L'Actuariat avec 

Arthur Charpentier, Christophe Dutang

Décembre 2012 – Version numérique

rédigé en LaTeX

"*Composing computer programs to solve scientific problems is like writing poetry. You must choose every word with care and link it with the other words in perfect syntax. There is no place for verbosity or carelessness. To become fluent in a computer language demands almost the antithesis of modern loose thinking. It requires many interactive sessions, the hands-on use of the device. You do not learn a foreign language from a book, rather you have to live in the country for year to let the language become an automatic part of you, and the same is true for computer languages.*"

James Lovelock, *Building Java Programs.*

Les actuaires ont toujours été des gros utilisateurs de statistiques, et des manipulateurs de données. Il n'est donc pas surprenant de vouloir écrire un ouvrage dédié à l'actuariat dans la collection *Utilisation de R*. Toutefois ce livre n'est pas un livre d'actuariat [1]. Ce n'est pas non plus un livre de programmation [2]. Nous ne présenterons pas ici les rudiments du langage R, et renvoyons à plusieurs ouvrages parus récemment [3]. Le but est plutôt de proposer un complément pratique et concret à ces ouvrages théoriques, à l'aide d'un langage de programmation qui présente l'avantage d'être simple à déchiffrer pour quiconque souhaite comprendre les algorithmes utilisés.

Nous avons ici la modestie de présenter quelques techniques universelles sans aucunement prétendre à l'exhaustivitié, tout en ayant l'ambition de montrer que R est un logiciel idéal pour les actuaires et les personnes intéressés aux problématiques de modélisation statistique des risques.

Il est néanmoins particulièrement délicat d'écrire un livre basé sur l'utilisation d'un logiciel. Ou en l'occurence sur un langage de programmation (le S) et sur une communauté mettant à disposition des libraries de fonctions (les *packages*). Délicat d'autant plus que la communauté est particulièrement active. Dès que le livre sera publié, il risque de devenir obsolète assez rapidement. Mais il nous a semblé que si les packages à venir pourraient simplifier la tâche des actuaires, le contenu du livre devrait garder une relative fraîcheur pendant quelques années encore. C'est tout du moins ce que l'on espère, et qui légitime le fait que l'ouvrage soit maintenant disponible dans une version reliée.

En R, nous proposons de discuter les modèles économétriques de tarification et de provisionnement, les calculs d'annuités en assurance vie, les méthodes d'estimation des coefficients de crédibilité, ou encore du lissage des tables de mortalité prospectives. Il est évident que d'autres logiciels pourraient faire exactement la même chose. Mais R présente l'avantage d'être simple à comprendre (de part la forme matricielle du langage), d'être libre (ce qui permet à tout à chacun de reprendre des codes existants et de les améliorer), et gratuit.

Ce livre est basé sur des notes écrites pour des cours dispensés depuis une petite dizaine d'années (à l'Université Laval à Québec, à l'Université de Montréal, à l'Université du Québec à Montréal, à l'ENSEA de Rabat, à l'ENSAE à Paris, à l'Université de Rennes 1, ou à l'Institut

1. Nous renvoyons aux ouvrages Bowers et al. (1997), Denuit & Charpentier (2004), Denuit & Charpentier (2005), Kaas et al. (2009), de Jong & Zeller (2008), Frees (2009), Dickson et al. (2009), Klugman et al. (2009), Ohlsson & Johansson (2010), Marceau (2012) (parmi tant d'autres) qui présentent les fondementaux théoriques que nous allons évoquer ici sans réellement les justifier.

2. Nous renvoyons aux ouvrages Chambers (2009), Gentle (2009), Mori (2009), Cormen et al. (2009), Venables & Ripley (2002*a*), ou encore Knuth (1997*a*,*b*, 1998) - pour une réfléxion plus profonde sur la programmation - qui proposent des algorithmes probablement plus efficaces et rapides que la majorité des codes que nous verrons ici.

3. Zuur et al. (2009), Maindonald & Braun (2007), Chambers (2009), Dalgaard (2009), ou encore Krause (2009) (là encore parmi tant d'autres) pour des introductions à la programmation en R.

de Sciences Financières et d'Assurance (ISFA) à Lyon, mais aussi lors de formations données à des actuaires de différentes compagnies et mutuelles d'assurance en France).

Cet ouvrage va proposer dans le Chapitre 1 un paranorama des distributions statistiques utilisées pour la modélisation des sinistres en actuariat (dans l'esprit de Klugman et al. (2009)). Différentes méthodes d'estimation de paramètres et d'ajustement de lois seront évoquées, dont la majorité sont implémentées dans le package **fitdistrplus**.

Dans le Chapitre 2, nous aborderons la tarification *a priori* et l'utilisation des modèles linéaires généralisés pour calculer la prime pure d'un contrat d'assurance (en l'occurence en responsabilité civile automobile). Nous verrons ainsi comment modéliser les fréquences de sinistres (régression de Poisson et ses extensions) et les coûts (en évoquant l'écrètement des grands sinistres).

Le Chapitre 3 sera dédié aux calculs de provisions pour sinistres à payer, à partir de la méthode dite *Chain Ladder*, avec diverses extensions plus récentes, dont l'approche de Mack, et l'utilisation de la régression Poisson. Ce chapitre s'appuira sur le package **ChainLadder** tout en insistant sur l'écriture des algorithmes.

Enfin les Chapitre 4 et 5 présenteront des applications en assurance-vie, avec des calculs de base dans le Chapitre 4 (proposant de programmer plusieurs grandeurs classiques présentées dans Bowers et al. (1997) ou Dickson et al. (2009)). Le Chapitre 5 proposera une application sur les tables de mortalités prospectives. Ce dernier chapitre s'appuiera essentiellement sur le package **demography**, mais mais aussi le package **gnm**.

Bien que ce livre aborde des sujets aussi divers que les algorithmes récursifs pour les calculs d'annuités, ou la régression Poissonnienne pour le calcul de provisions pour sinistres à payer, nous avons essayé d'avoir des notations aussi cohérentes que possibles entre les chapitres, mais aussi avec les notations internationales usuelles. Nous noterons ainsi x une valeur réelle, X une variable aléatoire réelle, $\boldsymbol{x}$ un vecteur de $\mathbb{R}^d$, et $\boldsymbol{X}$ une matrice $d \times k$. La version sous R sera alors notée `x` ou `X`. Si $\boldsymbol{X}$ est une matrice, sa transposée sera notée $\boldsymbol{X}'$. Pour les lois de probabilité, nous noterons F la fonction de répartition, et f la densité associée - si elle existe - ou la masse de probabilitée associée dans le cas discret. Dans les sections traîtant d'inférence statistique, $\widehat{\theta}$ désignera l'estimateur d'un paramètre θ ; et dans les sections où nous nous attacherons à prédire diverses quantitées, $\widehat{Y}$ désignera l'estimateur de $\mathbb{E}(Y)$, voire $\mathbb{E}(Y|\boldsymbol{X}=\boldsymbol{x})$ lorsque des variables explicatives seront utilisées. Dans le chapitre d'assurance vie, $\cdot$ désignera un produit, et sera utilisé afin de séparer clairement les termes (dont les indices de part et d'autre ne permettent souvent pas une lecture simple).

Avant de conclure cette rapide introduction, nous tenions à remercier plusieurs personnes. Nous remercions Bernard Mathieu qui a proposé dès 2005 d'organiser des formations à R dédiées aux actuaires, en France. Et nous remercions toutes les personnes qui ont suivi ces formations pour les questions qu'elles ont soulevées ! Nous remercions aussi Frédéric Planchet pour ses relectures des manuscrits, et pour avoir lancé l'idée de publier un livre de R en actuariat. De manière assez globale, nous remercions nos étudiants qui ont suivi (ou subi) depuis 7 ans l'évolution de ces notes qui ont servi de support au livre que vous tenez aujourd'hui entre vos mains. Nous remercions aussi nos collègues et amis qui ont accepté de relire certaines parties de livre.

Arthur Charpentier & Christophe Dutang, Décembre 2012.

Table des matières

Chapitre 1

Modèles de sinistres sans variables explicatives

Plusieurs des techniques actuarielles étudiées dans cet ouvrage requièrent de connaître la loi de probabilité du montant ou du nombre de sinistres dans un portefeuille d'assurance non-vie. Le présent chapitre passe en revue les techniques les plus couramment utilisées pour déterminer ces lois à partir d'échantillon de données. En dehors de données simulées pour évaluer la robustesses des estimateurs, nous étudierons deux jeux de données : `dental` et `vents` contenant des montants de réclamation en assurance dentaire et des vitesses de vent de deux stations en région Rhône-Alpes, respectivement.

Nous débutons le chapitre par un rappel des principales lois utilisées en assurance non-vie dans la section 1.1. En sections 1.2 et 1.3, nous présentons les deux grandes méthodes d'estimation, à savoir l'approche non-paramétrique et l'approche paramétrique, respectivement. Enfin, la section 1.4 termine ce chapitre en présentant les méthodes de calibration standard pour les copules.

1.1 Rappels des lois usuelles en actuariat

De la définition même des risques d'assurance (et leur caractère incertain), les actuaires ont besoin d'utiliser les outils probabilistiques pour modéliser les phénomènes aléatoires. Les lois de probabilités s'attachent à préciser, formaliser et différencier les phénomènes aléatoires. Cette section a pour but de rappeler les lois de probabilités usuelles en actuariat non-vie. Pour une introduction aux probabilités, nous renvoyons le lecteur vers les ouvrages de références, par exemple, Amiot (1999), Moral et al. (2006), Delmas (2012).

Notons X une variable aléatoire représentant notre quantité d'intérêt, par exemple le montant du sinistre ou le nombre de sinistres au sein d'un portefeuille d'assurance. Une façon classique de caractériser X est d'en préciser sa fonction de répartition $F_X : x \mapsto \mathbb{P}(X \leq x)$ sur $\mathbb{R}$, ou un domaine $D \subset \mathbb{R}$ pouvant être borné ou non. Rappelons que F_X doit être une fonction croissante, continue à gauche de D dans $[0, 1]$. Deux cas doivent être distingués, soit la fonction F_X possède une dérivée notée f_X : cas des variables continues, soit elle n'en possède pas : cas des variables discrètes et/ou des variables mixtes. Ci-dessous, nous présentons donc dans un premier temps les lois continues. Ensuite, nous décrivons les lois discrètes et les mixtes. Enfin, nous terminons par les lois multivariées et les copules.

En commentaire général sur les distribution les plus classiques, R fournit la densité ou la fonction de masse de probabilité `d`, la fonction de répartition `p`, la fonction quantile `q` et un

générateur aléatoire `r` associées. Pour une loi de probabilité de racine `toto`, on a dons les 4 fonctions `dtoto`, `ptoto`, `qtoto`, `rtoto`. Si on souhaite utiliser une loi non-implémentée dans R, de nombreux packages comblent ce manque, voir la "task view" pour une liste exhaustive http://cran.r-project.org/web/views/Distributions.html. Dans cette longue liste, citons notamment le package **actuar** - dédié à l'actuariat - implémentant en plus les 18 lois de probabilités que nous détaillons dans la section suivante. De plus, **actuar** fournit également des fonctions auxiliaires pour les 18 lois et celles de R : les moments ordinaires $\mathbb{E}(X^k)$, les moments limités $\mathbb{E}(\min(X,x)^k)$, la fonction génératrice des moments $\mathbb{E}\left(e^{tX}\right)$ sous réserve que ces quantités existent. Trois préfixes ont donc été rajoutés `m`, `lev` et `mgf`. Par exemple, la fonction `mgfgamma` implémente la fonction génératrice des moments de la loi gamma.

1.1.1 Les lois continues

Dans cette sous-section, nous supposons que la loi de probabilité possède une densité f_X. En annexe A.1.1, nous rappelons la génèse des différentes densités proposées dans la littérature scientifique à l'aide du système de Pearson. Nous renvoyons le lecteur vers Kotz et al. (1994*a*,*b*) pour plus de détails sur les lois continues.

Les lois classiques en actuariat

Traditionnellement en actuariat, comme les principales quantités d'intérêt sont des coûts ou des durées, les lois de probabilités les plus utilisées sont celles à support positif. Les trois lois positives les plus courantes sont les suivantes :

– la loi gamma dont la densité `dgamma` s'écrit :

$$f_X(x) = \frac{\lambda^\alpha}{\Gamma(\alpha)} e^{-\lambda x} x^{\alpha-1},$$

où $x \geq 0$, $\alpha, \lambda > 0$ (les paramètres sont notés `shape` et `rate` sous R) et Γ représente la fonction Gamma. Si $\alpha = 1$, on retrouve la loi exponentielle. La fonction de répartition n'a de forme explicite puisqu'elle s'exprime à l'aide de la fonction Gamma incomplète $\gamma(,)$:

$$F_X(x) = \gamma(\alpha, \lambda x)/\Gamma(\alpha),$$

où $\gamma(\alpha, x) = \int_0^x t^{\alpha-1} e^{-t} dt$, voir Olver et al. (2010) pour les détails sur la fonction gamma incomplète inférieure.
Lorsque $\alpha = 1$, la distribution est appelée une exponentielle et lorsque $\alpha = r/2$ et $\lambda = 1/2$, la distribution est appelée loi du chi-deux avec r degrés de liberté. Le mode de la distribution est en $x = 0$ si $\alpha \leq 1$ et en $x > 0$ si $\alpha > 1$.
Enfin, une distribution gamma avec paramètre α entier est également nommée Erlang. Dans ce cas, on a

$$F_X(x) = 1 - \sum_{i=0}^{\alpha-1} \frac{(\lambda x)^i}{i!} e^{-\lambda x}.$$

– la loi log-normale dont la densité `dlnorm` s'écrit :

$$f_X(x) = \frac{1}{\sigma x \sqrt{2\pi}}\, e^{-\frac{(\log(x)-\mu)^2}{2\sigma^2}},$$

pour $x > 0$, $\mu \in \mathbb{R}$ et $\sigma^2 > 0$ (les paramètres sont notés `meanlog` et `sdlog` respectivement sous R). Sa fonction de répartition est simplement

$$F_X(x) = \Phi\left(\frac{\log(x) - \mu}{\sigma}\right),$$

où $x > 0$ et Φ dénote la fonction de répartition de la loi normale centrée réduite).

– la loi de Weibull dont la densité `dweibull` s'écrit :

$$f_X(x) = \frac{\beta}{\eta^\beta} x^{\beta-1} e^{-(\frac{x}{\eta})^\beta},$$

où $x > 0$ and $\eta, \beta > 0$ (notés `scale` et `shape` respectivement. Sa fonction de répartition possède l'expression suivante

$$F_X(x) = 1 - e^{-(\frac{x}{\beta})^\eta}.$$

Comme le montre la figure 1.1, ces lois des plus usuelles ont des densités assez différentes et possèdent des propriétés très différentes. Les paramètres ont été choisis de manière à ce que les trois lois soient d'espérance 1.

```
> x <- seq(0,5,.01)
> y <- dlnorm(x, -1/2, 1)
> y2 <- dgamma(x, 2, 2)
> y3 <- dweibull(x, 2, 2/sqrt(pi))
> leg.txt <- c("LN(-1/2,1)","G(2,2)","W(2,2/sqrt(pi))")
> plot(x, y, xlab="x", ylab="f(x)", main="Comparaison des densit\'es",
+ ylim=range(y, y2, y3), col="black", type="l")
> lines(x,y2, lty=2)
> lines(x,y3, lty=3)
> legend("topright",leg=leg.txt, col="black",lty=1:3)
```

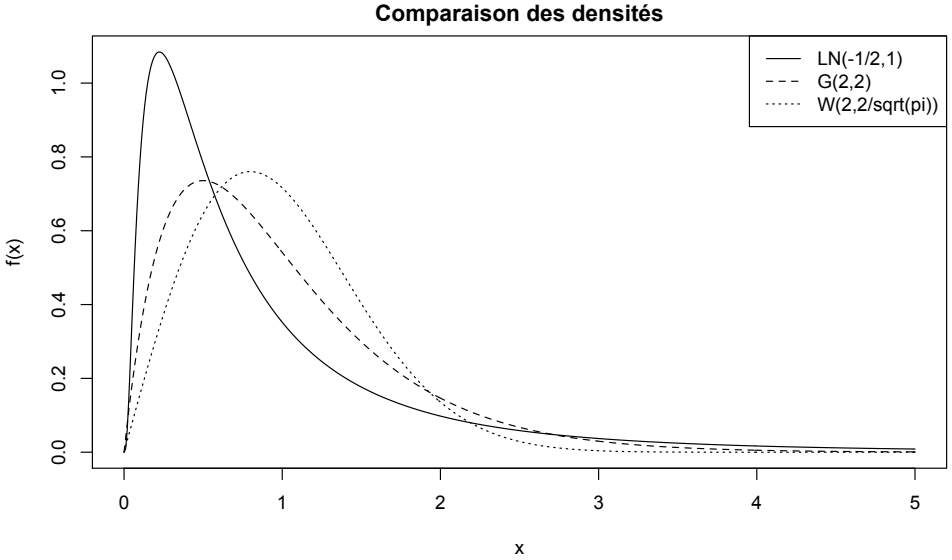


FIGURE 1.1 – Densités de lois usuelles pour des variables positives.

Dans le tableau 1.1, on a listé par ordre alphabétique les lois continues présentes avec R. Notons que ce tableau 1.1 contient des lois à support infini comme la loi normale, des lois à support borné comme la loi béta ou des lois à support positif comme la loi exponentielle.

Les familles de lois continues

Pour obtenir d'autres lois, on peut appliquer différentes transformations sur ces lois :

– une translation $X - c$ (par exemple la loi lognormale translatée pour X lognormale),

Lois de probabilité	Racine	Lois de probabilité	Racine
beta	`beta`	logistique	`logis`
Cauchy	`cauchy`	lognormale	`lnorm`
chi-2	`chisq`	normale	`norm`
exponentielle	`exp`	Student t	`t`
Fisher F	`f`	uniforme	`unif`
gamma	`gamma`	Weibull	`weibull`

TABLE 1.1 – Loi implémentées dans R.

- une mise à l'échelle λX (par exemple la loi normale pour X normale centrée réduite),
- une puissance X^α (par exemple la loi beta type 1 généralisée pour X de loi beta type 1),
- un inverse $1/X$ (par exemple la loi inverse gamma pour X gamma),
- un logarithme $\log(X)$ (par exemple la loi loglogistique pour X logistique),
- une exponentielle e^X (par exemple la loi Pareto pour X exponentiel),
- un ratio $X/(1-X)$ (par exemple la loi béta type 2 pour X une loi béta type 1).

Pour chacune des transformations ci-dessus, on peut facilement déduire la densité en calculant la transformée inverse. Par exemple, pour $Y = \lambda X$, on a $f_Y(y) = f_X(y/\lambda)$. Dans R, il est facile de générer des réalisations de telles transformations. Choisissons par exemple $Y = \log X$ où X est une loi uniforme sur [0,1].

```
> x <- runif(100)
> y <- -log(x)
> par(mar=c(4, 4, 2, 1), mfrow=c(1, 2))
> hist(y)
> plot(ecdf(y), do.points=FALSE)
> curve(pexp, 0, max(y), add=TRUE, col="grey50")
```

Comme nous le verrons plus tard, la variable Y est de loi exponentielle de paramètre 1, voir la figure 1.2.

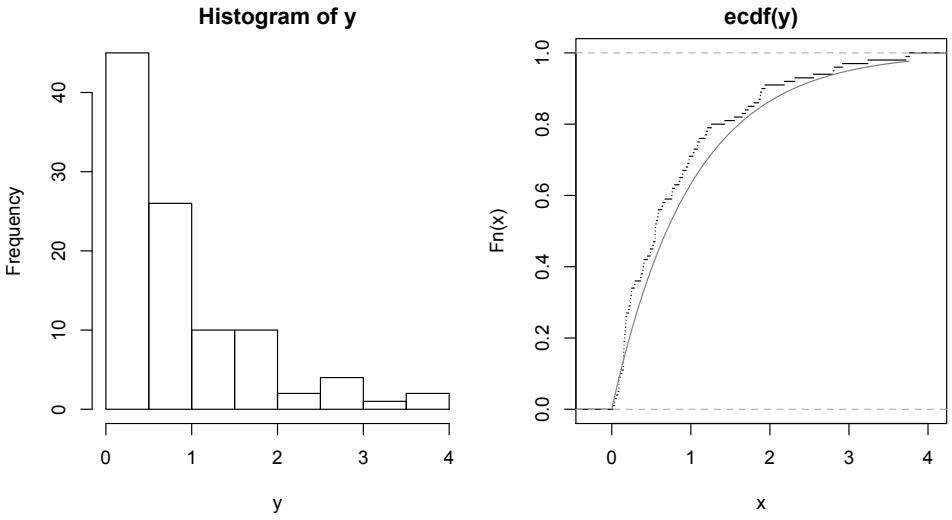


FIGURE 1.2 – Transformée logarithmique.

Nous présentons maintenant deux grandes familles de lois basées sur les transformées puissance et ratio, respectivement la famille gamma transformée et béta transformée.

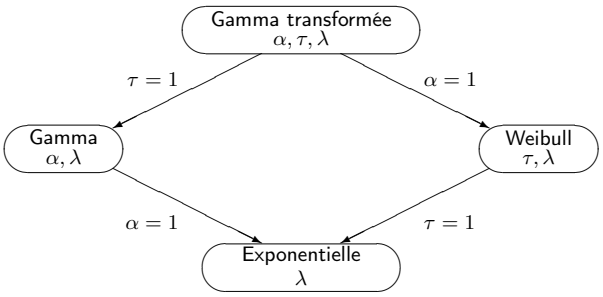


FIGURE 1.3 – Relations entre les membres de la famille gamma transformée

Gamma transformée Les trois lois de la figure **??** n'étant pas à queue de distribution épaisse, on utilise très souvent d'autres lois pour modéliser le montant des sinistres élevés. La famille gamma transformée est une extension de la famille gamma obtenue par transformation d'une variable aléatoire gamma. Soit $X \sim G(\alpha, 1)$ et

$$Y = X^{1/\tau}/\lambda,$$

alors Y suit une loi gamma transformée $GT(\alpha, \tau, \lambda)$ pour $\tau > 0$. Elle a pour densité et fonction de répartition

$$f_Y(y) = \frac{\lambda^{\tau\alpha}}{\Gamma(\alpha)} \tau y^{\alpha\tau - 1} e^{-(\lambda y)^\tau}, \quad \text{et} \quad F_Y(y) = \Gamma(\alpha, (\lambda y)^\tau)/\Gamma(\alpha).$$

Lorsque $\alpha\tau \leq 1$, le mode de la distribution est en $y = 0$. Lorsque $\alpha\tau > 1$, le mode de la densité est alors en $y > 0$.

Notons que pour $\alpha = 1$, on retrouve la loi de Weibull et pour $\tau = 1$ la loi gamma. Ces relations sont illutrées à la figure 1.3. Si $\tau < 0$ dans la transformation de X en posant $\tau^\star = -\tau$, alors on obtient

$$f_Y(y) = \frac{\tau^\star e^{-(\lambda y)^{-\tau^\star}}}{\lambda^{\tau^\star\alpha} y^{\alpha\tau^\star + 1} \Gamma(\alpha)}, \quad \text{et} \quad F_Y(y) = 1 - \Gamma(\alpha, (\lambda y)^{-\tau^\star})/\Gamma(\alpha).$$

Béta transformée La loi béta (de première espèce) est une variable aléatoire continue sur [0,1] et peut être utilisée quelque fois pour modéliser des taux de destruction. Néanmoins c'est surtout sa transformée logit $\frac{X}{1-X}$ à valeurs dans $\mathbb{R}_+$ qui est utilisé pour modéliser le montant des sinistres. Cette loi est appélée loi béta de seconde espèce.

Soit X une variable de loi Béta 1 $\beta_1(a, b)$. Sa densité est donnée par

$$f_X(x) = \frac{x^{a-1}(1-x)^{b-1}}{\beta(a, b)},$$

où $x \in [0, 1]$, $a, b > 0$ et $\beta(.,.)$ est la fonction béta, au sens où $\beta(a, b) = \Gamma(a)\Gamma(b)/\Gamma(a + b)$. Très logiquement sa fonction de répartition s'exprime en fonction d'une fonction béta incomplète $\beta(a, b, .)$

$$F_X(x) = \frac{\beta(a, b, x)}{\beta(a, b)}.$$

On en déduit que la variable $Z = \frac{X}{1-X}$ a pour densité

$$f_Z(x) = \frac{x^{a-1}}{\beta(a, b)(1 + x)^{a+b}}.$$

En appliquant deux transformations de plus, $Y = \theta\left(\frac{X}{1-X}\right)^{1/\gamma}$ a pour densité une loi béta transformée

$$f_Y(y) = \frac{1}{\beta(a,b)} \frac{\gamma(y/\theta)^{\gamma\tau}}{y(1+(y/\theta)^{\gamma})^{\alpha+\tau}}.$$

Sa fonction de réparatition s'exprime par

$$F_Y(y) = \frac{\beta(a,b,\frac{v}{1+v})}{\beta(a,b)}, \quad \text{avec} \quad v = (y/\theta)^{\gamma}.$$

La famille bêta transformée compte plusieurs membres dont, entre autres : la loi de Burr(b,γ,θ) lorsque $a=1$, la loi de Pareto généralisée(b,a,θ) lorsque $\gamma=1$, la loi de Pareto (b,θ) lorsque $\gamma=a=1$. Ces relations sont illustrées à la figure 1.4.

