La tarification a priori

Auteurs: ABDOUL OUDOUSS DIAKITE, OTHMANE ETTADLAOUI
Encadreur: LAHCEN DOUGE
Module: Assurance non vie
Université: Cadi Ayyad, Faculté des Sciences et Techniques
Code source: https://bookdown.org/abdouloudoussdiakite/Tarification/

12 June, 2022

Contents

1 Introduction

L'assurance est une opération de transfert d'un risque ou d'une partie d'un risque d'un assuré à un assureur. Ce dernier s'engage à indemniser son client en cas de survenance d'un sinistre pendant toute la période couverte par le contrat. La prime reçue par l'assureur doit refléter le risque qu'il est prêt à couvrir d'où la nécessité de se demander combien faut-il recevoir en prime pour assurer λ niveau de risque ?

1.1 Objectif

Dans ce projet, nous allons faire une étude sur des données que nous décrirons plus tard. Le but est d'appliquer différentes méthodes vues en assurance non-vie et de ressortir le meilleur modèle de tarification. Bien sûr nous allons commencer par une étude statistique de nos données ainsi qu'un ensemble de représentations graphiques.

1.2 Les données du projets

Cette base de données contient 16082 images d'une assurance automobile. (*Télécharger*). Le code suivant permet de charger les données qui se trouvaient au préalable dans le dossier *Data*.

```
library(haven)
database <- read_sas("Data/base5.sas7bdat",
    NULL)
```

Table 1: A table of the first 10 rows of our data.

NAP	PERMIS	DEB_IMAG	FIN_IMAG	SEX	STATUT	CSP	USAGE
83	332	2004-01-01	2004-02-01	M	A	50	3
916	333	2004-02-01	NA	M	A	50	3
550	173	2004-05-15	2004-12-03	M	A	50	2
89	364	2004-11-29	NA	F	A	55	2
233	426	2004-02-07	2004-05-01	M	A	60	1
666	429	2004-05-01	NA	M	A	60	1
80	461	2004-04-02	2004-05-01	M	A	48	3
666	462	2004-05-01	NA	M	A	48	3
173	405	2004-10-29	NA	F	A	50	2
474	386	2004-01-01	2004-06-22	M	A	55	2

1.3 Description des données

Évidemment, il est très difficile de comprendre certaines abréviations dans les données que nous venons de télécharger. Ne vous inquiétez surtout pas ! Le tableau suivant contient la description de chaque colonne de la base 5 que nous appellerons dorénavant ***database***.

Table 2: Descriptions de database.

Description	Code
age du conducteur	agecond
ancienneté de permis	permis
sexe du conducteur	sex

Description	Code
statut matrimonial	statut
catégorie socio-professionnelle	csp
usage du véhicule	usage
option kilométrage limité	k8000
zone géographique	zone
coefficient de réduction majoration (bonus/malus)	RM
date de début d'image	deb_imag
date de fin d'image	fin_imag
nombre d'années-police	nap
nombre de sinistres responsables dans les 4 années précédent l'image	sinap1
nombre de sinistres non responsables dans les 4 années précédent l'image	sinap2
nombre de sinistres parking dans les 4 années précédent l'image	sinap3
nombre de sinistres incendie/vol dans les 4 années précédent l'image	sinap4
nombre de sinistres bris de glace dans les 4 années précédent l'image	sinap5
nombre de mises en demeure dans les 4 années précédent l'image	sinap6
charge de sinistres	charge

Passons maintenant à l'étude statistique !

2 Études statistiques

2.1 Nettoyage des données

Cette partie de ce document sera consacrée à l'étude statistique de notre jeu de données.
Il existe quelques incohérences au niveau des données telles que des charges négatives ou nulles pour des nombres de sinistres positifs et des nombres de sinistres nuls pour des charges positives. Nous avons donc choisi de remplacer par 0 toutes les charges négatives pour des sinistres nuls, tous les nombres de sinistres positifs pour des charges nulles et par la valeur absolue des charges pour les nombre de sinistres positifs.

```
library(dplyr)
# Ajout de la somme des sinistres par police
database$SumSINAPS <- database %>% select(starts_with("SINAP")) %>%
                                   apply(., 1, sum)
# Transformation des données
database <-  database %>% mutate(SumSINAPS = case_when(CHARGE==0~0,
                                               TRUE~SumSINAPS)) %>%
                        mutate(CHARGE = case_when(SumSINAPS==0~0,
                                                  SumSINAPS>0~abs(CHARGE),
                                                  TRUE~0))
```

Nous venons de créer avec le code précédent une nouvelle colonne dans la base de données `database` que nous avons appelé `SumSINAPS`. On peut facilement faire un sommaire de la somme des sinistres ainsi que des charges avec la fonction `summary` afin de connaître les mesures de tendance de cette variable.

```
summary(database[,c("CHARGE","SumSINAPS")])
```

```
##      CHARGE          SumSINAPS
##  Min.   :    0.0   Min.   : 0.000
##  1st Qu.:    0.0   1st Qu.: 0.000
```

