Actuariat de l'Assurance Non-Vie # 3

A. Charpentier (Université de Rennes 1)

ENSAE ParisTech, Octobre/Décembre 2016.

http://freakonometrics.hypotheses.org

Modélisation économétrique d'une variable de comptage

Références: Frees (2010), chapitre 12 (p 343-361) Greene (2012), section 18.3 (p 802-828) de Jong & Heller (2008), chapitre 6, sur la régression de Poisson. Sur les méthodes de biais minimal, de Jong & Heller (2008), section 1.3, Cameron & Trivedi (1998), Denuit *et al.* (2007) et Hilbe (2007).

Remarque: la régression de Poisson est un cas particulier des modèles GLM, avec une loi de Poisson et une fonction de lien logarithmique.

Utilisation du '**modèle collectif**' $S_t = \sum_{i=1}^{N_t} Y_i$, et $\mathbb{E}[S_1|\boldsymbol{X}] = \mathbb{E}[N_1|\boldsymbol{X}] \cdot \mathbb{E}[Y|\boldsymbol{X}]$.

Base pour les données de comptage

On dispose de deux bases

- la base de souscription (avec des informations sur l'assuré et le véhicule)
- la base de sinistres avec les sinistres RC (assurance responsabilité civile, obligatoire) et DO (assurance dommage, non obligatoire)

```
> sinistre=read.table("http://freakonometrics.free.fr/sinistreACT2040.txt",header=TRUE,sep=";")
> contrat=read.table("http://freakonometrics.free.fr/contractACT2040.txt",header=TRUE,sep=";")
> contrat=contrat[,1:10]
> names(contrat)[10]="region"
```

La clé est le numéro de police, `nocontrat`.

Base pour les données de comptage

```
> sinistre_RC=sinistre[(sinistre$garantie=="1RC")&(sinistre$cout>0),]
> T_RC=table(sinistre_RC$nocontrat)
> T1_RC=as.numeric(names(T_RC))
> T2_RC=as.numeric(T_RC)
> nombre_1_RC = data.frame(nocontrat=T1_RC,nb_RC=T2_RC)
> I_RC = contrat$nocontrat%in%T1_RC
> T1_RC= contrat$nocontrat[I_RC==FALSE]
> nombre_2_RC = data.frame(nocontrat=T1_RC,nb_RC=0)
> nombre_RC=rbind(nombre_1_RC,nombre_2_RC)
```

On compte ici le nombre d'accidents RC, par contrat.

Remarque dans le modèle collectif, $Y_i > 0$ (on exclut les sinistres classés '*sans suite*')

Base pour les données de comptage

```
> sinistre_DO=sinistre[(sinistre$garantie=="2DO")&(sinistre$cout>0),]
> T_DO=table(sinistre_DO$nocontrat)
> T1_DO=as.numeric(names(T_DO))
> T2_DO=as.numeric(T_DO)
> nombre_1_DO = data.frame(nocontrat=T1_DO,nb_DO=T2_DO)
> I_DO = contrat$nocontrat%in%T1_DO
> T1_DO= contrat$nocontrat[I_DO==FALSE]
> nombre_2_DO = data.frame(nocontrat=T1_DO,nb_DO=0)
> nombre_DO=rbind(nombre_1_DO,nombre_2_DO)
```

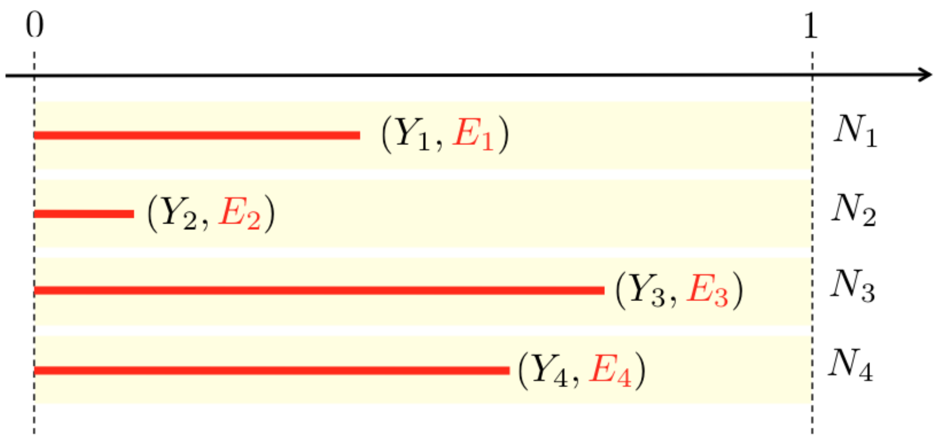
On compte ici le nombre d'accidents DO, par contrat.

Et on crée la base finale

```
> freq = merge(contrat,nombre_RC)
> freq = merge(freq,nombre_DO)
```

La notion d'exposition

Les contrats d'assurance sont annuels, mais dans la base, on peut avoir des données censurées (arrêt de la police ou image au 31 Décembre)



On observe Y et E, la variable d'intŕêt est N.

La notion d'exposition

Dans notre base, la fréquence pour les DO est de l'ordre de 6.5%,

```
> Y = freq$nb_DO
> E= freq$exposition
> sum(Y)/sum(E)
[1] 0.06564229
> weighted.mean(Y/E,E)
[1] 0.06564229
```

Fréquence de sinistre et segmentation

```
> X1 = freq$carburant
> tapply(Y,X1,sum)/tapply(E,X1,sum)
         D          E
0.07068945 0.06110016
> library(weights)
> wtd.t.test(x=(Y/E)[X1=="D"], y=(Y/E)[X1=="E"],
+            weight=E[X1=="D"], weighty=E[X1=="E"],samedata=FALSE)
$test
[1] "Two Sample Weighted T-Test (Welch)"

$coefficients
     t.value           df      p.value
1.768349e+00 2.631555e+04 7.701412e-02

$additional
 Difference      Mean.x      Mean.y    Std. Err
0.009589286 0.070689448 0.061100161 0.005422733
```

Fréquence de sinistre et segmentation

On peut aussi envisager une analyse de la variance

```
> summary(lm(Y/E~X1,weights=E))
Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.070689   0.002866  24.662   <2e-16 ***
X1E         -0.009589   0.003951  -2.427   0.0152 *

Residual standard error: 0.3207 on 49998 degrees of freedom
Multiple R-squared:  0.0001178, Adjusted R-squared:  9.781e-05
F-statistic: 5.891 on 1 and 49998 DF,  p-value: 0.01522
> anova(lm(Y/E~X1,weights=E))
Analysis of Variance Table

Response: Y/E
             Df Sum Sq Mean Sq F value  Pr(>F)
X1            1    0.6 0.60593   5.891 0.01522 *
Residuals 49998 5142.7 0.10286
```

Modèle binomial

Le premier modèle auquel on pourrait penser pour modéliser le nombre de sinistres est le modèle binomial $\mathcal{B}(n, p)$. Avec n connu, correspondant à l'exposition.

Pour être plus précis, on suppose que $Y_i \sim \mathcal{B}(E_i, p_i)$ où E_i est connu, et où p_i est fonction de variables explicatives (via un lien logistique).

On va exprimer l'exposition en semaine, p est alors la probabilité d'avoir un sinistre sur une semaine,

```
> freq_b=freq[freq$exposition<=1,]
> freq_b$sem=round(freq_b$exposition*52)
> freq_b=freq_b[freq_b$sem>=1,]
```

Pour faire une régression binomiale (et pas juste Bernoulli)

```
> reg1=glm(nb_DO/sem~1,family=binomial,weights=sem,data=freq_b)
```

ou encore

```
> reg2=glm(cbind(nb_DO, sem-nb_DO) ~ 1, data = freq_b, family =
    binomial)
```

La fréquence annuelle prédite est

```
> predict(reg1,type="response")[1]*52
         1
0.06574927
```

Si on utilise le carburant comme variable de segmentation

```
> reg2 <- glm(cbind(nb_DO, sem-nb_DO) ~ carburant, data = freq_b,
    family = binomial)
> predict(reg2,type="response",newdata=data.frame(carburant=c("D","E"
    )))*52
         1          2
0.07097405 0.06104802
```

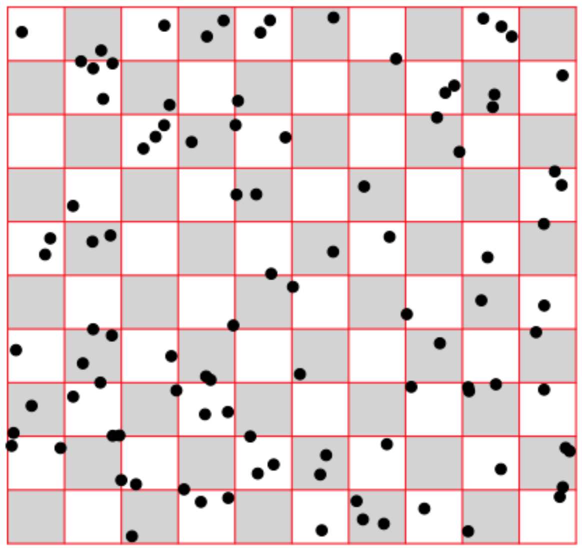
Remarque avec une loi binomiale $\mathbb{E}[N|\boldsymbol{X}] > \text{Var}[N|\boldsymbol{X}]$, sous-dispersion.

