

## Actuariat de l'Assurance Non-Vie # 3

A. Charpentier (Université de Rennes 1)

ENSAE 2017/2018



credit: Arnold Odermatt

## 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  $\mathbb{E}[S_1 | \mathbf{X}] = \mathbb{E}[N_1 | \mathbf{X}] \cdot \mathbb{E}[Y | \mathbf{X}]$ .

## 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)

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

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

## Base pour les données de comptage

```

1 > sinistre_RC=sinistre[(sinistre$garantie=="1RC")&(sinistre$cout>0),]
2 > T_RC=table(sinistre_RC$nocontrat)
3 > T1_RC=as.numeric(names(T_RC))
4 > T2_RC=as.numeric(T_RC)
5 > nombre_1_RC = data.frame(nocontrat=T1_RC,nb_RC=T2_RC)
6 > I_RC = contrat$nocontrat%in%T1_RC
7 > T1_RC= contrat$nocontrat[I_RC==FALSE]
8 > nombre_2_RC = data.frame(nocontrat=T1_RC,nb_RC=0)
9 > 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

```

1 > sinistre_D0=sinistre[(sinistre$garantie=="2D0")&(sinistre$cout>0),]
2 > T_D0=table(sinistre_D0$nocontrat)
3 > T1_D0=as.numeric(names(T_D0))
4 > T2_D0=as.numeric(T_D0)
5 > nombre_1_D0 = data.frame(nocontrat=T1_D0,nb_D0=T2_D0)
6 > I_D0 = contrat$nocontrat%in%T1_D0
7 > T1_D0= contrat$nocontrat[I_D0==FALSE]
8 > nombre_2_D0 = data.frame(nocontrat=T1_D0,nb_D0=0)
9 > nombre_D0=rbind(nombre_1_D0,nombre_2_D0)

```

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

Et on crée la base finale

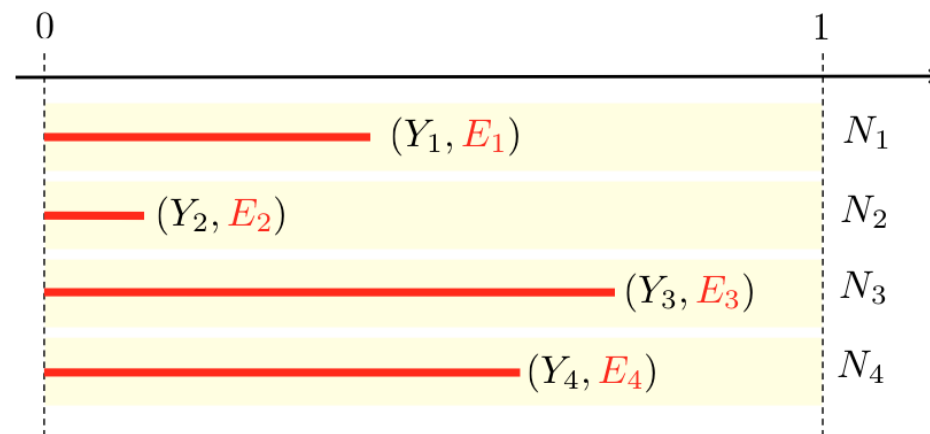
```

1 > freq = merge(contrat,nombre_RC)
2 > freq = merge(freq,nombre_D0)

```

## 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'intérêt est  $N$ .

## La notion d'exposition

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

```
1 > Y = freq$nb_DO
2 > E= freq$exposition
3 > sum(Y)/sum(E)
4 [1] 0.06564229
5 > weighted.mean(Y/E,E)
6 [1] 0.06564229
```

## Fréquence de sinistre et segmentation

```

1 > X1 = freq$carburant
2 > tapply(Y,X1,sum)/tapply(E,X1,sum)
3           D           E
4 0.07068945 0.06110016
5 > library(weights)
6 > wtd.t.test(x=(Y/E)[X1=="D"], y=(Y/E)[X1=="E"],
7 +           weight=E[X1=="D"], weighty=E[X1=="E"],samedata=FALSE)
8 $test
9 [1] "Two Sample Weighted T-Test (Welch)"
10
11 $coefficients
12           t.value           df           p.value
13 1.768349e+00 2.631555e+04 7.701412e-02
14
15 $additional
16   Difference      Mean.x      Mean.y      Std. Err
17 0.009589286 0.070689448 0.061100161 0.005422733

```



## Fréquence de sinistre et segmentation

On peut aussi envisager une **analyse de la variance**

```

1 > summary(lm(Y/E~X1,weights=E))
2 Coefficients:
3             Estimate Std. Error t value Pr(>|t|)
4 (Intercept)  0.070689   0.002866  24.662   <2e-16 ***
5 X1E          -0.009589   0.003951  -2.427   0.0152  *
6
7 Residual standard error: 0.3207 on 49998 degrees of freedom
8 Multiple R-squared:  0.0001178, Adjusted R-squared:  9.781e-05
9 F-statistic: 5.891 on 1 and 49998 DF,  p-value: 0.01522
10 > anova(lm(Y/E~X1,weights=E))
11 Analysis of Variance Table
12
13 Response: Y/E
14             Df Sum Sq Mean Sq F value   Pr(>F)
15 X1             1     0.6  0.60593    5.891 0.01522  *
16 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,

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

Pour faire une régression binomiale (et pas juste Bernoulli)

```
1 > reg1=glm(nb_D0/sem~1,family=binomial,weights=sem,data=freq_b)
```

ou encore

```
1 > reg2=glm(cbind(nb_D0, sem-nb_D0) ~ 1, data = freq_b, family =
  binomial)
```

La fréquence annuelle prédite est

```
1 > predict(reg1,type="response")[1]*52
2      1
3 0.06574927
```

Si on utilise le carburant comme variable de segmentation

```
1 > reg2 <- glm(cbind(nb_D0, sem-nb_D0) ~ carburant, data = freq_b,
  family = binomial)
2 > predict(reg2,type="response",newdata=data.frame(carburant=c("D","E"
  ))) * 52
3      1      2
4 0.07097405 0.06104802
```

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

## La loi de petits nombres

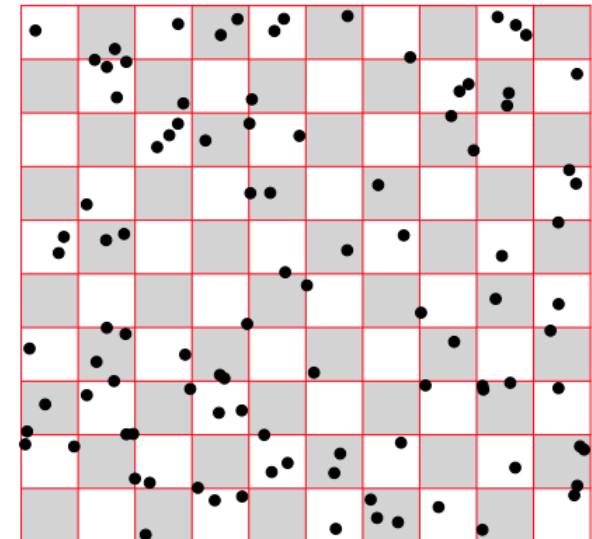
La loi de Poisson apparaît comme approximation de la loi binomiale quand

$$p \sim \lambda/n$$

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

```
1 > 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

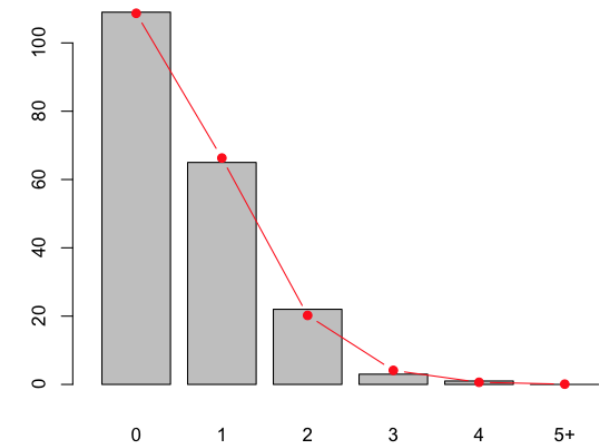


## La loi de petits nombres

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