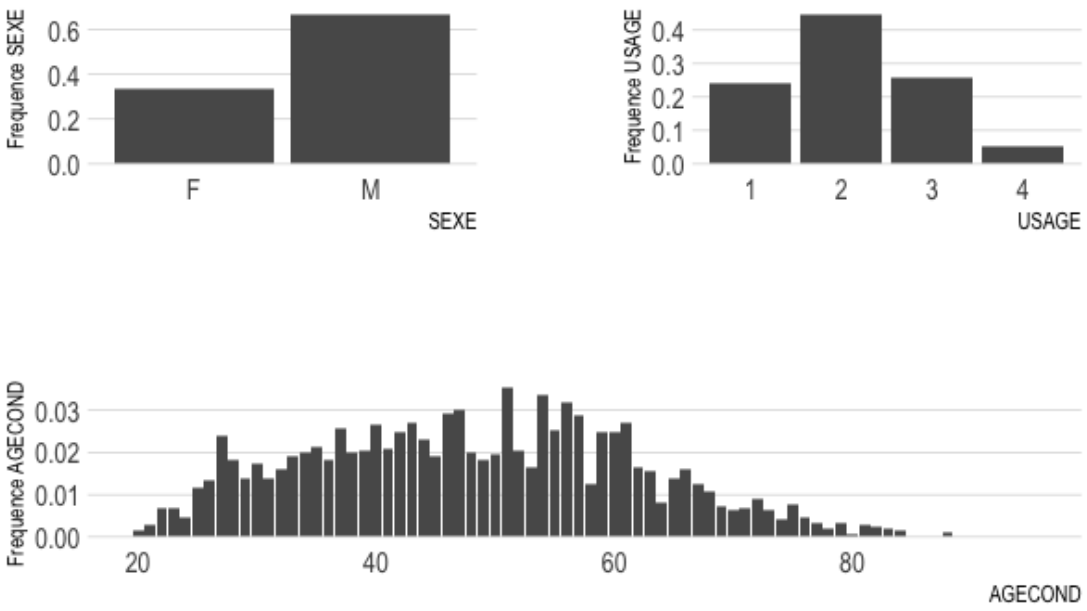
```
##  Median :    0.0   Median : 0.000
##  Mean   :  170.1   Mean   : 0.169
##  3rd Qu.:    0.0   3rd Qu.: 0.000
##  Max.   :95151.0   Max.   :10.000
```

2.2 Représentations graphiques

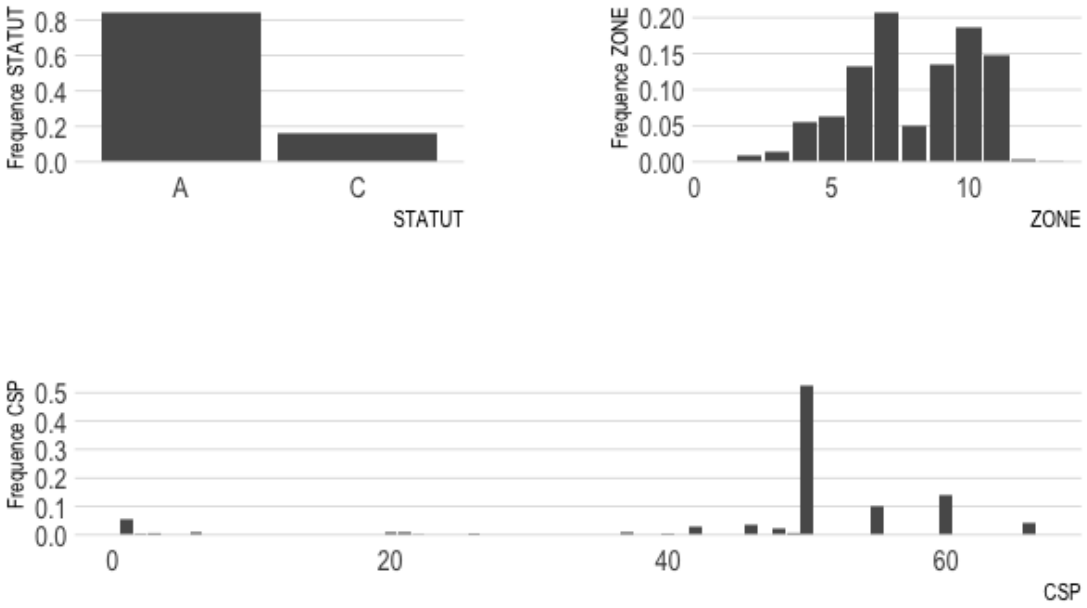
Dans cette partie du projet, nous allons effectuer un ensemble de représentations graphiques afin de savoir l'impact des variables sur les sinistres.
Ces graphes suivants indiquent la fréquence des sinistre en fonction des différentes variables : sexe, zone, catégorie socio-professionnelle, age, usage du véhicule et statut matrimonial.

```
library(dplyr)
library(ggplot2)
B1 = database %>% select(SEXE = SEX,STATUT,ZONE,CSP,USAGE,AGECOND) #selection des variables.
plot_ <- function(df, N){
  name = names(df)
  for (i in name){
    df0 <-  data.frame(df[[i]],N) # Creation d'une df(Variables,N)
    colnames(df0) <- c(i,'N')
    s <- df0 %>% group_by(valeur = df0[[i]]) %>%
      summarise(total = sum(N)) # Creation d'une df(groupe de Variables, Total)
    assign(paste0("table",i),s,.GlobalEnv)

    # debut du code pour la figure
    figure <- ggplot(data = s, aes(x = valeur, y = total/sum(total))) +
      geom_col() +
      xlab(i) +  # Ajout du label de x
      ylab(paste0("Frequence ",i)) + # Ajout du label de y
      hrbrthemes::theme_ipsum(grid = "Y") # Ajout d'un theme pour la figure
    # fin du code pour la figure

    assign(paste0("fig",i),figure,.GlobalEnv) #assigantion de la figure i
  }
}
SumSINAPS=database$SumSINAPS
plot_(B1,SumSINAPS) # Execution de la fonction precedente
```

```
require(patchwork)
(figSEXE|figUSAGE)/figAGECOND
```




```
(figSTATUT|figZONE)/figCSP
```