La loi de petits nombres

La loi de Poisson apparaît comme approximation de la loi binomiale quand $p \sim \lambda/n$

$$\binom{n}{k} p^k (1-p)^{n-k} \sim e^{-\lambda} \frac{\lambda^k}{k!}$$

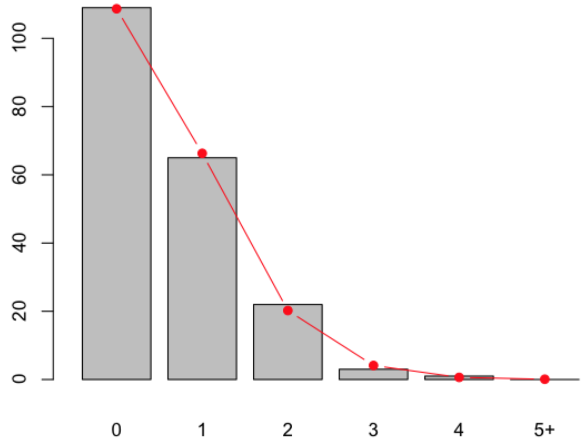
```
> data.frame(N,F=table(nb_cell),P=c(dpois
     (0:4,1),1-ppois(4,1)))
     N  F      P
1    0 36  36.78
2    1 39  36.78
3    2 16  18.39
4    3  7   6.13
5    4  2   1.53
6  5+   0   0.37
```



La loi de petits nombres

Nombre de soldats de cavaliers morts par ruade de cheval, entre 1875 et 1894, dans 10 corps (soit 200 corps annuels) Bortkiewicz (1898)

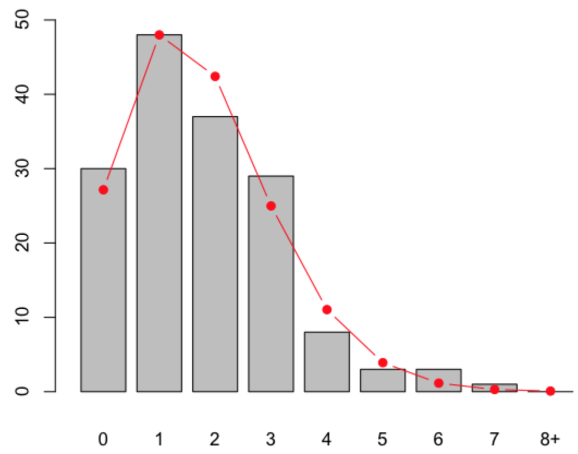
```
> data.frame(N,F=table(ruades),P=c(dpois
    (0:4,mean(ruades)),1-ppois(4,mean(ruades
    ))))
    N   F      P
1   0 109 108.67
2   1  65  66.21
3   2  22  20.22
4   3   3   4.11
5   4   1   0.63
6  5+   0   0.08
```



La loi de petits nombres

Nombre d'ouragans, par an, Lévi & Partrat (1989)

```
> data.frame(N,F=table(hurricanes),P=c(dpois
    (0:4,mean(hurricanes)),1-ppois(4,mean(
    hurricanes))))
     N   F      P
1    0  30  27.16
2    1  48  47.99
3    2  37  42.41
4    3  29  24.98
5    4   8  11.03
6    5   3   3.90
7    6   3   1.15
8    7   1   0.29
9   8+   0   0.08
```



La loi de Poisson et période de retour

Supposon qu'un évènement survienne avec une probabilité annuelle $p = 1/\tau$. Soit T le temps à attendre avant la première survenance,

$$\mathbb{P}[T > n] = \left(1 - \frac{1}{\tau}\right)^n \sim e^{-t/\tau}$$

et $\mathbb{E}[T] = \tau$ (notion de période de retour).

$t \setminus \tau$	10	20	50	100	200
10	34.86%	59.87%	81.70%	90.43%	95.11%
20	12.15%	35.84%	66.76%	81.79%	90.46%
50	0.51%	7.69%	36.41%	60.50%	77.83%
100	0.00%	0.59%	13.26%	36.60%	60.57%
200	0.00%	0.00%	1.75%	13.39%	36.69%

La loi de Poisson

La loi de Poisson est connue comme la loi des petits nombres,

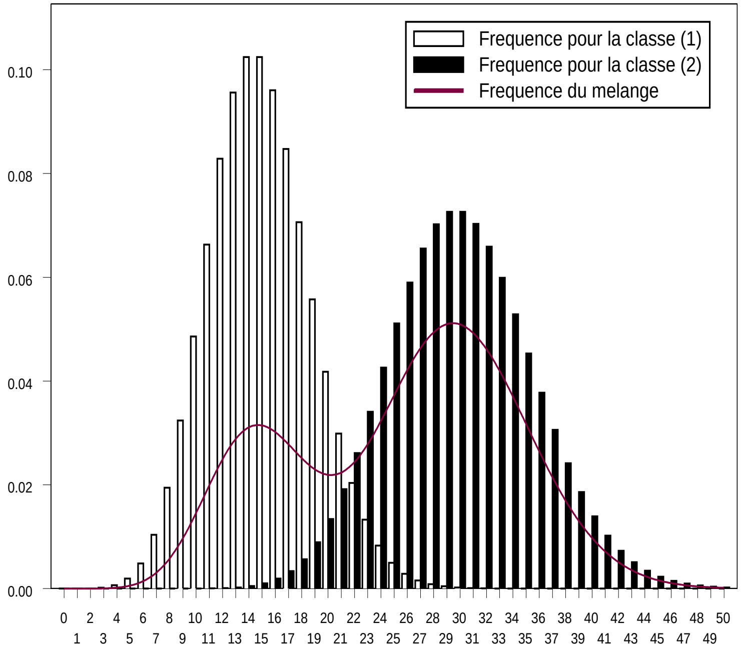
$$\mathbb{P}(N = k) = e^{-\lambda}\frac{\lambda^k}{k!}, \forall k \in \mathbb{N}.$$

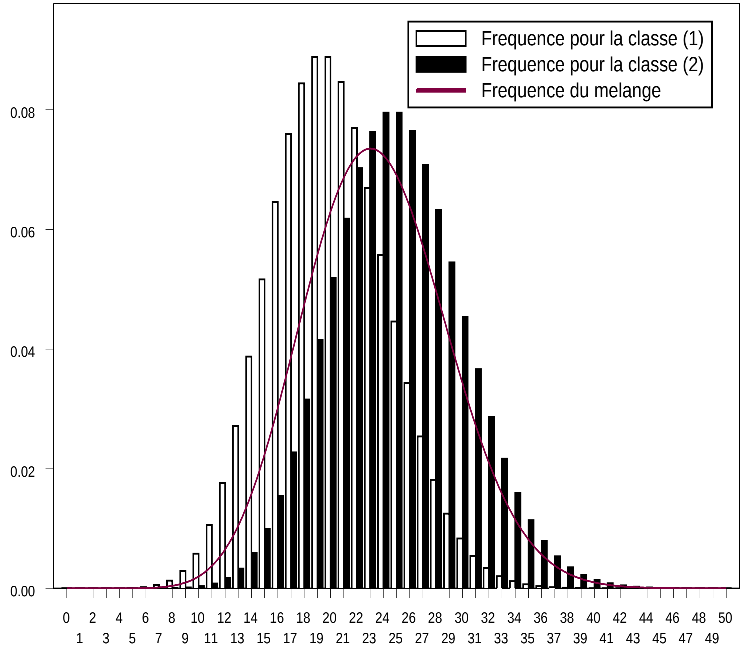
Pour rappel (cf premier cours), $\mathbb{E}(N) = \text{Var}(N) = \lambda \in \mathbb{R}_+$, équi-dispersion.

La loi Poisson mélange

En présence de sur-dispersion $\mathbb{E}(N) < \text{Var}(N)$, on peut penser à une loi Poisson mélange, i.e. il existe Θ, variable aléatoire positive, avec $\mathbb{E}(\Theta) = 1$, telle que

$$\mathbb{P}(N = k|\Theta = \theta) = e^{-\lambda\theta}\frac{[\lambda\theta]^k}{k!}, \forall k \in \mathbb{N}.$$





La loi Binomiale Négative

La loi binomiale négative apparît dans le modèle Poisson mélange, lorsque $\Theta \sim \mathcal{G}(\alpha, \alpha)$. Dans ce cas,

$$\pi(\theta) = x^{\alpha-1} \frac{\alpha^{\alpha} \exp(-\alpha x)}{\Gamma(\alpha)}$$

$$\mathbb{E}(\Theta) = 1 \text{ et } \text{Var}(\Theta) = \frac{1}{\alpha}.$$

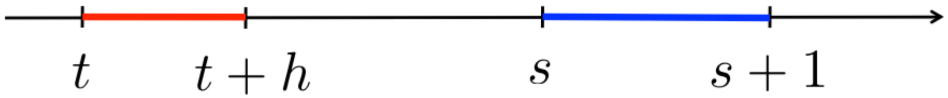
Dans ce cas, la loi (non conditionnelle) de N est

$$\mathbb{P}(N = k) = \int_0^{\infty} \mathbb{P}(N = k | \Theta = \theta) \pi(\theta) d\theta,$$

$$\mathbb{P}(N = k) = \frac{\Gamma(k + \alpha)}{\Gamma(k + 1)\Gamma(\alpha)} \left(\frac{1}{1 + \lambda/\alpha} \right)^{\alpha} \left(1 - \frac{1}{1 + \lambda/\alpha} \right)^{k}, \forall k \in \mathbb{N}$$

Le processus de Poisson

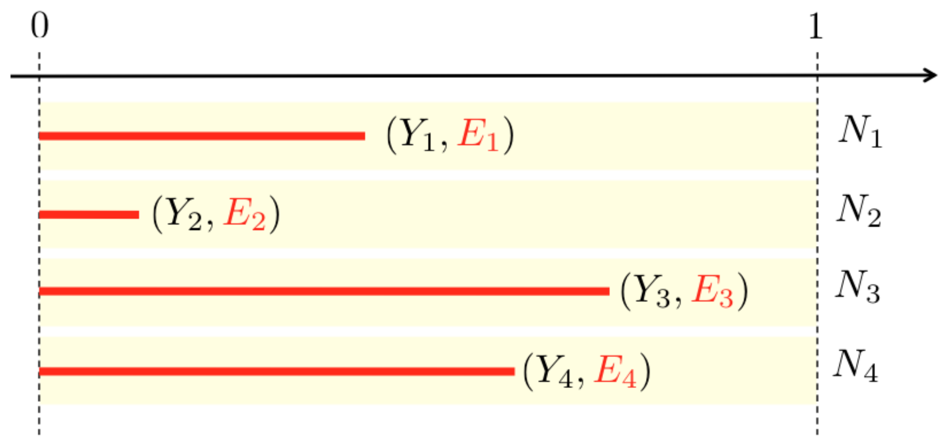
Pour rappel, $(N_t)_{t\geq 0}$ est un processus de Poisson homogène (de paramètre λ) s'il est à accroissements indépendants, et le nombre de sauts observés pendant la période $[t, t+h]$ suit une loi $\mathcal{P}(\lambda \cdot h)$.

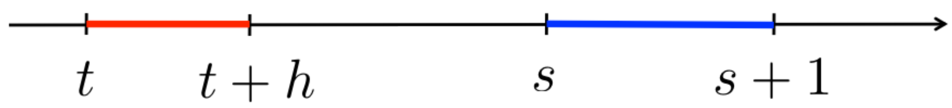


$N_{s+1} - N_s \sim \mathcal{P}(\lambda)$ est indépendant de $N_{t+h} - N_t \sim \mathcal{P}(\lambda \cdot h)$.

Exposition et durée d'observation

Soit N_i la frénquence annulisée de sinistre pour l'assuré i, et supposons $N_i \sim \mathcal{P}(\lambda)$. Si l'assuré i a été observé pendant une période E_i, le nombre de sinistre observé est $Y_i \sim \mathcal{P}(\lambda \cdot E_i)$.