```
1 > 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

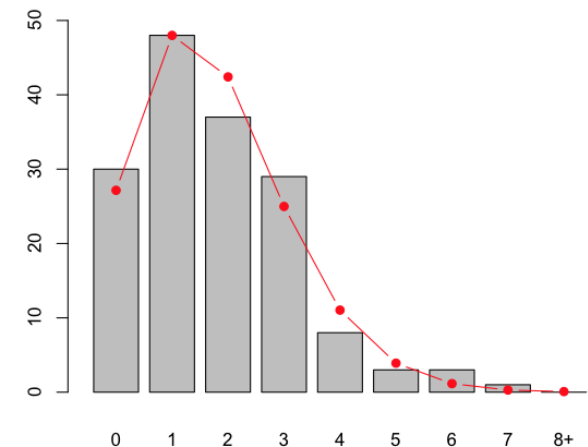


## La loi de petits nombres

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

```
1 > 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).

$t / \tau$	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**,

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

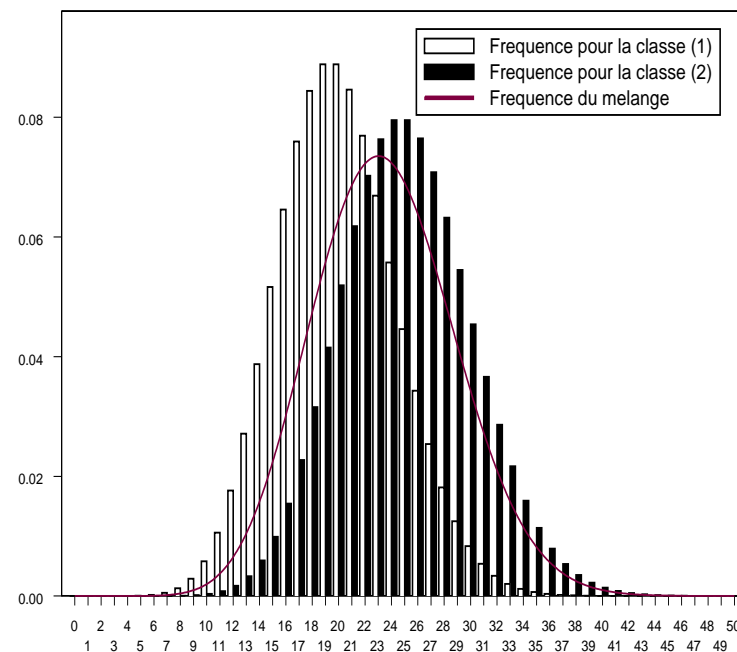
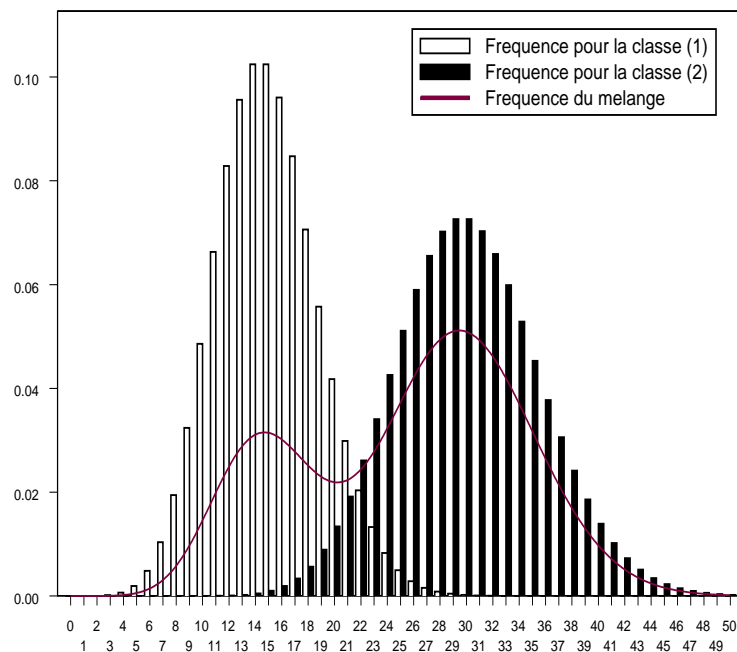
Pour rappel (cf premier cours),  $\mathbb{E}(N) = \text{Var}(N) = \lambda \in \mathbb{R}_+$ , **équi-dispersion**.



## 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  $\Theta$ , 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** apparaît dans le modèle Poisson mélange, lorsque  $\Theta \sim \mathcal{G}(\alpha, \alpha)$ . Dans ce cas,

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

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

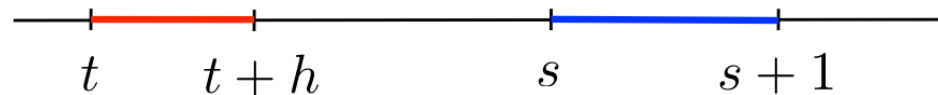
Dans ce cas, la loi (non conditionnelle) de  $N$  est

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

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

## Le processus de Poisson

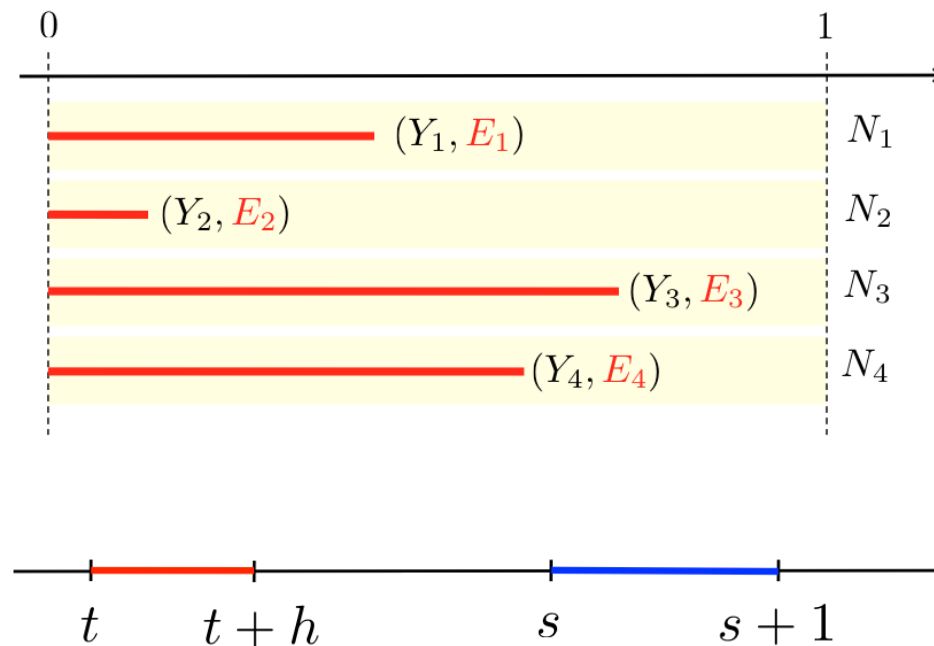
Pour rappel,  $(N_t)_{t \geq 0}$  est un **processus de Poisson homogène** (de paramètre  $\lambda$ ) s'il est à accroissements indépendants, et le nombre de sauts observés pendant la période  $[t, t + h]$  suit une loi  $\mathcal{P}(\lambda \cdot h)$ .



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

## Exposition et durée d'observation

Soit  $N_i$  la fréquence annulée 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

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

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

qui donne la condition du premier ordre

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

qui s'annule pour

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

## Maximum de Vraisemblance