On constate que pour la variable `SEX` la fréquence des sinistres des hommes(M) est plus que le double de celle des femmes(F).

Les clients du type d'usage de véhicule 2 ont les fréquences les plus élevées de sinistres, suivit dans cet ordre par les usages 1,3 et 4.
Les conducteurs de statut A ont une fréquence de sinistre à peu près cinq fois plus élevées que celle des conducteurs de statut C.
On peut utiliser ce même raisonnement pour interpréter tous les autres graphes.

2.3 Tests statistiques

Nous allons faire des tests statistiques histoire de connaître quelle loi suit la somme totale des sinistres `SumSINAPS`. Les tests d'adéquation servent à tester si un échantillon est distribué selon une loi de probabilité donnée. Ils permettent de décider, avec un seuil d'erreur α spécifié, si les écarts présentés par l'échantillon par rapport aux valeurs théoriques attendues sont dus au hasard ou sont au contraire significatifs.

Estimation par la méthode du maximum de vraisembance.

Soit X une variable aléatoire réelle de loi discrète ou continue dont on veut estimer le paramétre θ. Alors on définit une fonction f telle que:

$$f(x;\theta) = \begin{cases} f_\theta(x) : si\ X\ variable\ aléatoire\ continue \\ P_\theta(X = x) : si\ X\ variable\ aléatoire\ discrète \end{cases}$$

On appelle fonction de vraisemblance de θ pour une réalisation $(x_1, \ldots, x_n)$ d'un échantillon, la fonction de θ:

$$L(x_1, \ldots, x_n; \theta) = f(x_1, \ldots, x_n; \theta) = \prod_{i=1}^{n} f(x;\theta)$$

La méthode consiste à estimer θ par la valeur qui maximise L, cette méthode s'appelle méthode du maximum de vraisemblance(MLE[1]), on choisit la valeur θ qui réalise le maximum de $L(x_1, \ldots, x_n; \theta)$. Pour cela on cherche θ telle que:

$$\frac{\partial L}{\partial \theta} = 0 \ \ et \ \ \frac{\partial^2 L}{\partial \theta^2} \leq 0$$