Maximum de Vraisemblance

$$\mathcal{L}(\lambda,\boldsymbol{Y},\boldsymbol{E})=\prod_{i=1}^{n}\frac{e^{-\lambda E_i}[\lambda E_i]_i^Y}{Y_i!}$$

$$\log\mathcal{L}(\lambda,\boldsymbol{Y},\boldsymbol{E})=-\lambda\sum_{i=1}^{n}E_i+\sum_{i=1}^{n}Y_i\log[\lambda E_i]-\log\left(\prod_{i=1}^{n}Y_i!\right)$$

qui donne la condition du premier ordre

$$\frac{\partial}{\partial\lambda}\log\mathcal{L}(\lambda,\boldsymbol{Y},\boldsymbol{E})=-\sum_{i=1}^{n}E_i+\frac{1}{\lambda}\sum_{i=1}^{n}Y_i$$

qui s'annule pour

$$\widehat{\lambda}=\frac{\sum_{i=1}^{n}Y_i}{\sum_{i=1}^{n}E_i}=\sum_{i=1}^{n}\omega_i\frac{Y_i}{E_i}\text{ avec }\omega_i=\frac{E_i}{\sum_{i=1}^{n}E_i}$$

Maximum de Vraisemblance

```
> N = freq$nb_DO
> E= freq$exposition
> (lambda = sum(N)/sum(E))
[1] 0.06564229
> dpois(0:3,lambda)*100
[1] 93.646  6.147  0.201  0.004
```

Remarque: pour E_i on parlera d'exposition ou d'années police.

Fréquence annuelle et une variable tarifaire

```
> X1=freq$carburant
> tapply(N,X1,sum)
  D   E
998 926
> tapply(E,X1,sum)
       D         E
12519.55 13911.58
> (lambdas = tapply(N,X1,sum)/tapply(E,X1,sum))
         D          E
0.07971533 0.06656323
> cbind(dpois(0:3,lambdas[1]),dpois(0:3,lambdas[2]))*100
             [,1]         [,2]
[1,] 92.337916548 93.560375149
[2,]  7.360747674  6.227681197
[3,]  0.293382222  0.207267302
[4,]  0.007795687  0.004598794
```

Fréquence annuelle et tableau de contingence

Supposons que l'on prenne en compte ici deux classes de risques.

```
> X1=freq$carburant
> X2=cut(freq$agevehicule,c(0,3,10,101),right=FALSE)
> N_polices = table(X1,X2)
> E_agg=aggregate(E, by = list(X1 = X1, X2 = X2), sum)
> N_exposition=N_polices
> N_exposition[1:nrow(N_exposition),1:ncol(N_exposition)]=
+       matrix(E_agg$x,nrow(N_exposition),ncol(N_exposition))
> N_exposition
    X2
X1      [0,3)    [3,10)  [10,101)
  D 3078.938 5653.109 3787.503
  E 2735.014 5398.950 5777.619
>
> N_agg=aggregate(N, by = list(X1 = X1, X2 = X2), sum)
> N_sinistres=N_polices
> N_sinistres[1:nrow(N_sinistres),1:ncol(N_sinistres)]=
```

```
+       matrix(N_agg$x,nrow(N_sinistres),ncol(N_sinistres))
> N_sinistres
   X2
X1   [0,3) [3,10) [10,101)
  D    393    424       68
  E    343    419       88
> Freq_sinistres = N_sinistres/N_exposition
> Freq_sinistres
   X2
X1        [0,3)     [3,10)   [10,101)
  D 0.12764143 0.07500298 0.01795378
  E 0.12541067 0.07760768 0.01523119
```

Fréquence annuelle et tableau de contingence

Si on utilise la fonction en ligne sur blog

```
> freq_sin("nb_DO")$Freq
   X2
X1          [0,3)      [3,10)    [10,101)
  D 0.12764143 0.07500298 0.01795378
  E 0.12541067 0.07760768 0.01523119
> freq_sin("nb_RC")$Freq
   X2
X1          [0,3)      [3,10)    [10,101)
  D 0.07859854 0.08473214 0.07313526
  E 0.05959749 0.06816140 0.06836727
```

Tableau de contingence et biais minimial

Notons $Y_{i,j}$ le nombre de sinistres observés, $E_{i,j}$ l'exposition et $N_{i,j}$ la fréquence annualisée. La matrice $\boldsymbol{Y} = [Y_{i,j}]$ est ici la fréquence observée. On suppose qu'il est possible de modéliser Y à l'aide d'un modèle multiplicatif à deux facteurs, associés à chaque des des variables. On suppose que

$$\widehat{N}_{i,j} = L_i \cdot C_j, \text{ i.e. } \widehat{\boldsymbol{N}} = \boldsymbol{L}\boldsymbol{C}^\mathsf{T}$$

cf Bailey (1963) et Mildenhall (1999)

L'estimation de $\boldsymbol{L} = (L_i)$ et de $\boldsymbol{C} = (C_j)$ se fait généralement de trois manières: par moindres carrés, par minimisation d'une distance (e.g. du chi-deux) ou par un principe de balancement (ou méthode des marges).

Méthode des marges, Bailey (1963)

Dans la méthode des marges (selon la terminologie de Bailey (1963), formellement, on veut

$$\sum_j Y_{i,j} = \sum_j E_{i,j} N_{i,j} = \sum_j E_{i,j} L_i \cdot C_j,$$

en somment sur la ligne i, pour tout i, ou sur la colonne j,

$$\sum_i Y_{i,j} = \sum_i E_{i,j} N_{i,j} = \sum_i E_{i,j} L_i \cdot C_j,$$

La première équation donne

$$L_i = \frac{\sum_j Y_{i,j}}{\sum_j E_{i,j} C_j}$$

et la seconde

$$C_j = \frac{\sum_i Y_{i,j}}{\sum_i E_{i,j} L_i}.$$

Méthode des marges, Bailey (1963)

On résoud alors ce petit systeme de maniere itérative (car il n'y a pas de solution analytique simple).

```
> DO=freq_sin("nb_DO")
> m=sum(DO$Sin)/sum(DO$Exp)
> L<-matrix(NA,10,nrow(DO$Exp))
> C<-matrix(NA,10,ncol(DO$Exp))
> L[1,]<-rep(1,2);colnames(L)=rownames(DO$Sin)
> C[1,]<-rep(m,3);colnames(C)=colnames(DO$Sin)
>
> for(j in 2:10){
+    L[j,1]<-sum(DO$Sin[1,])/sum(DO$Exp[1,]*C[j-1,])
+    L[j,2]<-sum(DO$Sin[2,])/sum(DO$Exp[2,]*C[j-1,])
+    C[j,1]<-sum(DO$Sin[,1])/sum(DO$Exp[,1]*L[j,])
+    C[j,2]<-sum(DO$Sin[,2])/sum(DO$Exp[,2]*L[j,])
+    C[j,3]<-sum(DO$Sin[,3])/sum(DO$Exp[,3]*L[j,])
+    }
> L[10,]
```

```
        D        E
1.007697 1.002415
> C[10,]
      [0,3)      [3,10)    [10,101)
0.12593567 0.07588711 0.01623609
> PREDICTION=(L[10,])%*%t(C[10,])
> PREDICTION
           [0,3)      [3,10)    [10,101)
[1,] 0.1269050 0.07647120 0.01636106
[2,] 0.1262397 0.07607034 0.01627529
> sum(PREDICTION[1,]*DO$Exp[1,])
[1] 885
> sum(DO$Sin[1,])
[1] 885
```

Méthode des moindres carrés

Parmi les méthodes proches de celles évoquées auparavant sur la méthode des marges, il est aussi possible d'utiliser une méthode par moindres carrés (pondérée). On va chercher à minimiser la somme des carrés des erreurs, i.e.

$$D = \sum_{i,j} E_{i,j}(N_{ij} - L_i \cdot C_j)^2$$

La condition du premier ordre donne ici $\displaystyle\frac{\partial D}{\partial L_i} = -2\sum_j C_j E_{i,j}(N_{i.j} - L_i \cdot C_j) = 0$

soit

$$L_i = \frac{\sum_j C_j E_{i,j} N_{i.j}}{\sum_j E_{i,j} C_j^2} = \frac{\sum_j C_j Y_{i.j}}{\sum_j E_{i,j} C_j^2}$$

L'autre condition du premier ordre donne

$$C_j = \frac{\sum_i L_i E_{i,j} N_{i.j}}{\sum_i E_{i,j} L_i^2} = \frac{\sum_i L_i Y_{i.j}}{\sum_i E_{i,j} L_i^2}$$

On résoud alors ce petit systeme de maniere itérative (car il n'y a pas de solution analytique simple).

```
> DO=freq_sin("nb_DO")
> m=sum(DO$Sin)/sum(DO$Exp)
> L<-matrix(1,100,nrow(DO$Exp))
> C<-matrix(NA,100,ncol(DO$Exp))
> L[1,]<-rep(1,2);colnames(L)=rownames(DO$Sin)
> C[1,]<-rep(m,3);colnames(C)=colnames(DO$Sin)
>
>  for(j in 2:100){
+     L[j,1]=sum(DO$Sin[1,]*C[j-1,])/sum(DO$Exp[1,]*C[j-1,]^2)
+     L[j,2]=sum(DO$Sin[2,]*C[j-1,])/sum(DO$Exp[2,]*C[j-1,]^2)
+     C[j,1]=sum(DO$Sin[,1]*L[j,])/sum(DO$Exp[,1]*L[j,]^2)
+     C[j,2]=sum(DO$Sin[,2]*L[j,])/sum(DO$Exp[,2]*L[j,]^2)
+     C[j,3]=sum(DO$Sin[,3]*L[j,])/sum(DO$Exp[,3]*L[j,]^2)
+     }
>
> L[100,]
```

```
        D         E
1.011633 1.012599
> C[100,]
      [0,3)      [3,10)    [10,101)
0.12507961 0.07536373 0.01611180
> PREDICTION=(L[10,])%*%t(C[10,])
> PREDICTION
          [0,3)      [3,10)    [10,101)
[1,] 0.1265347 0.07624043 0.01629923
[2,] 0.1266554 0.07631321 0.01631479
> sum(PREDICTION[1,]*DO$Exp[1,])
[1] 882.3211
> sum(DO$Sin[1,])
[1] 885
```