```
1 > N = freq$nb_D0
2 > E= freq$exposition
3 > (lambda = sum(N) / sum(E))
4 [1] 0.06564229
5 > dpois(0:3, lambda)*100
6 [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

```

1 > X1=freq$carburant
2 > tapply(N,X1,sum)
3     D     E
4 998 926
5 > tapply(E,X1,sum)
6           D           E
7 12519.55 13911.58
8 > (lambdas = tapply(N,X1,sum)/tapply(E,X1,sum))
9           D           E
10 0.07971533 0.06656323
11 > cbind(dpois(0:3,lambdas[1]),dpois(0:3,lambdas[2]))*100
12           [,1]      [,2]
13 [1,] 92.337916548 93.560375149
14 [2,]  7.360747674  6.227681197
15 [3,]  0.293382222  0.207267302
16 [4,]  0.007795687  0.004598794

```

## Fréquence annuelle et tableau de contingence

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

```

1 > X1=freq$carburant
2 > X2=cut(freq$agevehicule ,c(0,3,10,101),right=FALSE)
3 > N_polices = table(X1,X2)
4 > E_agg=aggregate(E, by = list(X1 = X1, X2 = X2), sum)
5 > N_exposition=N_polices
6 > N_exposition[1:nrow(N_exposition),1:ncol(N_exposition)]=
7 +       matrix(E_agg$x,nrow(N_exposition),ncol(N_exposition))
8 > N_exposition
9       X2
10 X1      [0,3)   [3,10) [10,101)
11   D 3078.938 5653.109 3787.503
12   E 2735.014 5398.950 5777.619
13 >
14 > N_agg=aggregate(N, by = list(X1 = X1, X2 = X2), sum)
15 > N_sinistres=N_polices
16 > N_sinistres[1:nrow(N_sinistres),1:ncol(N_sinistres)]=

```



```

17 +      matrix(N_agg$x,nrow(N_sinistres),ncol(N_sinistres))
18 > N_sinistres
19      X2
20 X1    [0,3) [3,10) [10,101)
21   D    393    424     68
22   E    343    419     88
23 > Freq_sinistres = N_sinistres/N_exposition
24 > Freq_sinistres
25      X2
26 X1      [0,3)      [3,10)      [10,101)
27   D 0.12764143 0.07500298 0.01795378
28   E 0.12541067 0.07760768 0.01523119

```

## Fréquence annuelle et tableau de contingence

Si on utilise la fonction en ligne sur [blog](#)

```

1 > freq_sin("nb_D0")$Freq
2   X2
3 X1   [0,3)   [3,10)   [10,101)
4   D 0.12764143 0.07500298 0.01795378
5   E 0.12541067 0.07760768 0.01523119
6 > freq_sin("nb_RC")$Freq
7   X2
8 X1   [0,3)   [3,10)   [10,101)
9   D 0.07859854 0.08473214 0.07313526
10  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  $\mathbf{Y} = [Y_{i,j}]$  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

$$\hat{N}_{i,j} = L_i \cdot C_j, \text{ i.e. } \hat{\mathbf{N}} = \mathbf{L}\mathbf{C}^\top$$

cf [Bailey \(1963\)](#) et [Mildenhall \(1999\)](#)

L'estimation de  $\mathbf{L} = (L_i)$  et de  $\mathbf{C} = (C_j)$  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).

```

1 > D0=freq_sin("nb_D0")
2 > m=sum(D0$Sin)/sum(D0$Exp)
3 > L<-matrix(NA,10,nrow(D0$Exp))
4 > C<-matrix(NA,10,ncol(D0$Exp))
5 > L[1,]<-rep(1,2);colnames(L)=rownames(D0$Sin)
6 > C[1,]<-rep(m,3);colnames(C)=colnames(D0$Sin)
7 >
8 > for(j in 2:10){
9 +   L[j,1]<-sum(D0$Sin[1,])/sum(D0$Exp[1,]*C[j-1,])
10 +   L[j,2]<-sum(D0$Sin[2,])/sum(D0$Exp[2,]*C[j-1,])
11 +   C[j,1]<-sum(D0$Sin[,1])/sum(D0$Exp[,1]*L[j,])
12 +   C[j,2]<-sum(D0$Sin[,2])/sum(D0$Exp[,2]*L[j,])
13 +   C[j,3]<-sum(D0$Sin[,3])/sum(D0$Exp[,3]*L[j,])
14 + }
15 > L[10,]
```

```

16      D      E
17 1.007697 1.002415
18 > C[10,]
19      [0,3)      [3,10)      [10,101)
20 0.12593567 0.07588711 0.01623609
21 > PREDICTION=(L[10,])%*%t(C[10,])
22 > PREDICTION
23      [0,3)      [3,10)      [10,101)
24 [1,] 0.1269050 0.07647120 0.01636106
25 [2,] 0.1262397 0.07607034 0.01627529
26 > sum(PREDICTION[1,]*D0$Exp[1,])
27 [1] 885
28 > sum(D0$Sin[1,])
29 [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.

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

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

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).

```

1 > D0=freq_sin("nb_D0")
2 > m=sum(D0$Sin)/sum(D0$Exp)
3 > L<-matrix(1,100,nrow(D0$Exp))
4 > C<-matrix(NA,100,ncol(D0$Exp))
5 > L[1,]<-rep(1,2);colnames(L)=rownames(D0$Sin)
6 > C[1,]<-rep(m,3);colnames(C)=colnames(D0$Sin)
7 >
8 > for(j in 2:100){
9 +   L[j,1]=sum(D0$Sin[1,]*C[j-1,])/sum(D0$Exp[1,]*C[j-1,]^2)
10 +   L[j,2]=sum(D0$Sin[2,]*C[j-1,])/sum(D0$Exp[2,]*C[j-1,]^2)
11 +   C[j,1]=sum(D0$Sin[,1]*L[j,])/sum(D0$Exp[,1]*L[j,]^2)
12 +   C[j,2]=sum(D0$Sin[,2]*L[j,])/sum(D0$Exp[,2]*L[j,]^2)
13 +   C[j,3]=sum(D0$Sin[,3]*L[j,])/sum(D0$Exp[,3]*L[j,]^2)
14 + }
15 >
16 > L[100,]

```



```

17      D      E
18 1.011633 1.012599
19 > C[100,]
20      [0,3)      [3,10)      [10,101)
21 0.12507961 0.07536373 0.01611180
22 > PREDICTION=(L[10,])%*%t(C[10,])
23 > PREDICTION
24      [0,3)      [3,10)      [10,101)
25 [1,] 0.1265347 0.07624043 0.01629923
26 [2,] 0.1266554 0.07631321 0.01631479
27 > sum(PREDICTION[1,]*D0$Exp[1,])
28 [1] 882.3211
29 > sum(D0$Sin[1,])
30 [1] 885