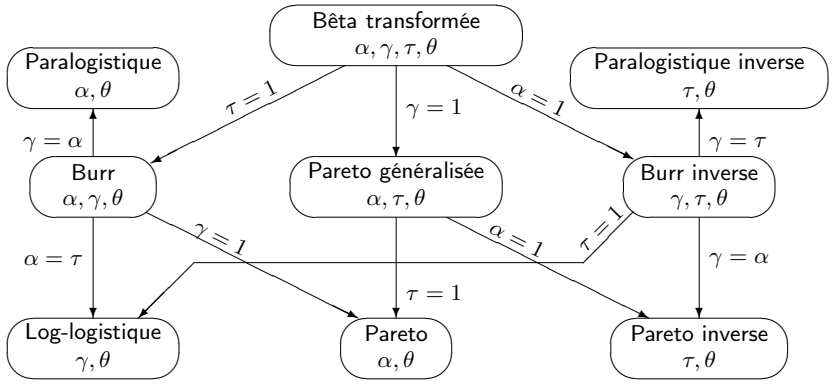
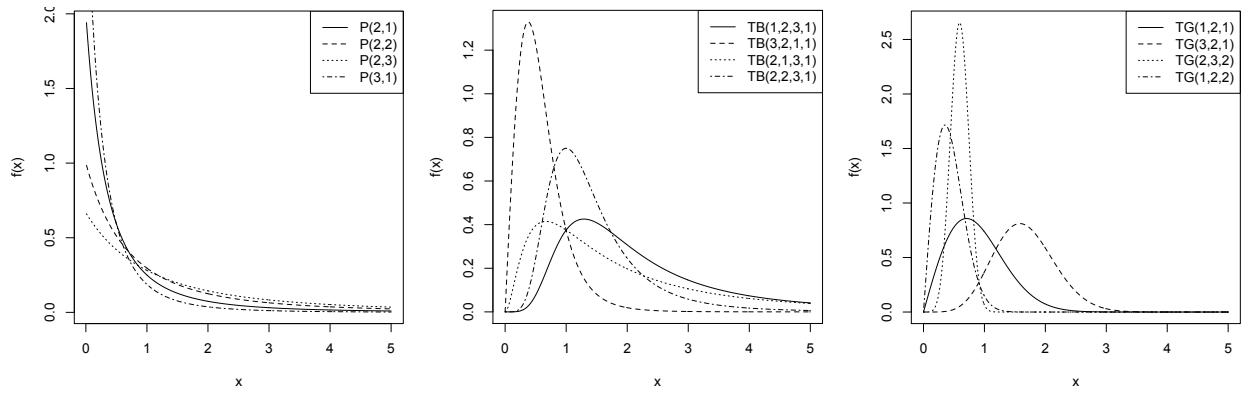


FIGURE 1.4 – Relations entre les membres de la famille béta transformée

Comparaison de lois actuarielles

La figure 1.5 trace la densité de trois grandes lois utilisées en actuariat non vie, à savoir la loi de Pareto, la Béta transformée et la gamma transformée.



(a) Loi Pareto (b) Loi Béta transformée (c) Loi gamma transformée

FIGURE 1.5 – Densités de lois actuarielles

Le tableau 1.1 présentait les lois de base de R. Dans le tableau 1.2, on trouve la liste de lois très spécifiques et très adaptées à l'actuariat non-vie, proposées dans le package **actuar**,

(Dutang et al. 2008). Il est composé de colonnes comportant le nom de la famille de lois, le nom de la loi de probabilité et de la racine de la fonction R correspondante.

Famille	Lois de probabilité	Racine
Transformed beta	Transformed beta	`trbeta`
	Burr	`burr`
	Loglogistic	`llogis`
	Paralogistic	`paralogis`
	Generalized Pareto [1]	`genpareto`
	Pareto	`pareto`
	Inverse Burr	`invburr`
	Inverse Pareto	`invpareto`
	Inverse paralogistic	`invparalogis`
Transformed gamma	Transformed gamma	`trgamma`
	Inverse transformed gamma	`invtrgamma`
	Inverse gamma	`invgamma`
	Inverse Weibull	`invweibull`
	Inverse exponential	`invexp`
Other	Loggamma	`lgamma`
	Single parameter Pareto	`pareto1`
	Generalized beta	`genbeta`

TABLE 1.2 – Loi implémentées dans **actuar**

1.1.2 Les lois discrètes

Considérons maintenant une variable aléatoire X que l'on associera à un comptage. On caractérisera ces variables discrètes par leur probabilité élémentaire, ou leur fonction de masse de probabilité. Les 3 lois usuelles discrètes sont :

- la loi binomiale de fonction de probabilité `dbinom` donnée par

$$\mathbb{P}(X = k) = \binom{n}{k} p^k (1-p)^{n-k},$$

où $\binom{n}{k}$ est le nombre de combinaison de k éléments parmi n (i.e. $\frac{n!}{k!(n-k)!}$), $k \in \mathbb{N}$ et $0 \leq p \leq 1$ la probabilité de "succès". Cette loi vérifie $\mathbb{E}X > \mathbb{V}[X]$.

- la loi de Poisson de fonction de probabilité `dpois` donnée par

$$\mathbb{P}(X = k) = \frac{\lambda^k}{k!} e^{-\lambda},$$

où $\lambda > 0$ est le paramètre de forme et $k \in \mathbb{N}$. Cette loi vérifie $\mathbb{E}X = \mathbb{V}[X]$.

- la loi binomiale négative de fonction de probabilité `dnbinom` donnée par

$$\mathbb{P}(X = k) = \binom{m+k-1}{k} p^m (1-p)^k,$$

où $k \in \mathbb{N}$ et $p \in [0, 1]$. Lorsque $m = 1$, on trouve la loi géométrique de paramètre p. Cette loi vérifie $\mathbb{E}X < \mathrm{Var}X$.

1. Attention ceci ne correspond à la loi de Pareto généralisée de la théorie des valeurs extrêmes.

Ces 3 lois permettent de modéliser une majorité des variables discrètes. La figure 1.6 compare les lois discrètes à espérance égale ($\mathbb{E}(X) = 5$).

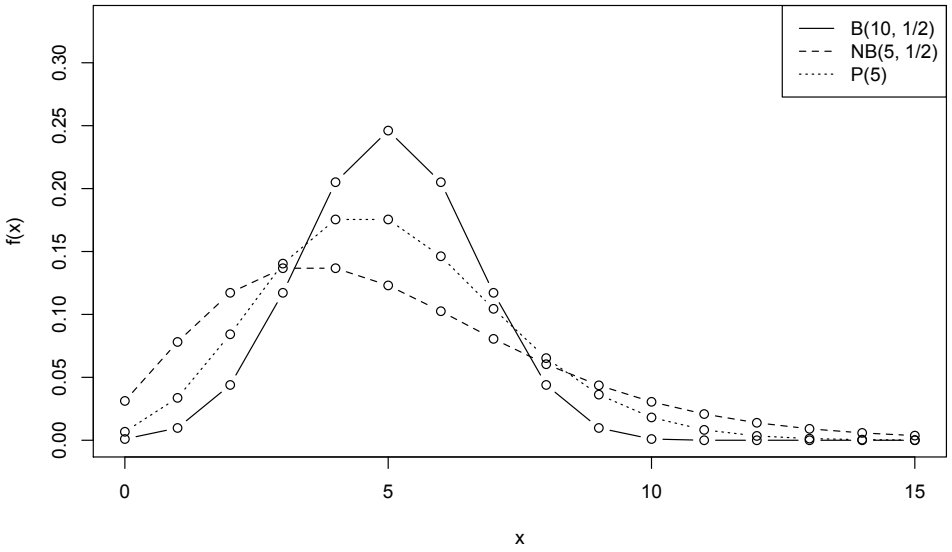


FIGURE 1.6 – Fonctions de masse de probabilité des lois discrètes usuelles

En fait, ces trois lois font partie de la famille dite de Sundt $(a, b, 0)$, dont les probabilités élémentaires vérifient

$$\frac{\mathbb{P}(X = k+1)}{\mathbb{P}(X = k)} = a + \frac{b}{k},$$

pour $k \geq 0$ et a, b deux paramètres. On retrouve la loi binomiale avec

$$a = \frac{-p}{1-p} \quad \text{et} \quad b = \frac{p(n+1)}{1-p},$$

la loi de Poisson avec

$$a = 0 \quad \text{et} \quad b = \lambda,$$

et enfin la loi Binomiale Négative avec

$$a = 1 - p \quad \text{et} \quad b = (1-p)(m-1).$$

La famille $(a, b, 0)$ va être utilisée pour les lois composées. De manière plus générale, on peut définir la famille (a, b, n) en tronquant la variable aléatoire pour les valeurs plus petites ou égales à $n-1$. C'est à dire, on a

$$\mathbb{P}(X = k) = \begin{cases} 0 & \text{si} \quad k < n \\ \mathbb{P}(X = k-1)\left(a + \dfrac{b}{k}\right) & \text{si} \quad k \geq n \end{cases},$$

De plus, on peut parfois appliquer des transformations à ces lois usuelles comme supprimer la valeur en $k = 0$ ou en modifiant la valeur en $k = 0$. Pour obtenir les lois zéro-tronqués, il suffit de considérer la famille $(a, b, 1)$.

Les versions zéros-modifiées s'obtiennent à partir des versions zéro-tronquées $(a, b, 1)$. Notons X^M la variable zéro-modifiée obtenu par d'une variable X. On définit les probabilités élémentaire

par

$$\mathbb{P}(X^M = k) = \begin{cases} p_0^M & \text{si } \ k = 0 \\ \dfrac{1 - p_0^M}{1 - \mathbb{P}(X = 0)} \mathbb{P}(X = k) & \text{sinon} \end{cases},$$

où p_0^M est la probabilité en 0 et X est la variable aléatoire sous-jacente que l'on considère, e.g. la loi de Poisson $\mathbb{P}(\tilde{X} = k) = \dfrac{\lambda^k}{k!} e^{-\lambda}$.

Des packages implémentent ces lois usuelles, néanmoins il est facile de les construire à la main ! Créons la fonction de masse de probabilite

```
> dpoism <- function(x, p0, ...)
+        ifelse(x == 0, p0, (1-p0)/(1-dpois(0, ...))*dpois(x, ...))
```

Ensuite, il est facile d'afficher cette fonction en appelant le code suivant :

```
> x <- 0:10
> y <- dpoism(x, dpois(0, 1), 1)
> y2 <- dpoism(x, 1/2, 1)
> y3 <- dpoism(x, dpois(0, 2), 2)
> y4 <- dpoism(x, 1/3, 2)
> leg.txt <- c("P(1)","P-0M(1)", "P(2)","P-0M(2)")
> plot(x, y, xlab="x", ylab="f(x)", ylim=range(y, y2, y3, y4[-(1:15)]),
+ col="black", type="b")
> lines(x, y2, col="blue", type="b")
> lines(x, y3, col="red", type="b")
> lines(x, y4, col="green", type="b")
> legend("topright",leg=leg.txt, col=c("black","blue","red","green"),lty=1)
```

Sur la figure 1.7, on peut observer la décrochage en 0 pour la loi de Poisson zéro-modifiée de paramètre 1 et 2.

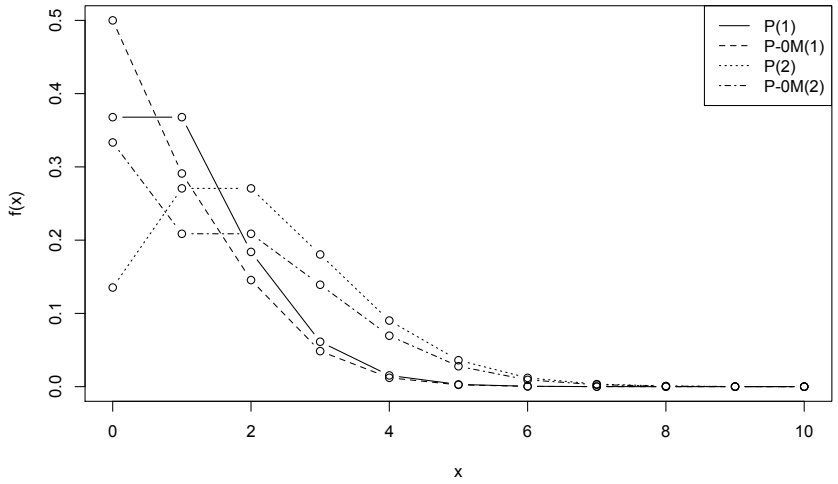


FIGURE 1.7 – La loi de Poisson zéro-modifiée.

Enfin, des lois plus paramétrées comme la loi hypergéométrique peuvent parfois être utilisées. Pour aller plus loin sur les lois discrètes, nous renvoyons le lecteur intéressé vers Johnson et al. (2005).

1.1.3 Les lois mixtes

Zéro-modifié

Les lois mixtes sont des lois de probabilité qui sont ni entièrement continues ni entièrement discrètes. Ce sont généralement des lois continues auxquelles on a rajouté des masses de probabilités. Typiquement pour modéliser le montant des sinistres, il peut être intéressant de considérer une masse en zéro (i.e. aucun sinistre) et une loi continue au delà de zéro (i.e. occurence d'un sinistre). Dans la littérature, on appelle ces lois du nom de la loi continue complétée de zéro-modifié.

Par exemple, la loi exponentielle zéro-modifiée a pour fonction de répartition

$$F_X(x) = q + (1-q)(1-e^{-\lambda x})$$

pour $x \geq 0$. Cette loi ne possède pas densité puisqu'il y a une discontinuité en zéro : $\mathbb{P}(X=0) = q \neq \lim_{x \to 0^-} F_X(x) = 0$. Néanmoins, la variable aléatoire X conditionellement à $X > 0$ possède la densité de la loi exponentielle.

Cette approche peut aussi s'appliquer pour les variables discrètes et la modélisation du nombre de sinistre. Ainsi on peut choisir d'utiliser la loi Poisson zéro-modifiée (ou à inflation de zéros, qui sera utilisée dans la Section 2.4.6), puisqu'il parait logique que la probabilité de n'avoir aucun sinistre soit bien différente et plus élevée que d'avoir des sinistres (voir la section précédente).

MBBEFD

Un exemple plus intéressant de loi mixte sont les lois MBBEFD, introduites et popularisées Bernegger (1997). MBBEFD est un sigle pour Maxwell-Boltzmann, Bore-Einstein et Fermi-Dirac. La loi MBBEFD est caractérisée par la fonction de répartition suivante

$$F(x) = \begin{cases} a\left(\dfrac{a+1}{a+b^x} - 1\right) & \text{if } \ 0 \leq x < 1 \\ 1 & \text{if } \ x \geq 1 \end{cases},$$

pour $x \in \mathbb{R}_+$. C'est donc un mélange d'une loi continue sur $]0,1[$ et d'une masse de Dirac en 1. On a en effet une masse de probabilité en 1 :

$$p = 1 - F(1) = \frac{(a+1)b}{a+b}.$$

Les paramètres (a,b) sont définis pour un grand choix d'intervalles : $]-1,0[\times]1,+\infty[$ et $]-\infty,-1[\cup]0,+\infty[\times]0,1[$. La forme de la fonction de répartition F a les propriétés suivantes :

- pour $(a,b) \in I_1 =]-1,0[\times]1,+\infty[$, F est concave,
- pour $(a,b) \in I_2 =]-\infty,-1[\times]0,1[$, F est concave,
- pour $(a,b) \in I_3 =]0,b[\times]0,1[$, F est concave,
- pour $(a,b) \in I_4 = [b,1[\times]0,1[$, F est convexe puis concave,
- pour $(a,b) \in I_4 = [1,+\infty[\times]0,1[$, F est convexe.

On peut exprimer la fonction de masse de probabilités à l'aide de la fonction de Dirac δ. On obtient l'expression suivante :

$$f(x) = \frac{-a(a+1)b^x \ln(b)}{(a+b^x)^2} \mathbb{1}_{]0,1[}(x) + p\delta_1(x).$$

Actuellement, il n'y a pas de packages concernant cette loi. Mais, il est facile de l'implémenter.

```
> dMBBEFD <- function(x, a, b)
+         -a * (a+1) * b^x * log(b) / (a + b^x)^2 + (a+1) * b / (a+b) * (x == 1)
> pMBBEFD <- function(x, a, b)
+         a * ( (a+1) / (a + b^x) - 1) * (x < 1) + 1 * (x >= 1)
```

La loi MBBEFD a été introduite pour la modélisation des courbes d'exposition et des taux de destruction pour les traités de réassurance non proportionnelles. Nous renvoyons le lecteur intéressé vers Bernegger (1997).

Les lois composées

Nous considérons la variable S défini par

$$S = \sum_{i=1}^{N} X_i,$$

où X_i sont des variables aléatoires i.i.d. et avec comme convention que la somme est nulle si $N = 0$. En pratique S représente la charge totale de sinistre et X_i des montants individuels de sinistres. En conditionnant par rapport au nombre de sinitres, on a

$$F_S(x) = \sum_{n=0}^{\infty} \mathbb{P}(S \leq x | N = n)\mathbb{P}(N = n) = \sum_{n=0}^{\infty} F_X^{*n}(x) p_n, \tag{1.1}$$

où $F_X(x) = \mathbb{P}(X \leq x$ est la fonction de répartition (commune) des $X_1, \ldots, X_n$, $p_n = \mathbb{P}(N = n)$ et $F_X^{*n}(x) = \mathbb{P}(X_1 + \cdots + X_n \leq x)$ est le produit de convolution d'ordre n de $F_X(\cdot)$.

Il existe différentes stratégies pour calculer la loi de la somme : une formule exacte si les variables aléatoires sont discrètes (algorithme de Panjer ou FFT), des approximations (normale ou normal-power ou gamma) et une approche par simulation. Toutes ces méthodes sont disponibles dans la fonction `aggregateDist` du package **actuar**.

Si X est une variable aléatoire discrète sur $\mathbb{N}$ alors l'équation 1.1 devient

$$F_X^{*k}(x) = \begin{cases} \mathbb{1}_{x \geq 0}, & k = 0 \\ F_X(x), & k = 1 \\ \sum_{y=0}^{x} F_X^{*(k-1)}(x-y) P(X = y), & k = 2, 3, \ldots \end{cases} \tag{1.2}$$

Le calcul récursif peut se faire avec l'algorithme de Panjer (1981) si la loi de la variable aléatoire N appartient à la famille $(a, b, 0)$ ou $(a, b, 1)$. La formula de récursion est la suivante

$$\mathbb{P}(S = x) = \frac{[\mathbb{P}(X = 1) - (a+b)\mathbb{P}(X = 0)]\mathbb{P}(X = x) + \sum_{y=1}^{x \wedge m} (a + by/x)\mathbb{P}(X = y)\mathbb{P}(S = x - y)}{1 - a\mathbb{P}(X = 0)},$$

où la récursion démarre par $\mathbb{P}(S = 0) = G_N(\mathbb{P}(X = 0))$ et G_N est la fonction génératrice des probabilités, i.e. $G_N(z) = \mathbb{E}(z^N)$. La récursion s'arrête lorsque les probabilités sont arbitrairement proche de 0.

La formule est implémentée en C pour diminuer le temps de calcul. Ne connaissant par avance la valeur x telle que $P(S = x) \approx 0$, on démarre avec une taille fixe du tableau contenant les probabilités élémentaires de S, que l'on double à chaque fois que c'est nécessaire.

En pratique, les montants de sinistres sont rarements discrets, mais on peut discrétiser la fonction de répartition pour retomber dans le cadre d'application de l'algorithme de Panjer.

Comme pour l'algorithme de Panjer, on suppose que X est à valeurs discrètes. La convolée d'ordre n $F_X^{*n}(x)$ de la fonction de répartition de X, utilisée dans l'équation 1.1, peut se calculer à l'aide de la transformée de Fourrier discrète. Cette transformée est implémentée par l'algorithme FFT (Fast Fourrier Transform). Dans R, la fonction `convolve` réalise ceci.

Différentes approximations sont possibles pour évaluer la fonction de répartition de la variable S. Nous présentons ci-dessous les plus connues.

Approximation normal consiste à calibrer une loi normale sur S par la méthode des moments :

$$F_S(x) \approx \Phi\left(\frac{x-\mu_S}{\sigma_S}\right),$$

où $\mu_S = E(S)$ et $\sigma_S^2 = Var(S)$. Nul ne va sans dire que cette approximation est plutot brutale et peu conservatrice sur la queue de distribution.

Approximation normale-puissance considère la formule suivante

$$F_S(x) \approx \Phi\left(-\frac{3}{\gamma_S} + \sqrt{\frac{9}{\gamma_S^2} + 1 + \frac{6}{\gamma_S}\frac{x-\mu_S}{\sigma_S}}\right),$$

où $\gamma_S = E((S-\mu_S)^3)/\sigma_S^{3/2}$. L'approximation est valide pour $x > \mu_S$ seulement et est raisonablement bonne pour $\gamma_S < 1$, voir Daykin et al. (n.d.) pour plus de détails.

Simulations L'approche par simulation est simple, cela consiste à simuler un certain nombre de réalisations. La fonction `aggregateDist` est même plus général que le modèle décrit plus, car elle accepte un modèle hiérarchique pour la fréquence et la sévérité des sinistres.

Exemple Présentons maintenant un exemple où le montant de sinistre est de loi gamma et le nombre de sinistre suit une loi de Poisson.

Considérons le cas où N suit une loi de Poisson de paramètre λ et que les variables X_i sont i.i.d. de loi gamma de moyenne $\mathcal{G}(\alpha, \beta)$. Dans ce cas, les moments ont pour expression

$$\mathbb{E}[S] = \mathbb{E}[N] \cdot \mathbb{E}[X] = \lambda\alpha/\beta,$$

$$\mathbb{V}[S] = \mathbb{V}[N]\mathbb{E}[X]^2 + \mathbb{E}[N]\,\mathbb{V}[X] = \lambda\left(\frac{\alpha+\alpha(\alpha+1)}{\beta^2}\right),$$

et

$$\gamma_S = \frac{\mathbb{E}\left[X^3\right]}{\sqrt{\lambda}\mathbb{E}[X^2]^{3/2}} = \frac{1}{\sqrt{\lambda}} \cdot \frac{\alpha}{\sqrt{\alpha(\alpha+1)}}.$$

Comme précisé ci-dessus, l'algorithme de Panjer ne fonctionne qu'avec des lois discrètes, il convient donc de discrétise la fonction de répartition des montants de sinistre et ensuite d'utiliser la fonction `aggregateDist`.

```
> fx.u <- discretize(pgamma(x, 2, 1), from = 0, to = 22, step = 0.5,
+                    method = "upper")
> Fs.u <- aggregateDist("recursive", model.freq = "poisson",
+                       model.sev = fx.u, lambda = 10, x.scale = 0.5)
```

```
> fx.l <- discretize(pgamma(x, 2, 1), from = 0, to = 22, step = 0.5,
+                    method = "lower")
> Fs.l <- aggregateDist("recursive", model.freq = "poisson",
+                       model.sev = fx.l, lambda = 10, x.scale = 0.5)
> Fs.n <- aggregateDist("normal", moments = c(20, 60))
> Fs.np <- aggregateDist("npower", moments = c(20, 60, 4/sqrt(60)))
> Fs.s <- aggregateDist("simulation",
+                       model.freq = expression(y = rpois(10)),
+                       model.sev = expression(y = rgamma(2, 1)),
+                       nb.simul = 10000)
```

Sur la figure 1.8, on a tracé la queue de distribution de la somme agrégée S pour toutes les méthodes. On constate que la méthode par simulation, l'algorithme de Panjer couplé à une discrétisation sans biais et l'approximation normale-puissance sont toutes très proche. L'approximation normale et l'algorithme de Panjer couplé à une discrétisation supérieure sur-estiment la fonction de répartition tandis que l'algorithme de Panjer couplé à une discrétisation inférieure la sous-estime.

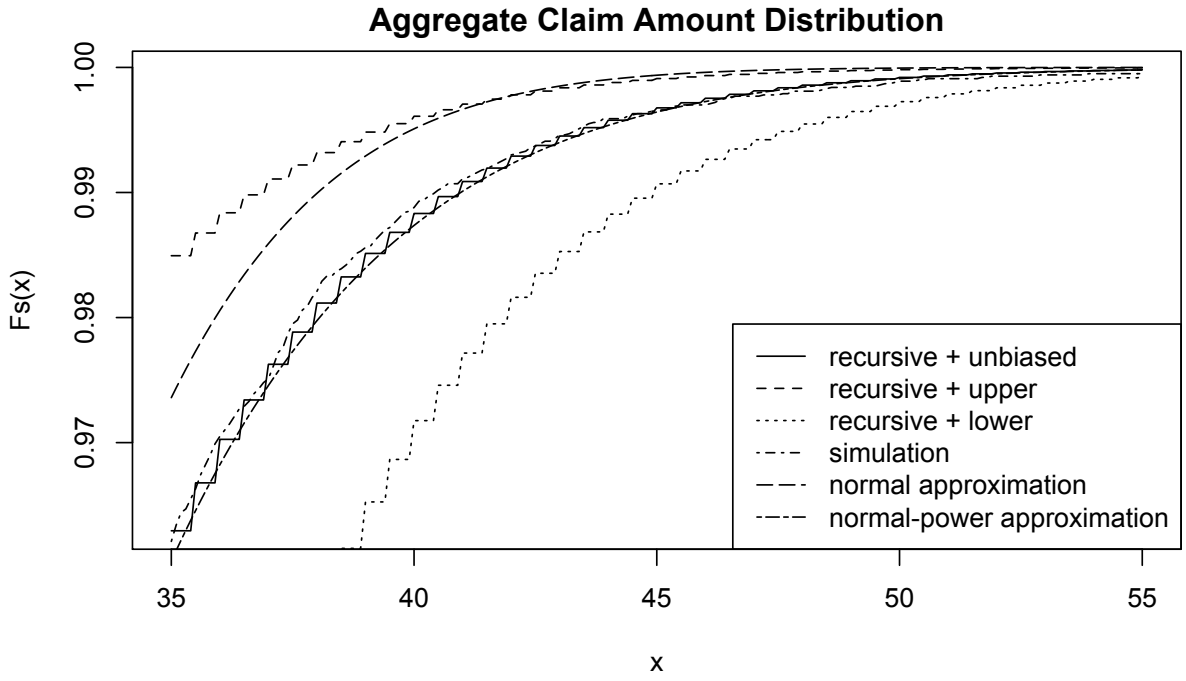


FIGURE 1.8 – Comparaison de méthodes

1.1.4 Les lois multivariées et les copules

Nous abordons dans cette sous-section très rapidement les lois multivariés, c'est à dire la loi d'un vecteur aléatoire. De plus, il parait difficile de ne pas parler des copules tant leur utilisation en actuariat n'a cessé d'augmenter ses dernières années, voir, par exemple, Frees & Valdez (1998), Embrechts et al. (2001), Frees & Wang (2006). Ainsi, nous présentons aussi rapidement les copules dans cette sous-section.