Méthode du χ^2

Parmi les méthodes proches de celles évoquées dans la section ?? sur la méthode des marges, il est aussi possible d'utiliser une méthode basée sur la distance du chi-deux. On va chercher à minimiser

$$Q = \sum_{i,j} \frac{E_{i,j}(N_{i,j} - L_i \cdot C_j)^2}{L_i \cdot C_j}$$

Là encore on utilise les conditions du premier ordre, et on obtient

$$L_i = \left(\frac{\sum_j \left(\frac{E_{i,j} Y_{i,j}^2}{C_j} \right)}{\sum_j E_{i,j} C_j} \right)^{\frac{1}{2}}$$

et une expression du même genre pour C_j.

```
> DO=freq_sin("nb_DO")
> m=sum(DO$Sin)/sum(DO$Exp)
```

```
> L<-matrix(1,100,nrow(DO$Exp))
> C<-matrix(NA,100,ncol(DO$Exp))
> L[1,]<-rep(1,2);colnames(L)=rownames(DO$Sin)
> C[1,]<-rep(m,3);colnames(C)=colnames(DO$Sin)
>
> for(j in 2:100){
+  L[j,1]=sqrt(sum(DO$Exp[1,]*DO$Freq[1,]^2/C[j-1,])/sum(DO$Exp[1,]*C[j-1,]))
+  L[j,2]=sqrt(sum(DO$Exp[2,]*DO$Freq[2,]^2/C[j-1,])/sum(DO$Exp[2,]*C[j-1,]))
+  C[j,1]=sqrt(sum(DO$Exp[,1]*DO$Freq[,1]^2/L[j,])/sum(DO$Exp[,1]*L[j,]))
+  C[j,2]=sqrt(sum(DO$Exp[,2]*DO$Freq[,2]^2/L[j,])/sum(DO$Exp[,2]*L[j,]))
+  C[j,3]=sqrt(sum(DO$Exp[,3]*DO$Freq[,3]^2/L[j,])/sum(DO$Exp[,3]*L[j,]))
+  }
>
```

```
> L[100,]
       D        E
1.19012 1.18367
> C[100,]
     [0,3)     [3,10)    [10,101)
0.10664299 0.06427321 0.01379165
> PREDICTION=(L[10,])%*%t(C[10,])
> PREDICTION
         [0,3)     [3,10)    [10,101)
[1,] 0.1269180 0.07649285 0.01641373
[2,] 0.1262301 0.07607824 0.01632476
> sum(PREDICTION[1,]*DO$Exp[1,])
[1] 885.362
> sum(DO$Sin[1,])
[1] 885
```

(on est très proche ici de la méthode des marges, de Bailey).

Approche(s) économétrique(s) du biais minimal

Ici, on considère $\boldsymbol{y} = [N_{i,j}]$ et $\widehat{\boldsymbol{y}} = [L_i C_j] = [e^{\ell_i + c_j}]$.

Rappel Dans un modèle linéaire - i.e. $\mathbb{E}[Y] = \boldsymbol{X\beta}$ - avec homoscédasticité, les équations normales sont

$$\boldsymbol{X}^{\mathsf{T}}[\boldsymbol{y} - \boldsymbol{X\beta}] = \boldsymbol{0}$$

et dans un modèle avec hétéroscédasticité, si $\text{Var}[\boldsymbol{\varepsilon}] = \sigma^2\boldsymbol{\Omega}$,

$$\boldsymbol{X}^{\mathsf{T}}\boldsymbol{\Omega}^{-1}[\boldsymbol{y} - \boldsymbol{X\beta}] = \boldsymbol{0}$$

Dans un modèle multiplicatif - $\mathbb{E}[Y] = e^{\boldsymbol{X\beta}}$ avec homoscédasticité, les équations normales sont

$$\boldsymbol{X}^{\mathsf{T}}e^{\boldsymbol{X\beta}}[\boldsymbol{y} - e^{\boldsymbol{X\beta}}] = \boldsymbol{0}$$

et dans un modèle avec hétéroscédasticité, si $\text{Var}[\boldsymbol{\varepsilon}] = \sigma^2\boldsymbol{\Omega}$,

$$\boldsymbol{X}^{\mathsf{T}}\boldsymbol{\Omega}^{-1}e^{\boldsymbol{X\beta}}[\boldsymbol{y} - e^{\boldsymbol{X\beta}}] = \boldsymbol{0}$$

Approche(s) économétrique(s) du biais minimal

Résoudre $\boldsymbol{X}^{\mathsf{T}}[\boldsymbol{y}-e^{\boldsymbol{X}\boldsymbol{\beta}}]=\boldsymbol{0}$ revient à considérer un modèle multiplicatif, hétéroscedastique, avec $\boldsymbol{\Omega}\propto e^{\boldsymbol{X}\boldsymbol{\beta}}$, i.e. $\text{Var}(Y)\propto\mathbb{E}[Y]$ (cf loi de Poisson).

```
> df_agg=data.frame(N=as.numeric(DO$Sin),E=as.numeric(DO$Exp), X1=rep
    (levels(X1),ncol(DO$Sin)),X2=rep(levels(X2),each=nrow(DO$Sin)))
> regpoislog_agg <- glm(N~X1+X2,offset=log(E),data=df_agg, family=
    poisson(link="log"))
> ndf_agg=df_agg
> ndf_agg$E=1
> matrix(predict(regpoislog_agg,type="response",newdata=ndf_agg),nrow
    (PREDICTION),ncol(PREDICTION))
           [,1]        [,2]        [,3]
[1,] 0.1269050 0.07647120 0.01636106
[2,] 0.1262397 0.07607034 0.01627529
```

Approche(s) économétrique(s) du biais minimal

ou au niveau individuel

```
> df=data.frame(N=freq[,"nb_DO"],E,X1,X2)
> regpoislog <- glm(N~X1+X2,offset=log(E),data=df, family=poisson(
    link="log"))
> rownames(PREDICTION)=c("D","E")
> newd <- data.frame(X1=factor(rep(rownames(PREDICTION),ncol(
    PREDICTION))), E=rep(1,6), X1=factor(rep(rownames(PREDICTION),ncol
    (PREDICTION))), X2=factor(rep(colnames(PREDICTION),each=nrow(
    PREDICTION))))
> matrix(predict(regpoislog,newdata=newd,
+           type="response"),nrow(PREDICTION),ncol(PREDICTION))
           [,1]       [,2]       [,3]
[1,] 0.1269050 0.07647120 0.01636106
[2,] 0.1262397 0.07607034 0.01627529
```

Approche(s) économétrique(s) du biais minimal

On peut aussi envisager un modèle homosc´dastique

```
> df_agg=data.frame(N=as.numeric(DO$Sin),E=as.numeric(DO$Exp), X1=rep
    (levels(X1),ncol(DO$Sin)),X2=rep(levels(X2),each=nrow(DO$Sin)))
> reggauss_agg <- glm(N~X1+X2,offset=log(E),family=gaussian(link="log
    "),data=df_agg)
> ndf_agg=df_agg
> ndf_agg$E=1
> matrix(predict(reggauss_agg,type="response",newdata=ndf_agg),nrow(
    PREDICTION),ncol(PREDICTION))
          [,1]       [,2]       [,3]
[1,] 0.1261529 0.07592604 0.01594451
[2,] 0.1272804 0.07660463 0.01608701
```

La régression de Poisson

L'idée est la même que pour la régression logistique: on cherche un modèle linéaire pour la moyenne. En l'occurence,

$$Y_i \sim \mathcal{P}(\lambda_i) \text{ avec } \lambda_i = \exp[\boldsymbol{X}_i^\mathsf{T}\boldsymbol{\beta}].$$

Dans ce modèle, $\mathbb{E}(Y_i|\boldsymbol{X}_i) = \text{Var}(Y_i|\boldsymbol{X}_i) = \lambda_i = \exp[\boldsymbol{X}_i^\mathsf{T}\boldsymbol{\beta}]$.

Remarque: on posera parfois $\theta_i = \eta_i = \boldsymbol{X}_i^\mathsf{T}\boldsymbol{\beta}$

La log-vraisemblance est ici

$$\log \mathcal{L}(\boldsymbol{\beta}; \boldsymbol{Y}) = \sum_{i=1}^n [Y_i \log(\lambda_i) - \lambda_i - \log(Y_i!)]$$

ou encore

$$\log \mathcal{L}(\boldsymbol{\beta}; \boldsymbol{Y}) = \sum_{i=1}^n Y_i \cdot [\boldsymbol{X}_i^\mathsf{T}\boldsymbol{\beta}] - \exp[\boldsymbol{X}_i^\mathsf{T}\boldsymbol{\beta}] - \log(Y_i!)$$

Le gradient est ici

$$\nabla \log \mathcal{L}(\boldsymbol{\beta}; \boldsymbol{Y}) = \frac{\partial \log \mathcal{L}(\boldsymbol{\beta}; \boldsymbol{Y})}{\partial \boldsymbol{\beta}} = \sum_{i=1}^{n} (Y_i - \exp[\boldsymbol{X}_i^\mathsf{T} \boldsymbol{\beta}]) \boldsymbol{X}_i^\mathsf{T}$$

alors que la matrice Hessienne s'écrit

$$H(\boldsymbol{\beta}) = \frac{\partial^2 \log \mathcal{L}(\boldsymbol{\beta}; \boldsymbol{Y})}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}^\mathsf{T}} = - \sum_{i=1}^{n} (Y_i - \exp[\boldsymbol{X}_i^\mathsf{T} \boldsymbol{\beta}]) \boldsymbol{X}_i \boldsymbol{X}_i^\mathsf{T}$$

La recherche du maximum de $\log \mathcal{L}(\boldsymbol{\beta}; \boldsymbol{Y})$ est obtenu (numériquement) par l'algorithme de Newton-Raphson,

1. partir d'une valeur initiale $\boldsymbol{\beta}_0$
2. poser $\boldsymbol{\beta}_k = \boldsymbol{\beta}_{k-1} - H(\boldsymbol{\beta}_{k-1})^{-1} \nabla \log \mathcal{L}(\boldsymbol{\beta}_{k-1})$

où $\nabla \log \mathcal{L}(\boldsymbol{\beta})$ est le gradient, et $H(\boldsymbol{\beta})$ la matrice Hessienne (on parle parfois de Score de Fisher).