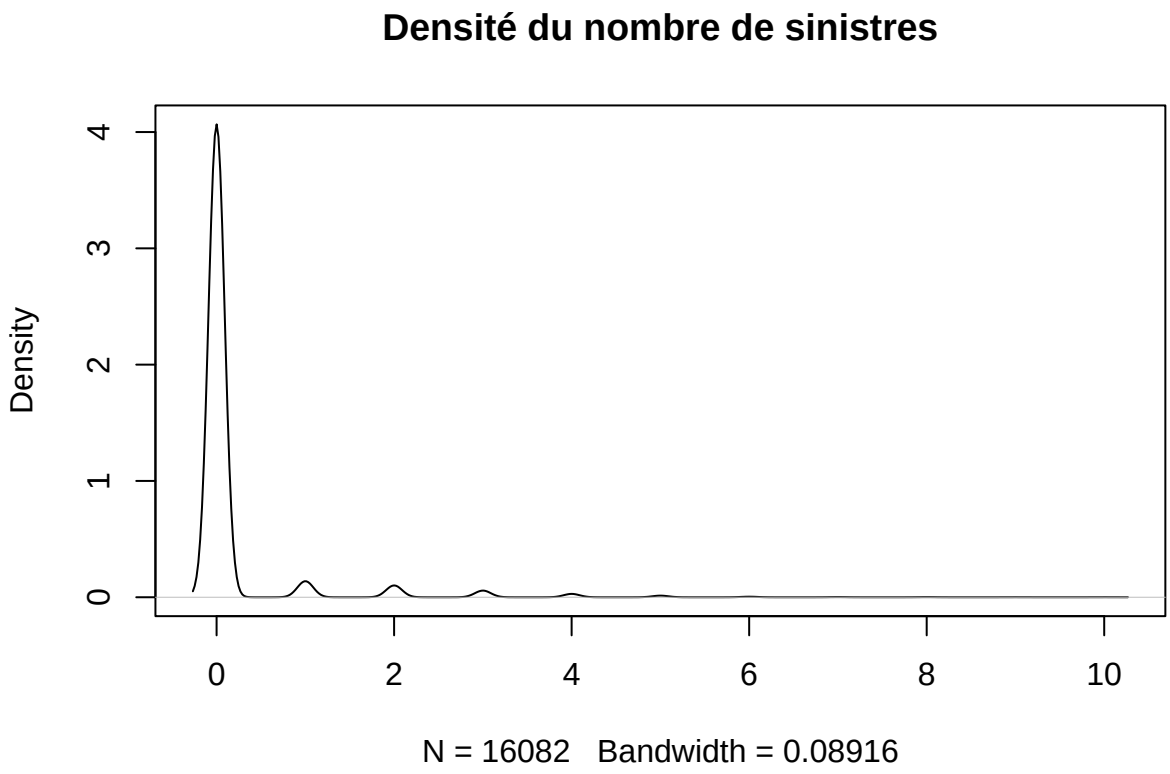
On passe en général au logarithme, c'est à dire on cherche θ telle que:

$$\frac{\partial \big(ln(L)\big)}{\partial \theta} = 0 \ \ et \ \ \frac{\partial^2 \big(ln(L)\big)}{\partial \theta^2} \leq 0$$

Nombre de sinistres

Pour faire les tests, nous allons générer une variable aléatoire X suivant une loi choisie, puis tester si la somme totale des sinistres `SumSINAPS` est de même loi.

[1]MLE : *Maximum likelihood estimation*



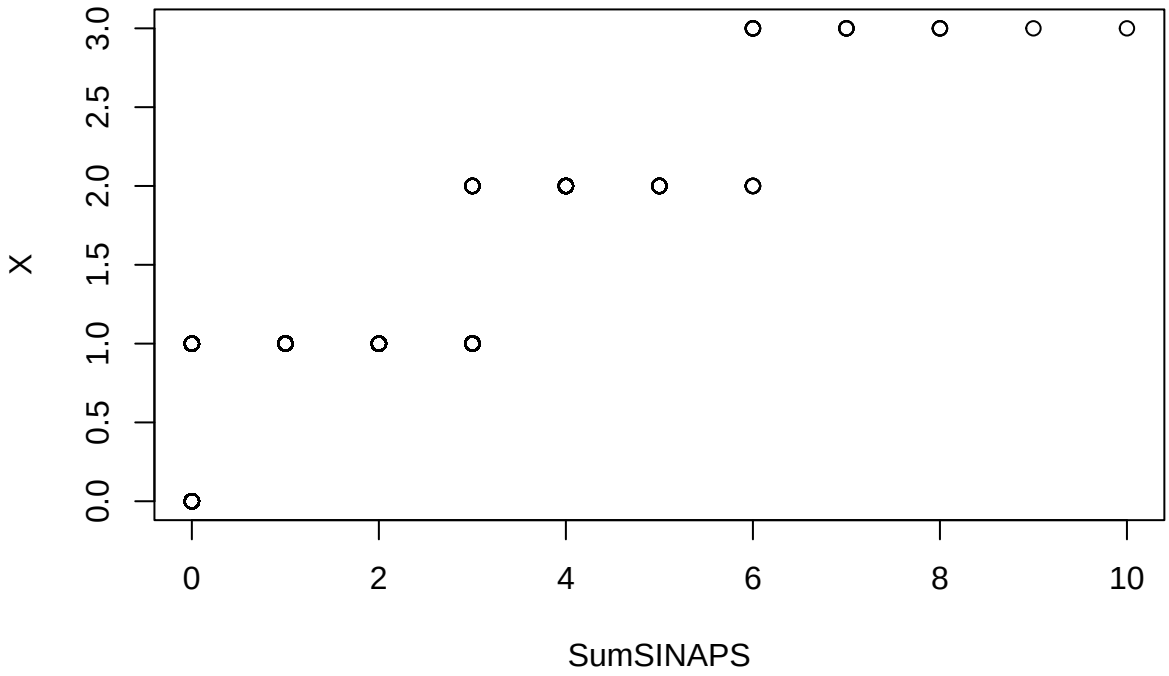
Loi de poisson Dans le cas habituel théorique, le nombre de sinistres suivrait la loi de poisson ce qui nous pousse à faire ce premier test.

```
#Generer une v.a de longueur indentique a celle de SumSINAPS et suivant la loi de poisson
X=rpois(length(SumSINAPS),mean(SumSINAPS))
#test de poisson
ks.test(SumSINAPS,X)
```

```
##
##  Two-sample Kolmogorov-Smirnov test
##
## data:  SumSINAPS and X
## D = 0.074244, p-value < 2.2e-16
## alternative hypothesis: two-sided
```