Les lois multivariées

Notons d la dimension. Comme dans le cas univarié, la loi du vecteur $\boldsymbol{X} = (X_1, \ldots, X_d)$ peut se décrire par la fonction de répartition

$$F_{\boldsymbol{X}}(\boldsymbol{x}) = \mathbb{P}(X_1 \leq x_1, \ldots, X_d \leq x_d).$$

De plus, si $\boldsymbol{X}$ est à valeurs discrètes, on peut utiliser la fonction de masse de probabilité $\mathbb{P}(\boldsymbol{X} = \boldsymbol{x})$. Si $\boldsymbol{X}$ est à support continu, on peut définir une fonction de densité $f_X(x)$. Nous présentons ci-dessous deux grandes familles de lois : la loi normale et la loi de Pareto.

Loi normale Notons $\boldsymbol{\mu} \in \mathbb{R}^d$ et $\Sigma M_d(\mathbb{R})$ les paramètres. La densité est donnée par

$$f_{\boldsymbol{X}}(\boldsymbol{x}) = \frac{1}{(2\pi)^{\frac{d}{2}}|\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})'\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu})}$$

pour $\boldsymbol{x} \in \mathbb{R}^d$ et $|.|$ désignant le déterminant. $\boldsymbol{\mu}$ est la moyenne du vecteur aléatoire et Σ sa matrice de variance-covariance. La fonction de répartition a là aussi pas d'expression explicite. Par définition, on a

$$F_{\boldsymbol{X}}(\boldsymbol{x}) = \int_{-\infty}^{x_1} \ldots \int_{-\infty}^{x_d} \frac{1}{(2\pi)^{\frac{d}{2}}|\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})'\Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu})} dx_1 \ldots dx_d$$

Loi Pareto Dans le cas univarié, la loi de Pareto se caracterise très souvent par sa queue de distribution à décroissance polynomiale. Considérons la loi de Pareto IV, on a

$$P(X > x) = \left(1 + \left(\frac{x-\mu}{\sigma}\right)^{\frac{1}{\gamma}}\right)^{-\alpha}.$$

L'extension multivariée est donnée dans Arnold (1983) :

$$P(\boldsymbol{X} > \boldsymbol{x}) = \left(1 + \sum_{i=1}^{d} \left(\frac{x_i - \mu_i}{\sigma_i}\right)^{\frac{1}{\gamma_i}}\right)^{-\alpha}$$

pour $\boldsymbol{x} \geq \boldsymbol{\mu}$, $\boldsymbol{\sigma}$ le vecteur des paramètres d'échelle, $\boldsymbol{\gamma}$ ceux de forme et $\boldsymbol{\alpha}$ ceux de décroissance. Il est possible d'obtenir la densité en dérivant par rapport à chaque variable. Pour plus de détails, voir Arnold (1983) et sa version plus récente Arnold (2008).

Autres : D'autres lois multivariées existent, voir Kotz et al. (1994*a*,*b*) pour les lois continues et Johnson et al. (1997) pour les lois discrètes.

Les copules

L'utilisation des copules permet de construire de lois multivariées de manière générique en spécifiant des lois marginales et une structure de dépendance. Les copules ont été introduits par Sklar (1959) dans le but de caractériser un vecteur $\boldsymbol{X} = (X_1, \ldots, X_d)$ ayant des lois marginales (i.e. $P(X_i \leq x_i)$) données. Par le théorème de Sklar, on a que pour toutes fonctions de répartitions multivariées F à marginales $F_1, \ldots, F_d$, il existe une fonction copule C telle que

$$F(x_1, \ldots, x_d) = P(X_1 \leq x_1, \ldots, X_d \leq x_d) = C(F_1(x_1), \ldots, F_d(x_d)).$$

La fonction multivariée $C : [0,1]^d \mapsto [0,1]$ appelée doit remplir les conditions suivantes pour que l'expression reste une fonction de répartition : condition au bord $C(\dots, u_{i-1}, 0, u_{i+1}, \dots) = 0$, $C(\dots, 1, u_i, 1, \dots) = u_i$ et d-croissance. Ces contraintes garantissent que C est une fonction de répartition multivariée dont les lois marginales sont uniformes sur $[0,1]$.

Cette représentation a l'énorme avantage de séparer les marginales F_i de la structure "interne" de dépendance C. La copule la plus simple est la copule indépendance

$$C(u_1, \dots, u_d) = \prod_{i=1}^{d} u_i = C^{\perp}(u_1, \dots, u_d) = \Pi(u_1, \dots, u_d).$$

Deux autres fonctions tout aussi importantes sont les bornes de Fréchet :

$$M(u_1, \dots, u_d) = \min(u_1, \dots, u_d) \quad \text{et} \quad W(u_1, \dots, u_d) = \left(\sum_{i=1}^{d} u_i - (d-1) \right)_+ .$$

La première est une copule quelle que soit la dimension d, alors que la seconde n'est une copule que si $d = 2$.Elles sont telles que toute copule C vérifie $W \leq C \leq M$. Il existe plusieurs familles de copules possédant des propriétés intéressantes pour certaines applications. Nous présentons les deux plus courantes.

Famille elliptique Les copules elliptiques se construisent à partir des lois ellyptiques (dont font partie la loi Gaussienne, ou la loi de Student). Notons E_d (resp. E_1) une fonction de répartition de dimension d (resp. 1) d'une telle famille. Une copule elliptique se définit par

$$C(u_1, \dots, u_d) = E_d(E_1^{-1}(u_1), \dots, E_1^{-1}(u_d)).$$

On retrouvera comme cas particulier est la copule Gaussienne, pour laquelle $E_d = F_{\mathcal{N}(0,\Sigma)}$ et $E_1 = F_{\mathcal{N}(0,1)}$. On trouve aussi dans cette famille, la copule de Student, voir Embrechts et al. (2001).

Famille archimédienne Une autre grande famille de copules, popularisée grace aux livres de Nelsen (1999, 2006), est la famille archimédienne. Les copules sont construites de la manière suivante :

$$C(u_1, \dots, u_d) = \phi^{-1} \left(\sum_{i=1}^{d} \phi(u_i) \right),$$

où $\phi : [0,1] \mapsto [0, \infty]$ est une fonction infiniment continue, complètement monotone et inversible (des conditions plus faibles peuvent être demandé si la dimension d est fixée : par exemple en dimension 2, ϕ doit être décroissante et convexe). Pour plus de détails sur cette construction, nous renvoyons le lecteur vers le théorème 2.1 de Marshall & Olkin (1988). Les trois copules les plus connues sont celles de Clayton $\phi(t) = t^{-\alpha} - 1$, celle de Gumbel $\phi(t) = (-\log(t))^{-\alpha}$ et celle de Frank $\phi(t) = -\log\left(\dfrac{e^{\alpha t} - 1}{e^{\alpha} - 1} \right)$.

Famille des copules extrêmes Ce sont les copules C qui vérifient la propriété suivante dite de max-stabilité

$$C(u_1, \dots, u_d) = \left(C(u_1^{1/k}, \dots, u_d^{1/k}) \right)^k,$$

pour tout $k > 0$. Cette propriété est issue de la théorie des valeurs extrêmes (si $k \in \mathbb{N}$, la copule de droite est la copule du vecteur $(\max\{X_{1,1}, \ldots, X_{1,k}\}, \ldots, \max\{X_{d,1}, \ldots, X_{d,k}\})$ pour des vecteurs $(X_1, \ldots, X_d)$ i.i.d. de copule sous-jacente C).

Parmi les lois des copules extrêmes, nous ne présentons que la copule de Gumbel, mais d'autres copules existens, notamment la copule de Galambos, Huler-Reiss, Marshall-Olkin, ...La copule de Gumbel qui est aussi archimédienne est donnée par

$$C(u_1, \ldots, u_n) = \exp\left(-\left[\sum_{i=1}^{n} (-\ln u_i)^{\alpha}\right]^{1/\alpha}\right),$$

où $\alpha > 0$. C'est la copule que nous allons utilisé dans la section 1.4.6. Il est important de noter que la copule gaussienne n'appartient **pas** à la famille des copules extrêmes.

1.2 Estimation non-paramétrique

Pour le reste du chapitre, on pose que X représente la variable aléatoire (continue) du montant d'un sinistre avec fonction de répartition $F(x)$. L'assureur dispose d'observations $X_1, \ldots, X_n$ (sous forme individuelles ou groupées) que l'on suppose former un échantillon aléatoire de la variable aléatoire X. Dans cette section, on va chercher à construire, à partir des données, des estimateurs de $F(x)$, de la fonction de densité de probabilité $f(x)$ et de certaines quantités liées sans faire d'hypothèse quant à la distribution de X. Cette approche sera dite non paramétrique. Elle a comme avantages d'être flexible et de bien prendre en compte la disparité des données. De plus, elle peut être très précise lorsque le nombre de données est grand. Par contre, elle est souvent moins efficace qu'une approche paramétrique et l'inférence statistique qui en résulte est plus compliquée.

1.2.1 Fonctions de répartition et densité empiriques

La première étape d'un processus de modélisation consiste souvent à tracer des graphiques tels que ceux présentés à la figure 1.2 permettant de déterminer globalement la distribution des données. Les fonctions sous-jacentes à ces graphiques constituent en fait des estimateurs de la fonction de répartition et de la fonction de densité de probabilité.

Dans le cas de données individuelles, on construit la fonction de répartition empirique, $F_n(x)$, et la fonction de masse de probabilité empirique, $f_n(x)$, en attribuant à chacune des n données un poids de $1/n$. On a donc

$$F_n(x) = \frac{1}{n} \sum_{j=1}^{n} \mathbb{1}_{\{X_j \leq x\}},$$

et par "différentiation",

$$f_n(x) = \frac{1}{n} \sum_{j=1}^{n} \mathbb{1}_{\{X_j = x\}},$$

où $\mathbb{1}_A$ est une fonction indicatrice valant 1 lorsque la condition A est vraie, et 0 sinon. Pour la densité empirique, l'estimateur ci-dessus est une somme de masse de Dirac, c'est pourquoi on considère en général un estimateur lissé à l'aide de fonction noyau.

$$f_{K,n}(x) = \frac{1}{nh_n} \sum_{j=1}^{n} K\left(\frac{x - X_j}{h_n}\right),$$

où K est une fonction noyau (fonction positive d'intégrale 1) et h_n une taille de fenêtre. Si on prend un noyau rectangulaire $K(u) = 1/2\mathbb{1}_{[-1,1[}(u)$, alors on obtient un histogramme glissant. D'autres noyaus existent : le noyau gaussien (celui par défaut dans R, via la fonction `density`) se definit par $K(u) = \frac{1}{\sqrt{2\pi}} e^{-\frac{u^2}{2}}$, le noyau d'Epanechnikov $K(u) = \frac{3}{4\sqrt{5}}(1 - u^2/5)\mathbb{1}_{[-\sqrt{5},\sqrt{5}[}(u)$.

Le théorème de Glivencko-Cantelli assure la convergence presque sûre de ces deux estimateurs. La fonction `ecdf` de R retourne une fonction pour calculer $F_n(x)$ en tout x, tandis que la fonction `density` permet de calculer $f_{K,n}$ sur une grille.

Aux fins d'illustration, nous allons utiliser les données `dental` distribuées avec **actuar**. Il s'agit de 10 montants de réclamation en assurance dentaire avec une franchise de 50 :

```
> data(dental, package = "actuar")
> dental
 [1]  141   16   46   40  351  259  317 1511  107  567
```

On définit une fonction R pour calculer $F_n(x)$ avec

```
> Fn <- ecdf(dental)
```

Les méthodes de `summary` et `knots` fournissent de l'information plus détaillées sur l'objet :

```
> summary(Fn)
Empirical CDF:          10 unique values with summary
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   16.0    61.2   200.0   336.0   342.0  1510.0
> knots(Fn)
 [1]   16   40   46  107  141  259  317  351  567 1511
```

On peut évaluer la fonction de répartition empirique à ses noeuds ou en tout autre point :

```
> Fn(knots(Fn))
 [1] 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9 1.0
> Fn(c(20, 150, 1000))
[1] 0.1 0.5 0.9
```

Enfin, le graphique de gauche de la figure 1.9 a été tracé tout simplement avec

```
> plot(Fn)
```

Pour la densité empirique, le principe est le même. On se contente ici de l'afficher sur l'histogramme.

```
> hist(dental, prob=TRUE, breaks=c(0, 25, 100, 500, 1000, 2000))
> lines(density(dental), lty=2)
```

Voir le graphique de droite de la figure 1.9.

1.2.2 Quantiles

La fonction quantile d'une variable aléatoire X de fonction de répartition F est définie à l'aide de l'inverse généralisée

$$q_X(p) = \inf_{x \in \mathbb{R}} (F(x) \geq p),$$

aussi noté $F^{-1}(p)$. Hyndman & Fan (1996) propose une approche unifiée pour le calcul des quantiles empiriques où le quantile empirique une combinaison convexe des valeurs observées encadrant le quantile recherché.

Pour des variables continues, le quantile de type 7, celui utilisé par défaut dans R, est une interpolation linéaire entre la $\lfloor (n+1)p \rfloor$ e et la $\lceil (n+1)p \rceil$ e statistique d'ordre :

$$\hat{q}_p = (1 - h)x_{(j)} + hx_{(j+1)}$$

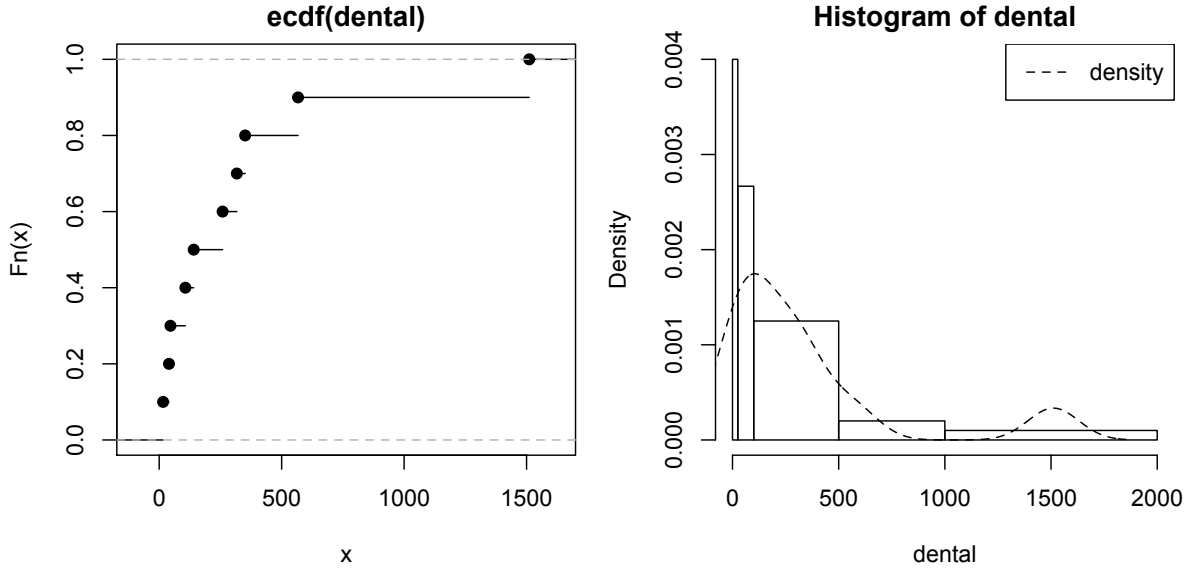


FIGURE 1.9 – Exemple de fonction de répartition empirique (gauche) et d'histogramme (droite) de données individuelles

avec

$$j = \lfloor 1 + (n-1)p \rfloor,$$

$$h = 1 + (n-1)p - j,$$

où $x_{(j)}$ est la j^{e} valeur de l'échantillon trié en ordre croissant, $\lfloor x \rfloor$ est le plus grand entier inférieur ou égal à x, et $\lceil x \rceil$ est le plus petit entier supérieur ou égal à x. Le théorème de Mosteller nous assure la convergence en loi de ces estimateurs.

Pour des variables discrètes, le quantile empirique de type 2 est défini par la moyenne des 2 valeurs les plus proches :

$$\hat{q}_p = \begin{cases} \frac{1}{2}(x_{(j)} + x_{(j+1)}) & \text{si } \ np \in \mathbb{N} \\ x_{(j)} & \text{sinon} \end{cases},$$

avec $j = \lfloor np \rfloor$. Avec l'algorithme ci-dessus, sur les données `dental` on a par exemple les quantiles suivant pour $p = 0.1, 0.5, 0.9$.

```
> quantile(dental, c(0.1, 0.5, 0.9))
  10%   50%   90%
 37.6 200.0 661.4
```

On remarquera que ces valeurs ne correspondent pas avec celles de la fonction de répartition empirique (non lissée) calculées à l'exemple 1.2.1.

1.2.3 Moments

Les moments constituent des mesures des tendances centrales d'une distribution et, par le fait même, de la forme générale de celle-ci. Il n'est donc pas rare de comparer les moments empiriques d'un échantillon aux moments théoriques d'une distribution. On nomme, lorsqu'il existe, moment (ordinaire) d'ordre k de la variable aléatoire X l'espérance de cette dernière

élevée à la puissance k :

$$\mu'_k = \mathbb{E}\left[X^k\right] = \int_0^\infty x^k dF(x),$$

tandis que le moment centré d'ordre k est définit par

$$\mu_k = \mathbb{E}\left[(X - \mathbb{E}[X])^k\right] = \int_0^\infty (x - \mathbb{E}[X])^k dF(x).$$

En particulier, on a $\mathbb{E}[X] = \mu'_1$ et $\mu_1 = 0$.

Pour développer des estimateurs des moments théoriques, il suffit de remplacer la fonction de répartition $F(x)$ par la fonction de répartition empirique $F_n(x)$. Pour un ensemble de données individuelles, on a donc

$$\hat{\mu}'_k = \int_0^\infty x^k dF_n(x) = \frac{1}{n}\sum_{j=1}^n (x_j)^k.$$

Le théorème de Slutsky garantit la convergence en loi des estimateurs empiriques vers leur équivalent théorique. Dans `R`, les premiers moments s'obtiennent à l'aide des fonctions `mean`, `var` ou en calculant explicitement les sommes pour les autres moments.

1.2.4 Espérances limitée et résiduelle

On a déjà définit X, la variable aléatoire du montant d'un sinistre. On définit maintenant $X \wedge u$, la variable aléatoire du montant limité à u :

$$X \wedge u = \min(X, u) = \begin{cases} X, & X < u \\ u, & X \geq u. \end{cases}$$

Ainsi, le moment limité d'ordre k de la variable aléatoire X est :

$$\begin{aligned} \mathbb{E}\left[(X \wedge u)^k\right] &= \int_0^\infty \min(x, u)^k \, dF(x) \\ &= \int_0^u x^k \, dF(x) + u^k(1 - F(u)). \end{aligned} \tag{1.3}$$

Dans la suite, on s'intéressera plus particulièrement au premier moment de $X \wedge u$, soit l'espérance limitée de X :

$$\mathbb{E}\left[X \wedge u\right] = \int_0^u x \, dF(x) + u(1 - F(u)). \tag{1.4}$$

L'espérance limitée peut s'interpréter comme l'espérance d'un sinistre avec une limite de souscription u.

Une valeur liée est l'espérance résiduelle. Celle-ci représente l'excédent moyen des valeurs d'une variable aléatoire supérieures à x. En termes de durée de vie, l'espérance résiduelle est appelée espérance de vie résiduelle (ou future) d'un individu d'âge x. Mathématiquement, on a

$$\begin{aligned} e(x) &= \mathbb{E}\left[X - x | X > x\right] \\ &= \frac{1}{1 - F(x)} \int_x^\infty (y - x) \, dF(y). \end{aligned}$$

Il n'est pas difficile de démontrer que l'on a entre l'espérance résiduelle et l'espérance limitée la relation

$$e(x) = \frac{\mathbb{E}[X] - \mathbb{E}[X \wedge x]}{1 - F_X(x)}. \tag{1.5}$$

L'espérance résiduelle s'interprète aussi comme la prime stop-loss pour une franchise x.

La version empirique de l'espérance limitée pour des données individuelle est

$$\begin{aligned}\hat{\mathbb{E}}[X \wedge u] &= \int_0^u x\, dF_n(x) + u(1 - F_n(u)) \\ &= \frac{1}{n} \sum_{x_j < u} x_j + u(1 - F_n(u)) \\ &= \frac{1}{n} \sum_{j=1}^{n} \min(x_j, u).\end{aligned}$$

Pour une limite fixe, il est simple de calculer cette quantité avec R :

```
> mean(pmin(dental, 1200))
[1] 304.4
```

1.3 Estimation paramétrique

L'approche paramétrique consiste à choisir un modèle connu pour le phénomène sous étude. Ce modèle comportera habituellement des paramètres qu'il faudra dé- terminer d'une manière ou une autre. En général, on optera pour une technique ayant un certain degré d'objectivité et se basant sur des observations du phéno- mène. En termes plus statistiques, on cherche à estimer le vecteur de paramètres $\theta = (\theta_1, \ldots, \theta_p)^T$ d'une fonction de densité de probabilité ou fonction de masse de probabilité $f(x, \theta)$ à partir d'un échantillon aléatoire $X_1, \ldots, X_n$ de la population. On note $(x_1, \ldots, x_n)$ les observations correspondantes.

1.3.1 Maximum de vraisemblance

La vraisemblance de l'échantillon s'exprime de la manière suivante

$$\mathcal{L}(\theta, x_1, \ldots, x_n) = \prod_{i=1}^{n} f_{X_i}(x_i, \theta) = \prod_{i=1}^{n} f(x_i, \theta),$$

où f désigne la fonction de masse de probabilité ou la densité suivant la loi retenue. L'estimateur du maximum de vraisemblance (EMV) de θ est la valeur $\hat{\theta}$ qui maximise la fonction de vraisemblance $\mathcal{L}(\theta, x_1, \ldots, x_n, \theta)$ par rapport à θ (pour un jeux d'observation donné) sur son domaine de définition. De plus, ceci est équivalent à maximiser le logarithme de la vraisemblance (appelée log-vraisemblance) :

$$l(\theta) = \sum_{i=1}^{n} \ln f(x_i, \theta).$$

On définit les fonctions de score

$$S_j(\theta) = \frac{\partial}{\partial \theta_j} \ln \mathcal{L}(\theta), \quad \text{pour} \quad j = 1, \ldots, p.$$

La maximisation de $\mathcal{L}(\theta)$ se résume donc à résoudre les équations normales

$$S_j(\theta) = 0, \quad \text{pour} \quad j = 1, \ldots, p.$$

Généralement, il n'existe pas de formules fermées pour ces équations, on les résout numériquement.

Dans le cas de données groupées, où nj représente le nombre de données dans la classe $]c_{j-1}, c_j]$, $j = 1, \ldots, r$, la probabilité qu'une donnée tombe dans l'intervalle $]c_{j-1}, c_j]$ est $F(c_j) - F(c_{j-1})$. La fonction de vraisemblance est donc

$$\mathcal{L}(\theta, x_1, \ldots, x_n) = \prod_{i=1}^{r} \left[F(c_j, \theta) - F(c_{j-1}, \theta)\right]^{n_j}.$$

Ainsi la log-vraisemblance s'écrit

$$l(\theta) = \sum_{i=1}^{r} n_j \ln \left[F(c_j, \theta) - F(c_{j-1}, \theta)\right].$$

On trouvera dans tout bon livre de statistique mathématique (par exemple Hogg et al. (2005), Saporta (2006), Dagnelie (2007), Dalgaard (2008)) une étude détaillée des propriétés de l'estimateur du maximum de vraisemblance. Aussi nous contenterons-nous, ici, de ne présenter que les principaux résultats.

Invariance Pour toute fonction bijective g, si $\hat{\theta}$ est l'EMV de θ, alors $g(\hat{\theta})$ est l'EMV de $g(\theta)$, soit

$$\widehat{g}(\theta) = g(\hat{\theta}).$$

Biais et efficacité asymptotique Sous des conditions de régularité, l'EMV est asymptotiquement sans biais et efficace. C'est-à-dire que si $\hat{\theta}_n$ est l'EMV de θ pour un échantillon (i.i.d.) de taille n, alors

$$\mathbb{E}\left[\hat{\theta}_{n,i}\right] \underset{n\to+\infty}{\longrightarrow} \theta_i \quad \text{et} \quad \mathbb{V}[\hat{\theta}_{n,i}] \underset{n\to+\infty}{\longrightarrow} \frac{1}{I_n(\theta)_{ii}}, \quad \text{pour} \quad i \in \{1, \ldots, p\},$$

où $I_n(\theta)$ désigne la matrice d'information de Fisher de taille $p \times p$ dont l'élément (i, j) est donné par

$$I_n(\theta)_{ij} = -n\mathbb{E}\left[\frac{\partial^2}{\partial\theta_i \partial\theta_j} \ln f(X, \theta)\right].$$

$(I_n(\theta)_{ii})^{-1}$ est appélée borne de Cramer-Rao.

Normalité asymptotique La distribution asymptotique de l'EMV est une loi normale multivariée de moyenne θ et avec matrice de variance-covariance $I_n(\theta)^{-1}$, i.e.

$$\hat{\theta}_n \to \mathcal{N}(\theta, I_n(\theta)^{-1}),$$

où $I_n(\theta)$ est la matrice d'information de Fisher donnée ci-dessus.

Concentrons-nous pour un instant sur le cas univarié ($p = 1$), plus simple. Par la propriété de normalité asymptotique, on a que, pour de grands échantillons,

$$P\left[-z_{\alpha/2} < \frac{\hat{\theta}_n - \theta}{\sqrt{I_n(\theta)^{-1}}} < z_{\alpha/2}\right] = 1 - \alpha,$$

où z_α est le $100(1-\alpha)^{\text{e}}$ centile d'une $N(0,1)$. On peut réécrire l'expression ci-dessus sous la forme

$$P\left[\hat{\theta}_n - z_{\alpha/2}\sqrt{I_n(\theta)^{-1}} < \theta < \hat{\theta}_n + z_{\alpha/2}\sqrt{I_n(\theta)^{-1}}\right] = 1-\alpha,$$

d'où

$$\left]\hat{\theta}_n - z_{\alpha/2}\sqrt{I_n(\theta)^{-1}}\ ,\ \hat{\theta}_n + z_{\alpha/2}\sqrt{I_n(\theta)^{-1}}\right[$$

est un intervalle de confiance de niveau $1-\alpha$ pour θ.