La régression de Poisson

Par exemple, si on régresse sur l'âge du véhicule

```
> Y <- freq$nb_DO
> X1 <- freq$agevehicule
> X <- cbind(rep(1,length(X1)),X1)
```

on part d'une valeur initiale (e.g. une estimation classique de modèle linéaire)

```
> beta=lm(Y~0+X)$coefficients
```

On fait ensuite une boucle (avec 50,000 lignes, l'algorithme du cours #2. ne fonctionne pas)

```
> for(s in 1:20){
+ gradient=t(X)%*%(Y-exp(X%*%beta))
+ hessienne=matrix(0,ncol(X),ncol(X))
+ for(i in 1:nrow(X)){
+ hessienne=hessienne + as.numeric(exp(X[i,]%*%beta))* (X[i,]%*%t(X[i
    ,]))}
```

```
+ beta=beta+solve(hessienne)%*%gradient
+ }
```

On peut montrer que $\widehat{\boldsymbol{\beta}} \xrightarrow{\mathbb{P}} \boldsymbol{\beta}$ et

$$\sqrt{n}(\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}) \xrightarrow{\mathcal{L}} \mathcal{N}(\mathbf{0}, I(\boldsymbol{\beta})^{-1}).$$

Numériquement, la encore, on peut approcher $I(\boldsymbol{\beta})^{-1}$ qui est la variance (asymptotique) de notre estimateur. Or $I(\boldsymbol{\beta}) = H(\boldsymbol{\beta})$, donc les écart-types de $\widehat{\boldsymbol{\beta}}$ donnés à droite

```
> cbind(beta,sqrt(diag(solve(hessienne))))
           [,1]        [,2]
   -2.6949729 0.03382691
X1 -0.1219873 0.00575687
```

On retrouve toutes ces valeurs en utilisant

```
> regPoisson=glm(nb_DO~X1,data=freq,family=poisson)
> summary(regPoisson)

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -2.694973   0.033827  -79.67   <2e-16 ***
X1          -0.121987   0.005757  -21.19   <2e-16 ***

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 11893  on 49999  degrees of freedom
Residual deviance: 11334  on 49998  degrees of freedom
AIC: 14694

Number of Fisher Scoring iterations: 6
```

Loi de Poisson vs. Conditions du Premier Ordre

D'un point de vue computationnel, l'ordinateur cherche à résoudre

$$\boldsymbol{X}^\mathsf{T}[\boldsymbol{y} - \exp(\boldsymbol{X}\boldsymbol{\beta})] = \boldsymbol{0},\ \text{i.e.}\ \sum_{i=1}^{n} \boldsymbol{X}_i^\mathsf{T}[\boldsymbol{y} - \exp(\boldsymbol{X}_i^\mathsf{T}\boldsymbol{\beta})] = \boldsymbol{0}$$

À aucun moment, on a besoin d'avoir $y_i \in \mathbb{N}$. En fait, on peut faire une 'régression de Poisson' pour des variables non-entières.

```
> reg=glm(nb_RC~1,data=freq,family=poisson)
> predict(reg,type="response")[1]
      1
0.03848
> reg=glm(nb_RC/10~1,data=freq,family=poisson)
There were 50 or more warnings (use warnings() to see the first 50)
> predict(reg,type="response")[1]
       1
0.003848
```

Propriété de la Régression de Poisson

Les conditions du premier ordre $\boldsymbol{X}^\mathsf{T}[\boldsymbol{y} - \exp(\boldsymbol{X}\boldsymbol{\beta})] = \boldsymbol{0}$ peuvent s'écrire $\boldsymbol{X}^\mathsf{T}\boldsymbol{y} = \boldsymbol{X}^\mathsf{T}\exp(\boldsymbol{X}\boldsymbol{\beta})$ ou encore $\boldsymbol{X}^\mathsf{T}\widehat{\boldsymbol{y}}$.

S'il y a une constante, la première colonne implique

$$\boldsymbol{1}^\mathsf{T}\boldsymbol{y} = \boldsymbol{1}^\mathsf{T}\widehat{\boldsymbol{y}} \text{ i.e. } \sum_{i=1}^n y_i = \sum_{i=1}^n \widehat{y}_i$$

Si $\boldsymbol{X}$ est une variable factorielle de modalités $a_1, \cdots, a_J$,

$$\boldsymbol{1}_{a_j}^\mathsf{T}\boldsymbol{y} = \boldsymbol{1}_{a_j}^\mathsf{T}\widehat{\boldsymbol{y}} \text{ i.e. } \sum_{i;x_i=a_j} y_i = \sum_{i;x_i=a_j} \widehat{y}_i$$

Prise en compte de l'exposition (offset)

Oups, on a oublié de prendre l'exposition dans notre modèle. On a ajusté un modèle de la forme

$$Y_i \sim \mathcal{P}(\lambda_i) \text{ avec } \lambda_i = \exp[\beta_0 + \beta_1 X_{1,i}]$$

mais on voudrait

$$Y_i \sim \mathcal{P}(\lambda_i \cdot E_i) \text{ avec } \lambda_i = \exp[\beta_0 + \beta_1 X_{1,i}]$$

ou encore

$$Y_i \sim \mathcal{P}(\tilde{\lambda}_i) \text{ avec } \tilde{\lambda}_i = E_i \cdot \exp[\beta_0 + \beta_1 X_{1,i}] = \exp[\beta_0 + \beta_1 X_{1,i} + \log(E_i)]$$

Aussi, l'exposition intervient comme une variable de la régression, mais en prenant le logarithme de l'exposition, et en forçant le paramètre à être unitaire, i.e.

$$Y_i \sim \mathcal{P}(\tilde{\lambda}_i) \text{ avec } \tilde{\lambda}_i = E_i \cdot \exp[\beta_0 + \beta_1 X_{1,i}] = \exp[\beta_0 + \beta_1 X_{1,i} + 1 \log(E_i)]$$

```
> Y <- freq$nb_DO
> X1 <- freq$agevehicule
> E <- freq$exposition
> X=cbind(rep(1,length(X1)),X1)
> beta=lm(Y~0+X)$coefficients
> for(s in 1:20){
+   gradient=t(X)%*%(Y-exp(X%*%beta+log(E)))
+   hessienne=matrix(0,ncol(X),ncol(X))
+   for(i in 1:nrow(X)){
+     hessienne=hessienne + as.numeric(exp(X[i,]%*%beta+log(E[i])))*(
    X[i,]%*%t(X[i,]))}
+   beta=beta+solve(hessienne)%*%gradient
+ }
>
> cbind(beta,sqrt(diag(solve(hessienne))))
          [,1]        [,2]
    -1.8256863 0.035138680
X1  -0.1578027 0.006198479
```

```
> regPoisson=glm(nb_DO~ageconducteur+offset(log(exposition)),data=
    freq,family=poisson)
> summary(regPoisson)

Coefficients:
                 Estimate Std. Error z value Pr(>|z|)
(Intercept)     -1.825686   0.035139  -51.96   <2e-16 ***
ageconducteur   -0.157803   0.006198  -25.46   <2e-16 ***

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 11671  on 49999  degrees of freedom
Residual deviance: 10822  on 49998  degrees of freedom
AIC: 14183

Number of Fisher Scoring iterations: 7
```

Régression de Poisson multiple

On peut (bien entendu) régresser sur plusieurs variables explicatives

```
> model_RC=glm(nb_RC~zone+as.factor(puissance)+agevehicule+
    ageconducteur+carburant+offset(log(exposition)),data=freq,family=
    poisson)
> summary(model_RC)
Coefficients:
                       Estimate Std. Error z value Pr(>|z|)
(Intercept)           -2.546970   0.122723 -20.754  < 2e-16 ***
zoneB                  0.011487   0.097559   0.118   0.9063
zoneC                  0.196208   0.077090   2.545   0.0109 *
zoneD                  0.403382   0.078788   5.120 3.06e-07 ***
zoneE                  0.594872   0.079207   7.510 5.90e-14 ***
zoneF                  0.684673   0.143612   4.768 1.87e-06 ***
as.factor(puissance)5  0.135072   0.081393   1.659   0.0970 .
as.factor(puissance)6  0.161305   0.079692   2.024   0.0430 *
as.factor(puissance)7  0.164168   0.079039   2.077   0.0378 *
as.factor(puissance)8  0.122254   0.110876   1.103   0.2702
```

```
as.factor(puissance)9    0.181978   0.123996   1.468   0.1422
as.factor(puissance)10   0.254358   0.119777   2.124   0.0337 *
as.factor(puissance)11   0.001156   0.170163   0.007   0.9946
as.factor(puissance)12   0.243677   0.223207   1.092   0.2750
as.factor(puissance)13   0.513950   0.284159   1.809   0.0705 .
as.factor(puissance)14   0.582564   0.295482   1.972   0.0487 *
as.factor(puissance)15   0.173748   0.383322   0.453   0.6504
agevehicule              0.001467   0.004191   0.350   0.7264
ageconducteur           -0.008844   0.001658  -5.335 9.58e-08 ***
carburantE              -0.201780   0.049265  -4.096 4.21e-05 ***

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 12680  on 49999  degrees of freedom
Residual deviance: 12524  on 49980  degrees of freedom
AIC: 16235

Number of Fisher Scoring iterations: 6
```

Effets marginaux, et élasticité

Les effets marginaux de la variable k pour l'individu i sont donnés par

$$\frac{\partial \mathbb{E}(Y_i|\boldsymbol{X}_i)}{\partial X_{i,k}} = \frac{\partial \exp[\boldsymbol{X}_i^\top \boldsymbol{\beta}]}{\partial X_k} = \exp[\boldsymbol{X}_i^\top \boldsymbol{\beta}] \cdot \beta_k$$

estimés pas $\exp[\boldsymbol{X}_i^\top \widehat{\boldsymbol{\beta}}] \cdot \widehat{\beta}_k$

Par exemple, pour avoir l'effet marginal de la variable `ageconducteur`,

```
> coef(model_RC)[19]
ageconducteur
 -0.008844096
> effet19=predict(model_DO,type="response")*coef(model_RC)[19]
> effet19[1:4]
            1             2             3             4
-1.188996e-03 -1.062340e-03 -7.076257e-05 -2.367111e-04
```

Effets marginaux, et élasticité

On peut aussi calculer les effets marginaux moyens de la variable k, $\overline{Y} \cdot \widehat{\beta}_k$

```
> mean(predict(model_RC,type="response"))*coef(model_RC)[19]
ageconducteur
-0.0003403208
```

Autrement dit, en vieillissant d'un an, il y aura (en moyenne) 0.0003 accident de moins, par an, par assuré.

Ici, on utilise des changements en unité ($\partial X_{i,k}$), mais il est possible détudier l'impact de changement en proportion. Au lieu de varier d'une utilité, on va considérer un changement de 1%.

Interprétation, suite

Dans la sortie, nous avons obtenu

```
Coefficients:
                        Estimate Std. Error z value Pr(>|z|)
```

...

```
carburantE             -0.201780   0.049265  -4.096 4.21e-05 ***
```

i.e.

```
> coefficients(model_RC)["carburantE"]
carburantE
-0.2017798
```

Autrement dit, à caractéristiques identiques, un assuré conduisant un véhicule essence a une fréquence de sinistres presque 20% plus faible qu'un assuré conduisant un véhicule diesel,

```
> exp(coefficients(model_RC)["carburantE"])
```

```
carburantE
 0.8172749
```

Dans la base, les zones prennent les valeurs A B C D E ou F, selon la densité en nombre d'habitants par km^2 de la commune de résidence (A ="1-50", B="50-100", C="100-500", D="500-2,000", E="2,000-10,000", F="10,000+").