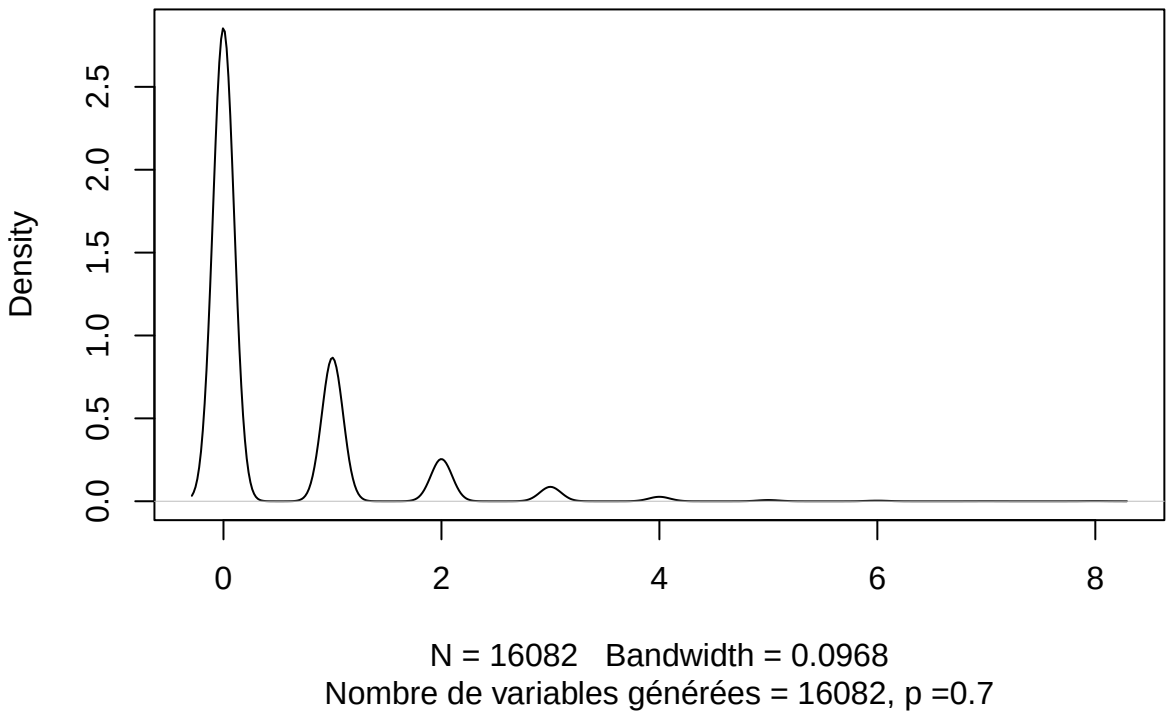
La *p-value* est inférieure à 0.5 donc on rejette l'hypothèse H_0 selon laquelle la somme des sinistres suit une loi de poisson. Pour consolider cette conclusion on peut faire un test visuel avec la fonction `qqplot()`. Si `SumSINAPS` et X suivent la même loi, alors le nuage de points doit s'apparenter à une droite.

```
qqplot(SumSINAPS,X)
```



On voit clairement que le nuage de points est loin d'être sur une même droite, ce qui conforte notre conclusion précédente.

Loi géométrique A part la loi de poisson, le nombre de sinistre pourrait suivre la loi géométrique. D'ailleur la densité de `SumSINAPS` que nous avons tracer a la même forme que le graphe d'une densité de la loi géométrique.



On peut tester graphiquement si une variable suit une loi géométrique toujours avec la même fonction `qqplot()`. Mais faudra trouver le paramètre p de la loi loi géométrique pour pouvoir générer notre échantillon X. Il existe un package permettant d'entrainer une donnée selon la loi loi géométrique du nom de fitdistrplus[2] afin de trouver l'estimateur du paramètre p par la méthode du maximum de vraisemblance.