En pratique, la forme de l'information $I_n(\theta)$ rend souvent le calcul de l'intervalle de confiance ci-dessus impossible. Deux cas de figure se présentent :

1. l'information est connue, mais dépend de θ d'une manière compliquée. On remplace alors θ par son estimation $\hat{\theta}$, ce qui résulte en une estimation de la variance et donc à l'intervalle de confiance
$$\left]\hat{\theta}_n - z_{\alpha/2}\sqrt{I_n(\hat{\theta}_n)^{-1}}, \hat{\theta}_n + z_{\alpha/2}\sqrt{I_n(\hat{\theta}_n)^{-1}}\right[.$$
2. l'information est inconnue, par exemple si l'espérance est trop compliquée. Dans un tel cas, on remplace l'espérance par une moyenne empirique : c'est **l'information observée**
$$\hat{I}_n(\hat{\theta}_n) = -\sum_{i=1}^{n} \frac{\partial^2}{\partial\theta^2}\ln f(x_i;\theta)\bigg|_{\theta=\hat{\theta}} = -\frac{\partial^2}{\partial\theta^2} l(\theta; x_i, \ldots, x_n)\bigg|_{\theta=\hat{\theta}_n}.$$
L'intervalle de confiance pour θ est alors
$$\left]\hat{\theta}_n - z_{\alpha/2}\sqrt{\hat{I}_n^{-1}(\hat{\theta}_n)}, \hat{\theta}_n + z_{\alpha/2}\sqrt{\hat{I}_n^{-1}(\hat{\theta}_n)}\right[.$$

Ces idées se généralisent au concept d'ellipse ou d'ellipsoïde de confiance dans le cas multivarié.

En pratique, il n'est pas rare que l'on souhaite estimer non pas θ, mais une fonction $h(\theta)$ de θ. On sait déjà que l'EMV de $h(\theta)$ est $h(\hat{\theta}_n)$, mais qu'en est-il d'un intervalle de confiance pour cette estimation ? En général, il s'agit d'un calcul difficile car la distribution de $h(\hat{\theta}_n)$ peut être très compliquée. On peut alors utiliser la **méthode delta**, qui est valide pour les grands échantillons. Ainsi dans le cas univarié et pour h continument différentiable, lorsque $n \to \infty$,

$$h(\hat{\theta}_n) \sim \mathcal{N}\left(h(\theta), [h'(\theta)]^2 I^{-1}(\theta)\right),$$

d'où un intervalle de confiance de $h(\theta)$ est

$$\left]h(\hat{\theta}_n) - z_{\alpha/2}\sqrt{[h'(\theta)]^2 I(\theta)}\ ,\ h(\hat{\theta}_n) + z_{\alpha/2}\sqrt{[h'(\theta)]^2 I(\theta)}\right[.$$

Ce résultat s'étend aussi au cas multivarié. Si l'on souhaite estimer une fonction $h(\theta_1, \ldots, \theta_p)$ des paramètres inconnus $\theta_1, \ldots, \theta_p$, alors par la méthode delta, on a asymptotiquement

$$h(\hat{\theta}_n) \sim \mathcal{N}(h(\theta), \nabla h^T I_n(\theta)^{-1} \nabla h),$$

où ∇h^T représente la transposée du gradient ∇h et ∇h est donné par

$$\nabla h(\theta) = \begin{bmatrix} \dfrac{\partial}{\partial\theta_1} h(\theta) \\ \vdots \\ \dfrac{\partial}{\partial\theta_p} h(\theta) \end{bmatrix}.$$

Un intervalle de confiance de niveau $1-\alpha$ pour $h(\theta)$ est donc

$$\left]h(\hat{\theta}_n) - z_{\alpha/2}\sqrt{\nabla h^T I_n(\theta)^{-1}\nabla h}, h(\hat{\theta}_n) + z_{\alpha/2}\sqrt{\nabla h^T I_n(\theta)^{-1}\nabla h}\right[.$$

Considérons l'exemple trivial : on choisit de calibrer la loi exponentielle de paramètre λ, alors on a

$$\log \mathcal{L}(\lambda, x_1, \ldots, x_n) = \log\left(\lambda^n \prod_{i=1}^{n} e^{-\lambda x_i}\right) = n\log(\lambda) + \sum_{i=1}^{n}(-\lambda)x_i.$$

D'où l'estimateur de maximum de vraisemblance est $n/\sum_{i=1}^{n} X_i$.

En pratique, il est plutôt rare de pouvoir obtenir explicitement l'expression du maximum de la fonction de vraisemblance. Dans de tels cas, il faut avoir recours à des méthodes numériques pour résoudre les équations normales (par exemple avec la méthode de Newton-Raphson) ou alors pour directement maximiser la fonction de vraisemblance ou, plus communément, de log-vraisemblance.

Le package **fitdistrplus** fournit une fonction `mledist` qui se charge d'appeler les algorithmes d'optimisation implémentés dans R (voir les fonctions `optim` ou `optimize` implémentant une méthode de quasi-Newton avec recherche linéaire et une méthode de recherche dichotomique, respectivement pour des fonctions multivariées et univariées). Dans la suite, nous allons tester l'estimateur de maximum de vraisemblance sur un échantillon gamma et un échantillon Pareto et tester leur robustesse.

```
> library(fitdistrplus)
> x <- rgamma(1000, 2, 3)
> mledist(x, "gamma")
$estimate
   shape     rate
2.031030 2.921142
$convergence
[1] 0
$loglik
[1] -516.0789
$hessian
          shape      rate
shape  632.6279 -342.3319
rate  -342.3319  238.0188
$optim.function
[1] "optim"
```

La fonction `fitdist` englobant la fonction `mledist` fournit des informations plus détaillées concernant les estimations, notamment les erreurs standards.

```
> fit1.gam <- fitdist(x, "gamma", method="mle")
> summary(fit1.gam)
Fitting of the distribution ' gamma ' by maximum likelihood
Parameters :
      estimate Std. Error
shape 2.031030 0.08443494
rate  2.921142 0.13765460
Loglikelihood:  -516.0789   AIC:  1036.158   BIC:  1045.973
Correlation matrix:
```

```
          shape      rate
shape 1.0000000 0.8822011
rate  0.8822011 1.0000000
```

Noter l'écart relativement grand entre les vraies valeurs des paramètre (2, 3) et leur estimation (2.031030, 2.921142). Sur le graphique 1.10, on peut constater la convergence relativement lente des estimateurs MLE.

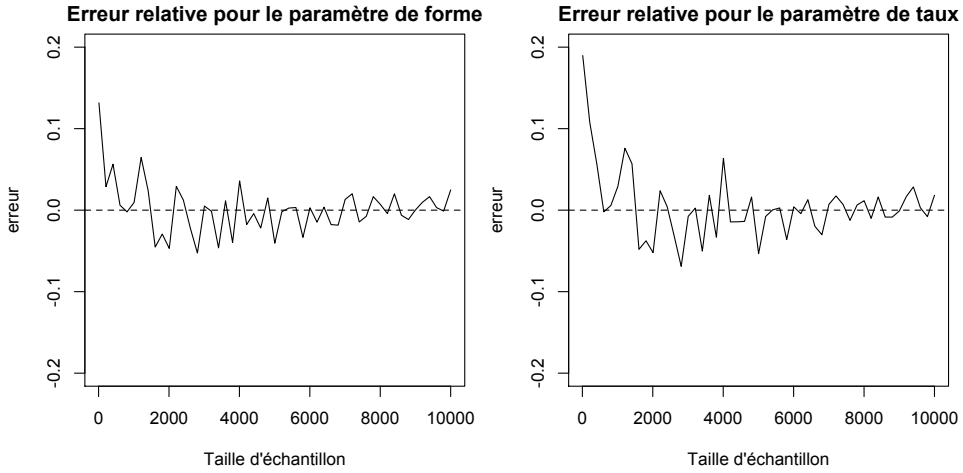


FIGURE 1.10 – Erreur relative sur les paramètres de la loi Gamma

On peut faire la même procédure pour un échantillon de loi de Pareto. A notre grand regret, l'estimateur de maximum de vraisemblance s'avère encore plus lent pour un échantillon de loi de Pareto, cf. figure 1.11.

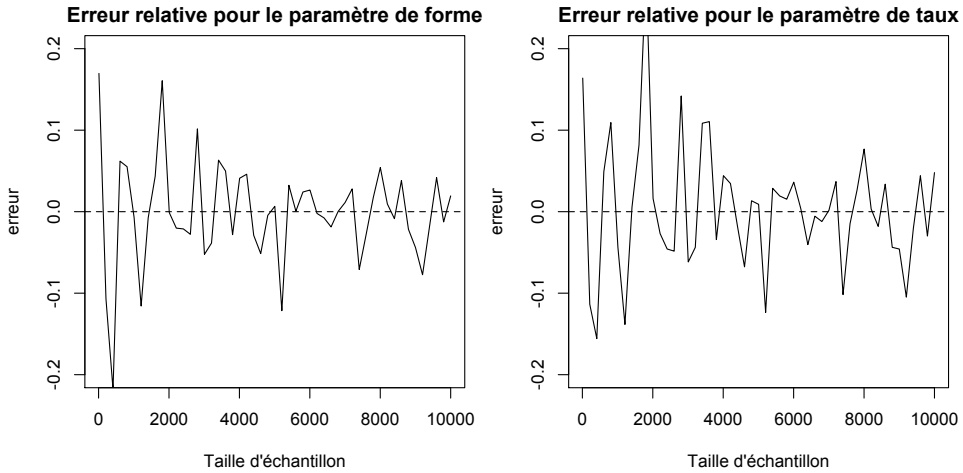


FIGURE 1.11 – Erreur relative sur les paramètres de la loi Pareto

1.3.2 Méthodes des moments

La méthode des moments est probablement la plus ancienne méthode utilisée pour faire de l'estimation ponctuelle. C'est une méthode d'estimation simple et intuitive, mais les estimateurs obtenus possèdent généralement peu de "belles" propriétés. Pour déterminer les estimateurs des

moments des paramètres $\theta_1, \ldots, \theta_p$, on impose que les p premiers moments théoriques soient identiques aux p premiers moments empiriques (au moins). On doit donc résoudre

$$\mathbb{E}[X^k] = \frac{1}{n}\sum_{i=1}^{n} X_i^k, \quad \text{pour} \quad k = 1, \ldots, p.$$

Il n'y a aucune garantie que la solution au système d'équations soit unique ou même qu'elle n'existe. Bien qu'ils ne réunissent peu de propriétés d'optimalité souhaitables pour des estimateurs ponctuels, les estimateurs des moments demeurent populaires, si ce n'est qu'à titre de points de départ pour d'autres méthodes. On remarquera que pour les lois inverse, il vaut souvent mieux utiliser les moments négatifs ($k = -1, -2, \ldots$).

Reprenons le cas de la loi exponentielle, l'espérance est donnée par $\frac{1}{\lambda}$. Le système d'équation se réduit à

$$\frac{1}{n}\sum_{i=1}^{n} X_i = \frac{1}{\lambda} \Leftrightarrow \lambda = \frac{n}{\sum_{i=1}^{n} X_i},$$

qui est aussi l'estimateur de maximum de vraisemblance. Ceci n'est évidemment qu'un pur hasard.

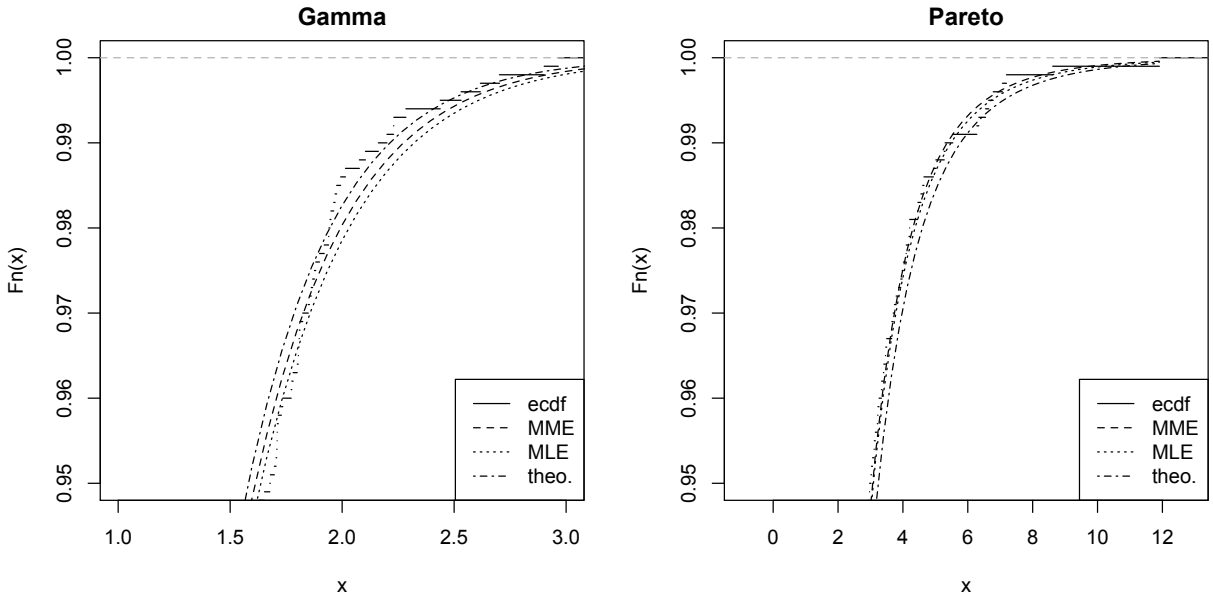
Le package **fitdistrplus** permet d'utiliser la méthode des moments soit directement avec la fonction `mmedist` soit via la fonction `fitdist`. Dans R, cela donne les commandes suivantes :

```
> library(fitdistrplus)
> x <- rexp(1000, 1)
> mmedist(x, "exp", order=1)
$estimate
    rate
0.991603
$convergence
[1] 0
$order
[1] 1
$memp
NULL
$loglik
[1] -1008.432
$method
[1] "closed formula"
```

Reprenons nos échantillons simulés gamma et Pareto, i.e. deux échantillons de taille 1000, dont on cherche à estimer les paramètres. Même si il peut être intéressant de tester des calibrations de moments d'ordre supérieurs, on se limite en pratique à égaliser les deux premiers moments.

Sur la figure 1.12, on a tracé la fonction de répartition empirique et les fonctions de répartition calibrées par maximum de vraisemblance et par la méthode des moments. Les deux sous-figures 1.12a et 1.12b montrent que les ajustements ne sont pas trop mauvais, même dans les queues de distribution.

Cependant, les estimations pour la loi de Pareto sont très loin de la vraie valeur des paramètres. Quant à la fonction de répartition empirique, elle sous estime probabilité la queue de distribution, puisque sur un échantillon de taille 1000, il y a peu de valeurs extrêmes pour une loi de Pareto. Ainsi, on sous-estime la queue de distribution de la vrai Pareto.



(a) Loi gamma (b) Loi Pareto

FIGURE 1.12 – Comparaison des calibrations – Echantillon taille 1000

1.3.3 Méthodes des quantiles

La méthodes des quantiles consiste à égaler les quantiles empiriques avec les quantiles théoriques. Dans l'esprit, elle est très similaire à la méthode des moments. Mais en pratique elle peut se révéler plus robuste, d'une part car les quantiles existent toujours et d'autre part cela permet de calibrer les données dans un endroit particulier de la fonction de répartition, le cœur ou les queues de distribution.

La méthode des quantiles consiste à résoudre les équations

$$q_{n,p_k} = F^{-1}(p_k), \quad \text{pour} \quad k = 1, \ldots, p$$

où q_{n,p_k} dénote le quantile empirique et $F^{-1}(p_k)$ le quantile théorique.

La fonction de quantile de la loi exponentielle est $Q(p) = -\frac{\log(1-p)}{\lambda}$. Il suffit donc de résoudre l'équation

$$Q_n(1/2) = -\frac{\log(1/2)}{\lambda} \Leftrightarrow \lambda = -\frac{\log(1/2)}{Q_n(1/2)},$$

dans le cas où l'on veut calibrer sur la médiane (i.e. $p = 1/2$).

Le package **fitdistrplus** permet aussi d'utiliser la méthode des quantiles soit directement avec la fonction `qmedist` soit via la fonction `fitdist`. Dans R, cela donne les commandes suivantes :

```
> x <- rexp(1000, 1)
> qmedist(x, "exp", probs=1/2)
$estimate
     rate
0.9577806
$convergence
[1] 0
```

```
$value
[1] 5.657381e-13
$hessian
         rate
rate 1.141883
$probs
[1] 0.5
$optim.function
[1] "optim"
$loglik
[1] -1009.352
> qmedist(x, "exp", probs=4/5)
$estimate
    rate
1.003066
$convergence
[1] 0
$value
[1] 2.279133e-12
$hessian
         rate
rate 5.117578
$probs
[1] 0.8
$optim.function
[1] "optim"
$loglik
[1] -1008.839
```

Contrairement à la méthode des moments, où égaliser des moments de hauts degrés était plutôt sans intérêt, pour la méthode des moments le choix de tels ou tels quantiles peut être totalement justifié. Sur l'exemple précédent, on a choisi la médiane et le quantile à 80%, et l'on constate des estimations assez différente.

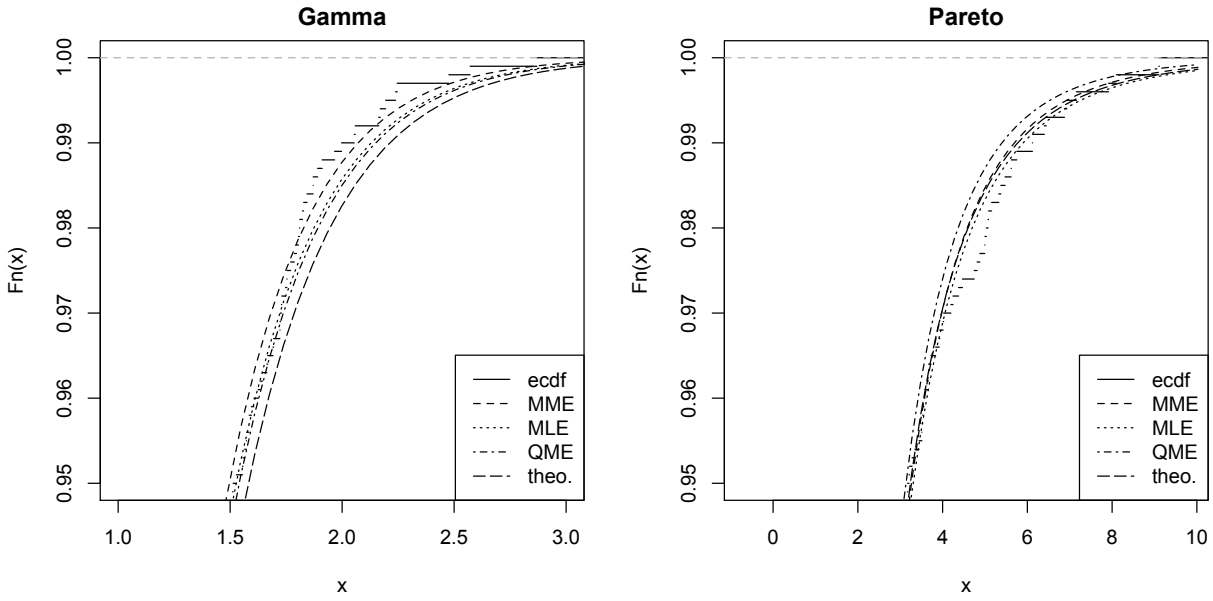
Enfin sur la figure 1.13, on a continué la comparaison des fonctions de répartition entre les trois méthodes paramétriques et la méthode non paramétrique. Notons que l'ordre entre le maximum de vraisemblances et la méthodes des moments semble de nouveau respecté. Par contre, pour le choix de quantiles considérés (1/3, 2/3), la méthode des quantiles peut ou ne pas être plus conservateur.

1.4 Estimation des copules

Dans cette section, nous présentons les méthodes d'estimation pour calibrer une copule. Nous renvoyons le lecteur à la section 1.4.6 pour un exemple d'application.

1.4.1 Méthode des moments

Cette méthode consiste à estimer les paramètres θ des lois marginales et le paramètre α de la copule par la méthode des moments :



(a) Loi gamma (b) Loi Pareto

FIGURE 1.13 – Comparaison des calibrations – Echantillon taille 1000

1. résoudre le système à d équations et d inconnues

$$\begin{cases} \overline{X}_n = f(\theta_1, \ldots, \theta_d) \\ S_n^2 = g(\theta_1, \ldots, \theta_d) \\ \mu_{3,n} = h(\theta_1, \ldots, \theta_d) \\ \vdots \end{cases},$$

où d désigne la dimension de θ, f, g et h sont les expressions des moments (ordinaires) d'ordre 1, 2 et 3 en fonction du paramètre θ. Répeter cette étape pour toutes les marginales,

2. égaler une mesure de dépendance avec son équivalent empirique. Si la copule possède une formule fermée, on peut directement inverser le tau de Kendall ou le rho de Spearman pour obtenir le paramètre α de la copule.

Notons que si la copule a plusieurs paramètres, il faut trouver plusieurs équations pour déterminer les paramètres de la copule, voir par exemple Joe (1997).

Pour des marginales exponentielles $\mathcal{E}(\lambda)$ et une copule de Gumbel, on trouve

$$\hat{\lambda}_n = \frac{1}{\overline{X}_n} \quad \text{et} \quad \hat{\alpha}_n = \frac{1}{1-\tau_n},$$

où τ_n désigne le tau de Kendall empirique, disponible dans la fonction `cor`.

1.4.2 Maximum de vraisemblance exact

Dans le cas où la densité de la copule existe, on peut utiliser les estimateurs de maximum de vraisemblance. Pour simplifier, on suppose qu'on utilise une copule bivariée C_α, ayant une densité et que les lois des marginales possèdent des densités. On note θ_1 et θ_2 les paramètres

des lois marginales. La log vraisemblance s'écrit :

$$\ln \mathcal{L}(\alpha, \theta_1, \theta_2, x_1, \ldots, x_n, y_1, \ldots, y_n) = \sum_{i=1}^{n} \ln \left(c \left(F_1(x_i, \theta_1), F_2(y_i, \theta_2), \alpha \right) \right) + \sum_{i=1}^{n} \ln \left(f_1(x_i, \theta_1) \right) + \sum_{i=1}^{n} \ln \left(f_2(y_i, \theta_2) \right).$$

Bien souvent, il n'existe pas d'expressions explicites des estimateurs maximisant $\ln \mathcal{L}$, et on réalise donc une maximisation numérique.

1.4.3 Inférence sur les marginales

Toujours dans l'hypothèse où la copule a une densité, on peut mélanger les deux premières approches, en estimant d'abord les paramètres des lois marginales, puis en estimant le paramètre de la copule. Cela consiste à :

1. estimer les paramètres θ_1 et θ_2 par maximum de vraisemblance,
2. construire les pseudo données $\forall 1 \leq i \leq n,\ u_i = F_1(x_i, \hat{\theta}_1)$ et $v_i = F_2(y_i, \hat{\theta}_2)$
3. estimer le(s) paramètre(s) α en maximisant la log-vraisemblance,

$$\ln \mathcal{L}(\alpha, u_1, \ldots, u_n, v_1, \ldots, v_n) = \sum_{i=1}^{n} \ln \left(c \left(u_i, v_i, \alpha \right) \right).$$

Cette méthode présente l'avantage d'utiliser les estimateurs "classiques" de maximum vraisemblance des marginales.

1.4.4 Maximum de vraisemblance canonique

C'est une méthode semi-paramétrique, qui se base sur la méthode précédente :

1. calculer les fonctions de répartition empirique $F_{1,n}$ et $F_{2,n}$,
2. construire les pseudo données $\forall 1 \leq i \leq n,\ u_i = F_{1,n}(x_i)$ et $v_i = F_{2,n}(y_i)$,
3. estimer le(s) paramètre(s) α en maximisant la log-vraisemblance,

$$\ln \mathcal{L}(\alpha, u_1, \ldots, u_n, v_1, \ldots, v_n) = \sum_{i=1}^{n} \ln \left(c \left(u_i, v_i, \alpha \right) \right).$$

1.4.5 Choix de la copule

Le choix de la copule doit être en relation avec l'élément modélisé. Par conséquent, si on cherche à modéliser un évènement extrême, on doit se concentrer la famille des copules extrêmes. Ceci exclut donc la copule gaussienne.

A une famille donnée, différents critères statistiques peuvent être comparés pour valider ou non une copule. Il existe des critères liés à la log-vraisemblance : la log-vraisemblance (simple) $\ln \mathcal{L}$, le critère d'Aikake (AIC) $2k - 2 \ln \mathcal{L}$ ou encore le critère de Schwarz (BIC) $-2 \ln \mathcal{L} + k \ln n$, où k est le nombre de paramètres (à estimer) et n la taille de l'échantillon.

Une autre catégorie de critères s'intéresse à des distances statistiques entre la distribution empirique et la distribution théorique (calibrée). Typiquement, on utilise la distance de Kolmogorov-Smirnov, d'Anderson-Darling ou encore la distance L^2, voir Saporta (2006).

1.4.6 Application aux couvertures de produits indiciels

Dans cette sous-section, nous présentons une application des copules à la couverture de produit indiciel, où nous allons prendre en compte la dépendance entre stations météorologiques pour la construction d'un indice. Nous nous concentrons sur l'aspect pédagogique de l'utilisation et ne cherchons pas à présenter le modèle parfait. Nous avons choisi la copule de Gumbel, car elle appartient aux copules extrêmes et aux copules Archimédiennes. Tout en ayant une expression simple et explicite, la copule de Gumbel a l'avantage de décrire les dépendances asymétriques, où les coefficients de queue inférieure et de queue supérieure diffèrent. Elle possède donc la caractéristique de pouvoir représenter des risques dont la structure de dépendance est accentuée sur la queue supérieure et est particulièrement adaptée en assurance et en finance pour étudier l'impact de la survenance d'événements de forte intensité sur la dépendance entre plusieurs variables d'intérêts.

Présentation

Nous avons utilisé la copule de Gumbel pour valoriser les couvertures indicielles cat[astrophe]. Ces contrats sont des dérivés climatiques adaptés à la réassurance d'évènement catastrophe (tempête, vague de froid,...) basé sur un indice climatique (force du vent, température,...). Cette application numérique est basée sur l'article Dubreuil & Vendé (2005).

