EMINES
School of Industrial Management

UNIVERSITÉ MOHAMMED VI
POLYTECHNIQUE

Rapport statistique : Modélisation des fréquences des sinistres

Groupe n°7

Réalisé par :
OUSSAMA ABBOUDI
HAMZA LBOUALAOUI
WALID NADIR
ALI YOULYOUZ

Table des matières

Introduction :

Etant élèves en deuxième année du cycle ingénieur de l'EMINES de l'université Mohammed VI Polytechnique , nous sommes amenés à suivre un cours en statistique, un cours théorique, qui porte sur les notions de bases des statistiques, nous sommes aussi amenés à appliquer ces notions acquises, en cours théorique dans des projets qui se relient au monde professionnel, afin de pouvoir faire la liaison entre tous ce qui est théorique et pratique. Notre projet s'articule sur la modélisation des fréquences des sinistres, un exercice typique et crucial pour toutes assurances, un exercice permettant le scoring de chaque contrat client de l'assurance, pour se faire, nous proposons le traitement suivant :

Problématique :

L'analyse prédictive n'est pas une idée neuve dans le monde de l'entreprise. Dans le secteur de l'Assurance, des actuaires utilisent quotidiennement des données du passé et construisent des modèles analytiques pour prédire la probabilité d'un événement futur. Les résultats obtenus via ces modèles statistiques servent directement à la prise de décision. L'Assurance a ainsi été un des premiers secteurs à utiliser l'analyse prédictive dans la gestion du risque client via des méthodes de scoring.

Pour autant, les avancées technologiques dans la capacité à traiter les données, accompagnées d'une baisse des coûts des solutions d'analyse décisionnelle, amènent les Assureurs à élargir leurs usages de l'analyse prédictive et à se doter de nouveaux outils. Ce changement technologique permettrait de décliner l'analyse prédictive sur l'ensemble de la chaîne de valeur et contribuerait même à réinventer le métier de l'assurance

Historiquement, l'Assurance est pionnière dans l'utilisation de l'analytique à travers la technique du scoring permettant d'associer un risque à un client. Les avancées dans les technologies et capacités de traitement de la donnée et l'accès à de nouvelles sources de données publiques ou privées ont permis d'élargir le champ d'utilisation de l'analyse prédictive dans l'Assurance. En 2013, Gartner affirmait que le « seuil de productivité » de l'analyse prédictive était atteint. Ce passage d'un gain potentiel à un gain estimé a convaincu les Assureurs de se lancer.

L'analyse prédictive approfondie permettrait ainsi de jouer sur l'ensemble des leviers de la chaîne de valeur de l'Assurance via des processus métiers plus performants définis à partir de modèles statistiques enrichis (incluant les réseaux neuronaux, les règles d'expert, les arbres de décisions,). Cette technique serait même au cœur de la création ou du maintien d'avantages concurrentiels dans un marché mature et fortement concurrentiel.

L'utilisation de modèles prédictifs contribue à :

- L'amélioration des prises de décisions dans la souscription et la lutte contre la fraude grâce une évaluation plus rapide et plus fiable du risque client et du risque de fraude ;
- Une meilleure identification de la « valeur client » et la communication ciblée vers les clients à fort potentiel
- L'optimisation des processus métier à travers leur automatisation via des règles métiers ;
- La réduction des coûts associée à la diminution des traitements manuels et l'identification de plus de cas de fraude ;
- La définition de produits personnalisés attractifs pour les clients (nouveaux produits, meilleur pricing model, usage-based insurance).

1-nettoyage-fusion des bases de données :

Nous réalisons un nettoyage primaire puis un autre avancé

1.1- nettoyage primaire :

➔ Nettoyage et fusion des bases de données sous R :

D'abord, Nous importons les deux bases de données à nettoyer et à fusionner :

➔ Les étapes pour nettoyer les bases de données sont les suivantes :

- Enlèvement des données manquantes :
- Suppression des observations répétées :
- Création des variables des dates (age, age_permis, age_vehicule…) :
- Détermination du nombre d'accidents par personne chaque année :
- Fusion des deux bases de données :
- Affectation de 0 pour les données perdues (charge et number)
- Suppression des variables suivantes après les avoir utiliser pour determiner les différents âges :

1.2. Nettoyage avancé de la base de données et discrétisation de la variable des fréquences :

Après avoir réalisé un premier filtrage de notre base de données, nous avons réalisé une étude préliminaire basée sur les arbres de décision afin d'éliminer les valeurs aberrantes qui y restent et qui peuvent influencer négativement notre model. Puis on a procédé par une discrétisation de la variable «freq » qui désigne la fréquence des sinistres pour qu'elle devienne significative (pas de raison de dire 2,3342432 accidents /an).

Le code suivant nous a servi pour la suppression des valeurs aberrantes et la discrétisation de la variable « freq » :

```
data=subset(a,SEXE!="" & d$exposition<=1 & d$charge>=0 &d$age_permis>=0 & d$age_vehicule>=0)

data<-data[!(data$freq>30 & data$X1==1),]
#or X1 est le vecteur associant à chaque observation le cluster correspondant
data<-data[!(data$freq>80 & data$X1==2),]
data<-data[!(data$freq>50 & data$X1==6),]
data<-data[!(data$freq>50 & data$X1==7),]
data<-data[!(data$freq>35 & data$X1==8),]
#discrétisation des frequences
data$freq=round(data$freq)
```

2- analyse de la base de données :

2.1. Statistique descriptive :

➔ Les deux graphiques camemberts ci-dessous représentent respectivement la proportion des deux sexes des conducteurs et la proportion des deux types de Combustible « Essence et Gasoil » dans la base de données.

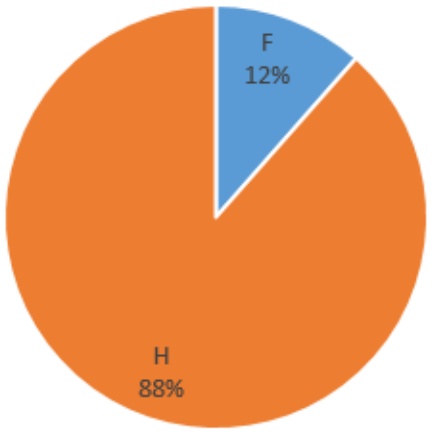


Figure 2 : sexe des conducteurs

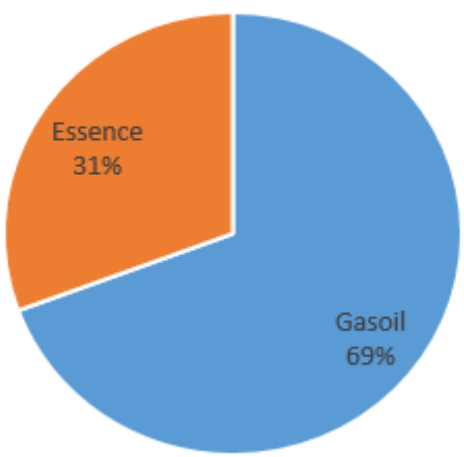
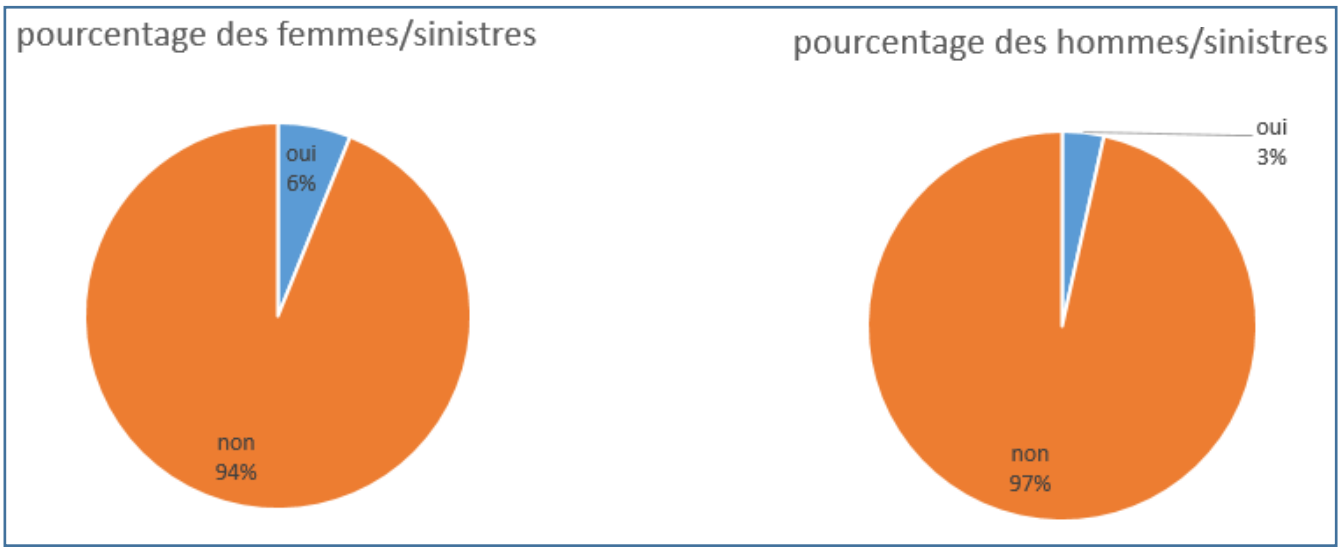