```

## Méthode du $\chi^2$

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  $C_j$ .

```
1 > D0=freq_sin("nb_D0")
2 > m=sum(D0$Sin)/sum(D0$Exp)
```

```

3 > L<-matrix(1,100,nrow(D0$Exp))
4 > C<-matrix(NA,100,ncol(D0$Exp))
5 > L[1,]<-rep(1,2);colnames(L)=rownames(D0$Sin)
6 > C[1,]<-rep(m,3);colnames(C)=colnames(D0$Sin)
7 >
8 > for(j in 2:100){
9 +   L[j,1]=sqrt(sum(D0$Exp[1,]*D0$Freq[1,]^2/C[j-1,])/sum(D0$Exp[1,]*C
10 +     [j-1,]))
11 +   L[j,2]=sqrt(sum(D0$Exp[2,]*D0$Freq[2,]^2/C[j-1,])/sum(D0$Exp[2,]*C
12 +     [j-1,]))
13 +   C[j,1]=sqrt(sum(D0$Exp[,1]*D0$Freq[,1]^2/L[j,])/sum(D0$Exp[,1]*L[j
14 +     [,1]))
15 +   C[j,2]=sqrt(sum(D0$Exp[,2]*D0$Freq[,2]^2/L[j,])/sum(D0$Exp[,2]*L[j
16 +     [,2]))
17 +   C[j,3]=sqrt(sum(D0$Exp[,3]*D0$Freq[,3]^2/L[j,])/sum(D0$Exp[,3]*L[j
18 +     [,3]))
19 + }
20 >

```

```

16 > L[100,]
17           D           E
18 1.19012 1.18367
19 > C[100,]
20           [0,3)           [3,10)           [10,101)
21 0.10664299 0.06427321 0.01379165
22 > PREDICTION=(L[10,])%*%t(C[10,])
23 > PREDICTION
24           [0,3)           [3,10)           [10,101)
25 [1,] 0.1269180 0.07649285 0.01641373
26 [2,] 0.1262301 0.07607824 0.01632476
27 > sum(PREDICTION[1,]*D0$Exp[1,])
28 [1] 885.362
29 > sum(D0$Sin[1,])
30 [1] 885

```

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

## Approche(s) économétrique(s) du biais minimal

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

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

$$\mathbf{X}^\top [\mathbf{y} - \mathbf{X}\beta] = \mathbf{0}$$

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

$$\mathbf{X}^\top \mathbf{\Omega}^{-1} [\mathbf{y} - \mathbf{X}\beta] = \mathbf{0}$$

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

$$\mathbf{X}^\top e^{\mathbf{X}\beta} [\mathbf{y} - e^{\mathbf{X}\beta}] = \mathbf{0}$$

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

$$\mathbf{X}^\top \mathbf{\Omega}^{-1} e^{\mathbf{X}\beta} [\mathbf{y} - e^{\mathbf{X}\beta}] = \mathbf{0}$$

## Approche(s) économétrique(s) du biais minimal

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

```

1 > df_agg=data.frame(N=as.numeric(D0$Sin),E=as.numeric(D0$Exp), X1=rep
      (levels(X1),ncol(D0$Sin)),X2=rep(levels(X2),each=nrow(D0$Sin)))
2 > regpoislog_agg <- glm(N~X1+X2,offset=log(E),data=df_agg, family=
      poisson(link="log"))
3 > ndf_agg=df_agg
4 > ndf_agg$E=1
5 > matrix(predict(regpoislog_agg,type="response",newdata=ndf_agg),nrow
      (PREDICTION),ncol(PREDICTION))
6           [,1]      [,2]      [,3]
7 [1,] 0.1269050 0.07647120 0.01636106
8 [2,] 0.1262397 0.07607034 0.01627529

```

## Approche(s) économétrique(s) du biais minimal

ou au niveau individuel

```

1 > df=data.frame(N=freq[, "nb_DO"], E, X1, X2)
2 > regpoislog <- glm(N~X1+X2, offset=log(E), data=df, family=poisson(
    link="log"))
3 > rownames(PREDICTION)=c("D", "E")
4 > 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))))
5 > matrix(predict(regpoislog, newdata=newd,
6 +           type="response"), nrow(PREDICTION), ncol(PREDICTION))
7           [,1]      [,2]      [,3]
8 [1,] 0.1269050 0.07647120 0.01636106
9 [2,] 0.1262397 0.07607034 0.01627529

```

## Approche(s) économétrique(s) du biais minimal

On peut aussi envisager un modèle homoscdastique

```

1 > df_agg=data.frame(N=as.numeric(D0$Sin),E=as.numeric(D0$Exp), X1=rep
      (levels(X1),ncol(D0$Sin)),X2=rep(levels(X2),each=nrow(D0$Sin)))
2 > reggauss_agg <- glm(N~X1+X2,offset=log(E),family=gaussian(link="log
      "),data=df_agg)
3 > ndf_agg=df_agg
4 > ndf_agg$E=1
5 > matrix(predict(reggauss_agg,type="response",newdata=ndf_agg),nrow(
      PREDICTION),ncol(PREDICTION))
6           [,1]      [,2]      [,3]
7 [1,] 0.1261529 0.07592604 0.01594451
8 [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'occurrence,

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

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

**Remarque:** on posera parfois  $\theta_i = \eta_i = \mathbf{X}_i^\top \boldsymbol{\beta}$

La log-vraisemblance est ici

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

ou encore

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

Le gradient est ici

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

alors que la matrice Hessienne s'écrit

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

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

1. partir d'une valeur initiale  $\beta_0$
2. poser  $\beta_k = \beta_{k-1} - H(\beta_{k-1})^{-1} \nabla \log \mathcal{L}(\beta_{k-1})$

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

## La régression de Poisson

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

```
1 > Y <- freq$nb_D0
2 > X1 <- freq$agevehicule
3 > X <- cbind(rep(1, length(X1)), X1)
```

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

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

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

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

```

6 + beta=beta+solve(hessienne)%*%gradient
7 + }

```

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

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

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

```

1 > cbind(beta,sqrt(diag(solve(hessienne))))
2           [,1]      [,2]
3    -2.6949729  0.03382691
4 X1  -0.1219873  0.00575687

```

On retrouve toutes ces valeurs en utilisant

```

1 > regPoisson=glm(nb_DO~X1,data=freq,family=poisson)
2 > summary(regPoisson)
3
4 Coefficients:
5             Estimate Std. Error z value Pr(>|z|)
6 (Intercept) -2.694973   0.033827  -79.67  <2e-16 ***
7 X1          -0.121987   0.005757  -21.19  <2e-16 ***
8
9 (Dispersion parameter for poisson family taken to be 1)
10
11 Null deviance: 11893  on 49999  degrees of freedom
12 Residual deviance: 11334  on 49998  degrees of freedom
13 AIC: 14694
14
15 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

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

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.

```

1 > reg=glm(nb_RC~1,data=freq,family=poisson)
2 > predict(reg,type="response")[1]
3      1
4 0.03848
5 > reg=glm(nb_RC/10~1,data=freq,family=poisson)
6 There were 50 or more warnings (use warnings() to see the first 50)
7 > predict(reg,type="response")[1]
8      1
9 0.003848

```

## Propriété de la Régression de Poisson

Les conditions du premier ordre  $\mathbf{X}^\top [\mathbf{y} - \exp(\mathbf{X}\boldsymbol{\beta})] = \mathbf{0}$  peuvent s'écrire  $\mathbf{X}^\top \mathbf{y} = \mathbf{X}^\top \exp(\mathbf{X}\boldsymbol{\beta})$  ou encore  $\mathbf{X}^\top \hat{\mathbf{y}}$ .