L'indice climatique doit refléter au mieux les caractérisques des montants des sinistres associés au risque météo pour diminuer le risque de base. En général, on choisit un panier de n stations (peu éloignées des régions assurées) dans lesquelles on mesure la variable climatique $X_i(t)$ au cours de la période $[t-1, t]$. Ensuite, l'indice journalier d'une station i est construit par $I_i(t) = \min(L_i - K_i, X_i(t) - K_i)$ où K_i et L_i sont le seuil et la limite par station. Sur une période T, l'indice d'une station est donc défini par $S_i(T) = \sum_{t=1}^{T} I_i(t)$ et l'indice cumulé par $S_T = \sum_{i=1}^{n} p_i S_i(T)$ pour une pondération $p_1, \ldots, p_n$. Enfin le flux engendré par la couverture indicielle est celui d'un call spread :

$$C_T = N \times \min\left(L - K, (S_T - K)_+\right),$$

où K et L sont la franchise et la limite du contrat, et N le montant nominal.

Pour notre exemple, on traite le risque "tempête" en Rhône Alpes. $X_i(t)$ désigne donc la force maximale du vent (en m/s) par jour. Nous avons choisi deux stations Saint Martin en Haut (variable X) et Echirolles (variable Y) avec les seuils respectifs 10 et 9, et les limites 16 et 15 [2]. On prend $T = 633$ jours, $N = 1$, $K = 50$ et $L = 200$.

Calibration

Il nous faut calibrer la copule de Gumbel sur nos données recueillies sur internet entre août 2005 et avril 2007. Les données sont livrées avec le package **gumbel** et il suffit de les charger avec la fonction `data()`. Comme les jeux de données `windEchirolles` et `windStMartin` possèdent un nombre d'enregistrements différents en 2007, on sélectionne le plus petit sous-ensemble. On enlève ausssi les données manquantes.

```
> library("gumbel")
> data(windEchirolles)
> data(windStMartin)
> n <- min(NROW(windStMartin), NROW(windEchirolles))
```

2. les seuils sont volontairement bas.

```
> id2keep <- !is.na(windStMartin$WIND.HIGH[1:n]) &
+ !is.na(windEchirolles$WIND.HIGH[1:n])
> x <- windStMartin$WIND.HIGH[1:n][id2keep]/3.6
> y <- windEchirolles$WIND.HIGH[1:n][id2keep]/3.6
```

Pour calibrer la copule de Gumbel, un choix de marginales s'impose. L'exemple de cette sous-section étant à but pédagogique, nous avons choisi de tester seulement deux lois : exponentielle et gamma. A l'aide de `mledist`, on obtient facilement les paramètres calibrés.

```
> library(fitdistrplus)
> xpar_gamma <- mledist(x, "gamma")
> ypar_gamma <- mledist(y, "gamma")
> xpar_exp <- mledist(x, "exp")
> ypar_exp <- mledist(y, "exp")
```

En traçant les fonctions de répartitions empiriques et calibrées pour chaque marginale, figure 1.14, on constate que la loi exponentielle n'est pas du tout adapté. On peut se convaincre par des tests d'adéquation : le test de Kolmogorov Smirnov rejette outrageusement l'hypothèse que les données suivent une loi exponentielle. On choisit donc la loi gamma. Notons les paramètres de forme et de taux sont notés α_X, λ_X pour Saint Martin en Haut et α_Y, λ_Y pour Echirolles.

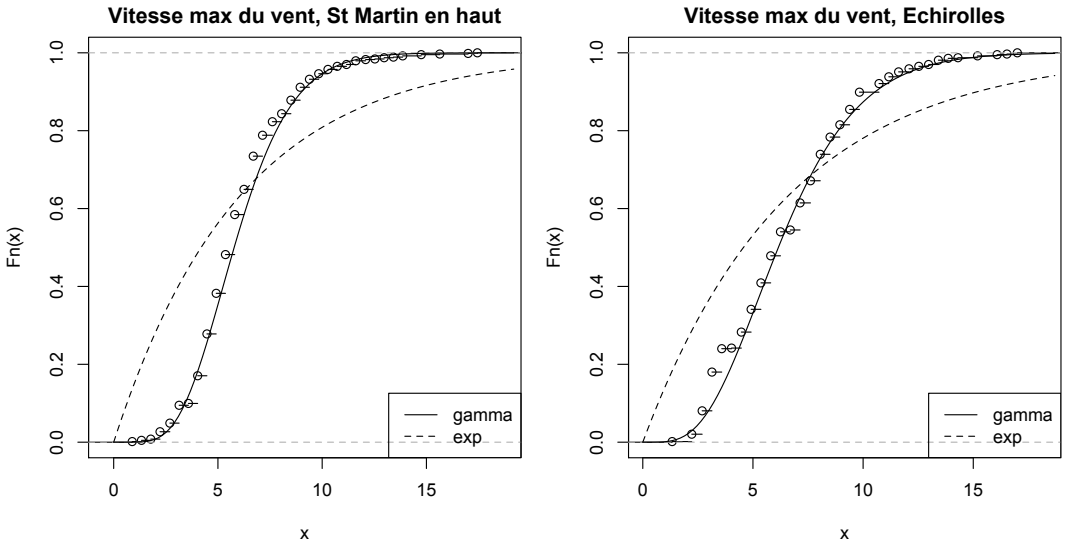


FIGURE 1.14 – Fonctions de répartitions empiriques (marche) et calibrées (courbes)

Calcul du paramètre de la copule On pose α_{cop} le paramètre de la copule de Gumbel. Nous utilisons les fonctions du package **gumbel**

- la méthode des moments ("Moment-Based Estimation") : `gumbel.MBE`,
- le maximum de vraisemblance exacte ("Exact Maximum Likelihood") : `gumbel.EML`,
- l'inférence sur les marginales ("Inference For Margins") : `gumbel.IFM`,
- le maximum de vraisemblance canonique ("Canonical Maximum Likelihood") : `gumbel.CML`.

Pour obtenir le paramètre de la copule de Gumbel, il suffit d'appeler les fonctions précédemment listées.

```
> res <- cbind(
+ MBE=gumbel.MBE(x, y, marg="gamma"),
```

```
+ EML=gumbel.EML(x, y, marg="gamma"),
+ IFM=gumbel.IFM(x, y, marg="gamma"),
+ CML=c(rep(NA, 4), gumbel.CML(x, y))
+ )
> rownames(res) <- c("shape-x","rate-x","shape-y","rate-y","copula")
> res <- cbind(res, avg=apply(res, 1, mean, na.rm=TRUE))
```

Le tableau 1.3 ci-dessous récapitule nos résultats d'estimation (variable `res` ci-dessus) pour les 4 méthodes présentées en section 1.4.

Méthodes	MBE	EML	IFM	CML	moyenne
$\hat{\alpha}_X$	6,99	7,022	7,395	-	7,135
$\hat{\lambda}_X$	1,156	1,155	1,223	-	1,178
$\hat{\alpha}_Y$	5,04	5,105	4,969	-	5,038
$\hat{\lambda}_Y$	0,7649	0,7712	0,7541	-	0,7634
$\hat{\alpha}_{cop}$	1,524	1,454	1,44	1.47	1,472

TABLE 1.3 – Estimations des paramètres

Le bon ajustement des marginales étant déjà été établi sur les graphes de la figure 1.14, il nous faut choisir une valeur pour les différents paramètres : celui de la copule et ceux des marginales. Pour la suite, nous choisissons les moyennes des estimations comme valeur de nos paramètres, c'est à dire la dernière colonne.

La bonne adéquation de la copule de Gumbel aux vitesses de vent maximales est confirmé par le tracé d'un qqplot empirique (cf. figure 1.15). C'est à dire, il nous faut tracer le nuage de points $(F_{X,n}(X_i), F_{Y,n}(Y_i))$ pour $1 \leq i \leq n$ où $F_{X,n}$ (resp. $F_{Y,n}$) représente la fonction de répartition empirique de X (resp. Y). Cela revient à tracer les rangs normalisés, puisque $F_{X,n}(X_i) = \text{rang}(X_i)/n$. Pour comparer ce nuage de points observés à la copule calibrée, nous simulons un échantillon de couples aléatoires $(U_i, V_i)_{1\leq i\leq n}$ de loi de Gumbel de paramètre α_{cop} et traçons les rangs des couples. Les nuages semblent équivalents.

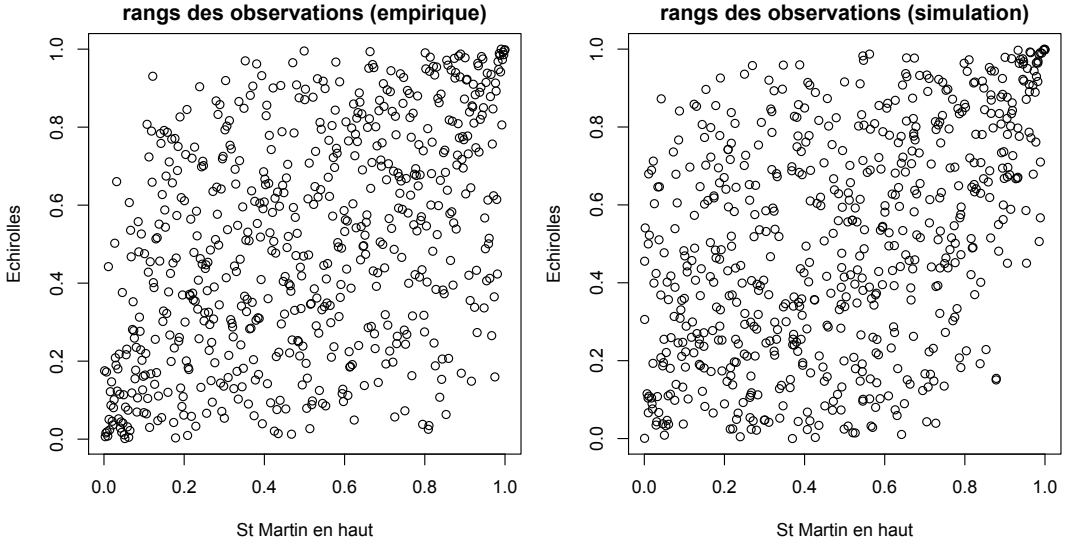


FIGURE 1.15 – qqplot empirique vs simulé

On peut aussi tester l'adéquation de nos données à la famille des copules extrêmes en se basant sur Genest et al. (2011). Le test statistique est implémenté dans le package **copula** par la fonction `gofEVCopula`. Comme on peut le constater ci-dessous, on ne rejette pas l'hypothèse que les données soient issues d'une copule de Gumbel. De plus, si l'on teste aussi la copule de Husler-Reiss, voir par exemple Joe (1997), la statistique de Cramer-von Mises est plus faible pour l'hypothèse Gumbel que l'hypothèse de Husler-Reiss. Par conséquent, on conclut que la copule de Gumbel est adaptée à nos données.

```
> library(copula)
> gofEVCopula(gumbelCopula(1), cbind(x, y), optim.method="BFGS", N=1000,
+ m=500, print.every=-1)
Parameter estimate(s): 1.477927
Cramer-von Mises statistic: 0.01389476 with p-value 0.3171828
> gofEVCopula(huslerReissCopula(1), cbind(x, y), optim.method="BFGS", N=1000,
+ m=500, print.every=-1)
Parameter estimate(s): 1.173151
Cramer-von Mises statistic: 0.01443714 with p-value 0.2482517
```

Evaluation du payoff

Maintenant que l'on a calibré notre copule (et les marginales), on va estimer le payoff C_T par une méthode de Monte Carlo. Nous avons réalisé 10000 simulations de période de $T = 633$ jours pour nos deux stations. Pour ce faire, nous créeons une fonction calculant C_T pour un échantillon donné. Notons qu'historiquement, le payoff est évalué à 80,64 unités monétaires.

```
> payoffIndice <- function(vent1, vent2, k1, k2, l1, l2, K, L, p1=1/2, p2=1/2)
+ {
+         S1 <- sum(pmin(vent1-k1, l1-k1)*(vent1 > k1))
+         S2 <- sum(pmin(vent2-k2, l2-k2)*(vent2 > k2))
+         S <- p1*S1 + p2*S2
+         return( min(max(S - K, 0), L - K) )
+ }
> payoffIndice(x, y, 9, 10, 16, 15, 50, 200)
[1] 80.63889
```

Ensuite, on simule des échantillons de vitesses de vent maximum pour lesquels on calcule les payoffs avec la fonction `payoffIndice`. Les paramètres du produit indiciel sont $K_1 = 9$, $K_2 = 10$, $L_1 = 16$, $L2 = 15$, $K = 50$ et $L = 200$. On constate que le payoff moyen est proche du payoff observé 80,64.

```
> priceIndHedCat <- function(nbday, nbsimu, param, k1, k2, l1, l2, K, L,
+ p1=1/2, p2=1/2)
+ {
+         f <- function()
+         {
+         ventSim <- rgumbel(nbday, param["copula"], dim=2)
+         ventSim[,1] <- qgamma(ventSim[,1], param["shape-x"], param["rate-x"])
+         ventSim[,2] <- qgamma(ventSim[,2], param["shape-y"], param["rate-y"])
+
+         payoffIndice(ventSim[,1], ventSim[,2], k1, k2, l1, l2, K, L, p1, p2)
+         }
+         replicate(nbsimu, f())
```

```
+ }
> finalpar <- res[,"avg"]
> payoff <- priceIndHedCat(633, 10000, finalpar, 9, 10, 16, 15, 50, 200)
> summary(payoff)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  19.35   68.53   79.10   79.75   90.61  144.30
```

Sur la figure 1.16 , nous avons tracé l'histogramme des payoff simulés. Nous pouvons constater que la distribution des sinistres est légèrement asymétrique, surtout autour de sa moyenne. Par ailleurs, nous avons ajouté l'estimation de la densité par la méthode du noyau d'Epanechnikov.

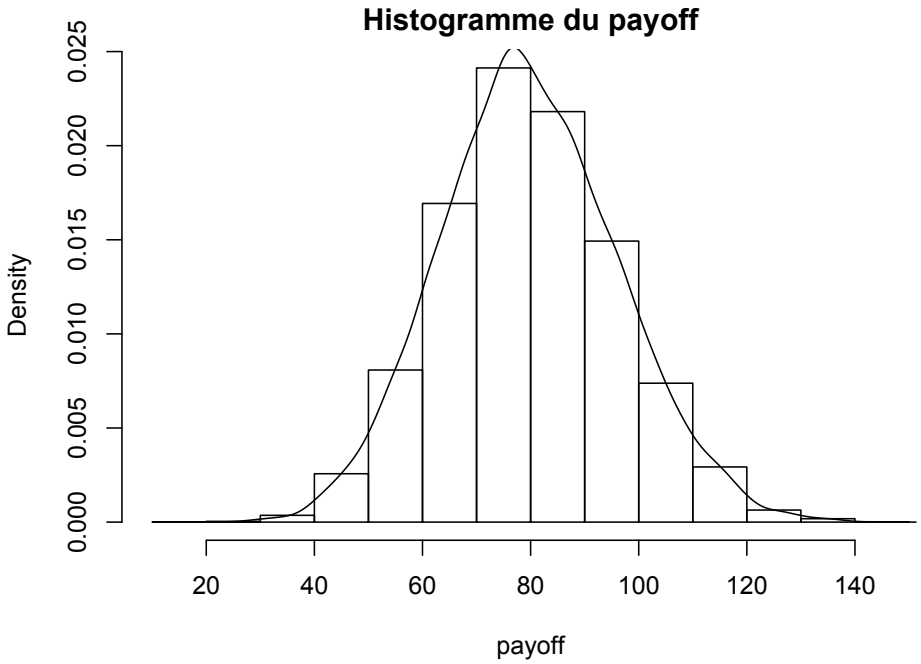


FIGURE 1.16 – Histogrammes à pas fixe et à même effectif.

Nous avons aussi fait une analyse de sensibilité au paramètre de la copule. La valeur calibrée est $\hat{\alpha}_{cop} = 1,472$. Nous obtenons les résultats donnés dans le tableau 1.4. On constate qu'une augmentation de la dépendance (i.e. $\hat{\alpha}_{cop}$ augmente) entraine une plus grande volatilité du payoff tant au niveau de l'écart-type que des quantiles à 75% et 90%. Cependant, la moyenne du payoff reste stable.

α_{cop}	-25%	-10%	(valeur estimée)	+10%	+25%
moyenne	79,93	79,68	79,75	79,61	79,69
écart-type	14,86	15,95	16,4	16,89	17,55
$\text{VaR}_{75\%}$	89,78	90,17	90,61	90,47	91,28
$\text{VaR}_{90\%}$	99,39	100,3	101,2	101.5	102,7

TABLE 1.4 – Statistiques des payoff C_T simulés

Maintenant, il ne reste plus qu'à choisir un principe de primes et calculer le prix de ce produit de couverture indiciel virtuel. Pour conclure sur cette application numérique, il est important de souligner qu'on ne tient pas compte de la dépendance sérielle entre les maximums des vitesses

de vent.

1.5 Exercices

Exercise 1.5.1. *Faire une fonction qui construit un histogramme à même effectif. Simuler ensuite un échantillon de taille 1000 de loi exponentielle et comparer les histogrammes à pas fixe et à même effectif.*

Exercise 1.5.2. *Simuler un échantillon de loi exponentielle avec un paramètre de valeur connue par avance. Estimer le paramètre avec la méthode de maximum de vraisemblance, des moments, des quantiles ou de distance minimale. Faire varier la taille de l'échantillon. Faire de même pour la loi log-normale et la loi de Pareto.*

Exercise 1.5.3. *Simuler une loi mélange pour laquelle* $1 - p = 99\%$ *des observations sont de loi lognornmale et* $p = 1\%$ *de loi Pareto, sans utiliser de boucles* `for`*. En choisissant des paramètres de lois tels que l'espérance de chacune des lois vaut 1, tracer les quantiles à 50% et 75% en fonction du paramètre de mélange* p*.*

Exercise 1.5.4. *Programmer l'algorithme de Panjer quand la variable de comptage* N *suit une loi binomiale négative, et que les coûtes de sinistres suivent une loi de Pareto. Comparer le quantile à 95% avec des simulations.*

Exercise 1.5.5. *Ajuster une loi de Pareto aux coûts de sinistres* `data(danish)` *de* `library(evir)` *avec la méthode de maximum de vraisemblance, des moments, des quantiles et de distance minimale.*

Exercise 1.5.6. *En utilisant la méthode du maximum de vraisemblance, ajuster une loi de Pareto, une loi de Weibull, une loi Gamma et une loi logmnormale aux coûts de sinistres* `data(danish)` *de* `library(evir)`*. Comparer les primes pures obtenues.*

Chapitre 2

La tarification a priori

L'assurance est un contrat par lequel, moyennant le versement d'une prime dont le montant est fixé a priori (en début de période de couverture), l'assureur s'engage à indemniser l'assuré pendant toute la période de couverture (disons un an). Cette prime est censée refléter le risque associé au contrat (on peut renvoyer à Denuit & Charpentier (2004) sur la théorie du calcul des primes, et à Denuit & Charpentier (2005) pour les considérations économiques). Pour chaque police d'assurance, la prime est fonction de variables dites de tarification (permettant de segmenter la population en fonction de son risque). Généralement, on considère

- des informations sur l'assuré, comme l'âge ou le sexe pour un particulier, ou le secteur d'activité et le nombre de salariés pour une entreprise,
- des informations sur le bien assuré, comme l'âge du véhicule, la puissance ou la marque en assurance auto, la surface du logement en multirisque habitation, le chiffre d'affaire de l'entreprise en perte d'exploitation,
- des informations géograhiques comme le revenu moyen dans la commune ou le département, la densité de médecins, la densité de population, etc.

La *fréquence* (annualisée) est le nombre de sinistres divisé par l'exposition (correspondant au nombre d'années police) pour une police d'assurance, ou un groupe de polices d'assurance. La plupart des contrats étant annuels, on ramènera toujours le nombre de sinistres à une exposition annuelle lors du calcul de la prime, et on notera N la variable aléatoire associée. Durant la période d'exposition, on notera Y_i les coûts des sinistres, c'est à dire les indemnités versées par l'assureur à l'assuré (ou une tierce personne). La charge totale par police est alors $S = 0$ s'il n'y a pas eu de sinistres, ou sinon :

$$S = Y_1 + \cdots + Y_N = \sum_{i=1}^{N} Y_i.$$

Classiquement (et ce point sera important pour constituer la base de données), les sinistres sont de coût positif,$Y_i > 0$, et N est alors le nombre de sinistres *en excluant* les sinistres classés sans suite (i.e. de coût nul).

La prime pure est $\mathbb{E}(S) = \mathbb{E}(N) \cdot \mathbb{E}(Y_i)$ dès lors que les coûts individuels sont i.i.d., et supposés indépendants du nombre de sinistres. Dans le cas où la fréquence et les charges sont hétérogènes, l'hétérogénéité étant caractérisée par une information Ω, la prime pure devrait être :

$$\mathbb{E}(S|\Omega) = \mathbb{E}(N|\Omega) \cdot \mathbb{E}(Y_i|\Omega).$$

Le facteur d'hétérogénéité Ω étant inconnu, on utilise les variables tarifaires à notre disposition pour obtenir un proxi de ces espérances conditionnelles. On cherche alors $\boldsymbol{X} = (X_1, \cdots, X_k)$ un

ensemble de variables explicatives telles que

$$\mathbb{E}(S|\boldsymbol{X}) = \mathbb{E}(N|\boldsymbol{X}) \cdot \mathbb{E}(Y_i|\boldsymbol{X}).$$

On dispose pour cela de deux bases : une base de souscription, `contrat`, contenant les informations sur les polices d'assurance, et une base de sinistre contenant des coûts de sinistres, `sinistre`. Les deux bases peuvent être reliées par un numéro de contrat. Pour constituer une base contenant les nombres de sinistres, la fonction `merge` [1] est particulièrement utile. Sur notre exemple, le code est le suivant :

```
> T<-table(sinistre$nocontrat)
> T1<-as.numeric(names(T))
> T2<-as.numeric(T)
> nombre1 <- data.frame(nocontrat=T1,nbre=T2)
> I <- contrat$nocontrat%in%T1
> T1<-contrat $nocontrat[I==FALSE]
> nombre2 <- data.frame(nocontrat=T1,nbre=0)
> nombre<-rbind(nombre1,nombre2)
> baseFREQ <- merge(contrat,nombre)
> baseCOUT <- merge(sinistre,contrat)
> tail(baseFREQ)
       nocontrat nombre exposition zone puissance agevehicule ageconducteur bonus
678008   6114325      0    0.00548    E         6           4            40    68
678009   6114326      0    0.00274    E         4           0            54    50
678010   6114327      0    0.00274    E         4           0            41    95
678011   6114328      0    0.00274    D         6           2            45    50
678012   6114329      0    0.00274    B         4           0            60    50
678013   6114330      0    0.00274    B         7           6            29    54
       marque carburant densite region nbre
678008     12         E    2733     93    0
678009     12         E    3317     93    0
678010     12         E    9850     11    0
678011     12         D    1323     82    0
678012     12         E      95     26    0
678013     12         D      65     72    0
> nrow(baseCOUT)
[1] 26444
```

La base `baseFREQ` contient, par police, le nombre de sinistres en responsabilité civile déclaré par l'assuré pendant l'année d'observation (en plus de toutes les variables de la base de souscription). Parmi les variables d'intérêt,

- `densite` est la densité de population dans la commune où habite le conducteur principal,
- `zone` : zone A B C D E ou F, selon la densité en nombre d'habitants par km^2 de la commune de résidence (A étant la moins dense, F la plus dense)
- `marque` : marque du véhicule selon la table suivante (1 Renault Nissan ; 2 Peugeot Citroën ; 3 Volkswagen Audi Skoda Seat ; 4 Opel GM ; 5 Ford ; 6 Fiat ; 10 Mercedes Chrysler ; 11 BMW Mini ; 12 Autres japonaises et coréennes ; 13 Autres européennes ; 14 Autres marques et marques inconnues)

1. Attention R possêde beaucoup de fonctions pour manipuler les données, néanmoins ce ne sera jamais un SGBD du type SQL.

– `region` : code à 2 chiffres donnant les 22 régions françaises (code INSEE)
– `ageconducteur` : âge du conducteur principal en début de la couverture,
– `agevehicule` : âge du véhicule en début de période.

Remark 2.0.7. *Il existe aussi une variable décrivant le taux de bonus de l'assuré en début de période. Cette variable est très explicative de la sinistralité future, mais elle ne sera pas utilisée ici.*

Nous disposons aussi d'un numéro de police `no` permettant de fusionner les deux bases, et donc d'associer à la charge d'un sinistre les caractéristiques du conducteur et du véhicule.

Example 2.0.8. *En utilisant la fonction* `GraphiqueExposition`*, on peut visualiser sommairement l'impact des différentes variables sur la fréquence de sinsitres. La Figure 2.1 permet de visualiser l'impact de âge du conducteur principal et la Figure 2.2 l'impact de la zone d'habitation.*

```
> GraphiqueExposition()
> GraphiqueExposition(nom="zone",continu=FALSE)
```

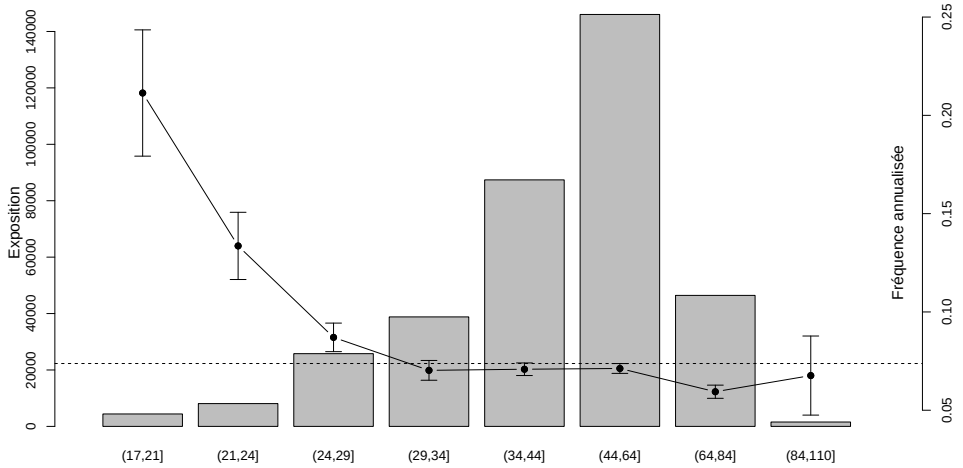


FIGURE 2.1 – Fréquence de sinistre en fonction de l'âge du conducteur principal

2.1 Les modèles linéaires généralisés

Depuis quelques années, l'outil principal utilisé en tarification est le modèle linéaire généralisé, développé par McCullagh & Nelder (1991), et dont la mise en oeuvre en assurance est détaillée dans Kaas et al. (2009), Denuit & Charpentier (2005), de Jong & Zeller (2008), Ohlsson & Johansson (2010) ou Frees (2009). Dans cette section, nous allons présenter le cadre des GLM, ainsi que leur mise en oeuvre sous R, avant de rentrer dans l'application en tarification dans les sections suivantes.