Figure 2 : Combustion

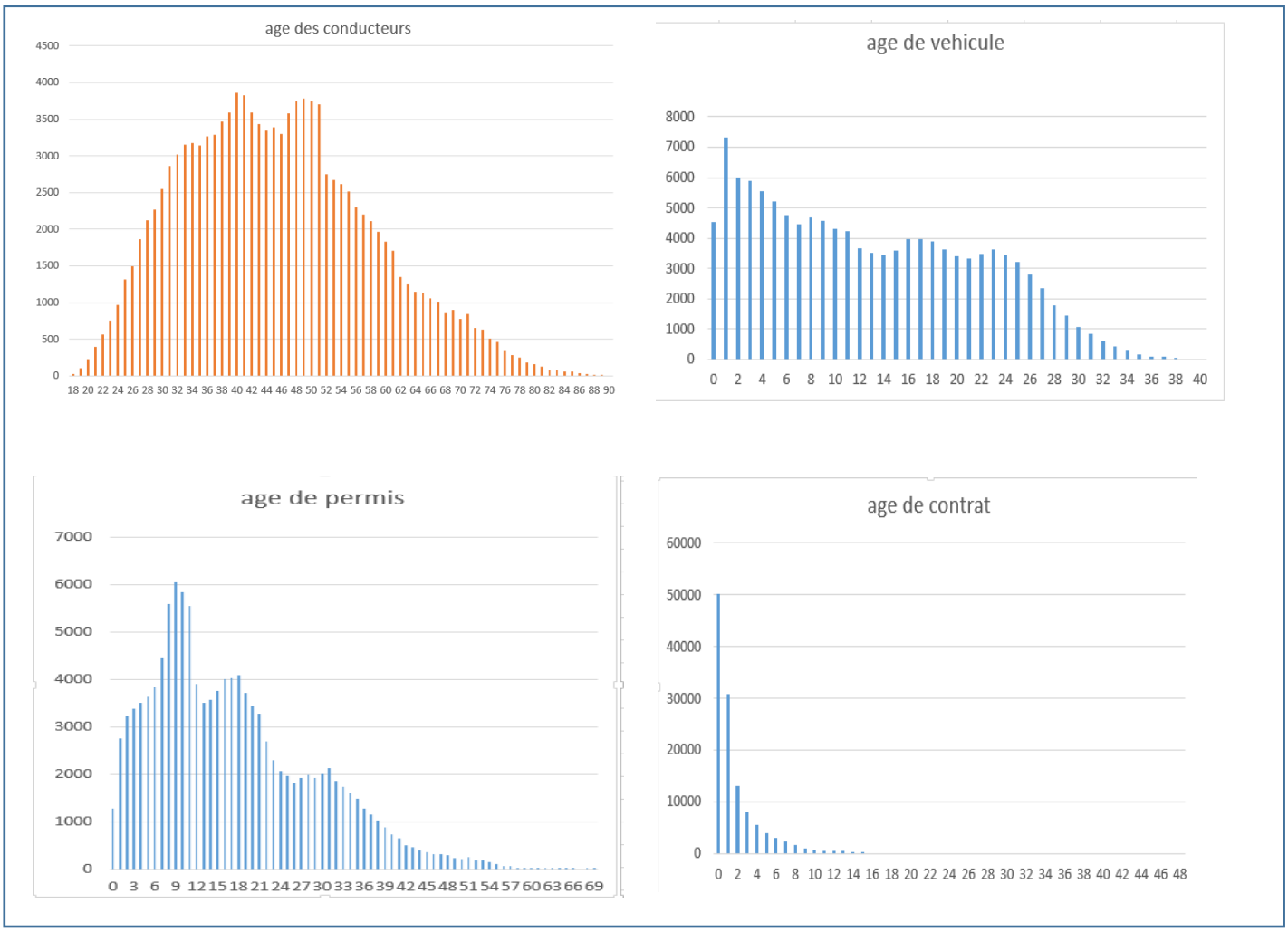
→ Il apparait donc que la plupart des conducteurs sont des hommes (88%) et que la majorité d'entre eux conduisent des véhicules à combustion Gasoil (69%).

➔ Les deux graphiques camemberts suivants donnent une idée sur la proportion des deux sexes qui ont eu au moins un accident et ceux dont on n'a jamais enregistré un sinistre.



➔ On observe donc que les femmes sont plus exposées au risque d'avoir un accident que les hommes.

➔ Les graphiques ci-dessous représentent respectivement les âges des conducteurs, les âges des véhicules ; les âges de permis et les âges des contrats des clients dans notre base de données.



➔ On observe donc que les âges des conducteurs varient de 18 à 90 ans et que la plupart d'entre eux sont jeunes (<=60 ans).

➔ On observe aussi que l'âge de véhicule est presque équiréparti sur le domaine [0,24 ans] et que le mode de cette série numérique est 1 an.

➔ Nous remarquons que de nombreux clients ont un permis d'ancienneté entre 5 et 10 ans et que la plupart des conducteurs sont des nouveaux clients (Age contrat =0).

2.1.2 Corrélations entre les variables de la base de données :

➔ Voici le tableau résumant les corrélations entre les différentes variables de notre base de données :

	puissance_fiscale	age	age_vehicule	age_permis	age_contrat	freq
puissance_fiscale	1.000000000	0.06385102	-0.03688162	0.0878889345	0.014684451	0.0021544235
age	0.063851022	1.00000000	-0.05685614	0.7159577361	0.253811069	-0.0129917341
age_vehicule	-0.036881617	-0.05685614	1.00000000	-0.1432124864	-0.027599189	-0.0474389489
age_permis	0.087888934	0.71595774	-0.14321249	1.0000000000	0.290837669	-0.0002949138
age_contrat	0.014684451	0.25381107	-0.02759919	0.2908376690	1.000000000	-0.0022091821
freq	0.002154423	-0.01299173	-0.04743895	-0.0002949138	-0.002209182	1.0000000000

➔ Le tableau ci-dessus révèle donc que les corrélations entre les variables de notre base de données sont assez faibles sauf entre l'âge du conducteur et l'âge du permis.

➔ Et en ce qui concerne la variable objective « freq » en particulier , elle n'est pas corrélée avec aucune variable explicative numérique

2.2. Analyse exploratoire

➔ Cette partie arrive juste après le nettoyage final de notre base de données, cette partie est cruciale, du moment que durant cette phase, où nous explorons notre DATA et où nous faisons de la statistique descriptive afin de tirer des constatations , des relations entre la variable objective et les différentes variables explicatives de notre base de données, à noter que durant cette étape nous utiliserons des modèles «glm.nb » simples avec une distribution du type binomiale négative de la variable « freq » qu'on justifie en utilisant la fonction « goodfit » du package « vcd » :

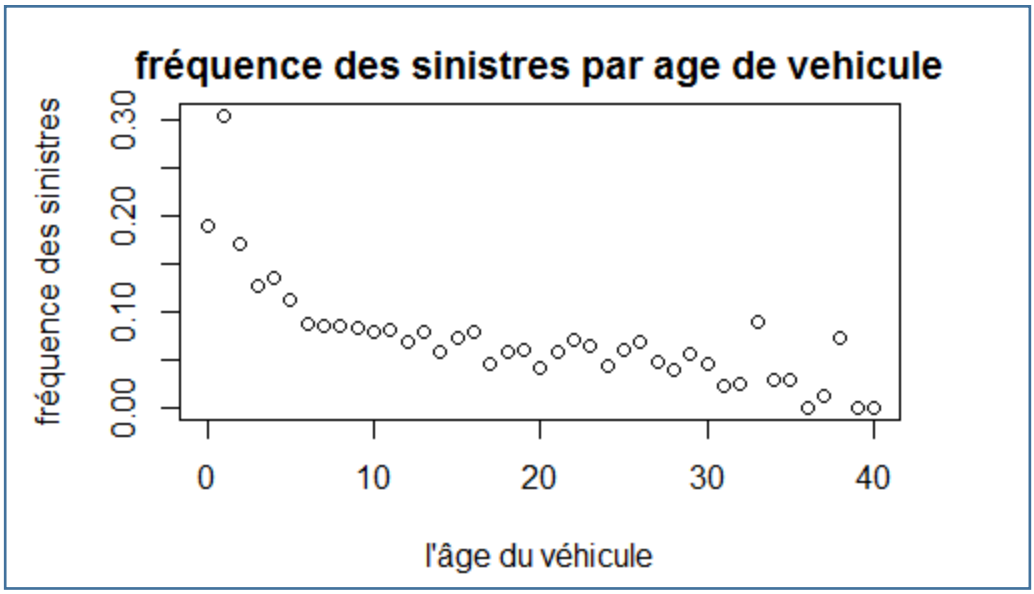
- D'après les résultats du goodfit en utilisant la méthode de minimisation du khi2 on trouve qu'une distribution binomiale négative des fréquences des sinistres dans la base de données est plus appropriée qu'une distribution de poisson vu que les valeurs de « fitted `pearson residual` » sur chaque valeur prise par la variable « freq » sont plus élevées pour une distribution de poisson que pour une distribution binomiale négative.

(voir annexes pour la visualisation du résultat obtenu)

2.2.1- fréquence des sinistres et âge de véhicule :

➔ Puisque la variable "age_vehicule" dans notre data est discrète, on peut procéder par ranger les valeurs que peut prendre notre variable et ensuite calculer la moyenne des sinistres pour chaque valeur.

```
age_veh=unique (data$age_vehicule)
age_veh=sort(age_veh)
plot(age_veh,tapply(data$freq,data$age_vehicule,mean),xlab="l'âge du véhicule",
ylab="fréquence des sinistres",main="fréquence des sinistres par age de vehicule")
```



➔ D'après le graphique il s'avère que généralement le nombre moyen des sinistres décroit avec l'âge du véhicule, sauf pour des valeurs non significatives de l'age_vehicule (age_vehicule > 85 ans)

➔ en utilisant un modèle "glm" pour une distribution binomiale négative du nombre de sinistres par rapport à l'âge du véhicule, on se retrouve avec le résultat suivant :

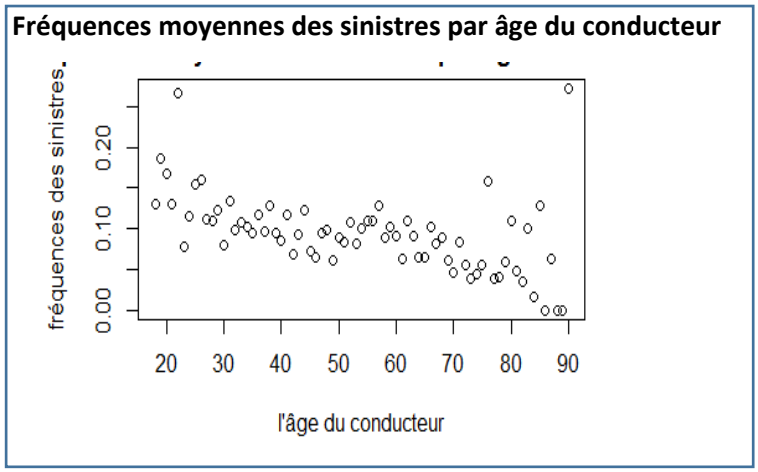
```
library(MASS)
model=glm.nb(data$freq~age_vehicule,data=data)
summary(model)

Call:
glm.nb(formula = data$freq ~ age_vehicule, data = data, init.theta = 0.02546993007,
    link = log)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-0.3224  -0.3040  -0.2739  -0.2434   5.0716

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)  -1.76897    0.03448  -51.30   <2e-16 ***
age_vehicule -0.05303    0.00236  -22.47   <2e-16 ***
---
Signif. codes:
0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for Negative Binomial(0.0255) family taken to be 1)

    Null deviance: 14302  on 123899  degrees of freedom
Residual deviance: 13732  on 123898  degrees of freedom
AIC: 53866

Number of Fisher Scoring iterations: 1

              Theta:  0.025470
          Std. Err.:  0.000534

 2 x log-likelihood:  -53859.916000
```

- Le coefficient "-0.05303" de Age_véhicule dans le modèle étant négatif et significatif, cela donc renforce l'idée que la fréquence moyenne des sinistres décroît avec l'âge du véhicule.