Si on regarde maintenant la zone, la zone A est la zone de référence. On notera que la zone B n'est pas significativement différente de la zone A.

```
zoneB                   0.011487   0.097559   0.118    0.9063
zoneC                   0.196208   0.077090   2.545    0.0109 *
zoneD                   0.403382   0.078788   5.120  3.06e-07 ***
zoneE                   0.594872   0.079207   7.510  5.90e-14 ***
zoneF                   0.684673   0.143612   4.768  1.87e-06 ***
```

Notons que l'on pourrait choisir une autre zone de référence

```
> freq$zone=relevel(freq$zone,"C")
> model_RC=glm(nb_RC~zone+as.factor(puissance)+agevehicule+
```

```
+                    ageconducteur+carburant+offset(log(exposition)),
    data=freq,family=poisson)
> summary(model_RC)
```

...

```
zoneA                  -0.196208   0.077090  -2.545 0.010921 *
zoneB                  -0.184722   0.086739  -2.130 0.033202 *
zoneD                   0.207174   0.064415   3.216 0.001299 **
zoneE                   0.398664   0.064569   6.174 6.65e-10 ***
zoneF                   0.488465   0.135765   3.598 0.000321 ***
```

Si on refait la régression, on trouve que tous les zones sont disinctes de la zone C.

Pareil pour la zone D. En revanche, si la modalité de référence devient la zone E, on note que la zone F ne se distingue pas,

```
> freq$zone=relevel(freq$zone,"E")
> model_RC=glm(nb_RC~zone+as.factor(puissance)+agevehicule+
+          ageconducteur+carburant+offset(log(exposition)),data=freq,
    family=poisson)
```

```
> summary(model_RC)
```

...

```
zoneC                   -0.398664   0.064569  -6.174 6.65e-10 ***
zoneA                   -0.594872   0.079207  -7.510 5.90e-14 ***
zoneB                   -0.583385   0.088478  -6.594 4.29e-11 ***
zoneD                   -0.191490   0.066185  -2.893  0.00381 **
zoneF                    0.089801   0.135986   0.660  0.50902
```

On pourrait être tenté de regroupe A et B, et E et F. En effet, les tests de Student suggèrent des regroupement (pour chacune des paires). Pour faire un regroupement des deux paires, on fait un test de Ficher,

```
> freq$zone=relevel(freq$zone,"C")
> model_RC=glm(nb_RC~zone+as.factor(puissance)+agevehicule+
+                 ageconducteur+carburant+offset(log(exposition)),data
    =freq,family=poisson)
> library(car)
> linearHypothesis(model_RC,c("zoneA=zoneB","zoneE=zoneF"))
```

```
Linear hypothesis test

Hypothesis:
zoneA - zoneB = 0
zoneE - zoneF = 0

Model 1: restricted model
Model 2: nb_RC ~ zone + as.factor(puissance) + agevehicule +
    ageconducteur +
    carburant + offset(log(exposition))

  Res.Df Df  Chisq Pr(>Chisq)
1  49982
2  49980  2 0.4498     0.7986
```

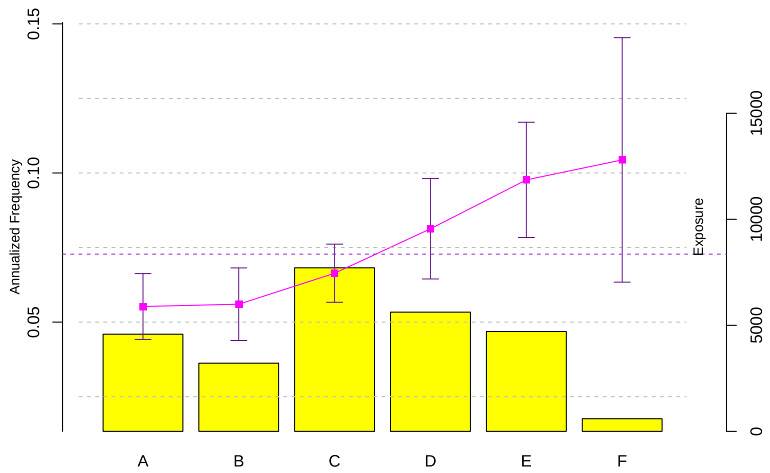
On peut accepter (avec une telle p-value) un regroupement. Construisons cette nouvelle variable

```
> levels(freq$zone)=c("AB","AB","C","D","EF","EF")
```

Régression de Poisson sur de variable factorielle, visualisation

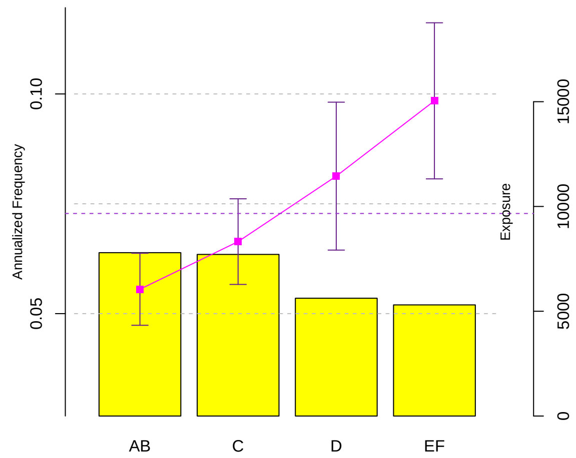
Avec la fonction `graph_freq` de blog

```
> graph_freq("zone",continuous=FALSE)
```



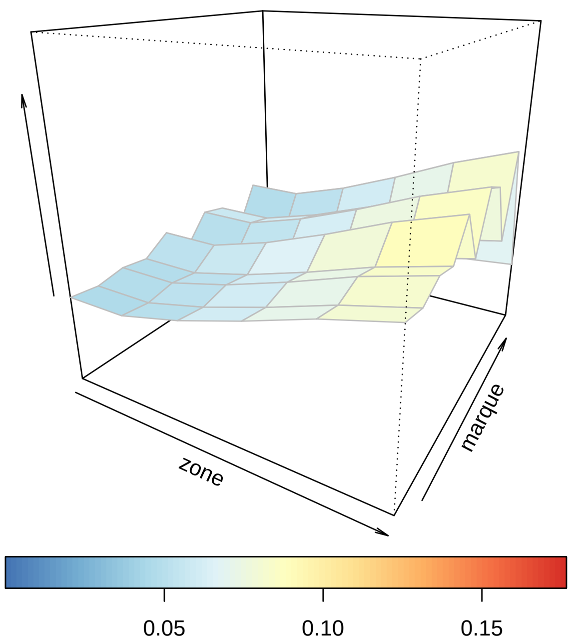
Régression de Poisson sur de variable factorielle, visualisation

```
> levels(freq$zone)=c("AB","AB","C","D","EF"
    ,"EF")
> graph_freq("zone",continuous=FALSE)
```



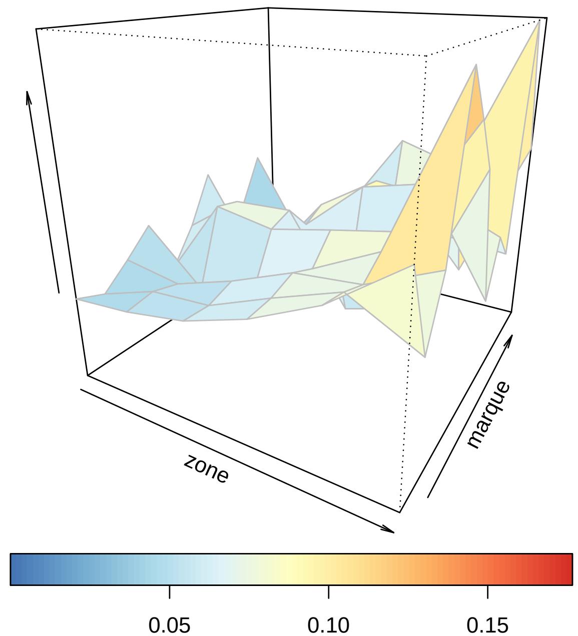
Effets Croisés

```r
lst_p=levels(freq$zone)
lst_m=levels(freq$marque)
REG1=glm(nb_RC~zone+marque+offset(
    exposition), data=freq,family=
    poisson)
nd=data.frame(
  zone=rep(lst_p,length(lst_m)),
  marque=rep(lst_m,each=length(lst_p)),
  exposition=1)
y1=predict(REG1,newdata=nd,type="
    response")
my1=matrix(y1,length(lst_p),length(lst_
    m))
persp(my1)
```



Effets Croisés

```
REG2=glm(nb_RC~zone+marque+zone*marque+offset(exposition), data=freq, family=poisson)
y2=predict(REG1,newdata=nd,type="response")
my2=matrix(y1,length(lst_p),length(lst_m))
persp(my2)
```

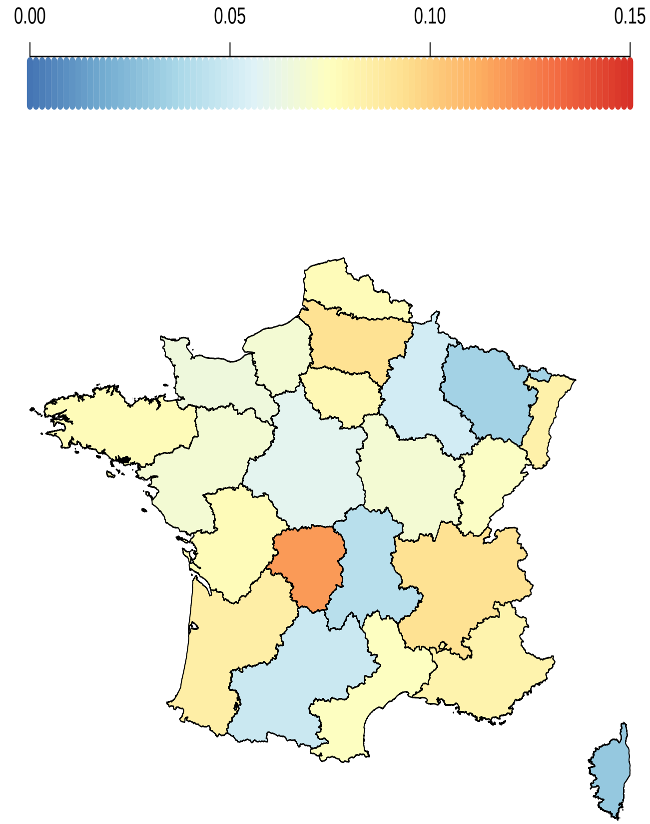


Variable Explicative Spatiale

Les régions utilisées sur la base sont reliées à la classification INSEE, data.gouv.fr

```
> loc="http://osm13.openstreetmap.fr/~cquest/openfla/export/regions
    -20140306-5m-shp.zip"
> download.file(loc,"region.zip")
> unzip("region.zip",exdir="regions")
> require(maptools)
> regions=readShapePoly("./regions/regions-20140306-5m.shp")
> LISTE=regions@data$nom[
  c(1,2,3,4,5,6,7,8,9,10,13,14,15,17,18,21,22,23,24,25,26,27)]
> corresp_insee=
c(42,72,83,25,26,53,24,21,94,43,23,11,91,74,41,73,31,52,22,54,93,82)
```