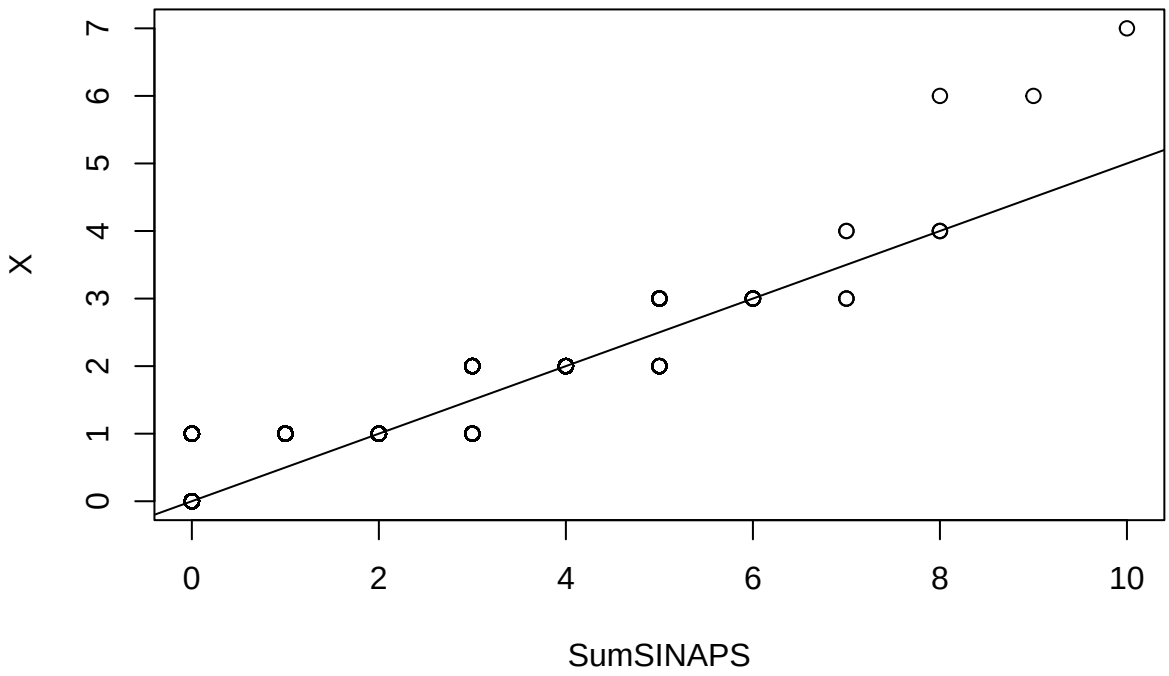
```
library(fitdistrplus)
fitSINAPS <- fitdist(data = SumSINAPS,distr = "geom",
                     method = "mle")
summary(fitSINAPS)
```

```
## Fitting of the distribution ' geom ' by maximum likelihood
## Parameters :
##       estimate  Std. Error
## prob 0.8554255 0.002564721
## Loglikelihood:  -7767.809   AIC:  15537.62   BIC:  15545.3
```

La sortie de la fonction `summary` indique que $p = 0.8554255$, on va l'utiliser pour générer X puis faire le test de `qqplot`.

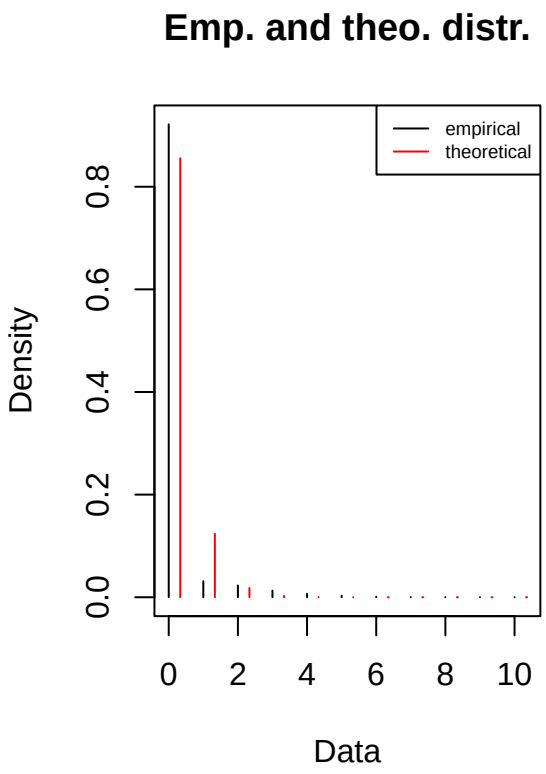
```
X = rgeom(length(SumSINAPS),prob =0.8554255)
qqplot(SumSINAPS,X)
abline(0,0.5)
```

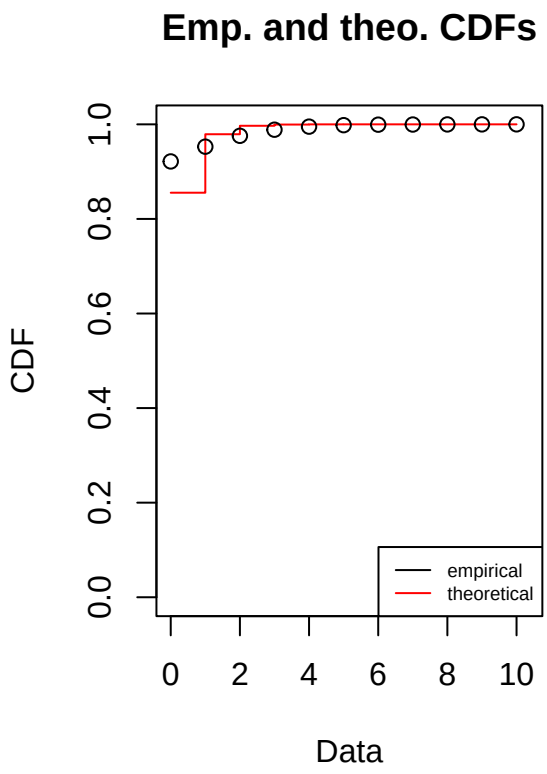
[2]https://cran.r-project.org/web/packages/fitdistrplus/index.html