2.2.2- fréquence des sinistres et âge du conducteur :

- La variable “age” dans notre data qui réfère à l’âge du conducteur principal étant discrète, on peut donc procéder par ranger les valeurs que peut prendre la variable “age” et ensuite calculer la moyenne des sinistres pour chaque valeur afin de visualiser la relation entre l’âge du conducteur et la fréquence moyenne des sinistres par âge.

```
age=unique (data$age)
age=sort(age)
plot(age,tapply(data$freq,data$age,mean),xlab="l'âge du conducteur",ylab="fréquences des
sinistres",main="fréquences moyennes des sinistres par âge du conducteur")
```



- D’après le graphique obtenu, nous observons que la fréquence des sinistres diminue avec l’âge du conducteur.

En utilisant le modèle glm suivant :

```
library(MASS)
model=glm.nb(data$freq~age,data=data)
summary(model)

Call:
glm.nb(formula = data$freq ~ age, data = data, init.theta = 0.02328989089,
    link = log)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-0.2952  -0.2828  -0.2766  -0.2696   5.6520

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.870516   0.077068 -24.271  < 2e-16 ***
age         -0.010272   0.001643  -6.252 4.06e-10 ***
---
```

```
Signif. codes:
0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for Negative Binomial(0.0233) family taken to be 1)

    Null deviance: 13520  on 123899  degrees of freedom
Residual deviance: 13480  on 123898  degrees of freedom
AIC: 54375

Number of Fisher Scoring iterations: 1

              Theta:  0.023290
          Std. Err.:  0.000481

 2  x log-likelihood:  -54369.057000
```

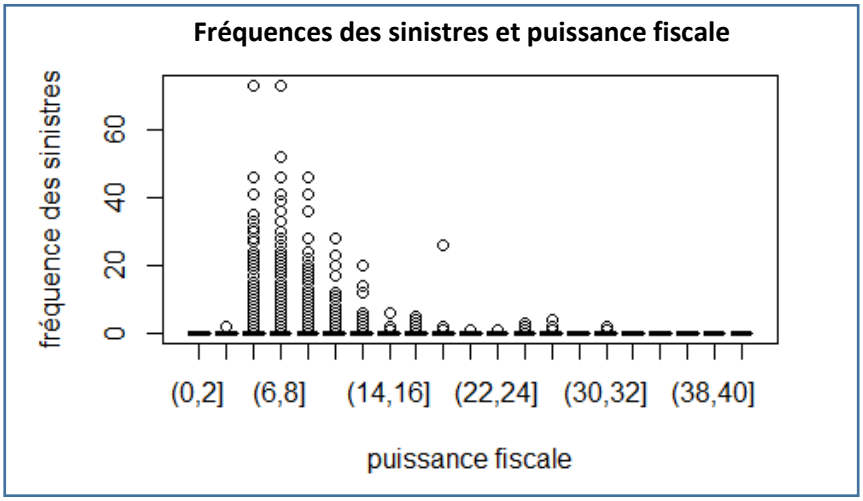
→Le coefficient “-0.010272” de l’âge du conducteur dans le modèle glm étant négatif et significatif on peut donc dire que les jeunes conducteurs sont les plus exposés à avoir des accidents.

→ Plus précisément si on a une première personne d’âge “x” et une deuxième personne d’âge “x+1” , alors le risque chez la deuxième personne d’avoir plus d’accidents diminue de “1-exp(-0.010272) ~ 1%” par rapport à la première personne.

2.2.3- fréquence des sinistres et puissance fiscale :

➔ Pour représenter l’effet de la puissance fiscale sur la fréquence des sinistres, on regroupe la variable “puissance_fiscale” en 21 groupes. la valeur maximale de la puissance fiscale dans notre base de donnée après cleaning étant 42.

```
seuils=seq(0,42,2)
data$cut=cut(data$puissance_fiscale,breaks=seuils)
boxplot(data$freq~data$cut,xlab="puissance fiscale",ylab="fréquence des
sinistres",main="fréquences moyennes des sinistres par puissance fiscale")
```



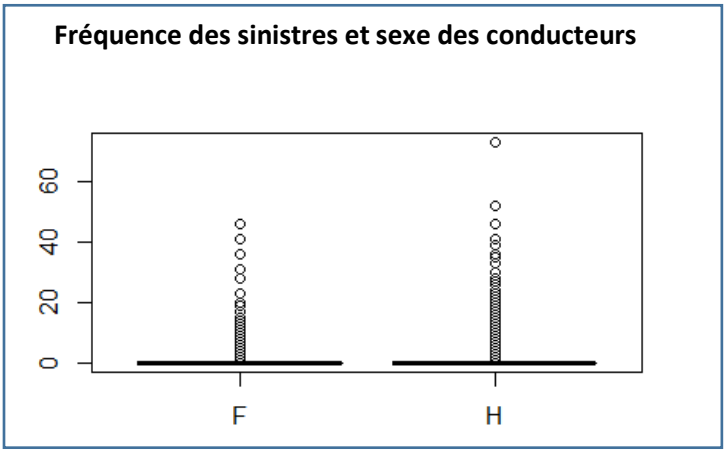
→ D’après la boîte à moustache obtenue, qui représente la fréquence moyenne des sinistres par intervalle de puissance fiscale, on peut observer que la variable puissance_fiscale a un effet explicatif sur les fréquences des sinistres puisqu’on observe que pour certaines valeurs de puissance fiscales on a beaucoup de valeurs aberrantes, portant pour

certaines d'autres on observe moins de valeurs aberrantes ou parfois on n'enregistre aucun sinistre (puissance fiscale>=33)

2.2.4- fréquence des sinistres et sexe des conducteurs :

➔ La variable "SEXE" étant qualitative et comprenant deux facteurs "M" et "F", et donc pour visualiser l'effet du SEXE sur la fréquence des sinistres, on envisage de faire un boxplot de la variable "freq" en fonction de "SEXE".

```
boxplot(data$freq~data$SEXE)
```



→ Le boite à moustache montre donc que les hommes et les femmes ont presque la même moyenne des fréquences des sinistres et donc il n'y a pas de différence significative entre les deux moyennes. Pour accepter cette hypothèse ou la rejeter, on réalise un test bilatéral de Student pour la comparaison des deux moyennes.

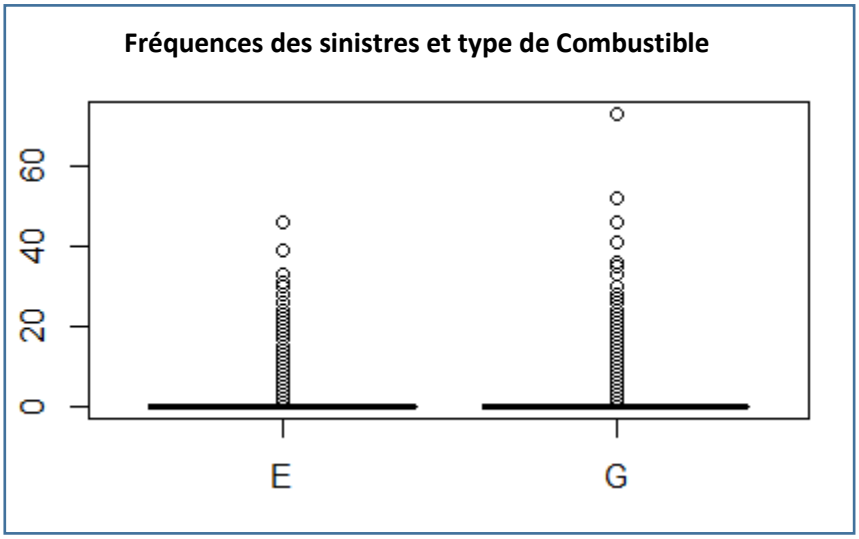
```
t.test(data$freq[data$SEXE== F], data$freq[data$SEXE== H alternative="two.sided")

Welch Two Sample t-test

data:  data$freq[data$SEXE == F and data$freq[data$SEXE == H]
t = 6.2762, df = 16475, p-value = 3.556e-10
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 0.04622172 0.08820405
sample estimates:
 mean of x  mean of y
0.15683252 0.08961963
```

→ La p-value= 9.537e-15 de test de Student étant très inférieure à 5%, on rejette donc l'hypothèse d'égalité des moyennes et on peut conclure que les hommes sont plus exposés à avoir des accidents que les femmes.

2.2.5- fréquence des sinistres et Combustion :

➔ De même pour la variable “Combustion” qui est aussi qualitative et comprenant deux facteurs “E” et “G”, on peut visualiser son effet sur la fréquence des sinistres à l’aide d’une boite à moustache reliant les deux variables.

```
boxplot(data$freq~data$Combustion)
```



➔ D’après la boîte à moustache, on peut assumer que la nature du carburant essence ou gasoil n’a pas un effet significatif sur les fréquences puisque la moyenne des fréquences des sinistres pour les deux facteurs “E” et “G” sont presque égales. Pour accepter cette hypothèse ou la rejeter on réalise le test de student bilatéral suivant :

```
t.test(data$freq[data$Combustion== E], data$freq[data$Combustion== G], alternative="two.sided")

    Welch Two Sample t-test

data:  data$freq[data$Combustion == E] and data$freq[data$Combustion == G]
t = -1.8689, df = 82973, p-value = 0.06164
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.0217889360  0.0005183267
sample estimates:
 mean of x  mean of y
0.08998499 0.10062029
```

→La P value étant supérieure à 5% on peut donc accepter l’hypothèse que la variable “Combustion” n’a pas d’influence significative sur les fréquences des sinistres.

2.2.6- interaction entre Sexe des conducteurs et combustion :

➔ Pour voir si on a une interaction entre les deux variables “SEXE” et “Combustion” qui pourrait influencer la fréquence des sinistres on réalise un test de khi^2 qui demande en premier de dresser un tableau de contingence dont les valeurs sont la moyenne des fréquences des sinistres pour chaque catégorie de la variable “SEXE”, intersection de chaque catégorie de la variable “Combustion”.