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

$$\mathbf{1}^\top \mathbf{y} = \mathbf{1}^\top \hat{\mathbf{y}} \text{ i.e. } \sum_{i=1}^n y_i = \sum_{i=1}^n \hat{y}_i$$

Si  $\mathbf{X}$  est une variable factorielle de modalités  $a_1, \dots, a_J$ ,

$$\mathbf{1}_{a_j}^\top \mathbf{y} = \mathbf{1}_{a_j}^\top \hat{\mathbf{y}} \text{ i.e. } \sum_{i; x_i = a_j} y_i = \sum_{i; x_i = a_j} \hat{y}_i$$

## 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.

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



```

1 > Y <- freq$nb_D0
2 > X1 <- freq$agevehicule
3 > E <- freq$exposition
4 > X=cbind(rep(1,length(X1)),X1)
5 > beta=lm(Y~0+X)$coefficients
6 > for(s in 1:20){
7 +   gradient=t(X)%*%(Y-exp(X*%beta+log(E)))
8 +   hessienne=matrix(0,ncol(X),ncol(X))
9 +   for(i in 1:nrow(X)){
10 +     hessienne=hessienne + as.numeric(exp(X[i,]%*%beta+log(E[i])))*(
      X[i,]%*%t(X[i,]))}
11 +   beta=beta+solve(hessienne)%*%gradient
12 + }
13 >
14 > cbind(beta,sqrt(diag(solve(hessienne))))
15           [,1]      [,2]
16 -1.8256863  0.035138680
17 X1 -0.1578027  0.006198479

```

```

1 > regPoisson=glm(nb_DO~ageconducteur+offset(log(exposition)),data=
    freq,family=poisson)
2 > summary(regPoisson)
3
4 Coefficients:
5             Estimate Std. Error z value Pr(>|z|)
6 (Intercept)   -1.825686    0.035139  -51.96   <2e-16 ***
7 ageconducteur -0.157803    0.006198  -25.46   <2e-16 ***
8
9 (Dispersion parameter for poisson family taken to be 1)
10
11 Null deviance: 11671  on 49999  degrees of freedom
12 Residual deviance: 10822  on 49998  degrees of freedom
13 AIC: 14183
14
15 Number of Fisher Scoring iterations: 7

```

## Régression de Poisson multiple

On peut (bien entendu) régresser sur plusieurs variables explicatives

```
1 > model_RC=glm(nb_RC~zone+as.factor(puissance)+agevehicule+
  ageconducteur+carburant+offset(log(exposition)),data=freq,family=
  poisson)
2 > summary(model_RC)
3 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	

```

15 as.factor(puissance)9      0.181978      0.123996      1.468      0.1422
16 as.factor(puissance)10     0.254358      0.119777      2.124      0.0337 *
17 as.factor(puissance)11     0.001156      0.170163      0.007      0.9946
18 as.factor(puissance)12     0.243677      0.223207      1.092      0.2750
19 as.factor(puissance)13     0.513950      0.284159      1.809      0.0705 .
20 as.factor(puissance)14     0.582564      0.295482      1.972      0.0487 *
21 as.factor(puissance)15     0.173748      0.383322      0.453      0.6504
22 agevehicule                0.001467      0.004191      0.350      0.7264
23 ageconducteur              -0.008844      0.001658     -5.335     9.58e-08 ***
24 carburantE                  -0.201780      0.049265     -4.096     4.21e-05 ***
25
26 (Dispersion parameter for poisson family taken to be 1)
27
28 Null deviance: 12680 on 49999 degrees of freedom
29 Residual deviance: 12524 on 49980 degrees of freedom
30 AIC: 16235
31
32 Number of Fisher Scoring iterations: 6

```

## Effets marginaux, et élasticité

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

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

estimés par  $\exp[\mathbf{X}_i^\top \hat{\boldsymbol{\beta}}] \cdot \hat{\beta}_k$

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

```

1 > coef(model_RC)[19]
2 ageconducteur
3 -0.008844096
4 > effet19=predict(model_D0, type="response")*coef(model_RC)[19]
5 > effet19[1:4]
6           1           2           3           4
7 -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  $k$ ,  $\bar{Y} \cdot \hat{\beta}_k$

```
1 > mean(predict(model_RC, type="response")) * coef(model_RC)[19]
2 ageconducteur
3 -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 d'étudier 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

```
1 Coefficients:
2
3 Estimate Std. Error z value Pr(>|z|)
4
5 ...
6
7 1 carburantE -0.201780 0.049265 -4.096 4.21e-05 ***
```

i.e.

```
23 > coefficients(model_RC)["carburantE"]
24 carburantE
25 -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,

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

```
2 carburantE
```

```
3 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<sup>2</sup> 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.

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

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

```
1 > freq$zone=relevel(freq$zone,"C")
```

```
2 > model_RC=glm(nb_RC~zone+as.factor(puissance)+agevehicule+
```



```

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

...

1 zoneA           -0.196208    0.077090   -2.545  0.010921  *
2 zoneB           -0.184722    0.086739   -2.130  0.033202  *
3 zoneD            0.207174    0.064415    3.216  0.001299  **
4 zoneE            0.398664    0.064569    6.174  6.65e-10  ***
5 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,

```

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

```

```
4 > summary(model_RC)
```

```
...
```

1	zoneC	-0.398664	0.064569	-6.174	6.65e-10	***
2	zoneA	-0.594872	0.079207	-7.510	5.90e-14	***
3	zoneB	-0.583385	0.088478	-6.594	4.29e-11	***
4	zoneD	-0.191490	0.066185	-2.893	0.00381	**
5	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 Fisher,

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

```

6 Linear hypothesis test
7
8 Hypothesis:
9 zoneA - zoneB = 0
10 zoneE - zoneF = 0
11
12 Model 1: restricted model
13 Model 2: nb_RC ~ zone + as.factor(puissance) + agevehicule +
      ageconducteur +
14      carburant + offset(log(exposition))
15
16      Res.Df Df    Chisq Pr(>Chisq)
17 1    49982
18 2    49980  2  0.4498    0.7986

```

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

```

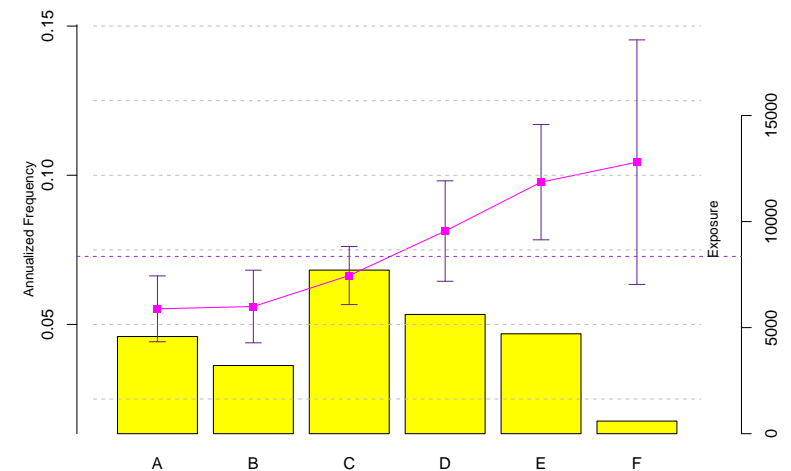
1 > levels(freq$zone)=c("AB", "AB", "C", "D", "EF", "EF")

