VehEnergy <- droplevels(dt$VehEnergy)
      }
    # Garage
    if (colnames(dt)[i]=="Garage"){
      levels(dt$Garage) <- c(levels(dt$Garage), "aucun", "garage indépendant",
                                  "concessionnaire")
      dt$Garage[dt$Garage == "None"]<-"aucun"
      dt$Garage[dt$Garage == "Private garage"]<-"garage indépendant"
      dt$Garage[dt$Garage == "Collective garage"]<-"concessionnaire"
      dt$Garage <- droplevels(dt$Garage)
    }
  }
  return (dt)
}
```

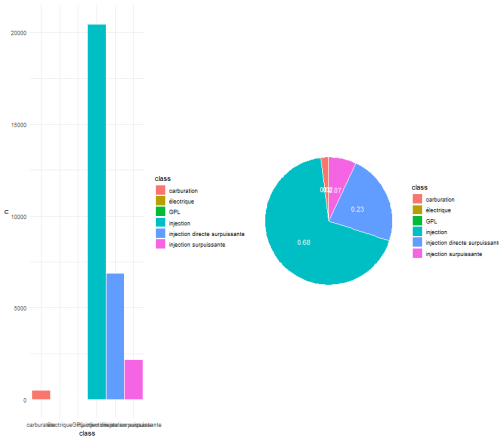
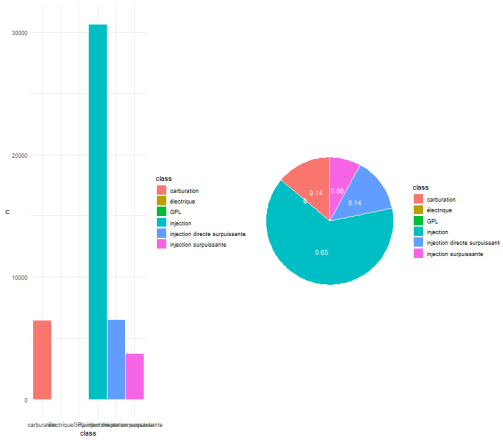
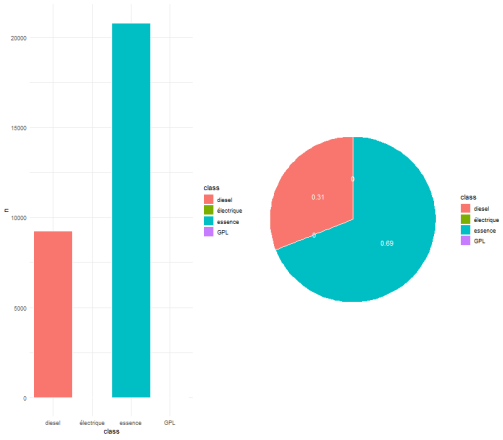
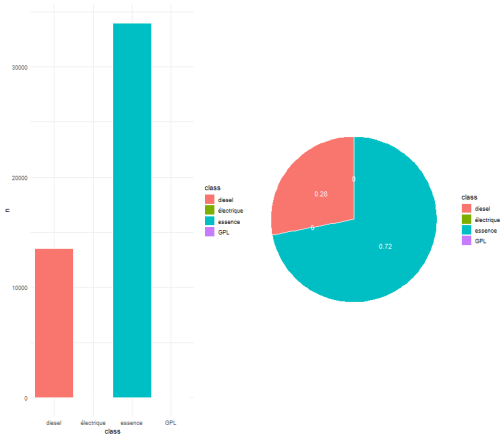