➔ Le tableau de contingence est comme suite :

```
a1=mean(subset(data,data$SEXE==F)$freq[subset(data,data$SEXE==F)$Combustion==E])
a2=mean(subset(data,data$SEXE==F)$freq[subset(data,data$SEXE==F)$Combustion==G])
b1=mean(subset(data,data$SEXE==H)$freq[subset(data,data$SEXE==H)$Combustion==E])
b2=mean(subset(data,data$SEXE==H)$freq[subset(data,data$SEXE==H)$Combustion==G])
M <- as.table(rbind(c(a1, a2), c(b1,b2)))
dimnames(M) <- list(gender=c("F","H"),Combustion=c("Essence", "Gasoil"))
test$observed
      Combustion
gender    Essence     Gasoil
     F 0.15728832 0.15633265
     H 0.07351929 0.09582791
```

→En acceptant l'hypothèse (H0: il n'y a pas d'interaction entre SEXE et Combustion) le tableau de contingence théorique est comme suite:

```
test$expected
      Combustion
gender    Essence     Gasoil
     F 0.14987759 0.16374338
     H 0.08093002 0.08841718
le resultat du test est :
(test <- chisq.test(M))
Chi-squared approximation may be incorrect

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

data:  M
X-squared = 9.6434e-32, df = 1, p-value = 1
```

→ Puisque la p-value du test est égale à 1 on est donc presque sûr qu'il n'y a pas d'interaction entre les variables "SEXE" et "combustion".

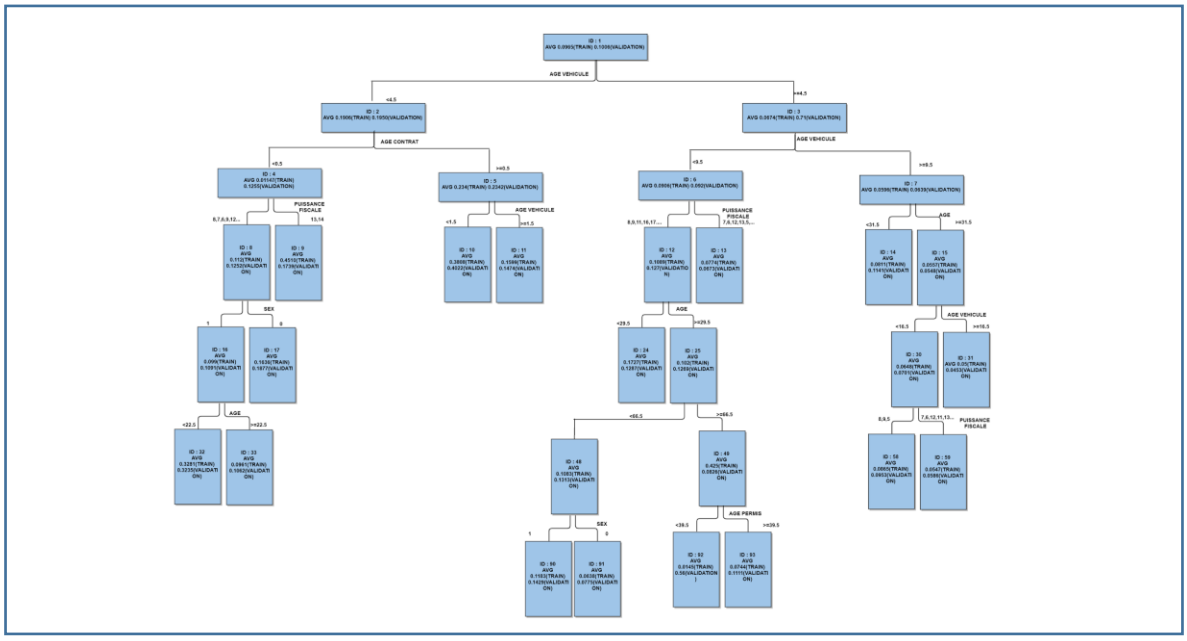
3-LES ARBRE DE DECISION :

➔ Nous avons réalisé deux études, la première porte sur toutes les observations de la base de données et la deuxième juste sur les personnes qui ont enregistré des sinistres, l'objectif de la deuxième étude sera de prévoir le nombre de sinistres que peut avoir un client durant l'exercice sachant qu'il aura au moins un sinistre.

3.1. Etude n°1 :

3.1.1. Générer l'arbre De décision :

➔ Notre arbre de décision est généré en se basant sur le principe de minimisation de la variance de la variable objective « freq » dans les différentes feuilles de l'arbre. L'arbre généré est le suivant :



➔ Le code utilisé pour générer le vecteur qui associe chaque observation au numéro du cluster qui le correspond est le suivant :

Voir annexes

3.1.2. Étude de la moyenne de la fréquence des sinistres par cluster:

➔ On génère les moyennes des fréquences des sinistres dans chaque cluster, l'objectif est d'avoir une idée sur le comportement des conducteurs dans chaque cluster.

```
tapply(data$freq,data$clust,mean)
         1          2          3          4          5          6          7          8          9
0.34883721 0.16963449 0.30188679 0.09774617 0.38053935 0.15760123 0.07519560 0.16666667 0.08148148
        10         11         12         13         14         15         16
0.02522936 0.06449376 0.12153454 0.09130873 0.08558868 0.05695838 0.04895766
```

→ Le cluster dont la moyenne de fréquence de sinistres est la plus élevée est le premier (clust==1) :

➔ Les caractéristiques de ce cluster :

➔ age de vehicule<4.5 age contrat=0

➔ Puissance fiscale =13,14

→Le cluster dont la moyenne de fréquence de sinistres est la plus basse est le 10eme (clust==10) :

➔ Les caractéristiques de ce cluster :

➔ puissance_fiscale IS ONE OF: 8, 9, 11, 16, 17, 10, 18, 4, 19, 21

➔ AND age_vehicule < 9.5 AND age_vehicule >= 4.5 AND

➔ age_permis < 39.5 AND age >= 66.5

3.1.3 – performance de l'arbre de décision :

➔ Afin d'avoir une idée sur la performance de notre arbre de décision nous calculons le SSE associé qui est la somme des carrés des erreurs et on le compare avec le SST qui est le total des carrés des erreurs.

Le code utilisé pour faire cette comparaison est comme suite :

```
predcart=rep(0,nrow(data))
tab=tapply(data$freq,data$clust,mean,data=data)
for (i in 1:nrow(data)){
  predcart[i]=tab[data$clust[i]]
}
SSE=sum((data$freq-predcart)^2) -> 117129.3
SST=sum((data$freq-mean(data$freq))^2) ->117836.5
```

➔ Nous observons donc que le SSE est très proche du SST, du coup l'arbre de décision ne servira pas d'une manière significative dans la prévision des fréquences des sinistres.

3.1.4 –modèle linéaire généralisé

a. good fit test :

➔ Le test de goodfit désigne le degré d'ajustement du modèle aux donnés observées.

➔ On importe la fonction goodfit qui existe dans la bibliothéque « vcd »

➔ D'après les résultats du goodfit en utilisant la méthode de minimisation du khi^2 on trouve qu'une distribution binomiale négative des fréquences des sinistres est plus significative qu'une distribution de poisson vu que la valeur de khi^2 sur chaque groupe sont plus élevées pour une distribution de poisson.

Voir annexes

b. GENMOD :

➔ Maintenant, en arrivant à l'étape de réalisation d'un modèle glm qui décrit notre variable discrétisé "freq" qui va nous donner une moyenne sur chaque cluster. Pour cela, nous allons comparer deux modèles de distribution de poisson et de binomiale négative.

1- MODELE GLM (BINOMIAL NEGATIVE)

➔ Pour une distribution de type binomiale négative le résultat de notre model GLM est comme suite :

```
glm.nb(formula = freq ~ as.factor(clust), data = data, init.theta = 0.02714444132,
    link = log)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-0.3835  -0.2879  -0.2684  -0.2366   5.3890

Coefficients:
                  Estimate Std. Error z value Pr(>|z|)
(Intercept)        -1.05315    0.67949  -1.550 0.121162
as.factor(clust)2  -0.72096    0.69407  -1.039 0.298924
as.factor(clust)3  -0.14455    0.91679  -0.158 0.874714
as.factor(clust)4  -1.27223    0.68365  -1.861 0.062753 .
as.factor(clust)5   0.08698    0.68405   0.127 0.898814
as.factor(clust)6  -0.79454    0.68205  -1.165 0.244051
as.factor(clust)7  -1.53451    0.68216  -2.250 0.024480 *
as.factor(clust)8  -0.73861    0.71258  -1.037 0.299955
as.factor(clust)9  -1.45423    0.76353  -1.905 0.056829 .
as.factor(clust)10 -2.62660    0.79819  -3.291 0.000999 ***
as.factor(clust)11 -1.68804    0.70569  -2.392 0.016756 *
as.factor(clust)12 -1.05441    0.68444  -1.541 0.123429
as.factor(clust)13 -1.34036    0.68270  -1.963 0.049610 *
as.factor(clust)14 -1.40505    0.68438  -2.053 0.040069 *
as.factor(clust)15 -1.81228    0.68204  -2.657 0.007881 **
as.factor(clust)16 -1.96365    0.68063  -2.885 0.003913 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for Negative Binomial(0.0271) family taken to be 1)

    Null deviance: 14879  on 123899  degrees of freedom
Residual deviance: 13923  on 123884  degrees of freedom
AIC: 53547
```

➔ Le résultat montre donc que les coefficients de notre modèle sont très significatifs puisque la plupart des p-values issues des tests sur les coefficients du modèle sont inférieure à 5% et donc on est sûr à 95% qu’ils ne sont pas nuls.

➔ Le cluster de référence étant le cluster n°1, on prévoit pour tous les autres clusters qui restent à part le cluster n°5, une moyenne de fréquence de sinistres moins élevée que celle du premier cluster puisque les coefficients correspondants sont négatifs.

➔ l’AIC pour ce premier modèle est : 53547, on comparera cette valeur avec celles issues du modèle utilisant une distribution de poisson.