Variable Explicative Spatiale

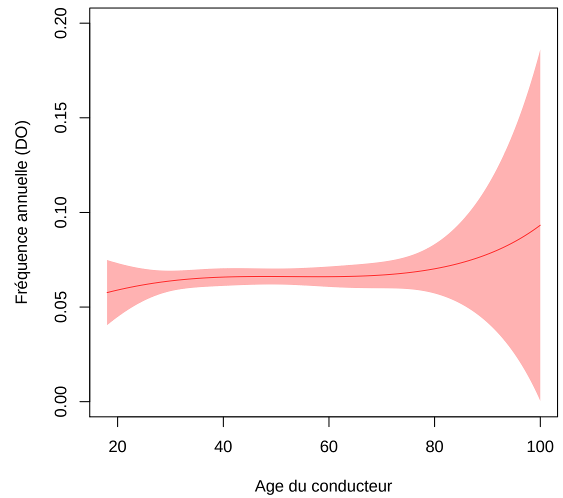



```
> N = freq$nb_RC
> E= freq$exposition
> X1=freq$region
> T=tapply(N,X1,sum)/tapply(E,X1,sum)
> T=T[as.character(corresp_insee)]
> library(RColorBrewer)
> CLpalette=colorRampPalette(rev(brewer.pal(n = 9, name = "RdYlBu")))(100)
> lst=which(regions@data$nom%in%LISTE)
> plot(regions[lst,],col=CLpalette[round(T/.15*100)])
```

Lissage et modèles non-paramétriques

Dans le modèle où seul l'âge du conducteur intervient, on a pour DO

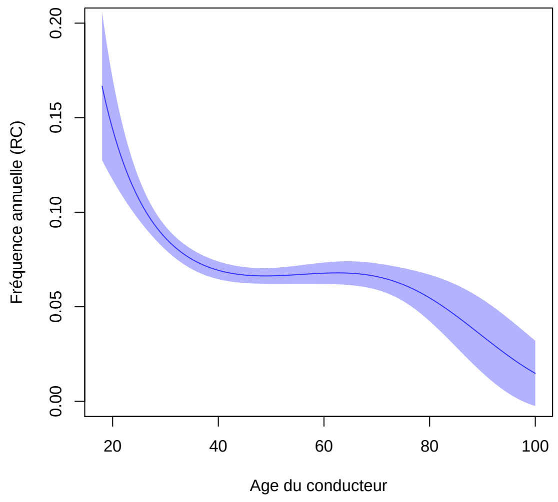
```
> library(splines)
> model_DO=glm(nb_DO~bs(ageconducteur)+offset(log(exposition)), data=freq, family=poisson)
> u=seq(18,100,by=.1)
> newd=data.frame(ageconducteur=u,exposition=1)
> y_DO=predict(model_DO,newdata=newd,type="response",se.fit =TRUE)
> plot(u,y_DO$fit,col="red")
> polygon(c(u,rev(u)),c(y_DO$fit+2*y_DO$se.fit,rev(y_DO$fit-2*y_DO$se.fit)),
+ col=rgb(1,0,0,.3),border=NA)
```



Lissage et modèles non-paramétriques

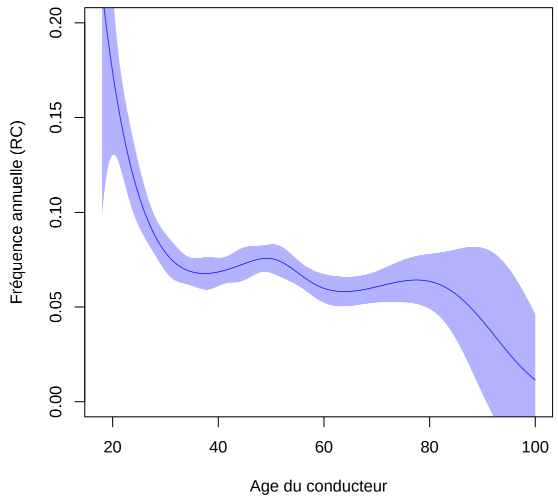
et pour RC

```
> library(splines)
> model_RC=glm(nb_RC~bs(ageconducteur)+
    offset(log(exposition)),
+ data=freq,family=poisson)
> u=seq(18,100,by=.1)
> newd=data.frame(ageconducteur=u,exposition
    =1)
> y_RC=predict(model_RC,newdata=newd,type="
    response",se.fit =TRUE)
> plot(u,y_DO$fit,col="blue")
> polygon(c(u,rev(u)),c(y_RC$fit+2*y_RC$se.
    fit,rev(y_RC$fit-2*y_RC$se.fit)),
+ col=rgb(0,0,1,.3),border=NA)
```



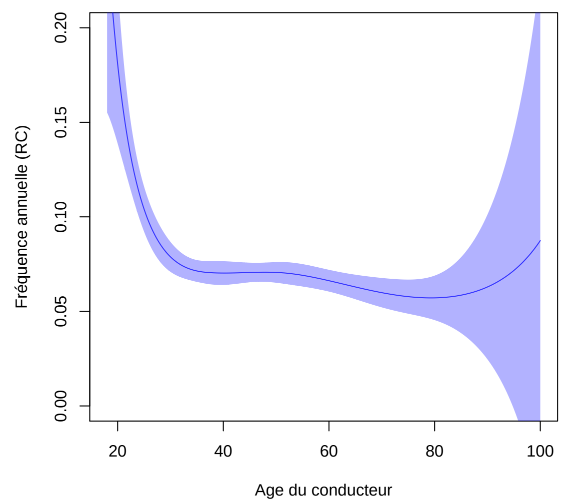
Lissage et modèles non-paramétriques

```
> library(splines)
> model_RC=glm(nb_RC~bs(ageconducteur,df=8)+
    offset(log(exposition)),
+ data=freq,family=poisson)
> u=seq(18,100,by=.1)
> newd=data.frame(ageconducteur=u,exposition
    =1)
> y_RC=predict(model_RC,newdata=newd,type="
    response",se.fit =TRUE)
> plot(u,y_DO$fit,col="blue")
> polygon(c(u,rev(u)),c(y_RC$fit+2*y_RC$se.
    fit,rev(y_RC$fit-2*y_RC$se.fit)),
+ col=rgb(0,0,1,.3),border=NA)
```



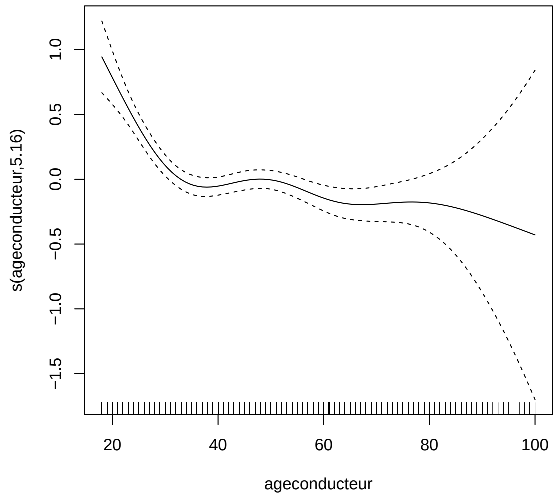
Lissage et modèles non-paramétriques

```
> library(splines)
> model_RC=glm(nb_RC~bs(ageconducteur,df=5)+
    offset(log(exposition)),
+ data=freq,family=poisson)
> u=seq(18,100,by=.1)
> newd=data.frame(ageconducteur=u,exposition
    =1)
> y_RC=predict(model_RC,newdata=newd,type="
    response",se.fit =TRUE)
> plot(u,y_DO$fit,col="blue")
> polygon(c(u,rev(u)),c(y_RC$fit+2*y_RC$se.
    fit,rev(y_RC$fit-2*y_RC$se.fit)),
+ col=rgb(0,0,1,.3),border=NA)
```



Lissage et modèles non-paramétriques

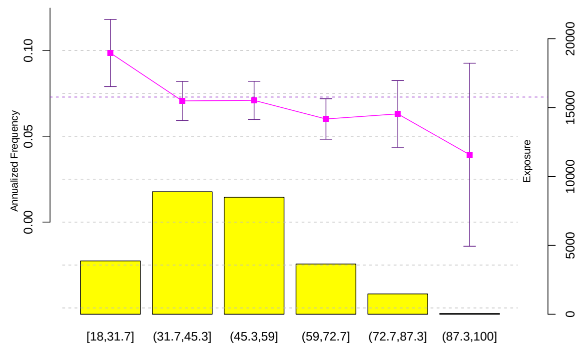
```
> library(mgcv)
> gam_RC=gam(nb_RC~s(ageconducteur)+offset(
    log(exposition)),data=freq,family=
    poisson)
> plot(gam_RC)
```



Transformer une variable continue en classes tarifaires

On peut envisager un découpage exogène, e.g. par intervalles de taille gale

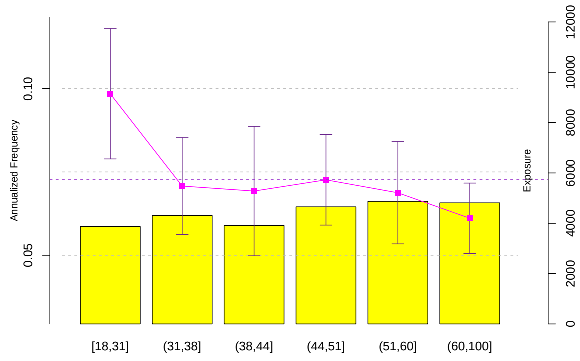
```
> library(classInt)
> CI=classIntervals(freq$ageconducteur, 6,
   style = "equal",intervalClosure="left")
> LV=CI$brks
> LV[6]=LV[6]+1
> graph_freq("ageconducteur",levels=LV)
```



Transformer une variable continue en classes tarifaires

ou par intervalles basés sur les quantiles

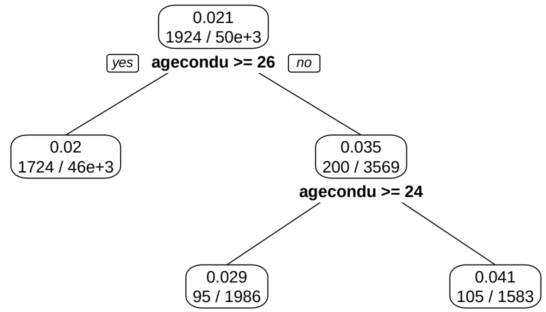
```
> library(classInt)
> CI=classIntervals(freq$ageconducteur, 6, style = "quantile",intervalClosure="left")
> LV=CI$brks
> LV[6]=LV[6]+1
> graph_freq("ageconducteur",levels=LV)
```