Ce graphe s'apparente plus avec une droite que celui tracer dans la section précédente. En plus le modèle entraîné présente des valeurs empériques et théoriques très proche.

```
plot(fitSINAPS)
```





On peut donc supposé que $SumSINAPS \sim \mathcal{G}(p := 0.8554255)$

Charges de sinistres

Comme nous l'avons fait précédemment, nous allons chercher le meilleur paramétré μlog et $sdlog$ pour la charge des sinistres qui suivrait une loi lognormale.

```
library(fitdistrplus)
Charge <- database$CHARGE
Charge <- Charge[which(Charge>0)]
fitCharge <- fitdistr(Charge,"log-normal")
fitCharge
```

```
##     meanlog       sdlog
##   6.66608145   1.38553971
##  (0.03901768) (0.02758967)
```

Autrement dit, si la charge de sinistres suit une loi lognormale alors ce sera:

$$log\mathcal{N}(6.66608145, 1.38553971)$$

On peut faire le test de Kolmogorov-Smirnov[3] pour conforter se résultat en tenant compte de l'écart type des erreurs d'estimation de la fonction `fitdistr`.

[3] https://en.wikipedia.org/wiki/Kolmogorov%E2%80%93Smirnov_test

```r
estimated = fitCharge$estimate + fitCharge$sd
X <- rlnorm(length(Charge),estimated[["meanlog"]],estimated[["sdlog"]])
ks.test(Charge,X)
```

```
##
##  Two-sample Kolmogorov-Smirnov test
##
## data:  Charge and X
## D = 0.055511, p-value = 0.04106
## alternative hypothesis: two-sided
```

La $p - value = 0.1046 > \alpha = 5\%$ alors on rejette pas l'hypothèse H_0, alors la charge de sinistres pourrait suivre la loi $log\mathcal{N}(6.7, 1.4)$.

```r
library(ggplot2)
dfdensity = data.frame(Charge,X)
ggplot(data = dfdensity)+
  geom_density(aes(x=Charge,color ="blue"))+
  geom_density(aes(x=X,color="red"))+
  scale_colour_manual(values = c( "red", "blue"),
                      labels= c("Charge","X"))+
  ggtitle("Densités",
          subtitle = "X a été généré aléatoirement\nSuivant la loi loi lognormale")+
  ggthemes::theme_tufte()
```

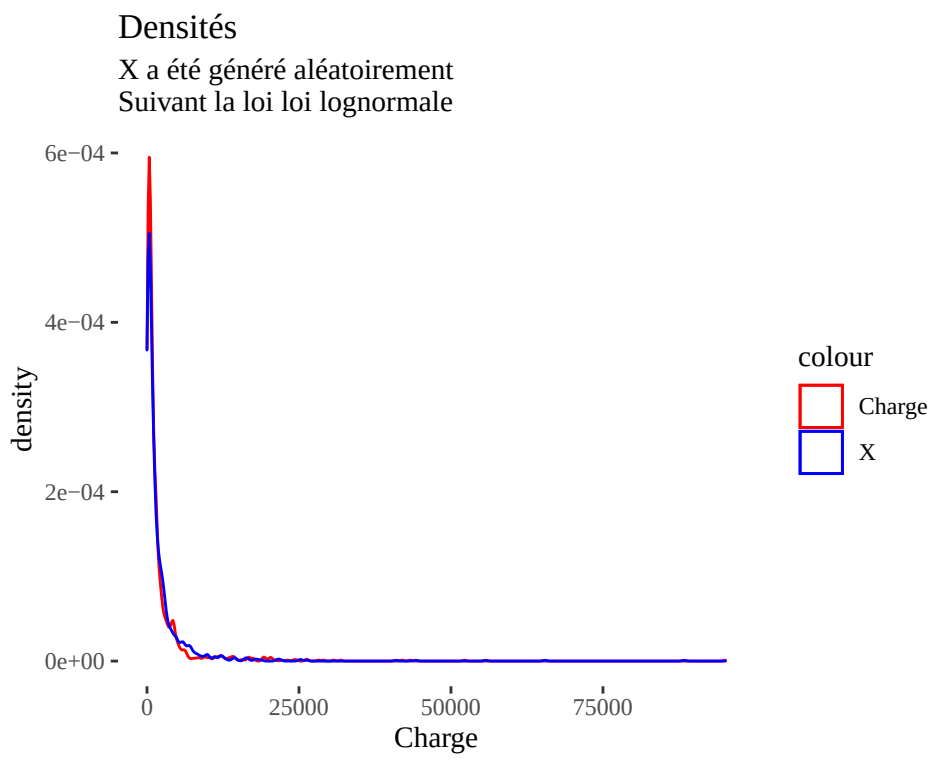


Figure 1: On peut constater que les densité de X et Charge sont presque les mêmes