2- MODELE GLM (POISSON):

En réalisant un modèle GLM avec une distribution de poisson, le code utilisé est:

```
glm(formula = freq ~ as.factor(clust), family = "poisson", data = data)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-0.8724  -0.4421  -0.3878  -0.3129  28.0776

Coefficients:
                  Estimate Std. Error z value Pr(>|z|)
(Intercept)        -1.05315    0.18257  -5.768 8.01e-09 ***
as.factor(clust)2  -0.72096    0.18999  -3.795 0.000148 ***
as.factor(clust)3  -0.14455    0.25413  -0.569 0.569485
as.factor(clust)4  -1.27223    0.18592  -6.843 7.76e-12 ***
as.factor(clust)5   0.08698    0.18371   0.473 0.635858
as.factor(clust)6  -0.79454    0.18397  -4.319 1.57e-05 ***
as.factor(clust)7  -1.53451    0.18519  -8.286  < 2e-16 ***
as.factor(clust)8  -0.73861    0.19946  -3.703 0.000213 ***
as.factor(clust)9  -1.45423    0.25226  -5.765 8.18e-09 ***
as.factor(clust)10 -2.62660    0.35248  -7.452 9.21e-14 ***
as.factor(clust)11 -1.68804    0.20997  -8.040 9.02e-16 ***
as.factor(clust)12 -1.05441    0.18592  -5.671 1.42e-08 ***
as.factor(clust)13 -1.34036    0.18530  -7.233 4.71e-13 ***
as.factor(clust)14 -1.40505    0.18692  -7.517 5.61e-14 ***
as.factor(clust)15 -1.81228    0.18562  -9.763  < 2e-16 ***
as.factor(clust)16 -1.96365    0.18408 -10.667  < 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: 93345  on 123899  degrees of freedom
Residual deviance: 88196  on 123884  degrees of freedom
AIC: 99771

Number of Fisher Scoring iterations: 7
```

- ➔ Les coefficients de ce modèle sont aussi significatifs (les p-values sont faibles) et le coefficient associé à chaque cluster est presque égal à celui associé au même cluster dans le modèle précédent.
- ➔ De même le cluster de référence est le cluster n°1, et on prévoit pour tous les autres clusters qui restent à part le cinquième cluster, une moyenne de fréquence de sinistres moins élevée que celle du premier cluster.
- ➔ l’AIC pour ce modèle est : 99771.

Comparaison :

➔ Nous observons donc la valeur de l'AIC (binomial négatif) est inférieure à AIC (poisson) ce qui signifie un minimum de perte d'information pour une distribution binomiale négative par rapport à une distribution de poisson.

➔ Ces résultats renforcent les résultats issus de la méthode "goodfit", ce qui justifie le choix du premier modèle utilisant une distribution binomial négative pour la fréquence des sinistres.

3.1.4. Modèle glm backward :

Nous avons aussi procédé par modéliser la variable « freq» en fonction des variables comprises dans la base de données par un modèle « glm » intelligent utilisant la méthode « backward » en considérant les deux types de distribution des fréquences (type poisson et binomial négatif) et sans tenant compte de la classification supervisée faite auparavant.

➔ Pour le modèle « glm » utilisant une distribution négative binomiale, le résultat de la dernière itération était comme suite :

```
model=glm.nb(freq~.,data=data)
m2=step(model,direction="backward")

                   Df Deviance   AIC
- puissance_fiscale  1     4342 21386
- age_permis         1     4343 21386
- age                1     4343 21387
<none>                     4342 21387
- age_vehicule       1     4347 21391
- age_contrat        1     4357 21400
- exp                1     6649 23693
- number             1    58253 75296

Step:  AIC=21385.92
freq ~ age + age_vehicule + age_permis + age_contrat + exp +
    number
```

➔ Et Pour le modèle « glm » utilisant une distribution de poisson, le résultat de la dernière itération est comme suit :

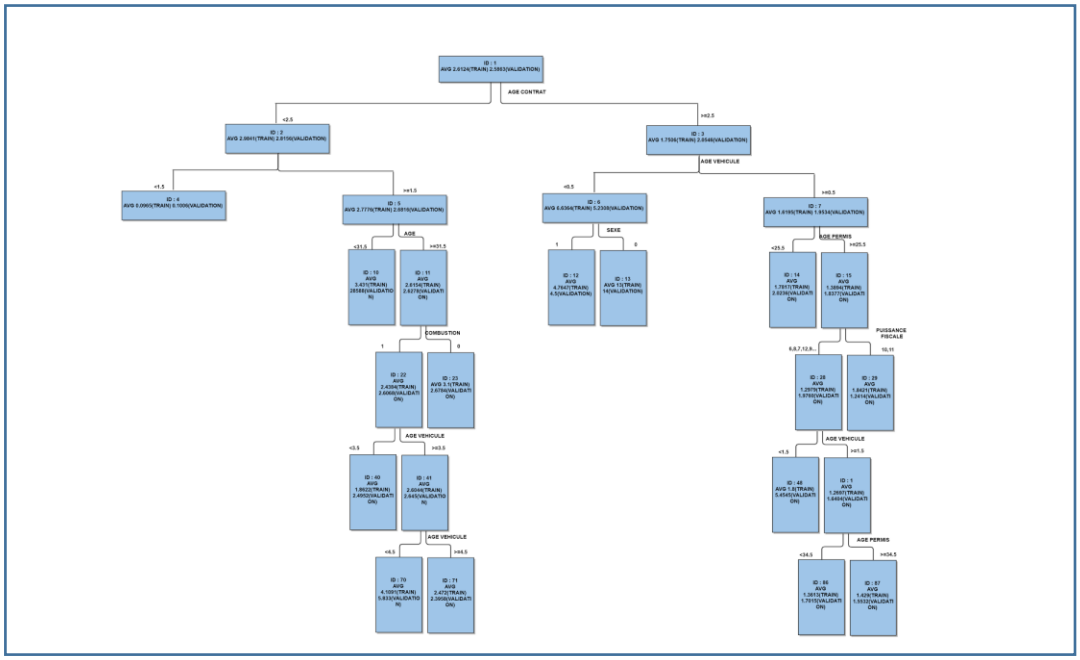
```
model=glm(freq~.,family=poisson,data=data)
m2=step(model,direction="backward")

                   Df Deviance   AIC
<none>                    13773 53773
- puissance_fiscale  1    13776 53775
- Combustion         1    13779 53777
- age_permis         1    13781 53779
- SEXE               1    13791 53789
- age                1    13832 53830
- age_vehicule       1    14230 54229

Step:  AIC=53773.17
freq ~ Combustion + puissance_fiscale + SEXE + age + age_vehicule +
    age_permis
```

➔ Nous observons donc que l'AIC du modèle glm avec une distribution binomiale négative est plus bas que celui avec une distribution de poisson mais ils restent tous les deux grands. Et donc il s'avère que même le modèle glm en utilisant la méthode backward n'est pas si significative.

3.2. Etude n°2 :

3.2.1. Générer l'arbre de décision :

➔ De même pour la première étude, on a réalisé pour la base de données qui ne comprend que les gens pour lesquels on a enregistré au moins un sinistre un arbre de décision comme suite :



➔ Le code utilisé pour générer le vecteur qui associe chaque observation au numéro du cluster correspondant est le suivant :

Voir annexe Partie d'étude n°2

3.2.2. Étude de la moyenne :

➔ On génère la moyenne de fréquence des sinistres pour chaque cluster:

```
tapply(sin_people$freq,sin_people$clust,mean)
       1        2        3        4        5        6        7        8        9       10
11.428571  4.700000  1.645833  3.214286  1.285714  1.526531  1.854839  3.520368  3.177143 2.803993
      11       12       13
 2.178248  4.569767  2.462834
```

→Le cluster dont la moyenne de fréquence de sinistres est la plus élevée est le premier (clust==1) :

➔ Les caractéristiques de ce cluster :

➔ age_vehicule < 0.5 et age_contrat >= 2.5 et SEXE = femme

→Le cluster dont la moyenne de fréquence de sinistres est la plus basse est le premier (clust==1) :

➔ Les caractéristiques de ce cluster :

➔ puissance_fiscale = 6, 8, 7, 12, 9, 17

➔ age_vehicule >= 1.5

➔ age_permis >= 34.5

3.2.3 – performance de l'arbre de décision :

➔ Pour tester la performance de ce dernier arbre de décision nous calculons de même son SSE associé et son SST.

Le code utilisé pour faire cette comparaison est comme suite :

```
predcartsin=rep(0,nrow(sin))
tabsin=tapply(sin$freq,sin$clust,mean)
for (i in 1:nrow(sin)){
  predcartsin[i]=tab[sin$clust[i]]
}
SSE=sum((sin$freq-predcartsin)^2) ->  84690.64
SST=sum((sin$freq-mean(sin$freq))^2)-> 87764.28
```

➔ Nous Observons de même que le SSE est très proche du SST, du coup l'arbre de décision de la deuxième étude ne servira pas d'une manière significative dans la prévision des fréquences des sinistres.

3.2.3 – modèle linéaire généralisé :

a. goodfit test :

➔ Le test de goodfit désigne le degré d'ajustement du modèle aux donnés observées.

➔ D'après les résultats du goodfit en utilisant la méthode de minimisation du khi2 on trouve qu'une distribution binomiale négative des fréquences des sinistres dans la deuxième base de données est plus appropriée qu'une distribution de poisson vu que les valeurs de khi^2 sur chaque cluster sont plus élevées pour une distribution de poisson.

Voir annexes étude n°2 goodfit test

b. GENMOD :

➔ De même que pour l'étude précédente, on réalisera deux modèles GLM, le premier avec une distribution binomiale négative des fréquences et le deuxième avec une distribution de poisson.

1- MODELE GLM (BINOMIAL NEGATIF) :

```
glm.nb(formula = freq ~ as.factor(clust), data = sin, init.theta = 1.801438382,
    link = log)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.9559  -0.7425  -0.5065  -0.0830   7.2288

Coefficients:
                  Estimate Std. Error z value Pr(>|z|)
(Intercept)         2.4361     0.3030   8.040 8.96e-16 ***
as.factor(clust)2  -0.8886     0.3426  -2.593 0.009506 **
as.factor(clust)3  -1.9379     0.3224  -6.012 1.84e-09 ***
as.factor(clust)4  -1.2685     0.3503  -3.621 0.000294 ***
as.factor(clust)5  -2.1848     0.3148  -6.939 3.94e-12 ***
as.factor(clust)6  -2.0131     0.3110  -6.472 9.65e-11 ***
as.factor(clust)7  -1.8183     0.3052  -5.957 2.56e-09 ***
as.factor(clust)8  -1.1776     0.3048  -3.863 0.000112 ***
as.factor(clust)9  -1.2801     0.3057  -4.187 2.82e-05 ***
as.factor(clust)10 -1.4051     0.3057  -4.596 4.30e-06 ***
as.factor(clust)11 -1.6576     0.3080  -5.382 7.38e-08 ***
as.factor(clust)12 -0.9167     0.3175  -2.887 0.003887 **
as.factor(clust)13 -1.5348     0.3046  -5.039 4.67e-07 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for Negative Binomial(1.8014) family taken to be 1)

    Null deviance: 4378.5  on 4656  degrees of freedom
Residual deviance: 3936.8  on 4644  degrees of freedom
AIC: 19091
```

➔ Le résultat du modèle montre que les coefficients de notre modèle sont significatifs puisque la plupart des p-values issues des tests sur les coefficients du modèle sont inférieure à 5% et donc on est sûr à 95% qu'ils ne sont pas nuls.