Transformer une variable continue en classes tarifaires

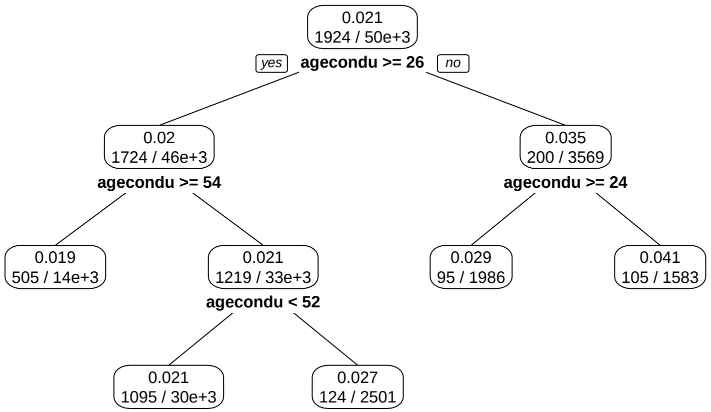
On peut aussi utiliser un découpage endogène, cf. arbre de régression

```
> library(rpart)
> arbre=rpart(nb_RC~ageconducteur,
    data=freq,method="poisson",cp=5e
    -4)
> library(rpart.plot)
> prp(arbre,type=2,extra=1)
```



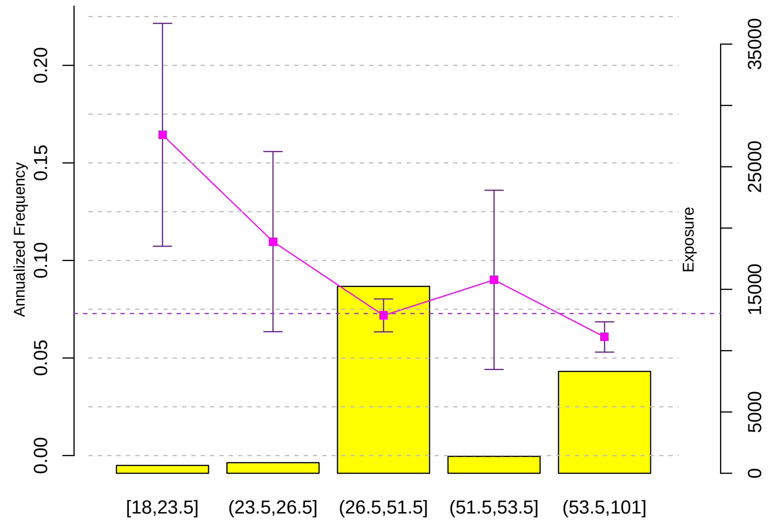
Transformer une variable continue en classes tarifaires

```
> arbre=rpart(nb_RC~ageconducteur,
    data=freq,method="poisson",cp=4e
    -4)
> prp(arbre,type=2,extra=1)
```



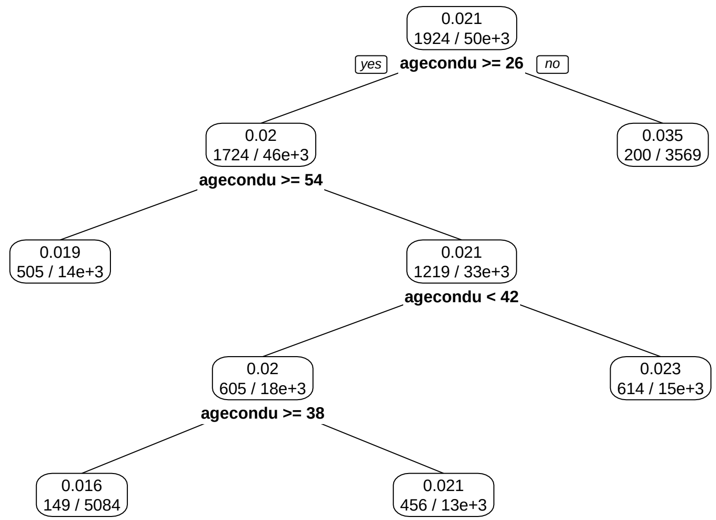
Transformer une variable continue en classes tarifaires

```
> lb=labels(arbre)
> cut_ages = substr(lb,nchar(lb)-3,
    nchar(lb))
> cut_ages=as.numeric(cut_ages)
> LV=c(18,sort(unique(cut_ages[!is.na
    (cut_ages)])),101)
> graph_freq("ageconducteur",levels=
    LV)
```



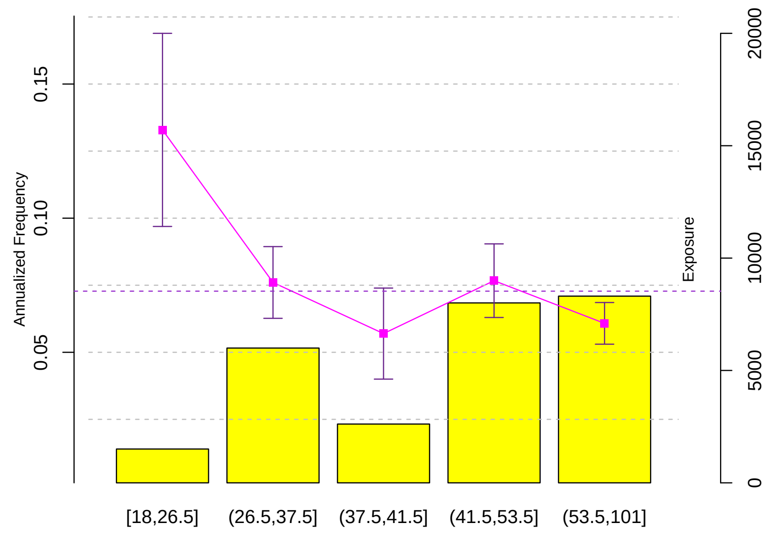
Transformer une variable continue en classes tarifaires

```
> arbre=rpart(nb_RC~ageconducteur,
    data=freq,method="poisson",cp=4e
    -4,minsplit=10000)
> prp(arbre,type=2,extra=1)
```



Transformer une variable continue en classes tarifaires

```
> lb=labels(arbre)
> cut_ages = substr(lb,nchar(lb)-3,
    nchar(lb))
> cut_ages=as.numeric(cut_ages)
> LV=c(18,sort(unique(cut_ages[!is.na
    (cut_ages)])),101)
> graph_freq("ageconducteur",levels=
    LV)
```



Arbre pour une loi de Poisson ?

On utilise ici une fonction d'impureté $\mathcal{I}(\cdot)$ basé sur la déviance* de la loi de Poisson. Pour un noeud N,

$$\mathcal{I}(N)=\sum_{i\in\{N\}}\left(Y_i\log\left(\frac{Y_i}{\widehat{\lambda}_N E_i}\right)-[Y_i-\widehat{\lambda}_N E_i]\right)\text{ avec }\widehat{\lambda}_N=\frac{\sum_{i\in\{N\}}Y_i}{\sum_{i\in\{N\}}E_i}$$

La méthode est ensuite la même que pour un arbre de classification.

* la log-vraisemblance pour une loi de Poisson est

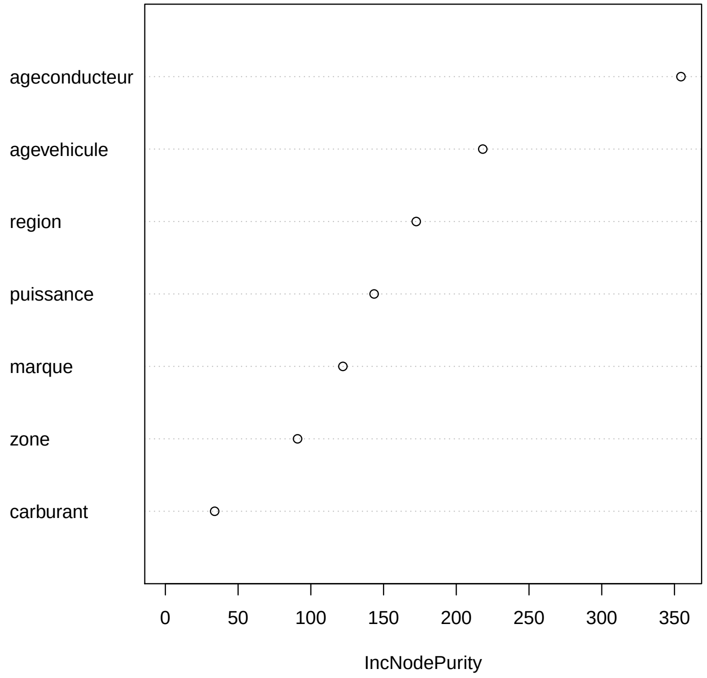
$$\log\mathcal{L}(\boldsymbol{\lambda})=\sum_{i=1}^n y_i\log\lambda_i-\lambda_i-\log(y_i!)$$

et la déviance est alors la différence

$$\log\mathcal{L}(\boldsymbol{y})-\log\mathcal{L}(\boldsymbol{\lambda})=\sum_{i=1}^n y_i\log\frac{y_i}{\lambda_i}-[y_i-\lambda_i]$$

Des arbres aux forêts

```
> library(randomForest)
> RF=randomForest(nb_RC~ageconducteur
    +agevehicule+region+puissance+
    marque+zone+carburant+offset(
    exposition),data=freq)
> varImpPlot(RF)
```



Procédure stepwise, AIC

```
> step(glm(nb_RC~ageconducteur+agevehicule+region+puissance+marque+
    zone+carburant+offset(exposition), data=sub_freq,family=poisson))
Start:  AIC=16162.21
nb_RC ~ ageconducteur + agevehicule + as.factor(region) + puissance +
    marque + zone + carburant + offset(exposition)

                    Df Deviance   AIC
- agevehicule        1    12409 16160
<none>                    12409 16162
- puissance          1    12412 16163
- ageconducteur      1    12418 16169
- marque            10    12438 16171
- carburant          1    12423 16174
- region            21    12464 16175
- zone               5    12469 16212
```

Procédure stepwise, AIC

ou avec des splines sur les variables continues,

```
                    Df  Deviance    AIC
- bs(puissance)      3     12379  16138
<none>                     12374  16139
- marque            10     12396  16141
- bs(agevehicule)    3     12383  16142
- carburant          1     12387  16150
- region            21     12428  16151
- bs(ageconducteur)  3     12405  16164
- zone               5     12432  16187
```