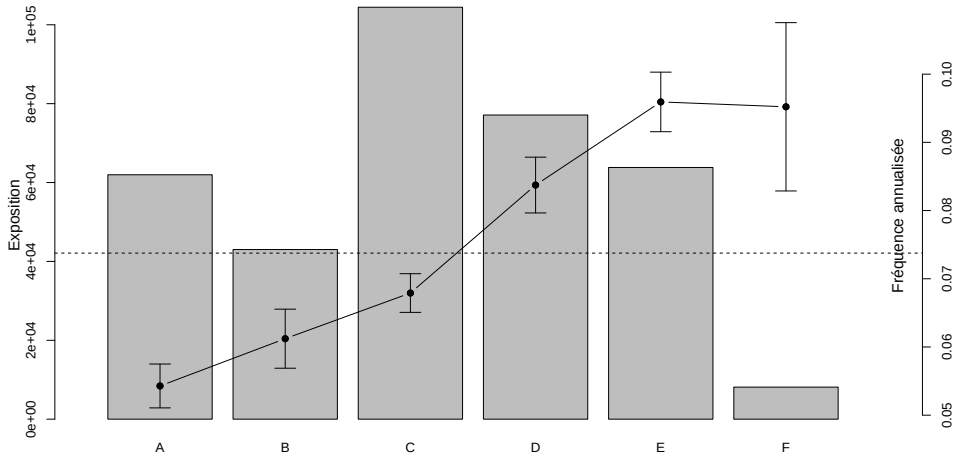


FIGURE 2.2 – Fréquence de sinistre en fonction de la zone d'habitation

2.1.1 Le cadre général des GLM

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

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

où a, b et c sont des fonctions et θ, ϕ les paramètres.

Example 2.1.1.

La *loi normale* $\mathcal{N}(\mu,\sigma^2)$ appartient à cette famille, avec $\theta = \mu$, $\phi = \sigma^2$, $a(\phi) = \phi$, $b(\theta) = \theta^2/2$ et

$$c(y,\phi) = -\frac{1}{2}\left(\frac{y^2}{\sigma^2} + \log(2\pi\sigma^2)\right), \quad y \in \mathbb{R},$$

Example 2.1.2.

La *loi de Poisson* $\mathcal{P}(\lambda)$ appartient à cette famille,

$$f(y|\lambda) = \exp(-\lambda)\frac{\lambda^y}{y!} = \exp\Big(y\log\lambda - \lambda - \log y!\Big), \quad y \in \mathbb{N},$$

avec $\theta = \log\lambda$, $\phi = 1$, $a(\phi) = 1$, $b(\theta) = \exp\theta = \lambda$ et $c(y,\phi) = -\log y!$.

Example 2.1.3.

La *loi binomiale* $\mathcal{B}(n,p)$ correspond au cas $\theta = \log\{p/(1-p)\}$, $a(\phi) = 1$, $b(\theta) = n\log(1+\exp(\theta))$, $\phi = 1$ et $c(y,\phi) = \log\binom{n}{y}$.

Example 2.1.4.

La *loi Gamma* caractèrisèe par la densitè suivante,

$$f(y|\mu,\nu) = \frac{1}{\Gamma(\nu)}\left(\frac{\nu}{\mu}\right)^{\nu} y^{\nu-1}\exp\left(-\frac{\nu}{\mu}y\right),\ \ y\in\mathbb{R}_+,$$

est également dans la famille exponentielle. Il faut choisir $\theta = -\dfrac{1}{\mu}$, $a(\phi)=\phi$, $b(\theta)=-\log(-\theta)$,

$$c(y,\phi) = \left(\frac{1}{\phi}-1\right)\log(y) - \log\left(\Gamma\left(\frac{1}{\phi}\right)\right)$$

et $\phi=\nu^{-1}$.

Pour une variable aléatoire Y dont la densité est de la forme exponentielle, alors

$$\mathbb{E}(Y) = b'(\theta) \quad \text{et} \quad \mathbb{V}(Y) = b''(\theta)\phi,$$

de telle sorte que la variance de Y apparaît comme le produit de deux fonctions,

- la première, $b''(\theta)$, qui dépend uniquement du paramètre θ est appelée *fonction variance*,
- la seconde est indépendante de θ et dépend uniquement de ϕ.

En notant $\mu=\mathbb{E}(Y)$, on voit que le paramètre θ est lié à la moyenne μ. La fonction variance peut donc être définie en fonction de μ , nous la noterons dorénavant $V(\mu)$.

Example 2.1.5.

Dans le cas de la loi normale, $V(\mu)=1$, dans le cas de la loi de Poisson, $V(\mu)=\mu$ alors que dans le cas de la loi Gamma, $V(\mu)=\mu^2$.

Notons que la fonction variance caractérise complètement la loi de la famille exponentielle. Chacune des lois de la famille exponentielle possède une fonction de lien spécifique, dite *fonction de lien canonique*, permettant de relier l'espérance μ au paramètre naturel (ou canonique) θ. Le lien canonique est tel que $g_\star(\mu)=\theta$. Or, $\mu=b'(\theta)$ donc $g_\star(\cdot)=b'(\cdot)^{-1}$.

Example 2.1.6.

Dans le cas de la loi normale, $\theta=\mu$ (`link='identity'`), dans le cas de la loi de Poisson, $\theta=\log(\mu)$ (`link='log'`) alors que dans le cas de la loi Gamma, $\theta=1/\mu$ (`link='inverse'`).

Sous R, la syntaxe des modèles linéaires généralisées est (par exemple) :

```
> glm(Y~X1+X2+X3+offset(log(Z)), family = quasipoisson(link='log'),
+ data = base, weights)
```

ce qui correspond à un modèle

$$\mathbb{E}(Y_i|\boldsymbol{X}_i) = \mu_i = g^{-1}\left(\boldsymbol{X}_i'\boldsymbol{\beta}+\xi_i\right) \text{ et } \mathbb{V}(Y_i|\boldsymbol{X}_i) = \frac{\phi V(\mu_i)}{\omega_i}$$

où `Y` est le vecteur des Y_i que l'on cherche à modéliser (le nombre de sinistres de la police i par exemple), `X1`, `X2` et `X3` sont les variables explicatives qui peuvent être qualitatives (on parlera de facteurs) ou quantitatives, `link='log'` indique que g est la fonction log, `family=poisson` revient à choisir une fonction variance V identité, alors que `family=quasipoisson` revient à choisir une fonction variance V identité avec un paramètre de dispersion ϕ à estimer, `offset` correspond à la variable ξ_i, et `weights` le vecteur ω_i.

Cette fonction `glm` calcule alors des estimateurs de $\boldsymbol{\beta}$ et ϕ, entre autres, car comme pour le modèle linéaire gaussien (la fonction `lm`) on peut obtenir des prédictions, des erreurs, ainsi qu'un grand nombre d'indicateurs relatifs à la qualité de l'ajustement.

2.1.2 Approche économétrique de la tarification

Cette famille de lois (dite *exponentielle*) va s'avérer être particulièrement utile pour construire des modèles économétriques beaucoup plus généraux que le modèle Gaussien usuel. On suppose disposer d'un échantillon $(Y_i, \boldsymbol{X}_i)$, où les variables $\boldsymbol{X}_i$ sont des informations exogènes sur l'assuré ou sur le bien assuré, et où Y_i est la variable d'intérêt, qui sera

- une variable booléenne prenant les valeurs $\{0, 1\}$, par exemple l'assuré i a-t-il été victime d'un accident l'an dernier, ou le sinistre i était-il très important,
- une variable de comptage, à valeurs dans $\mathbb{N}$, par exemple le nombre d'accidents de l'assuré i l'an passé,
- une variable positive, à valeurs dans $\mathbb{R}^+$, par exemple le coût du sinistre i, ou bien la durée entre la survenance et la déclaration du sinistre.

On supposera que, conditionnellement aux variables explicatives $\boldsymbol{X}$, les variables Y sont indépendantes et identiquement distribuées. En particulier, on partira d'un modèle de la forme

$$f(y_i|\theta_i, \phi) = \exp\left(\frac{y_i\theta_i - b(\theta_i)}{a(\phi)} + c(y_i, \phi)\right),$$

où l'on supposera que

$$g(\mu_i) = \eta_i = \boldsymbol{X}_i'\beta,$$

pour une fonction de lien $g(\cdot)$ donnée (on gardera ainsi un score *linéaire* en les variables explicatives), et où, pour rappel,

$$\mu_i = \mathbb{E}(Y_i|\boldsymbol{X}_i).$$

La fonction lien est la fonction qui permet de lier les variables explicatives $\boldsymbol{X}$ à la prédiction μ, alors que la loi apparaît via la fonction variance, sur la forme de l'hétéroscédasticité et l'incertitude associée à la prédiction. Le petit exemple ci-dessous permet de visualiser sur un petit de données simple six régressions GLM différentes,

```
> x <- c(1,2,3,4,5)
> y <- c(1,2,4,2,6)
> base <- data.frame(x,y)
> regNId <- glm(y~x,family=gaussian(link="identity"))
> regNlog <- glm(y~x,family=gaussian(link="log"))
> regPId <- glm(y~x,family=poisson(link="identity"))
> regPlog <- glm(y~x,family=poisson(link="log"))
> regGId <- glm(y~x,family=Gamma(link="identity"))
> regGlog <- glm(y~x,family=Gamma(link="log"))
```

La prédiction (ainsi qu'un intervalle de confiance) pour chacun de ces modèles est présentée sur la Figure 2.3. Le code de base pour obtenir la prédiction avec un intervalle de confiance (à 95%) est simplement

```
> visuel=function(regression,titre){
+ plot(x,y,pch=19,cex=1.5,main=titre,xlab="",ylab="")
+ abs <- seq(0,7,by=.1)
+ yp <- predict(regression,newdata=data.frame(x=abs),se.fit = TRUE,
+ type="response")
+ polygon(c(abs,rev(abs)),c(yp$fit+2*yp$se.fit,rev(yp$fit-2*yp$se.fit)),
+ col="light grey",border=NA)
+ points(x,y,pch=19,cex=1.5)
+ lines(abs,yp$fit,lwd=2)
```

```
+ lines(abs,yp$fit+2*yp$se.fit,lty=2)
+ lines(abs,yp$fit-2*yp$se.fit,lty=2)}
```

Pour les 6 modèles ajustés sur le petit jeu de données,

```
> par(mfrow = c(2, 3))
> visuel(regNId,"Gaussienne, lien identité")
> visuel(regPId,"Poisson, lien identité")
> visuel(regGId,"Gamma, lien identité")
> visuel(regNlog,"Gaussienne, lien logarithmique")
> visuel(regPlog,"Poisson, lien logarithmique")
> visuel(regGlog,"Gamma, lien logarithmique")
```

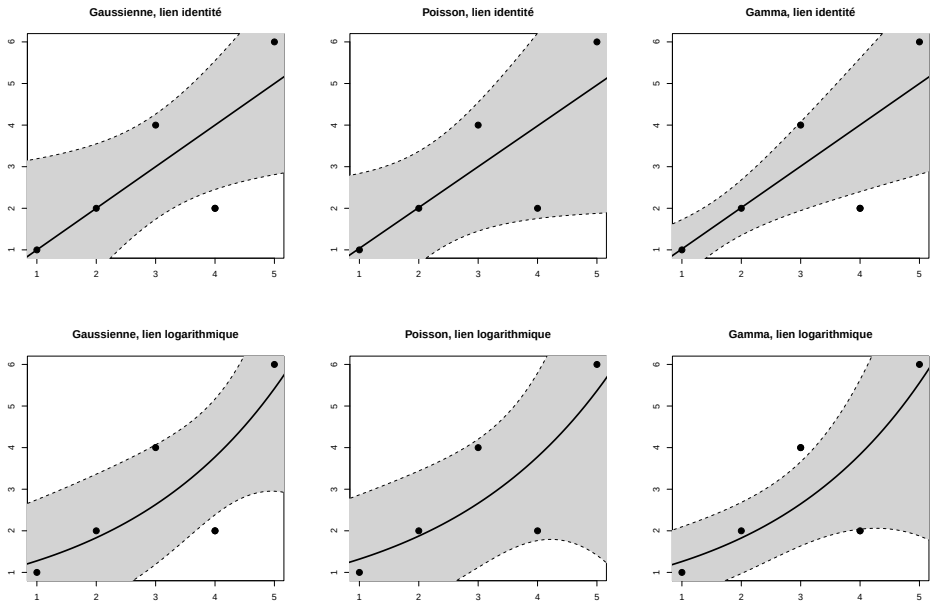


FIGURE 2.3 – Prédiction par 6 modèles linéaires différents, 3 lois et 2 fonctions de lien, avec les intervalles de confiance de prédiction.

Remark 2.1.7. *De la même manière qu'en économétrie linéaire, il est aussi possible d'allouer des poids à chacune des observations ω_i. Mais nous n'en parlerons pas trop ici. Il peut s'agir de pondération décroisantes avec le temps, attribuées à des années trop anciennes, si l'on utilise des données sur une période plus longue, par exemple.*

2.1.3 Estimation des paramètres

La loi de Y sachant $\boldsymbol{X}$ étant spécifiée, on peut obtenir numériquement les estimateurs de $\boldsymbol{\beta}$ et ϕ par maximisation de la vraisemblance, manuellement (en travaillant toutefois sur un échantillon de la base)

```
> set.seed(1)
> echantillon=sample(1:nrow(baseFREQ),size=100)
> logvraisemblance <- function(beta){
+ L=beta[1]+beta[2]*baseFREQ[echantillon,"ageconducteur"]
+ -sum(log(dpois(baseFREQ[echantillon,"nbre"],exp(L))))}
```

```
> optim(par=c(-3,.01),fn=logvraisemblance)$par
[1] -3.414 -0.027
```
ou directement via la fonction `glm` implèmentant un algorithme spècifique aux GLMs :
```
> glm(nbre~ ageconducteur,data=baseFREQ[echantillon,],family=poisson)

Call:  glm(formula = nbre ~ ageconducteur, family = poisson,
data = baseFREQ[echantillon,])

Coefficients:
  (Intercept)  ageconducteur
      -3.4154        -0.0269

Degrees of Freedom: 99 Total (i.e. Null);  98 Residual
Null Deviance:            9.21
Residual Deviance: 9.09         AIC: 15.1
```

Notons qu'il est aussi possible d'utiliser une régression linéaire pondérée. En effet, on cherche à maximiser ici une (log)-vraisemblance (ou une déviance comme nous le verrons plus tard), qui s'écrit dans le cas des modèles exponentiels,

$$\log(\mathcal{L}(\theta_1,\ldots,\theta_n,\phi,y_1,\ldots,y_n)) = \sum_{i=1}^{n}\left[\frac{y_i\theta_i - b(\theta_i)}{a(\phi)} + c(y_i,\phi)\right]. \tag{2.1}$$

On cherche les paramètres $\boldsymbol{\beta}$, il nous suffit de dèriver la log-vraisemblance par rapport au paramètre $\boldsymbol{\beta}$.

Notons $\mu_i = E(Y_i)$ et $\eta_i = g(\mu_i) = X_i\beta$, le prèdicteur linèaire pour la $i$ème observation parmi n.

Pour i et j donnè, on a

$$\frac{\partial \ln(\mathcal{L}_i)}{\partial \beta_j} = \frac{\partial \ln(\mathcal{L}_i)}{\partial \mu_i} \times \frac{\partial \mu_i}{\partial \beta_j} = (g^{-1})'(g(\mu_i)) \times \frac{y_i - \mu_i}{\mathrm{V}(Y_i)} X_{ij}.$$

Ainsi on obtient les èquations du score :

$$\sum_i \frac{\partial \ln(\mathcal{L}_i)}{\partial \beta_j} = \sum_i (g^{-1})'(X_i\beta) \times \frac{y_i - \mu_i}{\mathrm{V}(Y_i)} X_{ij} = 0,$$

pour tout j.

Ce qui correspondrait à la condition du premier ordre dans une régression pondérée, où la matrice de poids serait $W = [w_{i,j}]$, où $w_{i,j} = 0$ si $i \neq j$, et sinon

$$w_{i,i} = \frac{1}{\mathrm{V}(Y_i)}$$

Mais cette matrice de poids étant inconnue (elle dépend des paramètres que l'on cherche à estimer), on met en place une itération de régression pondérée, la matrice de poids étant calculée à partir des coefficients de l'étape précédante.

Dans le cas d'une régression log-Poisson, le code devient,

```
> X=baseFREQ[echantillon,"ageconducteur"]
> Y=baseFREQ[echantillon,"nbre"]
> beta=c(-1,1)
```

```
> BETA=matrix(NA,101,2)
> BETA[1,]=beta
> for(i in 2:101){
+ eta=beta[1]+beta[2]*X
+ mu=exp(eta)
+ w=mu
+ z=eta+(Y-mu)/mu
+ REG=lm(z~X,weights=w)
+ beta=REG$coefficients
+ BETA[i,]=beta
+ }
> BETA[85:101,]
       [,1]    [,2]
 [1,] -9.01  0.0960
 [2,] -6.64  0.0505
 [3,] -4.85  0.0111
 [4,] -3.83 -0.0149
 [5,] -3.47 -0.0254
 [6,] -3.42 -0.0269
 [7,] -3.42 -0.0269
 [8,] -3.42 -0.0269
 [9,] -3.42 -0.0269
[10,] -3.42 -0.0269
[11,] -3.42 -0.0269
[12,] -3.42 -0.0269
[13,] -3.42 -0.0269
[14,] -3.42 -0.0269
[15,] -3.42 -0.0269
[16,] -3.42 -0.0269
[17,] -3.42 -0.0269
```

qui converge très rapidement (vers les valeurs trouvées par la fonction `glm`).

2.1.4 Interprétation d'une régression

Considérons tout simplement une régression de la fréquence annuelle de sinistre sur l'âge du conducteur. On supposera un modèle Poissonnien.

```
> reg1 <- glm(nbre~ageconducteur+offset(log(exposition)),
+ data=baseFREQ,family=poisson(link="log"))
> summary(reg1)

Call:
glm(formula = nbre ~ ageconducteur + offset(log(exposition)),
    family = poisson(link = "log"), data = baseFREQ)

Deviance Residuals:
   Min      1Q  Median      3Q     Max
-0.568  -0.353  -0.261  -0.142  13.326
```

```
Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)   -2.13774    0.02078  -102.9   <2e-16 ***
ageconducteur -0.01017    0.00044   -23.1   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 171819  on 678012  degrees of freedom
Residual deviance: 171273  on 678011  degrees of freedom
AIC: 222045

Number of Fisher Scoring iterations: 6
```

Avec un lien logarithmique, le modèle est multplicatif. Le multiplicateur est ici

```
> exp(coefficients(reg1)[2])
ageconducteur
    0.9898836
```

Autrement dit, tous les ans, la probabilité d'avoir un accident diminue de $1 - 0.9898 = 1.011\%$.

Si l'on considère des classes d'âges (définies *a priori*, nous reviendrons par la suite sur la construction *optimale* des classes), on obtient la régression suivante :

```
> seuils <- c(17,21,25,30,40,50,60,70,80,120)
> baseFREQ$agecut <- cut(baseFREQ$ageconducteur,breaks=seuils)
> reg2 <- glm(nombre~agecut+offset(log(exposition)),data=
+ baseFREQ,family=poisson(link="log"))
> summary(reg2)

Call:
glm(formula = nombre ~ agecut + offset(log(exposition)),
    family = poisson(link = "log"),
    data = baseFREQ)

Deviance Residuals:
   Min      1Q  Median      3Q     Max
-0.657  -0.351  -0.260  -0.140  13.332

Coefficients:
               Estimate Std. Error z value Pr(>|z|)
(Intercept)     -1.5542     0.0328   -47.4   <2e-16 ***
agecut(21,25]   -0.5272     0.0419   -12.6   <2e-16 ***
agecut(25,30]   -0.9518     0.0387   -24.6   <2e-16 ***
agecut(30,40]   -1.1175     0.0353   -31.6   <2e-16 ***
agecut(40,50]   -1.0277     0.0350   -29.4   <2e-16 ***
agecut(50,60]   -1.1172     0.0356   -31.4   <2e-16 ***
agecut(60,70]   -1.2318     0.0387   -31.8   <2e-16 ***
agecut(70,80]   -1.2689     0.0428   -29.7   <2e-16 ***
agecut(80,120]  -1.2402     0.0674   -18.4   <2e-16 ***
---
```

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

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 171919  on 678012  degrees of freedom
Residual deviance: 170594  on 678004  degrees of freedom
AIC: 221425

Number of Fisher Scoring iterations: 6
```

Notons qu'il est aussi possible de taper directement

```
> reg2 <- glm(nbre~cut(ageconducteur,breaks=seuils)+offset(log(exposition)),
+ data=baseFREQ,family=poisson(link="log"))
```

La classe de référence est ici celle des jeunes conducteurs `(17,21]`. Relativement à cette classe, on note que toutes les classes ont une probabilité d'avoir un accident plus faible. Pour un conducteur de la classe `(30,45]`, on note qu'il a 66% de chances en moins d'avoir un accident dans l'année qu'un jeune conducteur,

```
> exp(coefficients(reg2)[4])
cut(ageconducteur, breaks = seuils)(30,45]
                                 0.3373169
```

On peut changer la classe de référence, par exemple `(30,40]`,

```
> baseFREQ$agecut =relevel(baseFREQ$agecut,"(30,40]")
> reg2 <- glm(nbre~agecut+offset(log(exposition)),data=
+ baseFREQ,family=poisson(link="log"))
> summary(reg2)

Call:
glm(formula = nbre ~ agecut + offset(log(exposition)), family = poisson(link = "log"),
    data = baseFREQ)

Deviance Residuals:
   Min      1Q  Median      3Q     Max
-0.657  -0.351  -0.260  -0.140  13.332

Coefficients:
                Estimate Std. Error z value Pr(>|z|)
(Intercept)    -2.67e+00   1.31e-02 -203.52  < 2e-16 ***
agecut(17,21]   1.12e+00   3.53e-02   31.67  < 2e-16 ***
agecut(21,25]   5.89e-01   2.92e-02   20.17  < 2e-16 ***
agecut(25,30]   1.65e-01   2.43e-02    6.79  1.1e-11 ***
agecut(40,50]   8.94e-02   1.80e-02    4.98  6.4e-07 ***
agecut(50,60]   5.37e-05   1.91e-02    0.00    0.998
agecut(60,70]  -1.16e-01   2.44e-02   -4.74  2.1e-06 ***
agecut(70,80]  -1.51e-01   3.05e-02   -4.95  7.5e-07 ***
agecut(80,120] -1.22e-01   6.04e-02   -2.02    0.043 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for poisson family taken to be 1)
```

```
    Null deviance: 171819  on 678012  degrees of freedom
Residual deviance: 170496  on 678004  degrees of freedom
AIC: 221282

Number of Fisher Scoring iterations: 6
```

qui inciterait à fusionner les classes `(30,40]` et `(50,60]`, ou avec comme classe de référence `(70,90]`,

```
> baseFREQ$agecut =relevel(baseFREQ$agecut,"(70,80]")
> reg2 <- glm(nbre~agecut+offset(log(exposition)),data=
+ baseFREQ,family=poisson(link="log"))
> summary(reg2)

Call:
glm(formula = nbre ~ agecut + offset(log(exposition)), family = poisson(link = "log"),
    data = baseFREQ)

Deviance Residuals:
   Min      1Q  Median      3Q     Max
-0.657  -0.351  -0.260  -0.140  13.332

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)    -2.8230     0.0275 -102.64  < 2e-16 ***
agecut(30,40]   0.1508     0.0305    4.95  7.5e-07 ***
agecut(17,21]   1.2689     0.0428   29.66  < 2e-16 ***
agecut(21,25]   0.7396     0.0379   19.52  < 2e-16 ***
agecut(25,30]   0.3162     0.0343    9.22  < 2e-16 ***
agecut(40,50]   0.2402     0.0301    7.98  1.5e-15 ***
agecut(50,60]   0.1509     0.0308    4.90  9.5e-07 ***
agecut(60,70]   0.0350     0.0344    1.02     0.31
agecut(80,120]  0.0287     0.0650    0.44     0.66
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 171819  on 678012  degrees of freedom
Residual deviance: 170496  on 678004  degrees of freedom
AIC: 221282

Number of Fisher Scoring iterations: 6
```

qui inciterait ici à fusionner les dernières classes. Toutefois, comme il s'agit de fusionner 3 classes d'âge ensemble, il convient de faire ici un test multiple,

```
> library(car)
> linearHypothesis(reg2,c("agecut(60,70]=0","agecut(80,120]=0"))
Linear hypothesis test
```

```
Hypothesis:
agecut(60,70] = 0
agecut(80,120] = 0

Model 1: restricted model
Model 2: nbre ~ agecut + offset(log(exposition))

  Res.Df Df Chisq Pr(>Chisq)
1 678006
2 678004  2  1.05       0.59
```

ce qui autorise la fusion des trois classes (et définir une classe des plus de 60 ans).

Au lieu de comparer à la classe des jeunes conducteurs, on peut aussi comparer au conducteur moyen.