➔ Le cluster de référence étant le cluster n°1, on prévoit pour tous les autres clusters qui restent une moyenne de fréquence de sinistres plus basse puisque tous les coefficients du modèle sont négatifs.

➔ l'AIC pour ce premier modèle est :19091, on comparera cette valeur avec celles issues du modèle utilisant une distribution de poisson.

2- MODÈLE GLM (POISSON):

```
glm(formula = freq ~ as.factor(clust), family = "poisson", data = sin)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.9981  -1.0597  -0.6885  -0.1225  17.8322

Coefficients:
                  Estimate Std. Error z value Pr(>|z|)
(Intercept)         2.4361     0.1118  21.789  < 2e-16 ***
as.factor(clust)2  -0.8886     0.1400  -6.348 2.18e-10 ***
as.factor(clust)3  -1.9379     0.1372 -14.122  < 2e-16 ***
as.factor(clust)4  -1.2685     0.1537  -8.255  < 2e-16 ***
as.factor(clust)5  -2.1848     0.1295 -16.869  < 2e-16 ***
as.factor(clust)6  -2.0131     0.1232 -16.343  < 2e-16 ***
as.factor(clust)7  -1.8183     0.1148 -15.845  < 2e-16 ***
as.factor(clust)8  -1.1776     0.1135 -10.379  < 2e-16 ***
as.factor(clust)9  -1.2801     0.1145 -11.185  < 2e-16 ***
as.factor(clust)10 -1.4051     0.1147 -12.254  < 2e-16 ***
as.factor(clust)11 -1.6576     0.1178 -14.066  < 2e-16 ***
as.factor(clust)12 -0.9167     0.1227  -7.473 7.82e-14 ***
as.factor(clust)13 -1.5348     0.1136 -13.512  < 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: 14185  on 4656  degrees of freedom
Residual deviance: 13091  on 4644  degrees of freedom
AIC: 24660

Number of Fisher Scoring iterations: 6
```

- ➔ Les coefficients de ce modèle sont aussi significatifs (les p-values sont faibles) et sont presque égaux pour tous les groupes en se comparant avec les coefficients du modèle précédent.
- ➔ De même le cluster de référence est le cluster n°1, et on prévoit pour tous les autres clusters une moyenne de fréquence de sinistres plus basse que celle du premier cluster.
- ➔ l’AIC pour ce modèle est : 24660.

Comparaison :

- ➔ Nous observons encore cette fois pour la deuxième base de données que la valeur de l’AIC (binomial négatif) est inférieure à AIC (poisson).
- ➔ En jointure avec les résultats de la méthode “goodfit” on justifie le choix du premier modèle utilisant une distribution binomial négative pour la fréquence des sinistres.

3.2.4. Modèle glm backward :

- ➔ De même pour la deuxième étude, Nous avons aussi procédé par modéliser la variable « freq» en fonction des variables comprises dans la base de données par un modèle « glm » utilisant la méthode « backward » en

considérant les deux types de distribution des fréquences (type poisson et binomial négatif) et sans tenant compte de la classification supervisée déjà réalisée.

➔ Pour le modèle « glm » utilisant une distribution négative binomiale, le résultat de la dernière itération est comme suit :

```
model=glm.nb(freq~.,data=sin)
m2=step(model,direction="backward")

Step:  AIC=19350.47
freq ~ SEXE + age + age_vehicule + age_permis + age_contrat

              Df Deviance   AIC
<none>             4034.0 19351
- SEXE          1   4038.1 19353
- age_permis    1   4038.7 19353
- age           1   4038.7 19353
- age_vehicule  1   4050.9 19365
- age_contrat   1   4095.6 19410
```

Et Pour le modèle « glm » utilisant une distribution de poisson, le résultat de la dernière itération était comme suite :

```
model=glm(freq~.,family=poisson,data=sin)
m2=step(model,direction="backward")

Step:  AIC=25323
freq ~ SEXE + age + age_vehicule + age_permis + age_contrat

              Df Deviance   AIC
<none>             13768 25323
- SEXE          1    13778 25331
- age_permis    1    13779 25332
- age           1    13780 25333
- age_vehicule  1    13804 25358
- age_contrat   1    13952 25505
```

➔ nous observons encore une fois que l'AIC du modèle « glm » avec une distribution binomiale négative est plus bas que celui avec une distribution de poisson, mais ils restent tous les deux grands. Et donc il s'avère que même le modèle glm en utilisant la méthode backward n'est pas si significative même en utilisant une distribution binomiale négative.

Conclusion :

Ce projet nous a permis, d'appliquer nos connaissances acquises lors de notre cours théorique de statistiques, et nous avons abouti à des arbres de décisions qui permettent de modéliser les fréquences des sinistres, ainsi que des modèles statistiques qui décrivent ces fréquences en fonction des caractéristiques des clients, mais ne sont pas assez performantes, cela est dû peut-être à l'hétérogénéité des observations stockées dans la base de données que nous avons traitée.

Annexes

Cette partie du rapport , annexes est dédiée aux différents codes informatiques développés par notre équipe , lors de notre étude :

➔ Nettoyage et fusion des bases de données sous R :

D'abord, Nous importons les deux bases de données à nettoyer et à fusionner :

```
p=read.csv("P7production.csv",header = T,sep = ";")
s=read.csv("P7-sinistre.csv",header = T,sep = ";")
```

➔ Les étapes pour nettoyer les bases de données sont les suivantes :

- Enlèvement des données manquantes :

```
p=na.omit(p)
s=na.omit(s)
```

- Suppression des observations répétées :

```
p=unique.data.frame(p)
s=unique.data.frame(s)
```

- Création des variables des dates (age, age_permis, age_vehicule…) :

```
p$Date.obtention.du.permis.=as.character(p$Date.obtention.du.permis.)
p$Date.de.naissance=as.character(p$Date.de.naissance)
p$Date.du.premier.effet.=as.character(p$Date.du.premier.effet.)
p$Date.de.Mise.en.Circulation.=as.character(p$Date.de.Mise.en.Circulation.)
p$Date.obtention.du.permis.=as.Date(p$Date.obtention.du.permis., "%Y%m%d")
p$Date.de.naissance=as.Date(p$Date.de.naissance, "%Y%m%d")
p$Date.du.premier.effet.=as.Date(p$Date.du.premier.effet., "%Y%m%d")
p$Date.de.Mise.en.Circulation.=as.Date(p$Date.de.Mise.en.Circulation., "%Y%m%d")
p$age=p$Exercice-as.numeric(format(p$Date.de.naissance,'%Y'))
p$age_vehicule=p$Exercice-as.numeric(format(p$Date.de.Mise.en.Circulation.,'%Y'))
p$age_permis=p$Exercice-as.numeric(format(p$Date.obtention.du.permis.,'%Y'))
p$age_contrat=p$Exercice-as.numeric(format(p$Date.du.premier.effet.,'%Y'))
```

- Détermination du nombre d'accidents par personne chaque année :

```
library(plyr)
install.packages("dplyr")
sc=ddply(s,.(exercice,Police),summarize,charge=sum(Charge),number=length(Police))
write.csv(sc, file = "sc.csv")
```

- Fusion des deux bases de données :

```
a=left_join(p,sc , by = c("Exercice"="exercice","NUMERO_POLICE"="Police"))
write.csv(a, file = "merged2.csv")
```

- Affectation de 0 pour les données perdues (charge et number)

```
a$charge[is.na(a$charge)] <- 0
a$number[is.na(a$number)] <- 0
a$freq=(a$number/a$exposition)
```

- Suppression des variables suivantes après les avoir utiliser pour determiner les différents âges :

```
k$Date.obtention.du.permis.=NULL
k$Date.de.naissance=NULL
k$Date.du.premier.effet.=NULL
k$Date.de.Mise.en.Circulation.=NULL
k$NUMERO_POLICE=NULL
```

➔ Nettoyage avancé de la base de données et discrétisation de la variable des fréquences :

Le code suivant nous a servi pour la suppression des valeurs aberrantes et la discrétisation de la variable « freq » :

```
data=subset(a,SEXE!="" & d$exposition<=1 & d$charge>=0 &d$age_permis>=0 & d$age_vehicule>=0)

data<-data[!(data$freq>30 & data$X1==1),]
#or X1 est le vecteur associant à chaque observation le cluster correspondant
data<-data[!(data$freq>80 & data$X1==2),]
data<-data[!(data$freq>50 & data$X1==6),]
data<-data[!(data$freq>50 & data$X1==7),]
data<-data[!(data$freq>35 & data$X1==8),]
#discrétisation des frequences
data$freq=round(data$freq)
```

.

➔ Durant cette étape nous utiliserons des modèles « glm.nb » simples avec une distribution du type binomiale négative de la variable « freq » qu'on justifie en utilisant la fonction « goodfit » du package « vcd » :

```
goodfit(data$freq,type="nbinomial",method="MinChisq")

Observed and fitted values for nbinomial distribution
with parameters estimated by `MinChisq'

 count observed       fitted pearson residual
     0   119236 1.187069e+05       1.53562420
     1     2695 2.000424e+03      15.52953523
     2      886 9.144731e+02      -0.94156355
     3      260 5.522523e+02     -12.43624166
     4      325 3.740313e+02      -2.53524339
     5      100 2.697934e+02     -10.33725157
     6       97 2.025244e+02      -7.41505047
     7       44 1.562740e+02      -8.98123246
     8       40 1.230432e+02      -7.48643914
     9       15 9.838358e+01      -8.40657748
    10       23 7.962869e+01      -6.34602457
```