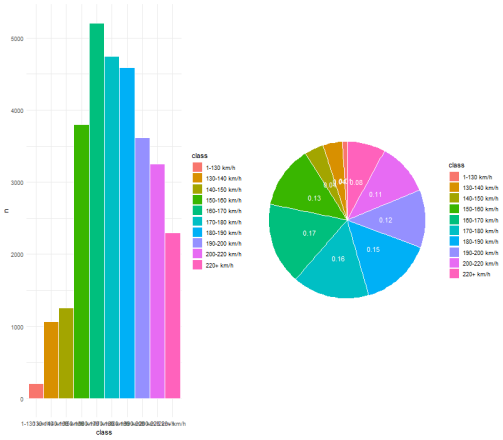
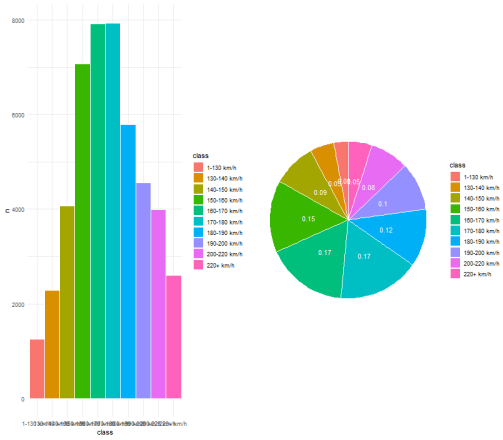
5.2 Affichage de l'ensemble des représentations graphiques

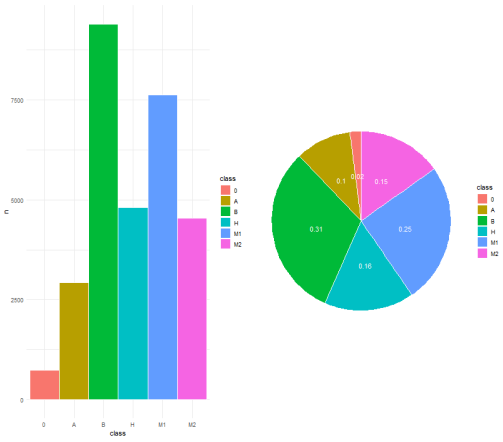
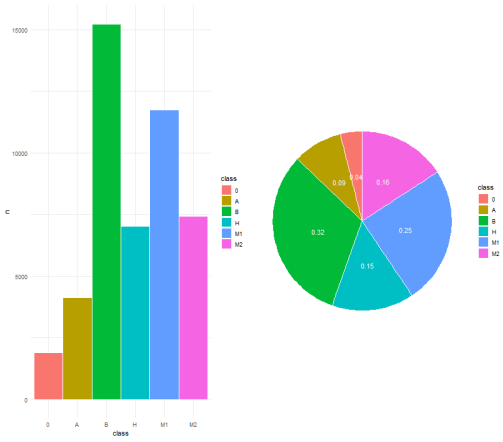
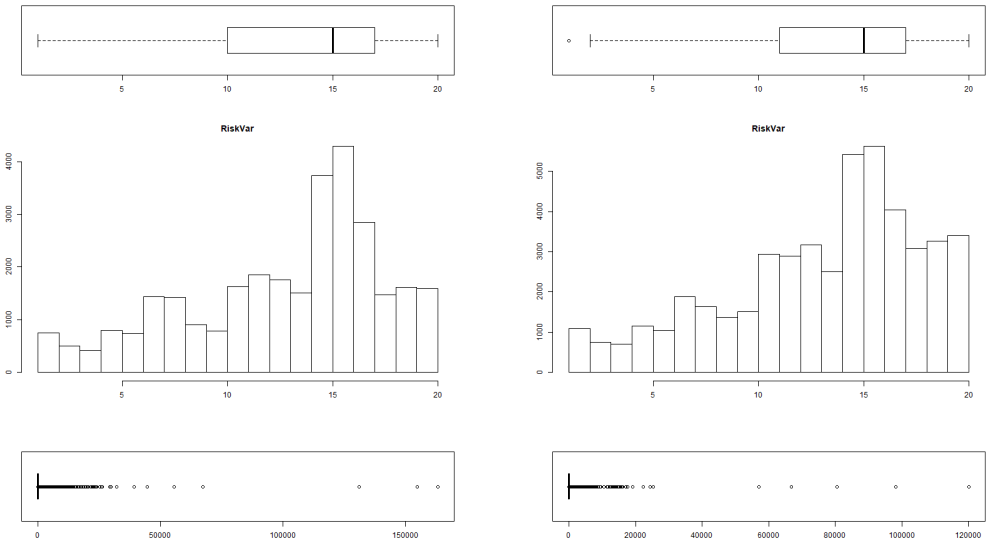
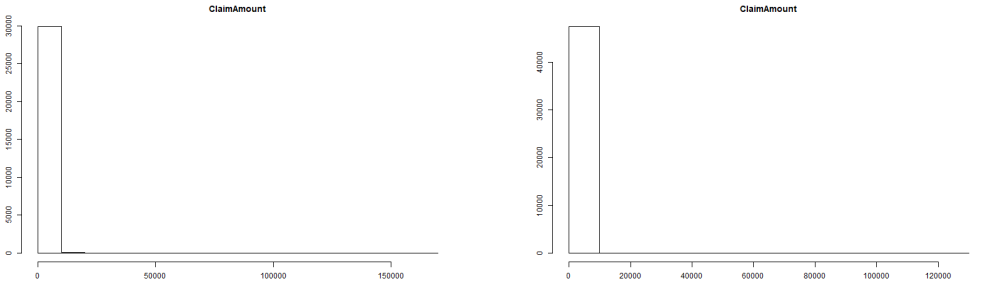
	freMPL1	freMPL2
Exposure		
LicAge		

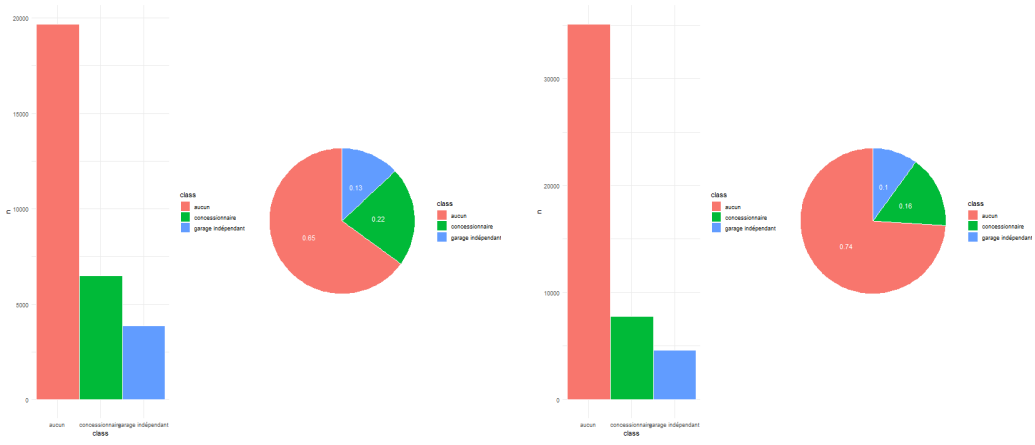
	freMPL1	freMPL2
VehAge		
MariStat		
SocioCateg		

	freMPL1	freMPL2
VehUsage		
DrivAge		
HasKmLimit		

	freMPL1	freMPL2
BonusMalus		
VehBody		
VehPrice		

	freMPL1	freMPL2
VehEngine		
VehEnergy		
VehMaxSpeed		

	freMPL1	freMPL2
VehClass		
RiskVar		
ClaimAmount		

	freMPL1	freMPL2
Garage		
ClaimInd