```
> seuils = c(17,21,25,30,45,55,65,80,120)
> reg2 = glm(nombre~0+cut(ageconducteur,breaks=seuils),
+ data=nombre,family=poisson(link="log"),offset=log(exposition))
```

Les multiplicateurs sont alors

```
> reg2b <- glm(nombre~1,data=nombre,family=poisson(link="log"),
+ offset=log(exposition))
> moyenne <- exp(coefficients(reg2b))
> reg2c <- glm(nombre~0+cut(ageconducteur,breaks=seuils),
+ data=nombre,family=poisson(link="log"),offset=log(exposition))
> exp(coefficients(reg2c))/moyenne
```

Une personne de la classe `(17,21]` a ainsi 2.86 fois plus de chance que l'assuré moyen d'avoir un accident.

2.1.5 Extension à d'autres familles de lois

Les modèles linéaires généralisés ont été définis pour des lois (de Y, conditionnelles aux variables explicatives $\boldsymbol{X}$) appartenant à la famille exponentielle. Il est toutefois possible de généraliser. Les lois de `library(gamlss)` sont des lois à quatre paramètres, (μ, σ, ν, τ), où μ est un paramètre de localisation (e.g. la moyenne), σ un paramètre d'échelle (e.g. l'écart-type), et où ν et τ sont des paramètres d'asymétrie et d'épaisseur de queue (e.g. la skewness et la kurtosis). Ces quatre paramètres peuvent être fonction des variables explicatives au travers d'une fonction de lien,

$$\begin{cases} \mu = g_\mu^{-1}(\boldsymbol{X}\boldsymbol{\alpha}) \\ \sigma = g_\sigma^{-1}(\boldsymbol{X}\boldsymbol{\beta}) \\ \nu = g_\nu^{-1}(\boldsymbol{X}\boldsymbol{\gamma}) \\ \tau = g_\tau^{-1}(\boldsymbol{X}\boldsymbol{\delta}) \end{cases}$$

Parmi les lois classiques, on retrouvera celles données dans la Table 2.1.

Dans sa version la plus simple, on retrouve le modèle proposé par Gerber & Shiu (1994),

$$\begin{cases} Y_i = \boldsymbol{X}_i'\boldsymbol{\beta} + \varepsilon_i, \text{modèle en moyenne} \\ \log \varepsilon_i^2 = \boldsymbol{Z}_i'\boldsymbol{\alpha} + u_i, \text{modèle en variance} \end{cases}$$

où u_i est un bruit i.i.d. suivant une loi Gamma. Cette fonction particulière est obtenue à l'aide de la fonction `lm.disp` de `library(dispmod)`.

loi	R	μ	σ	ν	τ
Binomiale	`BI`	logit	-	-	-
Normale	`NO`	identité	log	-	-
Poisson	`PO`	log	-	-	-
Gamma	`GA`	logit	-	-	-
inverse Gaussienne	`IG`	log	log	-	-
Gumbel	`GU`	identité	log	-	-
lognormale	`LNO`	log	log	-	-
binomiale négative (Poisson-Gamma)	`NBI`	log	log	-	-
Poisson-inverse Gaussien	`PIG`	log	log	-	-
Weibull	`WEI`	log	log	-	-
zero inflated Poisson	`ZIP`	log	logit	-	-

TABLE 2.1 – Les différentes lois et modèles de `library(gamlss)`.

2.1.6 De la qualité d'une régression

Pour mesurer les performances d'une régression, ou plus généralement d'un modèle quel qu'il soit, il faut se donner une *fonction de risque* $R(\cdot,\cdot)$ qui mesure la distance entre Y et sa prédiction $\widehat{Y}$ (on notera indifféremment $\widehat{Y}$ ou $\widehat{\mu}$). Classiquement, on utilise la norme L^2, correspond à l'erreur quadratique $R(Y,\widehat{Y}) = [Y-\widehat{Y}]^2$ ou la norme L^1 , correspondant à l'erreur absolue $R(Y,\widehat{Y}) = |Y-\widehat{Y}|$.

Si on reprend l'exemple de la section 2.1.2, les résidus sont représenté sur la Figure 2.4. Les résidus bruts correspondent à la différence entre Y_i et $\widehat{Y}_i$. Les résidus de Pearson sont des résidus standardisés,

$$\widehat{\varepsilon}_i = \frac{Y_i - \widehat{Y}_i}{\sqrt{V(\widehat{Y}_i)}}$$

où V est la fonction variance. Si on reprend le jeu de données utilisé pour introduire les GLM, utilisons la fonction suivante pour visualiser l'allure des résidus

```
> residus <- function(regression,titre){
+ RNIr <- residuals(regression,type="response")
+ RNIp <- residuals(regression,type="pearson")
+ RNId <- residuals(regression,type="deviance")
+ plot(x,RNIr,type="b",col="grey",main=titre,xlab="",ylab="")
+ lines(x, RNIp,type="b",pch=19)
+ lines(x, RNId,type="b",pch=3,lty=2)}
```

La Figure 2.4 permet de visualier les trois sortes de résidus, bruts en gris ($Y-\widehat{Y}$), et les résidus de Pearson et de déviance en noir.

```
> par(mfrow = c(2, 3))
> residus(regNId,"Gaussienne, lien identité")
> residus(regPId,"Poisson, lien identité")
> residus(regGId,"Gamma, lien identité")
> residus(regNlog,"Gaussienne, lien logarithmique")
> residus(regPlog,"Poisson, lien logarithmique")
> residus(regGlog,"Gamma, lien logarithmique")
```

Les résidus de Pearson permettent de prendre en compte de l'hétéroscédasticité qui ap-

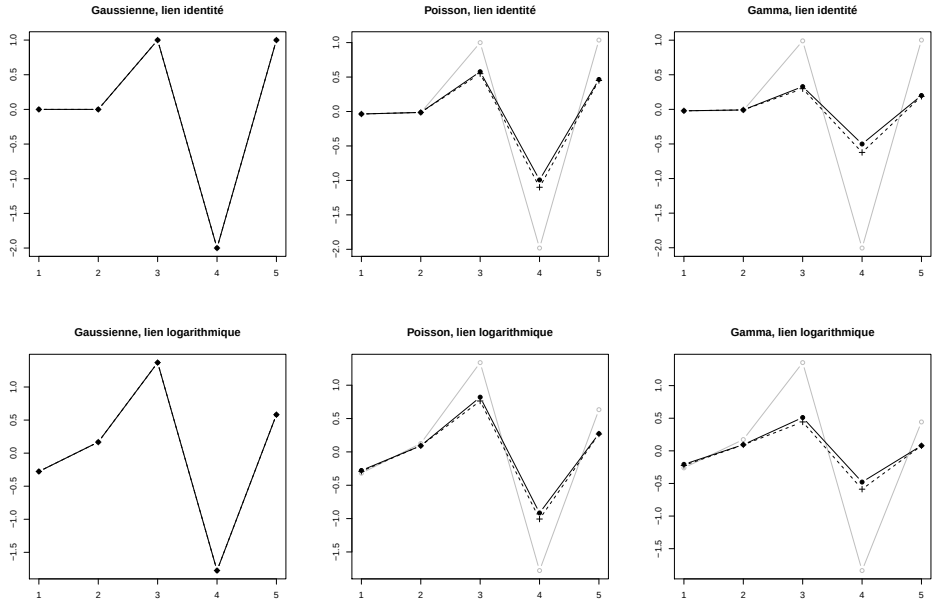


FIGURE 2.4 – Résidus bruts, de Pearson et de déviance sur 6 régressions GLM.

paraîtra dès lors que l'on quite le modèle Gaussien (la fonction variance ne sera alors plus constante). Davison & Snell (1991) revient longuement sur l'analyse des résidus dans le cadre de modèles linéaires généralisés. Rappelons que l'outil de base pour quantifier la qualité de la régression est la *déviance*

$$D(\boldsymbol{\beta}) = -2[\log \mathcal{L}(\widehat{\boldsymbol{\beta}}|Y) - \log \mathcal{L}_\star(Y)]$$

où $\log \mathcal{L}(\boldsymbol{\beta}|Y)$ désigne la log-vraisemblance du modèle, et où $\log \mathcal{L}_\star(Y)$ est la log-vraisemblance saturée (obtenue avec un modèle parfait).

```
> par(mfrow = c(1, 1))
>
> logLik(reg1)
'log Lik.' -111021 (df=2)
> deviance(reg1)
[1] 171273
> AIC(reg1)
[1] 222045
> -2*logLik(reg1)+2*2
[1] 222045
attr(,"nobs")
[1] 678013
attr(,"df")
[1] 2
attr(,"class")
[1] "logLik"
```

Dans un souci de partimonie, on pénalise souvent log-vraisemblance par le nombre de paramètres, ce qui correspond au critère d'information d'Akaike (AIC, en multipliant par 2). On

peut également définir le critère de Schwartz,

$$\begin{cases} AIC : -2\log\mathcal{L}(\widehat{\boldsymbol{\beta}}) + 2k \\ BIC : -2\log\mathcal{L}(\widehat{\boldsymbol{\beta}}) + k\log(n) \end{cases}$$

Il existe aussi un critère d'Aikaike *corrigé* (introduit par Hurvich & Tsai (1995)) dans le cas où l'on a trop peu d'observations. Toutes ces fonctions peuvent être obtenues à l'aide de la fonction `AIC` de `library(aod)` ou `BIC` de `library(BMA)`, ou encore `extractAIC` avec comme paramètre `k=log(nrow(base))`.

```
> extractAIC(reg1,k=2)[2]
[1] 222045
> extractAIC(reg1,k=log(nrow(baseFREQ)))[2]
[1] 222068
```

On peut comparer tous les modèles via

```
> AIC(reg1,reg2)
     df    AIC
reg1  2 222045
reg2  9 221282
```

2.2 Régression logistique et arbre de régression

Avant de modéliser la fréquence de sinistre, et le coût inidividuel des sinistres, nous allons évoquer rapidement les modèles binomiaux (avec une variable réponse Y de type 0 ou 1), en présentant la solution proposée par les GLM (régression logistique) mais aussi parler de l'utilisation des arbres de régression.

2.2.1 La régression logistique ou probit

La régression logistique suppose que si $\pi(Y|\boldsymbol{X}) = \mathbb{P}(Y = 1|\boldsymbol{X})$, alors

$$\frac{\pi(Y|\boldsymbol{X})}{1-\pi(Y|\boldsymbol{X})} = \frac{\mathbb{P}(Y=1|\boldsymbol{X})}{\mathbb{P}(Y=0|\boldsymbol{X})} = \exp\left(\boldsymbol{X}\boldsymbol{\beta}\right)$$

Dans le cas du modèle probit, on suppose qu'il existe un modèle latent Gaussien, tel que

$$Y_i^\star = \boldsymbol{X}_i'\boldsymbol{\beta} + \varepsilon_i$$

et que $Y_i = 0$ si $Y_i^\star < s$, et $Y_i = 1$ si $Y_i^\star > s$, et $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$.

La synthaxe de ces deux modèles est très proche, car seule la fonction de lien change.

```
> baseFREQ$touche=baseFREQ$nbre>0
> reglogit <- glm(touche~ageconducteur,
+ data= baseFREQ,family=binomial(link="logit"))
> regprobit <- glm(touche~ageconducteur,
+ data= baseFREQ,family=binomial(link="probit"))
> age <- seq(17,100)
> AGE <- data.frame(ageconducteur=age,exposition=1)
> Yl <- predict(reglogit,AGE,type="response")
> Yp <- predict(regprobit,AGE,type="response")
```

On notera que ces deux modèles donnent des prédictions très proches, comme le montre la Figure 2.5 (différence - en valeur absolue - inférieure à 0.5%).

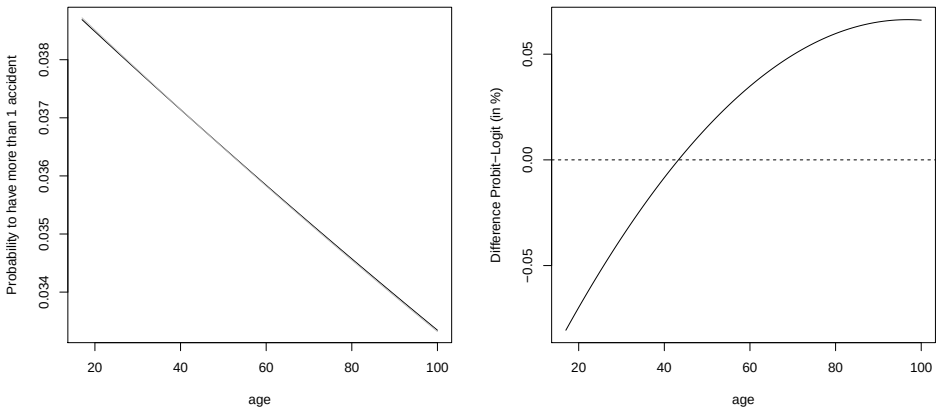


FIGURE 2.5 – Régression logistique (logit) versus modèle latent Gaussien (probit) pour prédire la probabilité d'avoir au moins un accident dans l'année, en fonction de l'âge du conducteur principal. .

2.2.2 Les arbres de régression

Les arbres de régression sont des outils nonparamétriques de segmentation. Dans un arbre de décision, on cherche à détecter des critères permettant de répartir les individus en 2 classes, caractérisées par $Y = 0$ et $Y = 1$. On commence par choisir la variable, qui, par ses modalités, sépare le mieux les individus de chacune des classes. On constitue alors un premier *noeud*. On réintère alors la procédure sur chaque nouveau noeud. Dans la méthode CART (*Classification And Regression Tree*), on regarde toutes les possibilités. On continue soit jusqu'à ce qu'il ne reste plus qu'un seul individu dans chaque noeud, soit suivant un critère d'arrêt. Les critères de discrimination et de constitution des noeuds sont généralement les suivants,

- lorsque les variables explicatives X_j sont qualitatives, ou discrètes, on utilise la distance du χ^2 (on parle d'arbre CHAID),
- en présence de variables de tous types, on peut utiliser l'indice de Gini (méthode CART),
- ou l'entropie (méthode C5.0),

Pour un variable continue, on distinguera $\{X_1 \leq s\}$ et $\{X_1 > s\}$. Pour une variable qualitative, on distinguera $\{X_1 \in \mathcal{A}\}$ et $\{X_1 \notin \mathcal{A}\}$.

Pour chacune des variables, on regarde l'ensemble des classifications possibles. Par exemple pour l'âge du conducteur, on posera

```
> ages <- sort(unique(baseFREQ$ageconducteur))
> k <- 5
> classe0 <- baseFREQ$ageconducteur<=ages[k]
> classe1 <- baseFREQ $ageconducteur> ages[k]
```

Une fois constituées les 2 classes, on calcule un des critères possibles (distance du chi-deux, critère de Gini, etc).

Si on regarde la décomposition obtenue sur le premier noeud, on observe que pour les conducteurs de moins de 25 ans, la probabilité d'avoir un accident est de 10%, contre 5% pour les conducteurs de plus de 25 ans. Dans le cas des régions, avec une distance du chi-deux, on

cherche à minimiser

$$\chi^2 = - \sum_{\text{classe}\in\{0,1\}} \sum_{y\in\{0,1\}} \frac{[n_{\text{classe},y} - n^{\perp}_{\text{classe},y}]^2}{n^{\perp}_{\text{classe},y}}$$

où $n_{\text{classe},y}$ désigne le nombre de personnes dans la classe considérée pour lesquelles la variable Y prend la modalité y.

```
> DISTANCE <- rep(NA,length(ages))
> names(DISTANCE)=ages
> for(k in 2:(length(ages)-1)){
+ classe0 <- baseFREQ$ageconducteur<=ages[k]
+ classe1 <- baseFREQ $ageconducteur> ages[k]
+ M=matrix(
+ rbind(c(sum(baseFREQ$touche[classe0]==FALSE),
+ sum(baseFREQ$touche[classe0]==TRUE)),
+ c(sum(baseFREQ$touche[classe1]==FALSE),
+  sum(baseFREQ$touche[classe1]==TRUE))),2,2)
+ DISTANCE[k] <- (-chisq.test(M)$statistic)}
```

Ici, le meilleur découpage possible est `(17,23]` et `(23,85]`,

```
> which.min(DISTANCE)
23
 6
```

ce que l'on peut visualiser sur la Figure 2.6,

```
> plot(ages,DISTANCE,type="b",ylab="distance du chi-deux",pch=3)
```

avec une borne supérieure entre 21 et 24 ans (optimale semble-t-il à 23 ans).

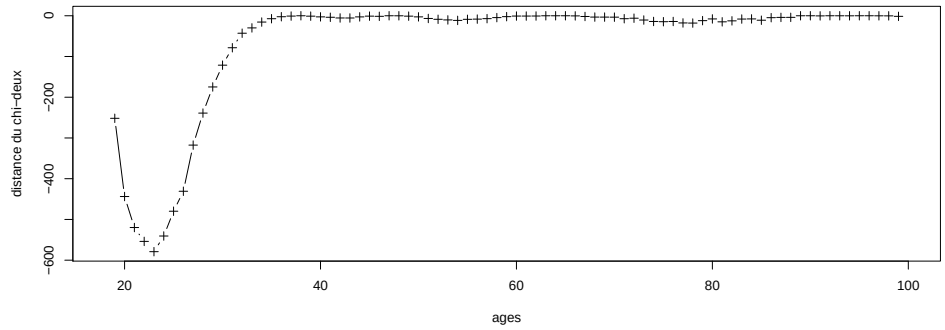


FIGURE 2.6 – Evolution de χ^2 lors du découpage en 2 classes$(17, k]$ et $(k, 100]$.

Manifestement, la première leçon que l'on peut tirer de ce graphique est que s'il convient de découper l'âge du conducteur principal en 2 classes, elles opposeront les jeunes conducteurs aux autres. A la seconde étape, on cherche une autre partition, en considérant la précédente comme acquise,

```
> k1 <- which.min(DISTANCE)
> DISTANCE <- rep(NA,length(ages))
> names(DISTANCE)=ages
> for(k in 2:(length(ages)-1)){
```

```
+ if(k!=k1){
+ classe0 <- (baseFREQ$ageconducteur<=ages[k])&(baseFREQ$ageconducteur<=ages[k1])
+ classe2 <- (baseFREQ$ageconducteur>ages[k])&(baseFREQ$ageconducteur>ages[k1])
+ classe1 <- 1-classe0-classe2
+  M=matrix(
+  rbind(c(sum(baseFREQ$touche[classe0]==FALSE),
+  sum(baseFREQ$touche[classe0]==TRUE)),
+  c(sum(baseFREQ$touche[classe1]==FALSE),
+  sum(baseFREQ$touche[classe1]==TRUE)),
+  c(sum(baseFREQ$touche[classe2]==FALSE),
+  sum(baseFREQ$touche[classe2]==TRUE))),3,2)
+  DISTANCE[k] <- (-chisq.test(M)$statistic)
+  }}
> which.min(DISTANCE)
99
82
```

En regardant la Figure 2.7, on observe qu'une fois fixé la borne supérieure caractérisant les '*jeunes*', la troisième classe constituée est une classe de personnes âgées (un peu au delà de 80 ans),

```
> plot(ages,DISTANCE,type="b",ylab="distance du chi-deux",pch=3)
```

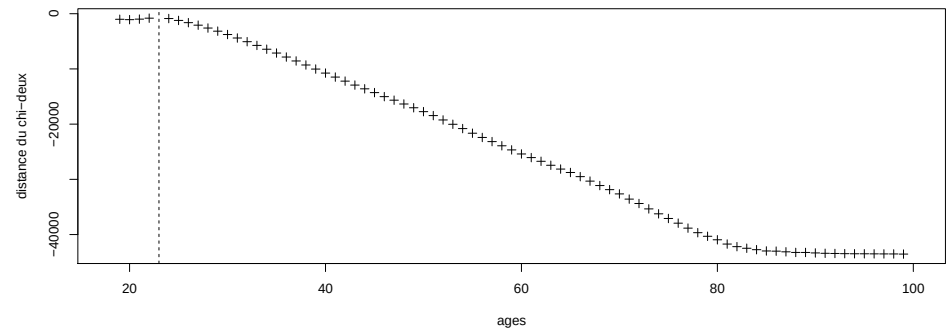


FIGURE 2.7 – Evolution de χ^2 lors du découpage en 3 classes $(17, k]$, $(17, 23]$ et $(23, 100]$, ou $(17, 23]$, $(23, k]$ et $(k, 100]$.

Parmi les autres critères, on peut aussi utiliser la distance de Gini,

$$G = -\sum_{\text{classe}\in\{0,1\}} \frac{n_{\text{classe}}}{n} \sum_{y\in\{0,1\}} \frac{n_{\text{classe},y}}{n_{\text{classe}}} \left(1 - \frac{n_{\text{classe},y}}{n_{\text{classe}}}\right)$$

ou l'entropie,

$$E = -\sum_{\text{classe}\in\{0,1\}} \frac{n_{\text{classe}}}{n} \sum_{y\in\{0,1\}} \frac{n_{\text{classe},y}}{n_{\text{classe}}} \log\left(\frac{n_{\text{classe},y}}{n_{\text{classe}}}\right)$$

Les arbres permettent une lecture relativement aisée pour l'utilisateur, et reposent sur des techniques nonparamétriques. Aussi, contrairement aux méthodes GLM que nous verrons par la suite, le choix des lois ou la recherche d'éventuelles nonlinéarités n'intervient pas ici. Les arbres

sont également peu sensibles aux points abèrrants (*outliers* en anglais). Mais les arbres, de par leur construction, posent aussi certains soucis. En particulier, on ne peut pas revenir en arrière, et le séquencement est très important.

2.2.3 Probabilité d'avoir (au moins) un sinistre dans l'année

A titre d'illustration, étudions la probabilité d'avoir au moins un sinistre dans l'année. Par défaut, l'arbre crée autant de classes que l'on a d'âges (vus en tant que variable discrete),

```
> library(tree)
> arbre=tree((nombre>0)~ageconducteur,data=baseFREQ,split="gini")
> age=data.frame(ageconducteur=18:90)
> y=predict(arbre,newdata=data.frame(ageconducteur=age))
> plot(age$ageconducteur,y,xlab="",ylab="")
```

Si l'on souhaite coupe les branches de l'arbre, on peut utiliser l'option `mincut` pour dire qu'on ne peut couper davantage qu'à condition de constituer des classes dont le nombre d'invidus à l'intérieur soit suffisamment élevé. ,

```
> arbre2=tree((nombre>0)~ageconducteur,data=baseFREQ,split="gini",mincut=50000)
> y2=predict(arbre2,newdata=data.frame(ageconducteur=age))
> lines(age$ageconducteur,y2,type="s",lwd=2)
> arbre3=tree((nombre>0)~ageconducteur,data=baseFREQ,split="gini",mincut=200000)
> y3=predict(arbre3,newdata=data.frame(ageconducteur=age))
> lines(age$ageconducteur,y3,type="s",col="grey",lwd=2)
```

On obtient alors les classes décrites sur la figure 2.8.

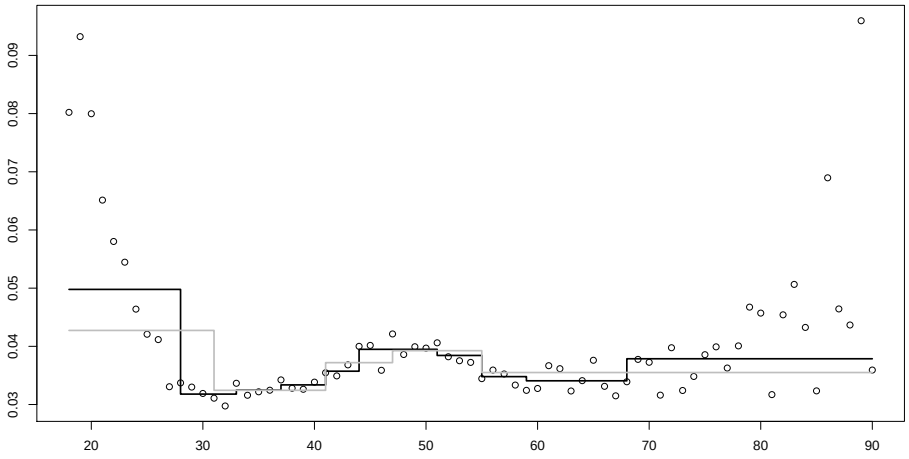


FIGURE 2.8 – Prédiction par arbre de régression, avec plus ou moins de classes d'âge, les points correspondent aux moyennes empiriques par âge (un noeud par âge), et les traits aux classes obtenues en imposant une taille suffisante par classe.

Mais le gros intérêt des arbres est de pouvoir visualiser le découpage et la structure d'arbre, comme sur la figure 2.9.