11	18	6.508650e+01	-5.83647964
12	38	5.363428e+01	-2.13479682
13	9	4.449986e+01	-5.32166255
14	6	3.713643e+01	-5.10938617
15	12	3.114715e+01	-3.43079673
16	1	2.623809e+01	-4.92708866
17	16	2.218768e+01	-1.31362718
18	2	1.882639e+01	-3.87799700
19	7	1.602279e+01	-2.25409274
20	8	1.367383e+01	-1.53437309
21	2	1.169789e+01	-2.83546004
22	4	1.002979e+01	-1.90395230
23	5	8.617005e+00	-1.23217078
24	7	7.416972e+00	-0.15310647
25	0	6.394945e+00	-2.52882288
26	2	5.522413e+00	-1.49891086
27	1	4.775857e+00	-1.72778560
28	10	4.135788e+00	2.88357047
29	0	3.585987e+00	-1.89367025
30	5	3.112905e+00	1.06957399
31	1	2.705182e+00	-1.03674625
32	0	2.353263e+00	-1.53403487
33	2	2.049088e+00	-0.03429202
34	0	1.785837e+00	-1.33635230
35	1	1.557729e+00	-0.44686590
36	2	1.359846e+00	0.54895885
37	0	1.187999e+00	-1.08995382
38	0	1.038613e+00	-1.01912389
39	1	9.086294e-01	0.09585460
40	0	7.954260e-01	-0.89186656
41	3	6.967533e-01	2.75931295
42	0	6.106772e-01	-0.78145840
43	0	5.355327e-01	-0.73180098
44	0	4.698840e-01	-0.68548085
45	0	4.124919e-01	-0.64225531
46	5	3.622852e-01	7.70510841
47	0	3.183370e-01	-0.56421359
48	0	2.798443e-01	-0.52900312
49	0	2.461109e-01	-0.49609562
50	0	2.165322e-01	-0.46533024
51	0	1.905833e-01	-0.43655842
52	1	1.678073e-01	2.03150810
53	0	1.478068e-01	-0.38445645
54	0	1.302355e-01	-0.36088162
55	0	1.147918e-01	-0.33880940
56	0	1.012123e-01	-0.31813873
57	0	8.926704e-02	-0.29877591
58	0	7.875540e-02	-0.28063393
59	0	6.950183e-02	-0.26363199
60	0	6.135281e-02	-0.24769499
61	0	5.417402e-02	-0.23275313
62	0	4.784782e-02	-0.21874145
63	0	4.227115e-02	-0.20559950
64	0	3.735368e-02	-0.19327100
65	0	3.301617e-02	-0.18170353
66	0	2.918911e-02	-0.17084821
67	0	2.581148e-02	-0.16065950
68	0	2.282967e-02	-0.15109492
69	0	2.019662e-02	-0.14211482
70	0	1.787093e-02	-0.13368221
71	0	1.581622e-02	-0.12576256

```
    72        0 1.400047e-02      -0.11832359
    73        3 1.239552e-02       8.74430532

goodfit(data$freq,type="poisson",method="MinChisq")

Observed and fitted values for poisson distribution
with parameters estimated by `MinChisq'

 count observed       fitted pearson residual
     0   119236 1.539174e-05     3.039230e+07
     1     2695 3.510678e-04     1.438345e+05
     2      886 4.003727e-03     1.400230e+04
     3      260 3.044014e-02     1.490044e+03
     4      325 1.735762e-01     7.796618e+02
     5      100 7.918148e-01     1.114899e+02
     6       97 3.010065e+00     5.417432e+01
     7       44 9.808017e+00     1.091776e+01
     8       40 2.796370e+01     2.276122e+00
     9       15 7.086889e+01    -6.636548e+00
    10       23 1.616438e+02    -1.090488e+01
    11       18 3.351736e+02    -1.732456e+01
    12       38 6.370772e+02    -2.373487e+01
    13        9 1.117769e+03    -3.316386e+01
    14        6 1.821074e+03    -4.253344e+01
    15       12 2.769107e+03    -5.239427e+01
    16        1 3.947510e+03    -6.281330e+01
    17       16 5.296364e+03    -7.255627e+01
    18        2 6.711332e+03    -8.189831e+01
    19        7 8.056726e+03    -8.968128e+01
    20        8 9.188234e+03    -9.577182e+01
    21        2 9.979671e+03    -9.987828e+01
    22        4 1.034658e+04    -1.016788e+02
    23        5 1.026060e+04    -1.012452e+02
    24        7 9.751352e+03    -9.867805e+01
    25        0 8.896686e+03    -9.432225e+01
    26        2 7.804739e+03    -8.832179e+01
    27        1 6.593228e+03    -8.118638e+01
    28       10 5.370856e+03    -7.314967e+01
    29        0 4.224243e+03    -6.499418e+01
    30        5 3.211672e+03    -5.658339e+01
    31        1 2.363050e+03    -4.859064e+01
    32        0 1.684327e+03    -4.104055e+01
    33        2 1.164169e+03    -3.406130e+01
    34        0 7.809809e+02    -2.794604e+01
    35        1 5.089508e+02    -2.251561e+01
    36        2 3.224607e+02    -1.784581e+01
    37        0 1.987827e+02    -1.409903e+01
    38        0 1.193159e+02    -1.092318e+01
    39        1 6.978101e+01    -8.233792e+00
    40        0 3.979062e+01    -6.307981e+00
    41        3 2.213606e+01    -4.067264e+00
    42        0 1.202138e+01    -3.467187e+00
    43        0 6.376604e+00    -2.525194e+00
    44        0 3.305523e+00    -1.818110e+00
    45        0 1.675449e+00    -1.294391e+00
    46        5 8.307623e-01     4.574233e+00
    47        0 4.031646e-01    -6.349524e-01
    48        0 1.915775e-01    -4.376957e-01
    49        0 8.917682e-02    -2.986249e-01
    50        0 4.068042e-02    -2.016939e-01
    51        0 1.819360e-02    -1.348837e-01
```

```
    52        1 7.980293e-03     1.110480e+01
    53        0 3.434365e-03    -5.860345e-02
    54        0 1.450628e-03    -3.808711e-02
    55        0 6.015848e-04    -2.452723e-02
    56        0 2.450261e-04    -1.565331e-02
    57        0 9.804849e-05    -9.901944e-03
    58        0 3.855816e-05    -6.209522e-03
    59        0 1.490623e-05    -3.860858e-03
    60        0 5.666567e-06    -2.380456e-03
    61        0 2.118818e-06    -1.455615e-03
    62        0 7.794807e-07    -8.828792e-04
    63        0 2.822073e-07    -5.312335e-04
    64        0 1.005754e-07    -3.171368e-04
    65        0 3.529245e-08    -1.878697e-04
    66        0 1.219667e-08    -1.103940e-04
    67        0 4.152121e-09    -6.445134e-05
    68        0 1.392722e-09    -3.745662e-05
    69        0 4.603826e-10    -2.130519e-05
    70        0 1.500114e-10    -1.230056e-05
    71        0 4.819139e-11    -6.423757e-06
    72        0 1.526653e-11    -3.708758e-06
    73        3 4.770029e-12     8.088961e+05
```

ETUDE n°1 :

➔ Le code utilisé pour générer le vecteur qui associe chaque observation au numéro du cluster qui le correspond est le suivant :

```
data emines.data;
    set disc;
LENGTH _ARBFMT_12 $     12; DROP _ARBFMT_12;
_ARBFMT_12 = ' ';
IF  NOT MISSING(age_vehicule ) AND
  age_vehicule  <                  4.5 THEN DO;
  IF  NOT MISSING(age_contrat ) AND
    age_contrat  <                  0.5 THEN DO;
    _ARBFMT_12 = PUT( puissance_fiscale , BEST12.);
     %DMNORMIP( _ARBFMT_12);
    IF _ARBFMT_12 IN ('13' ,'14' ) THEN DO;
      VAR1=1;
      END;
    ELSE DO;
      _ARBFMT_12 = PUT( SEXE , BEST12.);
       %DMNORMIP( _ARBFMT_12);
      IF _ARBFMT_12 IN ('0' ) THEN DO;
        VAR1=2;
        END;
      ELSE DO;
        IF  NOT MISSING(age ) AND
          age  <                 22.5 THEN DO;
          VAR1=3;
          END;
        ELSE DO;
          VAR1=4;
          END;
        END;
      END;
    END;
  ELSE DO;
```

```
    IF  NOT MISSING(age_vehicule ) AND
      age_vehicule  <                  1.5 THEN DO;
      VAR1=5;
      END;
    ELSE DO;
      VAR1=6;
      END;
    END;
  END;
ELSE DO;
  IF  NOT MISSING(age_vehicule ) AND
    age_vehicule  <                  9.5 THEN DO;
    _ARBFMT_12 = PUT( puissance_fiscale , BEST12.);
     %DMNORMIP( _ARBFMT_12);
    IF _ARBFMT_12 IN ('7' ,'6' ,'12' ,'13' ,'5' ,'14' ,'26' ,'20' ,'28' ,
    '15' ) THEN DO;
      VAR1=7;
      END;
    ELSE DO;
      IF  NOT MISSING(age ) AND
        age  <                 29.5 THEN DO;
        VAR1=8;
        END;
      ELSE DO;
        IF  NOT MISSING(age ) AND
                           66.5 <= age  THEN DO;
          IF  NOT MISSING(age_permis ) AND
                            39.5 <= age_permis  THEN DO;
            VAR1=9;
            END;
          ELSE DO;
            VAR1=10;
            END;
          END;
        ELSE DO;
          _ARBFMT_12 = PUT( SEXE , BEST12.);
           %DMNORMIP( _ARBFMT_12);
          IF _ARBFMT_12 IN ('0' ) THEN DO;
            VAR1=11;
            END;
          ELSE DO;
            VAR1=12;
            END;
          END;
        END;
      END;
    END;
  ELSE DO;
    IF  NOT MISSING(age ) AND
      age  <                 31.5 THEN DO;
      VAR1=13;
      END;
    ELSE DO;
      IF  NOT MISSING(age_vehicule ) AND
        age_vehicule  <                  16.5 THEN DO;
        _ARBFMT_12 = PUT( puissance_fiscale , BEST12.);
         %DMNORMIP( _ARBFMT_12);
        IF _ARBFMT_12 IN ('8' ,'9' ,'5' ) THEN DO;
          VAR1=14;
          END;
        ELSE DO;
```

```
          VAR1=15;
          END;
        END;
      ELSE DO;
        VAR1=16;
        END;
      END;
    END;
  END;
  clust=VAR1;
 run;
```

Good fit test :

➔ Le test de goodfit désigne le degré d'ajustement du modèle aux donnés observées.