3 Les modèles linéaires généralisés

Les modèles linéaires généralisés sont une généralisation du modèle linéaire Gaussien, obtenu en autorisant d'autres lois (conditionnelles) que la loi Gaussienne. Les lois possibles doivent appartenir à la famille exponentielle, c'est à dire dont la densité (ou mesure de probabilité dans le cas discret) s'écrit :

$$f(y|\theta,\phi) = \exp\left(\frac{y\theta - b(\theta)}{\phi} + c(y,\phi)\right)$$

La fonction lien est la fonction qui permet de lier les variables explicatives X à la prédiction μ, alors que la loi apparaît via la fonction variance, sur la forme de l'hétéroscédasticité et l'incertitude associée à la prédiction.

La cellule de code suivante permet d'entraîner 3 régressions GLM différentes puis de les comparer.

```
library(dplyr)
dataglm = database[,c("SumSINAPS","SEX","STATUT","ZONE","CSP","USAGE","AGECOND")]
set.seed(1234)
echantillon = sample(1:length(SumSINAPS))[1:(length(SumSINAPS)%/%4)]
datatrain = dataglm[-echantillon,]
datatest = dataglm[echantillon,]
#GLMs
# Gaussian identity
regNId  <- glm(SumSINAPS~.,family=gaussian(link="identity"),data = datatrain)
# Poisson identity
regPId  <- glm(SumSINAPS~.,family=poisson(link="identity"),data = datatrain)
# Poisson log
regPlog <- glm(SumSINAPS~.,family=poisson(link="log"),data = datatrain)
comparemodel <-  performance::compare_performance(regNId,regPId,regPlog)
comparemodel[,c("Name","AIC","AIC_wt","BIC")]
```

```
## # Comparison of Model Performance Indices
##
## Name    |       AIC | AIC weights |       BIC
## ------------------------------------------------
## regNId  | 24500.744 |     < 0.001 | 24559.927
## regPId  | 12971.160 |       0.926 | 13022.944
## regPlog | 12976.219 |       0.074 | 13028.004
```

Sous le critère de l'AIC et du BIC on peut choisir le modèle `regPlog` c'est à dire celui qui a pour fonction de lien *logpoisson*.

Régression logistique ou probit

On peut modéliser l'éxistance d'un sinistre pour un client. Pour cela il va falloire créer une nouvelle colonne `Indicatrice` dans notre `database` qui prend 0 si `SumSINAPS` est nul et 1 sinon.

```
database$Indicatrice <- ifelse(SumSINAPS>0,1,0)
```

Passons maintenant La régression logistique !

```r
dataglm = database[,c("Indicatrice","AGECOND")]
datatrain = dataglm[-echantillon,]
datatest = dataglm[echantillon,]
logistic <- glm(Indicatrice~.,data = datatrain,family = binomial(link = 'logit'))
Probit <- glm(Indicatrice~.,data = datatrain,family = binomial(link = 'probit'))
```

La cellule suivante permet de faire des prédictions pour chaque modèle :

```r
predLogistic <- predict(logistic,datatest[,-1],type = "response")
predProbit <- predict(Probit,datatest[,-1],type = "response")
```

4 Modèle collectif

Dans ce projet, on a trouver les lois que pourraient suivre la charge de sinistres(X) ainsi que le nombre de sinistres (N). On sait que la charge total de sinistres S vaut :

$$S = \sum_{i=0}^{N} X_i$$

Donc on peut écrire :

$$\mathbb{E}(S) = \mathbb{E}(N) \times \mathbb{E}(X)$$

Or $N \sim \mathcal{G}(p := 0.8554255)$ et X $log\mathcal{N}(6.66608145, 1.38553971)$ alors:

$$\mathbb{E}(N) = 1/p = 1.169009 \;\; \mathbb{E}(X) = \exp(\mu + \frac{\sigma^2}{2}) = 2050.71$$

d'où $\mathbb{E}(S) = 2050.71 \times 1.169009 = 2397.298$. Cette dernière correspond à la prime pure que devrait payer un assureur, on peut constater qu'elle est très proche de la moyenne empirique des Charge de sinistres de la base de données dont nous disposons.

```r
cat("Prime pure=",2397.298,";","mean(Charge)=",mean(Charge))
```

```
## Prime pure= 2397.298 ; mean(Charge)= 2169.361
```

5 Conclusion

Tout au long de ce projet, de différentes manières, nous avons pu modéliser le nombre et la sévérité des sinistres. On a commencer par une étude statistiques qui nous a permis de trouver des lois adéquates aux nombres et montants de sinistres. Par des modèles linéaires généralisés, il est aussi possible de tarifier des contrats d'assurance de ce type.