```

## Régression de Poisson sur de variable factorielle, visualisation

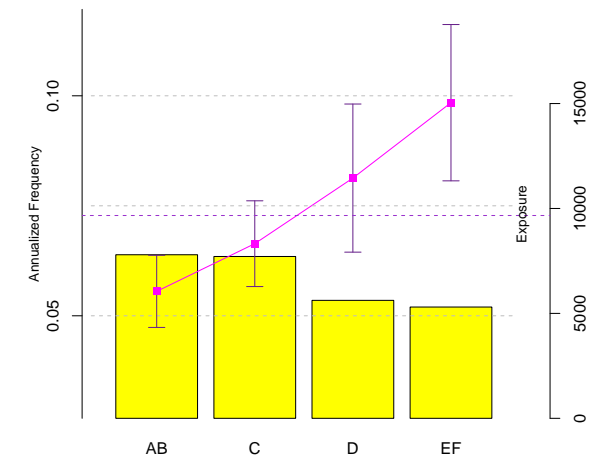
Avec la fonction `graph_freq` de [blog](#)

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



## Régression de Poisson sur de variable factorielle, visualisation

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

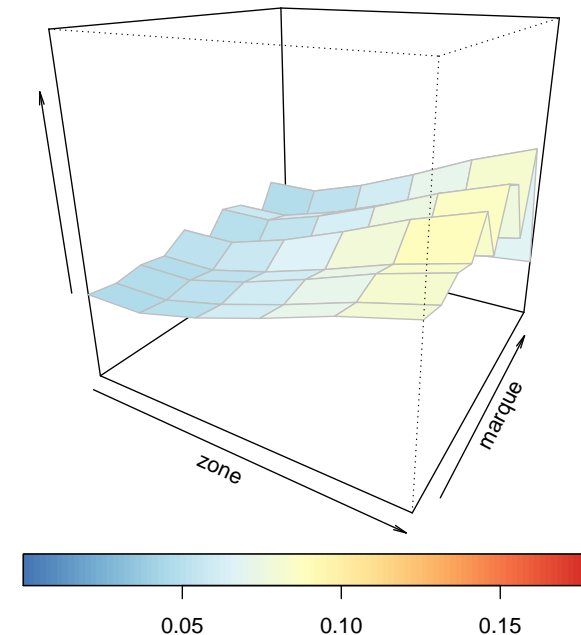


## Effets Croisés

```

1 lst_p=levels(freq$zone)
2 lst_m=levels(freq$marque)
3 REG1=glm(nb_RC~zone+marque+offset(
  exposition), data=freq,family=
  poisson)
4 nd=data.frame(
5   zone=rep(lst_p,length(lst_m)),
6   marque=rep(lst_m,each=length(lst_p)),
7   exposition=1)
8 y1=predict(REG1,newdata=nd,type="
  response")
9 my1=matrix(y1,length(lst_p),length(lst_
  m))
10 persp(my1)

```

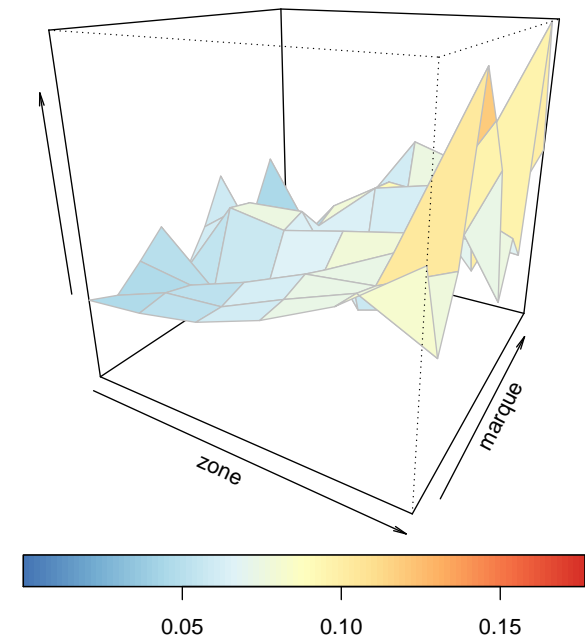


## Effets Croisés

```

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

```



## Variable Explicative Spatiale

Les régions utilisées sur la base sont reliées à la classification INSEE, [data.gouv.fr](http://data.gouv.fr)

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

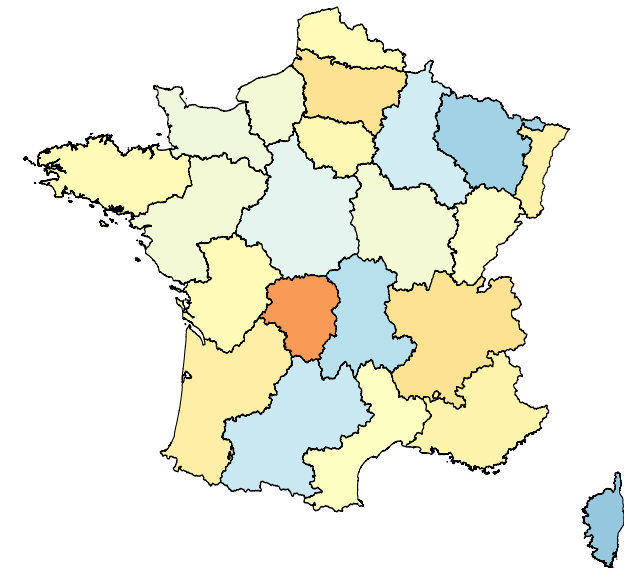
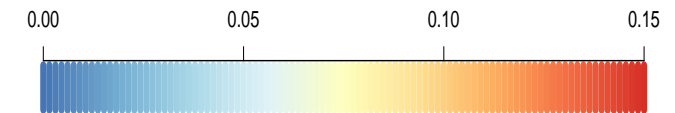


## Variable Explicative Spatiale

```

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

```



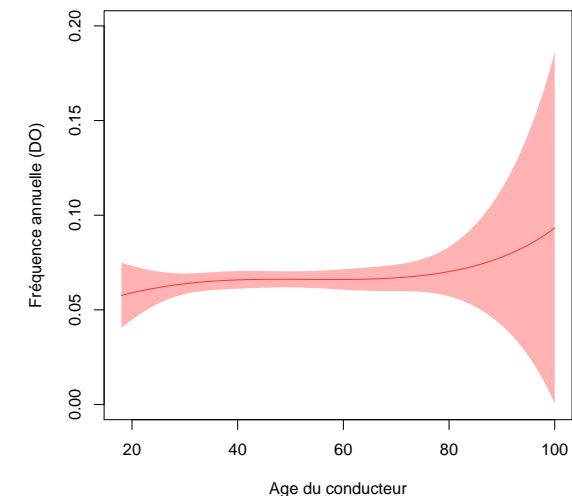
## Lissage et modèles non-paramétriques

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

```

1 > library(splines)
2 > model_D0=glm(nb_D0~bs(ageconducteur)+
  offset(log(exposition)), data=freq,
  family=poisson)
3 > u=seq(18,100,by=.1)
4 > newd=data.frame(ageconducteur=u,exposition
  =1)
5 > y_D0=predict(model_D0,newdata=newd,type="
  response",se.fit =TRUE)
6 > plot(u,y_D0$fit,col="red")
7 > polygon(c(u,rev(u)),c(y_D0$fit+2*y_D0$se.
  fit,rev(y_D0$fit-2*y_D0$se.fit)),
8 + col=rgb(1,0,0,.3),border=NA)