➔ On importe la fonction goodfit qui existe dans la bibliothéque « vcd »

```
library(vcd)
for (i in 1:16){
  minchisq = goodfit(subset(data,data$clust==i)$freq, type = "nbinomial", method = "MinChisq")
  summary(minchisq)
}
```

```
     Goodness-of-fit test for nbinomial distribution

             X^2 df  P(> X^2)
Pearson 4.327373 10 0.9313725

     Goodness-of-fit test for nbinomial distribution

             X^2 df   P(> X^2)
Pearson 25.46505 15 0.04403599

     Goodness-of-fit test for nbinomial distribution

             X^2 df P(> X^2)
Pearson 7.215007 10 0.705003


     Goodness-of-fit test for nbinomial distribution

             X^2 df     P(> X^2)
Pearson 80.01806 28 6.633667e-07

     Goodness-of-fit test for nbinomial distribution

             X^2 df     P(> X^2)
Pearson 92.33876 44 2.772796e-05

     Goodness-of-fit test for nbinomial distribution

             X^2 df     P(> X^2)
Pearson 658.9858 71 3.766682e-96

     Goodness-of-fit test for nbinomial distribution

             X^2 df     P(> X^2)
```

```
Pearson 100.8413 28 3.697983e-10


     Goodness-of-fit test for nbinomial distribution

             X^2 df    P(> X^2)
Pearson 30.96755 14 0.005601777


     Goodness-of-fit test for nbinomial distribution

              X^2 df  P(> X^2)
Pearson 0.5074063  1 0.4762638

     Goodness-of-fit test for nbinomial distribution

           X^2 df  P(> X^2)
Pearson 1.8727  3 0.5992439

     Goodness-of-fit test for nbinomial distribution

             X^2 df  P(> X^2)
Pearson 4.665781  4 0.3233401

     Goodness-of-fit test for nbinomial distribution

             X^2 df    P(> X^2)
Pearson 153.3421 71 5.41011e-08

     Goodness-of-fit test for nbinomial distribution

             X^2 df    P(> X^2)
Pearson 84.33389 29 2.64188e-07

     Goodness-of-fit test for nbinomial distribution

             X^2 df     P(> X^2)
Pearson 80.91912 39 9.263998e-05

     Goodness-of-fit test for nbinomial distribution

             X^2 df     P(> X^2)
Pearson 93.10036 26 1.752504e-09

     Goodness-of-fit test for nbinomial distribution

             X^2 df     P(> X^2)
Pearson 138.4356 26 2.645572e-17
```

Pour la loi de poisson. Nous obtenons, les résultats suivants :

```
for (i in 1:16){
  minchisq = goodfit(subset(data,data$clust==i)$freq, type = "poisson", method = "MinChisq")
  summary(minchisq)
}
Chi-squared approximation may be incorrect

     Goodness-of-fit test for poisson distribution
```

```
            X^2 df      P(> X^2)
Pearson 1359.246 11 7.459323e-285
Chi-squared approximation may be incorrect

     Goodness-of-fit test for poisson distribution

            X^2 df P(> X^2)
Pearson 53162.37 16        0

     Goodness-of-fit test for poisson distribution

            X^2 df P(> X^2)
Pearson 1599.185 11        0
Chi-squared approximation may be incorrect

     Goodness-of-fit test for poisson distribution

            X^2 df P(> X^2)
Pearson 12803658 29        0
Chi-squared approximation may be incorrect

     Goodness-of-fit test for poisson distribution

              X^2 df P(> X^2)
Pearson 4297465016 45        0

     Goodness-of-fit test for poisson distribution

                X^2 df P(> X^2)
Pearson 8.851195e+13 72        0
Chi-squared approximation may be incorrect

     Goodness-of-fit test for poisson distribution

            X^2 df P(> X^2)
Pearson 12267506 29        0
Chi-squared approximation may be incorrect

     Goodness-of-fit test for poisson distribution

            X^2 df P(> X^2)
Pearson 23449.61 15        0
Chi-squared approximation may be incorrect

     Goodness-of-fit test for poisson distribution

            X^2 df     P(> X^2)
Pearson 37.35593  2 7.731484e-09
Chi-squared approximation may be incorrect

     Goodness-of-fit test for poisson distribution

            X^2 df     P(> X^2)
Pearson 234.7816  4 1.233528e-49
Chi-squared approximation may be incorrect

     Goodness-of-fit test for poisson distribution

            X^2 df      P(> X^2)
Pearson 703.9196  5 6.978442e-150
```

```
     Goodness-of-fit test for poisson distribution

                X^2 df P(> X^2)
Pearson 4.97481e+13 72        0

     Goodness-of-fit test for poisson distribution

              X^2 df P(> X^2)
Pearson 16333632 30        0


     Goodness-of-fit test for poisson distribution

               X^2 df P(> X^2)
Pearson 398482493 40        0
Chi-squared approximation may be incorrect

     Goodness-of-fit test for poisson distribution

             X^2 df P(> X^2)
Pearson 9248696 27        0

     Goodness-of-fit test for poisson distribution

              X^2 df P(> X^2)
Pearson 15373206 27        0
```

Etude n° 2:

➔ Le code utilisé pour générer le vecteur qui associe chaque observation au numéro du cluster correspondant est le suivant :

```
data emines.sin;
 set set emines.ac;
LENGTH _ARBFMT_12 $     12; DROP _ARBFMT_12;
_ARBFMT_12 = ' ';
IF  NOT MISSING(age_contrat ) AND
                  2.5 <= age_contrat  THEN DO;
  IF  NOT MISSING(age_vehicule ) AND
    age_vehicule  <                 0.5 THEN DO;
    _ARBFMT_12 = PUT( SEXE , BEST12.);
     %DMNORMIP( _ARBFMT_12);
    IF _ARBFMT_12 IN ('0' ) THEN DO;
      VAR1=1;
      END;
    ELSE DO;
      VAR1=2;
      END;
    END;
  ELSE DO;
    IF  NOT MISSING(age_permis ) AND
                    25.5 <= age_permis  THEN DO;
      _ARBFMT_12 = PUT( puissance_fiscale , BEST12.);
       %DMNORMIP( _ARBFMT_12);
      IF _ARBFMT_12 IN ('10' ,'11' ) THEN DO;
        VAR1=3;
        END;
```

```
      ELSE DO;
        IF  NOT MISSING(age_vehicule ) AND
          age_vehicule  <                  1.5 THEN DO;
          VAR1=4;
          END;
        ELSE DO;
          IF  NOT MISSING(age_permis ) AND
                              34.5 <= age_permis  THEN DO;
            VAR1=5;
            END;
          ELSE DO;
            VAR1=6;
            END;
          END;
        END;
      END;
    ELSE DO;
      VAR1=7;
      END;
    END;
  END;
ELSE DO;
  IF  NOT MISSING(age_vehicule ) AND
    age_vehicule  <                  1.5 THEN DO;
    VAR1=8;
    END;
  ELSE DO;
    IF  NOT MISSING(age ) AND
      age  <                 31.5 THEN DO;
      VAR1=9;
      END;
    ELSE DO;
      _ARBFMT_12 = PUT( Combustion , BEST12.);
       %DMNORMIP( _ARBFMT_12);
      IF _ARBFMT_12 IN ('0' ) THEN DO;
        VAR1=10;
        END;
      ELSE DO;
        IF  NOT MISSING(age_vehicule ) AND
          age_vehicule  <                  3.5 THEN DO;
          VAR1=11;
          END;
        ELSE DO;
          IF  NOT MISSING(age_vehicule ) AND
            age_vehicule  <                  4.5 THEN DO;
            VAR1=12;
            END;
          ELSE DO;
            VAR1=13;
            END;
          END;
        END;
      END;
    END;
  END;
  clust=VAR1;
 run;
```

goodfit test :

➔ Le test de goodfit désigne le degré d'ajustement du modèle aux donnés observées.

```
for (i in 1:13){
  minchisq = goodfit(subset(sin_people,sin_people$clust==i)$freq, type = "nbinomial", method =
"MinChisq")
  summary(minchisq)
}


     Goodness-of-fit test for nbinomial distribution

           X^2 df  P(> X^2)
Pearson 21.7089 34 0.9492392


     Goodness-of-fit test for nbinomial distribution

            X^2 df    P(> X^2)
Pearson 54.00699 26 0.001013135


     Goodness-of-fit test for nbinomial distribution

            X^2 df     P(> X^2)
Pearson 158.0518 13 4.885394e-27


     Goodness-of-fit test for nbinomial distribution

            X^2 df     P(> X^2)
Pearson 55.55279 21 5.869758e-05


     Goodness-of-fit test for nbinomial distribution

            X^2 df     P(> X^2)
Pearson 436.1104 15 1.743665e-83


     Goodness-of-fit test for nbinomial distribution

            X^2 df      P(> X^2)
Pearson 663.8452 26 2.728348e-123

     Goodness-of-fit test for nbinomial distribution

            X^2 df P(> X^2)
Pearson 1646.825 34        0


     Goodness-of-fit test for nbinomial distribution

            X^2 df      P(> X^2)
Pearson 750.1491 44 3.021881e-129
```

```
     Goodness-of-fit test for nbinomial distribution

             X^2 df      P(> X^2)
Pearson 744.6256 71 6.373513e-113


     Goodness-of-fit test for nbinomial distribution

             X^2 df      P(> X^2)
Pearson 817.9039 44 3.581168e-143


     Goodness-of-fit test for nbinomial distribution

             X^2 df      P(> X^2)
Pearson 645.6037 39 6.411215e-111


     Goodness-of-fit test for nbinomial distribution

             X^2 df     P(> X^2)
Pearson 236.2521 71 1.263939e-19


     Goodness-of-fit test for nbinomial distribution

             X^2 df      P(> X^2)
Pearson 1297.633 39 6.555615e-247
```

Pour la loi de poisson :

```
for (i in 1:13){
  minchisq = goodfit(subset(sin_people,sin_people$clust==i)$freq, type = "poisson", method =
"MinChisq")
  summary(minchisq)
}


     Goodness-of-fit test for poisson distribution

             X^2 df P(> X^2)
Pearson 167393.8 35        0


     Goodness-of-fit test for poisson distribution

             X^2 df P(> X^2)
Pearson 9289.651 27        0


     Goodness-of-fit test for poisson distribution

             X^2 df      P(> X^2)
Pearson 1057.993 14 5.598189e-217


     Goodness-of-fit test for poisson distribution

            X^2 df P(> X^2)
Pearson 6347.07 22        0
```

```
     Goodness-of-fit test for poisson distribution

             X^2 df P(> X^2)
Pearson 3690.724 16        0


     Goodness-of-fit test for poisson distribution

             X^2 df P(> X^2)
Pearson 137629.1 27        0


     Goodness-of-fit test for poisson distribution

            X^2 df P(> X^2)
Pearson 2948397 35        0
     Goodness-of-fit test for poisson distribution

             X^2 df P(> X^2)
Pearson 87176560 45        0


     Goodness-of-fit test for poisson distribution

                X^2 df P(> X^2)
Pearson 595196745579 72        0


     Goodness-of-fit test for poisson distribution

             X^2 df P(> X^2)
Pearson 56034633 45        0


     Goodness-of-fit test for poisson distribution

            X^2 df P(> X^2)
Pearson 9825331 40        0
N

     Goodness-of-fit test for poisson distribution

                X^2 df P(> X^2)
Pearson 815008210873 72        0

     Goodness-of-fit test for poisson distribution

             X^2 df P(> X^2)
Pearson 12726433 40        0
```