```



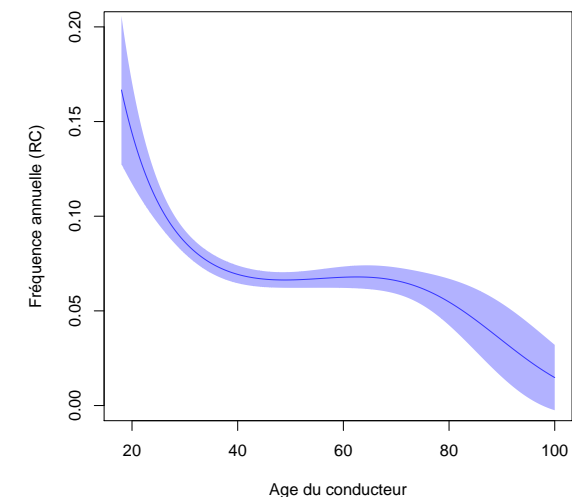
## Lissage et modèles non-paramétriques

et pour RC

```

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

```

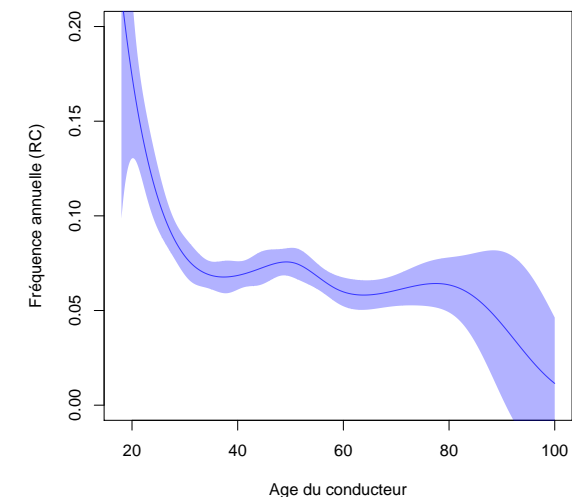


## Lissage et modèles non-paramétriques

```

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

```

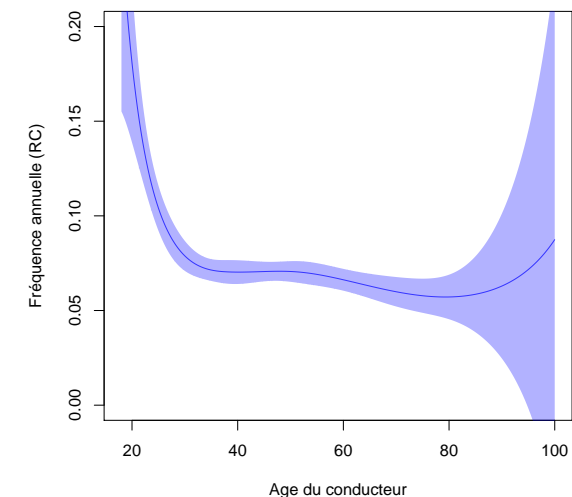


## Lissage et modèles non-paramétriques

```

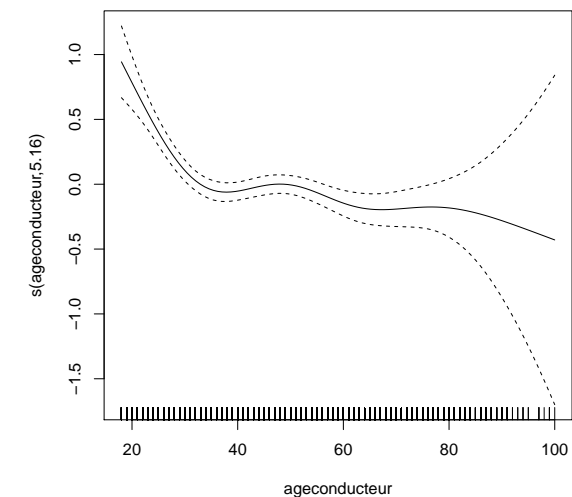
1 > library(splines)
2 > model_RC=glm(nb_RC~bs(ageconducteur,df=5)+
  offset(log(exposition)),
3 + data=freq,family=poisson)
4 > u=seq(18,100,by=.1)
5 > newd=data.frame(ageconducteur=u,exposition
  =1)
6 > y_RC=predict(model_RC,newdata=newd,type="
  response",se.fit =TRUE)
7 > plot(u,y_DO$fit,col="blue")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

```



## Lissage et modèles non-paramétriques

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



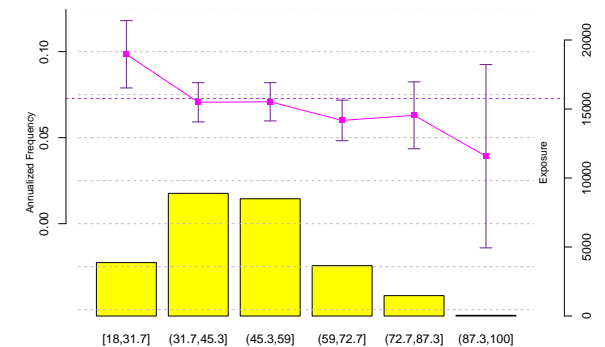
## Transformer une variable continue en classes tarifaires

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

```

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

```



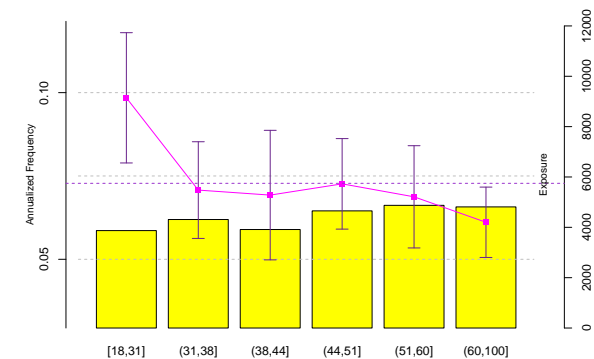
## Transformer une variable continue en classes tarifaires

ou par intervalles basés sur les quantiles

```

1 > library(classInt)
2 > CI=classIntervals(freq$ageconducateur, 6,
   style = "quantile",intervalClosure="left
   ")
3 > LV=CI$brks
4 > LV[6]=LV[6]+1
5 > graph_freq("ageconducateur",levels=LV)

```

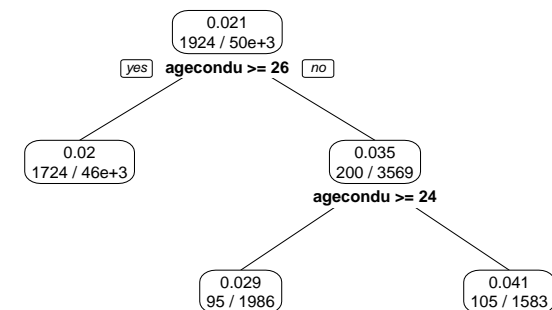




## Transformer une variable continue en classes tarifaires

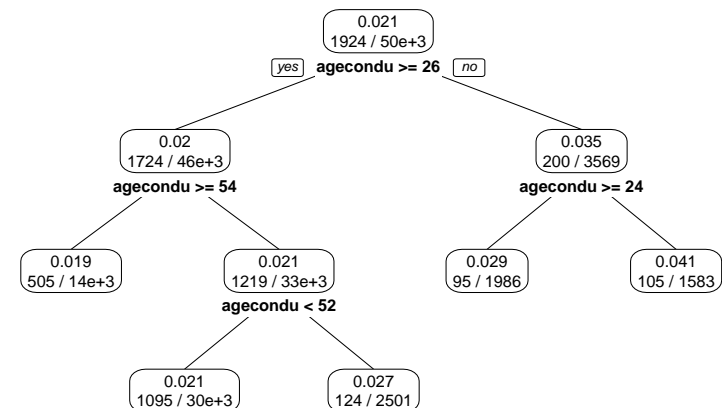
On peut aussi utiliser un découpage endogène, cf. **arbre de régression**

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



## Transformer une variable continue en classes tarifaires

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

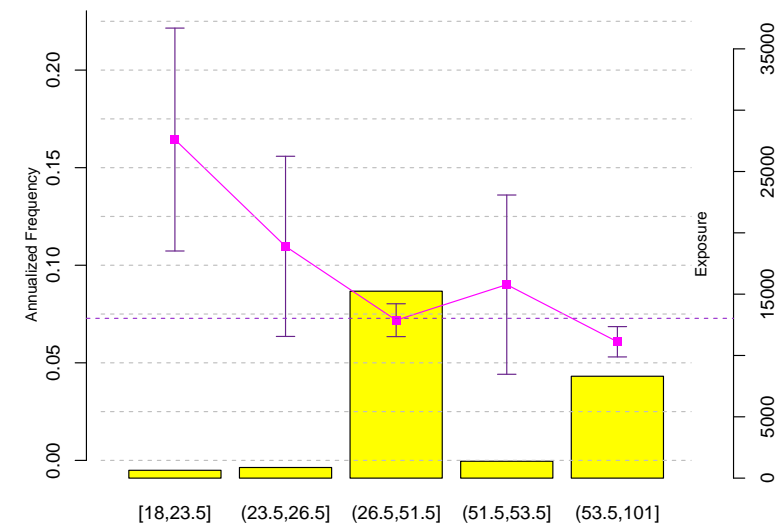


## Transformer une variable continue en classes tarifaires

```

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

```

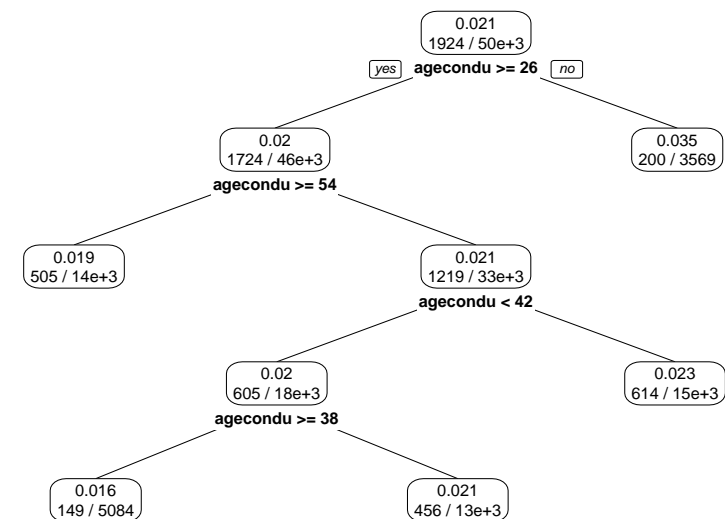


## Transformer une variable continue en classes tarifaires

```

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

```

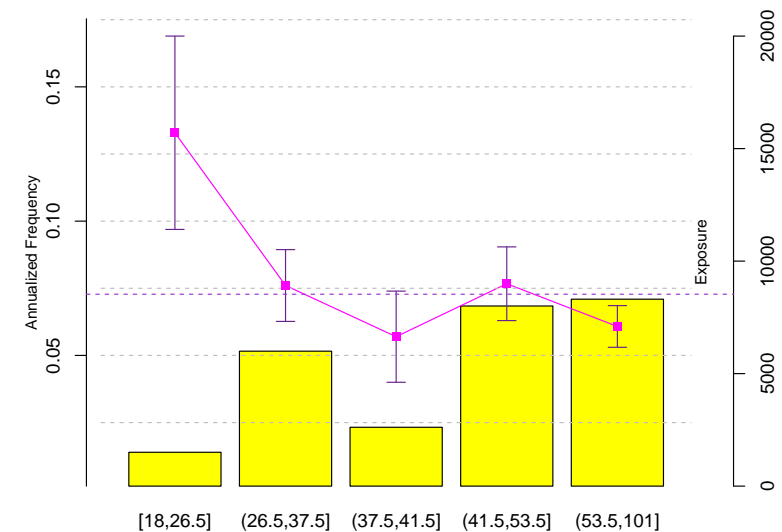


## Transformer une variable continue en classes tarifaires

```

1 > lb=labels(arbre)
2 > cut_ages = substr(lb,nchar(lb)-3,
    nchar(lb))
3 > cut_ages=as.numeric(cut_ages)
4 > LV=c(18,sort(unique(cut_ages[!is.na
    (cut_ages)])),101)
5 > 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<sup>\*</sup> de la loi de Poisson. Pour un noeud  $N$ ,

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

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

<sup>\*</sup> la log-vraisemblance pour une loi de Poisson est

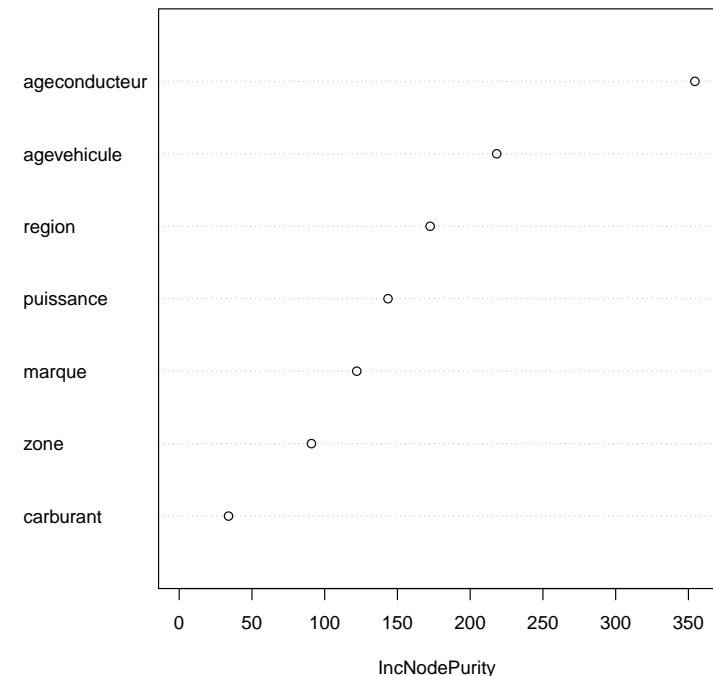
$$\log \mathcal{L}(\boldsymbol{\lambda}) = \sum_{i=1}^n y_i \log \lambda_i - \lambda_i - \log(y_i!)$$

et la déviance est alors la différence

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

## Des arbres aux forêts

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



## Procédure stepwise, AIC

```

1 > step(glm(nb_RC~ageconducteur+agevehicule+region+puissance+marque+
           zone+carburant+offset(exposition), data=sub_freq,family=poisson))
2 Start:  AIC=16162.21
3 nb_RC ~ ageconducteur + agevehicule + as.factor(region) + puissance +
4         marque + zone + carburant + offset(exposition)
5
6           Df Deviance   AIC
7 - agevehicule      1    12409 16160
8 <none>              12409 16162
9 - puissance        1    12412 16163
10 - ageconducteur     1    12418 16169
11 - marque           10    12438 16171
12 - carburant          1    12423 16174
13 - region            21    12464 16175
14 - zone              5    12469 16212

```



## Procédure stepwise, AIC

ou avec des splines sur les variables continues,

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