```
> plot(arbre2)
> text(arbre2)
> plot(arbre3)
> text(arbre3)
```

Sur ces arbres, la hauteur des branches reflète le gain (si on regarde l'arbre de manière descendante) ou la perte (si on le regarde de manière ascendante) en terme d'indice de Gini que l'on a en coupant les classes. Plus la branche est longue, plus la discrimination est forte.

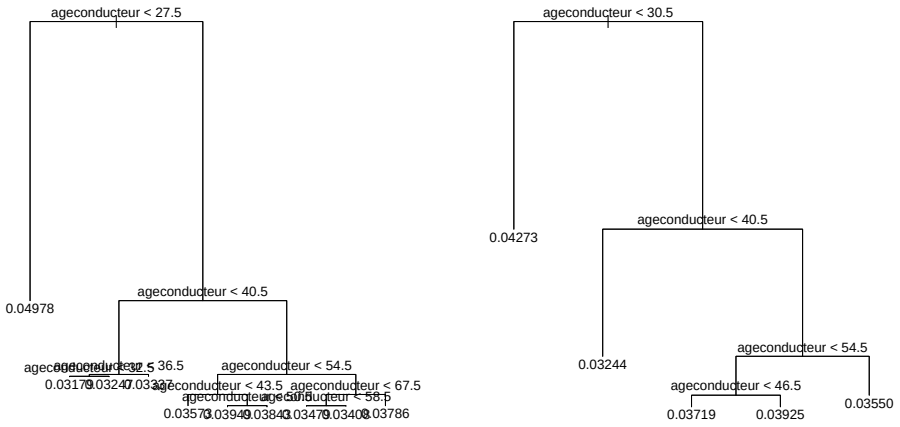


FIGURE 2.9 – Stucture des arbres de régression, avec `arbre2` à gauche, et `arbre3` à droite.

On notera toutefois que la stratégie optimale n'est peut être pas de supposer le risque constant par classe, comme le montre la Figure 2.10,,

```
> plot(age$ageconducteur,y,xlab="",ylab="")
> lines(age$ageconducteur,y3,type="s",col="grey",lwd=2)
> reg01.splines <- glm((nombre>0)~bs(ageconducteur,10),
+ family=binomial(link="logit"),data=baseFREQ)
> pred01.splines <- predict(reg01.splines,newdata=age,type="response")
> lines(age$ageconducteur,pred01.splines,lwd=2,col="black")
```

2.2.4 Probabilité d'avoir beaucoup de sinistres dans l'année

Une variable particulièrement intéressante est la probabilité d'avoir beaucoup d'accidents dans l'année. Mais une (grosse) partie des assurés n'étant dans la base que quelques semaines, ou quelques mois, il convient de recalculer les nombres annuels de sinistres, en divisant le nombre de sinistres observé par l'exposition, et en mettant un poids proportionnel à l'exposition (comme nous l'avions mentionné au début du chapitre).

Sur la Figure 2.11, on retrouve le fait que les jeunes conducteurs ont un comportement particulièrement risqué (à condition d'autoriser les classes de faible effectif).,,

```
> arbre1 <- tree((nbre/exposition>2) ~ ageconducteur ,
+ data=baseFREQ,weights=exposition,split="gini",mincut = 10000)
> plot(arbre1,type="proportional")
```

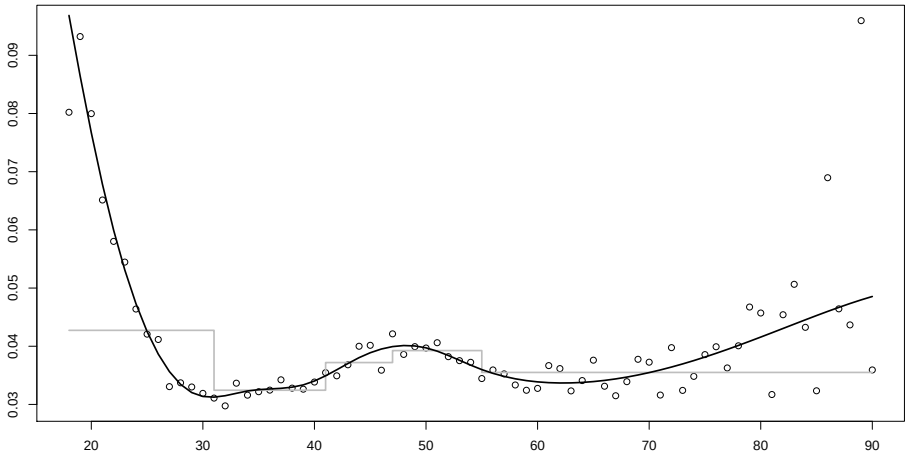


FIGURE 2.10 – Prédiction par arbre de régression, avec une régression logisitique sur une variable lissée (par splines).

```
> text(arbre1)
> arbre2 <- tree((nbre/exposition>2) ~ ageconducteur ,
+ data=baseFREQ,weights=exposition,split="gini",mincut = 100000)
> plot(arbre2,type="proportional")
> text(arbre2)
```

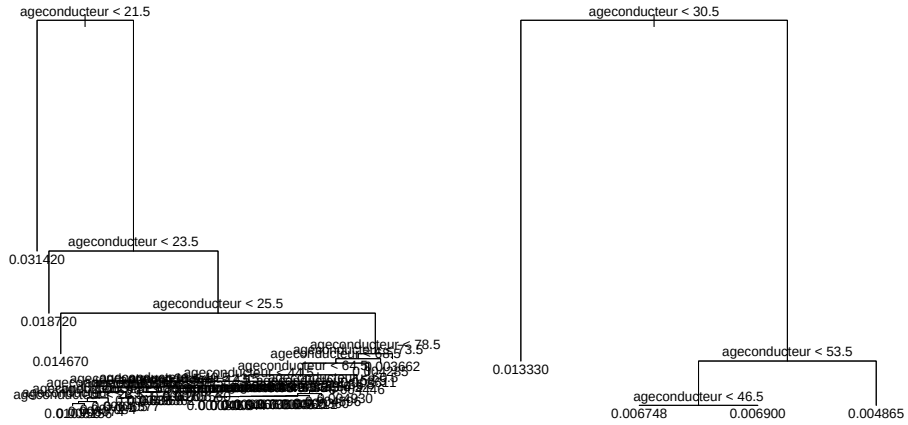


FIGURE 2.11 – Stucture des arbres de régression, avec `arbre1` à gauche, et `arbre2` à droite.

On peut d'ailleurs visualiser ces probabilités sur la Figure 2.12,

```
> ARBRE <- tree((nbre/exposition>2) ~ ageconducteur ,
```

```
+ data=baseFREQ,weights=exposition,split="gini",mincut = 10000)
> age=data.frame(ageconducteur=18:90)
> y=predict(ARBRE,newdata=data.frame(ageconducteur=age))
> y2=predict(ARBRE2,newdata=data.frame(ageconducteur=age))
> plot(age$ageconducteur,y,xlab="",ylab="")
> lines(age$ageconducteur,y2,type="s",col="grey",lwd=2)
> reg02.splines <- glm((nbre/exposition>2)~bs(ageconducteur,10),
+ family=binomial(link="logit"),weights=exposition,data=baseFREQ)
> pred02.splines <- predict(reg02.splines,newdata=age,type="response")
> lines(age$ageconducteur,pred02.splines,lwd=2,col="black")
```

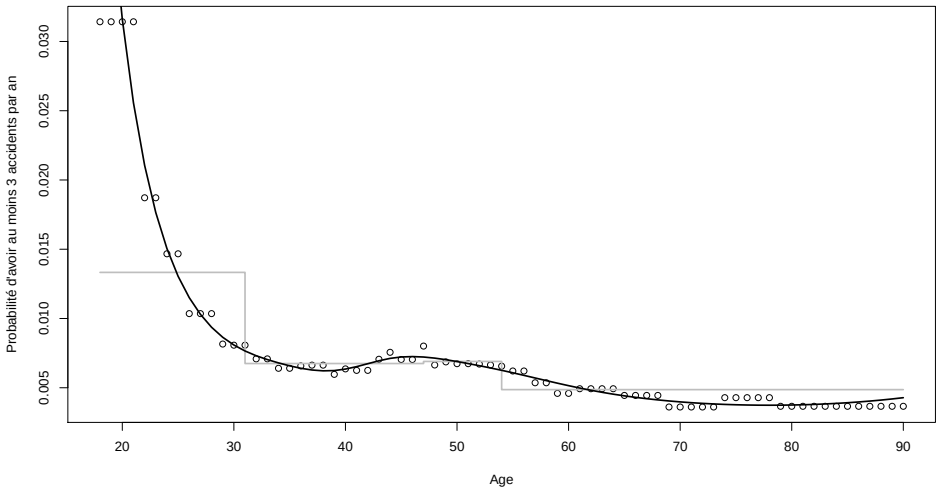


FIGURE 2.12 – Probabilité d'avoir au moins 3 accidents par an, arbre et régression logistique (sur variable lissée par splines).

Mais au lieu de n'étudier que l'âge du conducteur, on peut regarder aussi l'impact des autres varaibles. Sur la Figure 2.13, on retrouve le fait que les jeunes conducteurs ont un comportement particulièrement risqué (à condition d'autoriser les classes de faible effectif), mais que la zone d'habitation est aussi un facteur important, avec un risque plus élevé dans les villes : dans les zones `d`, `e` et `f` (plus de 500 habitants par km^2), la probabilité d'avoir au moins trois accidents est deux fois plus élevée - en tous les cas pour les conducteurs de plus de 30 ans - que dans les zones `a`, `b` et `c`

```
> arbre1 <- tree((nbre/exposition>2) ~ ageconducteur ,
+ data=baseFREQ,weights=exposition,split="gini",mincut = 10000)
> plot(arbre1,type="proportional")
> text(arbre1)
> arbre2 <- tree((nbre/exposition>2) ~ ageconducteur ,
+ data=baseFREQ,weights=exposition,split="gini",mincut = 100000)
> plot(arbre2,type="proportional")
> text(arbre2)
```

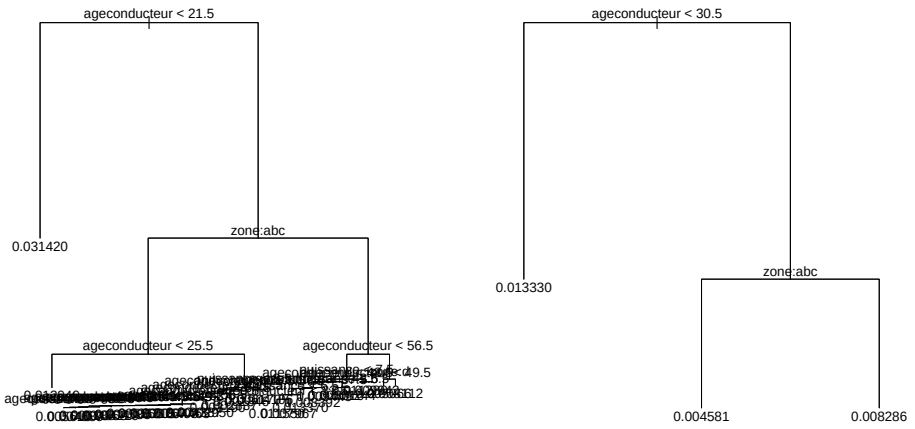


FIGURE 2.13 – Stucture des arbres de régression, avec `arbre1` à gauche, et `arbre2` à droite.

2.2.5 Probabilité d'avoir un gros sinistre dans l'année

Cette étude sera particulièrement intéressante pour écrêter les gros sinistres (nous reviendrons sur ce point dans la section 2.5.3), lors de la modélisation des coûts individuels de sinistres. On supposera (arbitrairement) que les *gros* sinistres sont ceux dont le montant dépasse 20 000 euros, ce qui concerne un peu plus de 200 sinistres,,,

```
> sum(baseCOUT$cout>20000)
[1] 215
```

Au lieu de modéliser les variables qui pourrait expliquer le fait d'avoir (ou pas) un accident, comme dans la section précédante, on va essayer de voir s'il y a des variables qui pourraient expliquer le fait d'avoir (ou pas) un très gros sinistre.

```
> gs <- baseCOUT$nocontrat[baseCOUT$cout>30000]
> baseFREQ$GS=0
> baseFREQ$GS[baseFREQ$nocontrat%in% gs]=1
> ARBRE <- tree(GS ~ puissance  + agevehicule + puissance + zone+ ageconducteur ,
+ data=baseFREQ,split="gini",mincut = 50000)
> ARBRE
node), split, n, deviance, yval
      * denotes terminal node

 1) root 678013 150.000 2.212e-04
   2) ageconducteur < 27.5 61023  25.990 4.261e-04 *
   3) ageconducteur > 27.5 616990 124.000 2.010e-04
     6) puissance < 5.5 212815  27.000 1.269e-04
      12) ageconducteur < 50.5 128654  12.000 9.327e-05
        24) ageconducteur < 36.5 51423   7.999 1.556e-04 *
        25) ageconducteur > 36.5 77231   4.000 5.179e-05 *
      13) ageconducteur > 50.5 84161  15.000 1.782e-04 *
     7) puissance > 5.5 404175  96.980 2.400e-04
```

```
14) zone: B,C,E,F 251268  49.990 1.990e-04
  28) agevehicule < 6.5 134878  20.000 1.483e-04
    56) ageconducteur < 49.5 76446   7.999 1.046e-04 *
    57) ageconducteur > 49.5 58432  12.000 2.054e-04 *
  29) agevehicule > 6.5 116390  29.990 2.578e-04
    58) agevehicule < 11.5 61902  17.990 2.908e-04 *
    59) agevehicule > 11.5 54488  12.000 2.202e-04 *
15) zone: A,D 152907  46.990 3.074e-04
  30) puissance < 6.5 52214  20.990 4.022e-04 *
  31) puissance > 6.5 100693  25.990 2.582e-04
    62) ageconducteur < 46.5 50219  13.000 2.589e-04 *
    63) ageconducteur > 46.5 50474  13.000 2.576e-04 *
```

On note qu'en fonction de la zone d'habitation, de la puissance du véhicule et de l'ancienneté du véhicule, on peut déterminer avec une bonne assurance la probabilité d'avoir un très gros sinistre. Et manifestement, une variable particulièrement importante est l'âge du conducteur (avec là encore un risque élevé pour les jeunes conducteurs) mais aussi la puissance du véhicule (si on veut suffisamment d'assurés par classes, l'âge du conducteur disparait au profit de la puissance, faute d'effectifs suffisants). Si on trace l'arbre, on obtient le dessin de la Figure 2.14

```
> ARBRE2 <- tree(GS ~ puissance  + agevehicule + puissance + zone+ ageconducteur ,
+ data=baseFREQ,split="gini",mincut = 100000)
> plot(ARBRE,type="proportional")
> text(ARBRE)
> plot(ARBRE2,type="proportional")
> text(ARBRE2)
```

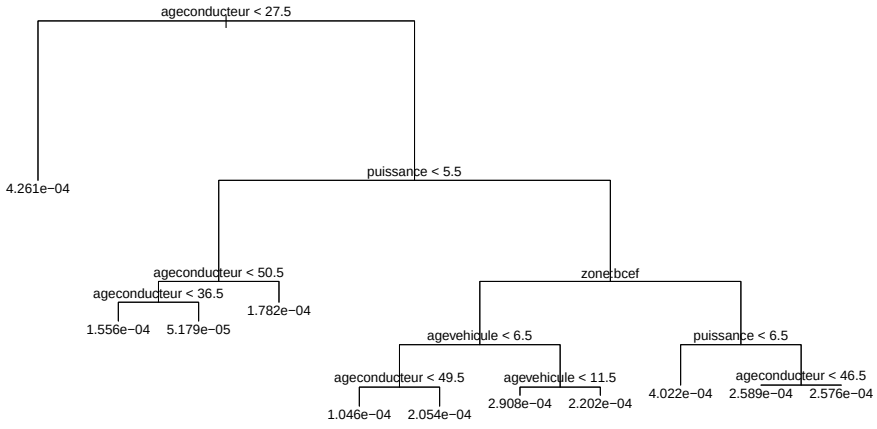


FIGURE 2.14 – Arbres de régression, pour expliquer la probabilité d'avoir (ou pas) un gros sinistre, en fonction de la densité de population, de l'ancienneté du véhicule, et de sa puissance.

2.3 Modéliser la fréquence de sinistralité

Nous avions vu en introduction à ce chapitre que la formule de base pour calculer une prime pure qui tiendrait compte de variables explicatives $\boldsymbol{X}$ est

$$\mathbb{E}(S|\boldsymbol{X}) = \mathbb{E}(N|\boldsymbol{X}) \cdot \mathbb{E}(Y_i|\boldsymbol{X}).$$

La première étape est de pouvoir modéliser la fréquence de sinistres $\mathbb{E}(N|\boldsymbol{X})$. Classiquement, les actuaires ont longtemps raisonné par *classes de risques*, c'est à dire en supposant les variables $\boldsymbol{X}$ qualitatives. Nous commencerons par évoquer ce cas (et en particulier la méthode des marges) pour introduire ensuite le cas où des variables explicatives sont continues.

2.3.1 Un peu d'analyse descriptive

Une hypothèse forte de la loi de Poisson est que $\mathbb{E}(N) = \mathbb{V}(N)$. Si l'on compare les valeurs numériques, cela donne l'ajustement suivant, si l'on estime le paramètre par la méthode des moments (ou par maximum de vraisemblance, `ML` qui ici coïncident) :

```
> N <- baseFREQ$nbre
> library(vcd)
> gof <- goodfit(N,type= "poisson",method= "ML")
> gof

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

 count observed       fitted
     0   653069 6.520780e+05
     1    23571 2.543248e+04
     2     1298 4.959614e+02
     3       62 6.447862e+00
     4        5 6.287020e-02
     5        2 4.904153e-04
     6        1 3.187879e-06
     7        0 1.776204e-08
     8        1 8.659485e-11
     9        1 3.752655e-13
    10        0 1.463618e-15
    11        2 5.189485e-18
    12        0 1.686678e-20
    13        0 5.060319e-23
    14        0 1.409740e-25
    15        0 3.665531e-28
    16        1 8.935236e-31
> plot(gof)
```

La Figure 2.15 permet de visualiser la qualité de l'ausjtement.

La différence entre la valeur prédite par le modèle Poissonnien et les valeurs observées nous poussent à essayer de mieux comprendre l'hétérogénéité qui existe au sein de nos données (sans prendre en compte la non-prise en compte de l'exposition).

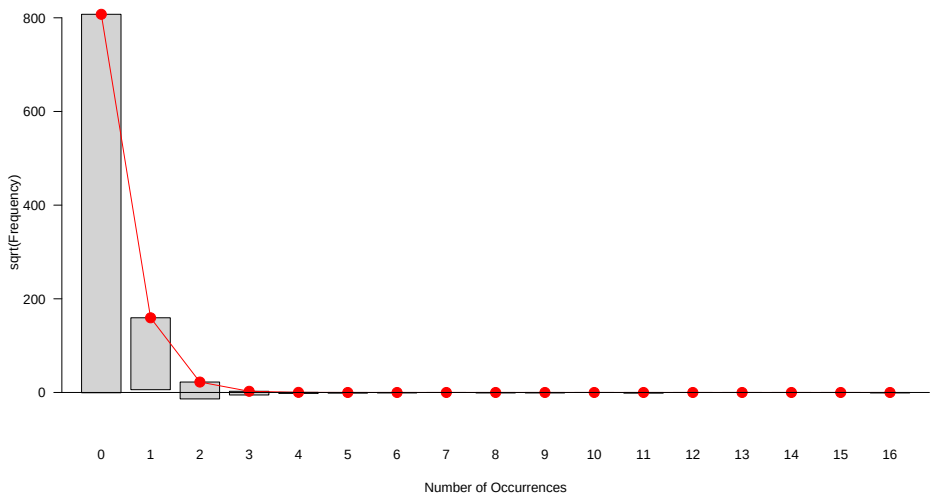


FIGURE 2.15 – Modélisation de la fréquence - globale - de sinistre par une loi de Poisson.

2.4 Les variables qualitatives ou facteurs

Les facteurs sont des codifications de variables *qualitatives*. Dans la base, nous disposons de plusieurs variables qualitatives comme le carburant `carburant` codé en `E` pour essence et `D` pour diesel, ou encore `region` pour la région.,

2.4.1 La méthode des marges

Bailey (1963) a proposé une méthode relativement simple pour faire de la tarification, appelée *method of marginal totals*. Avant de présenter cette méthode, notons que Jung (1968) a retrouvé cette méthode en faisant du maximum de vraisemblance sur un modèle Poissonnien. PlaÁons nous dans le cas où les variables exogène $\boldsymbol{X}$ sont qualitatifs, de telle sorte que l'on puisse définir des *classes de risques*. Alors,

$$\mathbb{P}(N=n|\boldsymbol{X}=\boldsymbol{X})=\exp[-\lambda_{\boldsymbol{X}}]\frac{\lambda_{\boldsymbol{X}}^n}{n!} \text{ où } \lambda_{\boldsymbol{X}}=\exp[-\boldsymbol{X}'\boldsymbol{\beta}]$$

ce qui donne une log-vraisemblance de la forme

$$\mathcal{L}(\boldsymbol{\beta}|n_i,\boldsymbol{X}_i)=\sum_{i=1}^{n}[-\lambda_{\boldsymbol{X}_i}]+n_i\log[\lambda_{\boldsymbol{X}_i}]-\log[n_i!]$$

dont la condition du premier ordre donne les équations normales,

$$\sum_{i,\boldsymbol{X}_i=\boldsymbol{X}} n_i=\sum_{i,\boldsymbol{X}_i=\boldsymbol{X}}\lambda_{\boldsymbol{X}}$$

pour toute classe de risque $\boldsymbol{X}$.,,,

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

```
> N <- baseFREQ$nbre
> E <- baseFREQ$exposition
> X1 <- baseFREQ$carburant
> X2 <- cut(baseFREQ$agevehicule,c(0,3,10,101),right=FALSE)
> names1 <- levels(X1)
> names2 <- levels(X2)
> (POPULATION=table(X1,X2))
   X2
X1   [0,3) [3,10) [10,101)
  D 102293 138763    91080
  E  85854 126926   133097
> EXPOSITION=POPULATION
> for(k in 1:nrow(EXPOSITION)){
+ EXPOSITION[k,]=tapply(E[X1==names1[k]],
+ X2[X1==names1[k]],sum)}
> EXPOSITION
   X2
X1     [0,3)   [3,10) [10,101)
  D 42160.49 76744.38 51756.02
  E 37015.56 73352.42 77470.57
> SINISTRE=POPULATION
> for(k in 1:nrow(SINISTRE)){
+ SINISTRE[k,]=tapply(N[X1==names1[k]],
+ X2[X1==names1[k]],sum)}
> SINISTRE
   X2
X1  [0,3) [3,10) [10,101)
  D  3273   6350     3827
  E  2474   5360     5160
> (FREQUENCE=SINISTRE/EXPOSITION)
   X2
X1       [0,3)     [3,10)   [10,101)
  D 0.07763192 0.08274222 0.07394309
  E 0.06683675 0.07307189 0.06660594
```

Notons $Y_{i,j}$ la fréquence empirique observée lorsque la première variable (i.e. X_1) prend la valeur i (ici i prend deux valeurs, homme ou femme) et la seconde variable (i.e. C) prend la valeur j (ici j prend trois valeurs, ville, banlieue et campagne). La matrice $\boldsymbol{Y} = [Y_{i,j}]$ est ici `FREQUENCE`. 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

$$Y_{i,j} = L_i \cdot C_j.$$

On notera $E_{i,j}$ l'exposition, i.e. `EXPOSITION`. 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). Les deux premières méthodes seront abordées en exercices. Dans la méthode des marges (selon la terminologie de Bailey (1963)), formellement, on veut

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

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

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

La première équation donne

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

et la seconde

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

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

```
> (m=sum(SINISTRE)/sum(EXPOSITION))
[1] 0.1020388
> L<-matrix(NA,100,2);C<-matrix(NA,100,3)
> L[1,]<-rep(m,2);colnames(L)=names1
> C[1,]<-rep(m,3);colnames(C)=names2
> for(j in 2:100){
+ L[j,1]<-sum(SINISTRE[1,])/sum(EXPOSITION[1,]*C[j-1,])
+ L[j,2]<-sum(SINISTRE[2,])/sum(EXPOSITION[2,]*C[j-1,])
+ C[j,1]<-sum(SINISTRE[,1])/sum(EXPOSITION[,1]*L[j,])
+ C[j,2]<-sum(SINISTRE[,2])/sum(EXPOSITION[,2]*L[j,])
+ C[j,3]<-sum(SINISTRE[,3])/sum(EXPOSITION[,3]*L[j,])
+ }
> L[1:5,]
              D          E
[1,] 0.07376302 0.07376302
[2,] 1.06843870 0.93781996
[3,] 1.06467985 0.94125969
[4,] 1.06463149 0.94130395
[5,] 1.06463087 0.94130452
> C[1:5,]
          [0,3)     [3,10)   [10,101)
[1,] 0.07376302 0.07376302 0.07376302
[2,] 0.07205381 0.07765869 0.07023750
[3,] 0.07208196 0.07767731 0.07019804
[4,] 0.07208232 0.07767755 0.07019753
[5,] 0.07208233 0.07767756 0.07019752
> PREDICTION2=SINISTRE
> PREDICTION2[1,]<-L[100,1]*C[100,]
> PREDICTION2[2,]<-L[100,2]*C[100,]
> PREDICTION2
   X2
X1       [0,3)     [3,10)   [10,101)
  D 0.07674107 0.08269792 0.07473445
  E 0.06785142 0.07311823 0.06607725
```

On notera que les marges sont identiques, par exemple pour la première ligne

```
> sum(PREDICTION2[1,]*EXPOSITION[1,])
[1] 13450
> sum(SINISTRE[1,])
[1] 13450
```

Cette technique est équivalente ùtiliser une régression log-Poisson sur les deux variables qualitatives,

```
> donnees <- data.frame(N,E,X1,X2)
> regpoislog <- glm(N~X1+X2,offset=log(E),data=donnees,
+ family=poisson(link="log"))
> newdonnees <- data.frame(X1=factor(rep(names1,3)),E=rep(1,6),
+ X2=factor(rep(names2,each=2)))
> matrix(predict(regpoislog,newdata=newdonnees,
+ type="response"),2,3)
           [,1]       [,2]       [,3]
[1,] 0.07674107 0.08269792 0.07473445
[2,] 0.06785142 0.07311823 0.06607725
```

Parmi les autres variables que l'on considèrera comme qualitatitive, il y a la région d'habitation, dont l'influence sur la fréquence de sinistre peut être visualisé sur la Figure 2.16.,,

```
> library(maptools)
> library(maps)
>  departements<-readShapeSpatial("DEPARTEMENT.SHP")
> legend(166963,6561753,legend=names(attr(colcode,"table")),
+ fill=attr(colcode, "palette"), cex=0.6, bty="n")
> region<-tapply(baseFREQ[,"nbre"],as.factor(baseFREQ[,"region"]),sum)/
+ tapply(baseFREQ[,"exposition"],as.factor(baseFREQ[,"region"]),sum)
> depFREQ=rep(NA,nrow(departements))
> names(depFREQ)=as.character(departements$CODE_REG)
> for(nom in names(region)){
+ depFREQ[names(depFREQ)==nom] <- region[nom]
+}
> plot(departements,col=gray((depFREQ-.05)*20))
> legend(166963,6561753,legend=seq(1,0,by=-.1)/20+.05,
+ fill=gray(seq(1,0,by=-.1)),cex=1.25, bty="n")
```

2.4.2 Prise en compte de l'exposition et variable offset

Dans un modèle collectif, on a besoin de connaître le nombre de sinistres survenus sur une police d'assurance. Dans l'optique de tarifer un contrat, il faut pouvoir prédire le nombre de sinistres qui surviendront, en moyenne, l'année suivante. Or si certains polices n'ont été observées que 6 mois dans la base, il convient de pondérer la fréquence de sinistre par l'exposition. Compte tenu de la propriété multiplicative d'un processus de Poisson, une police observée 1 an aura, en moyenne, 4 fois plus de sinistres qu'une police observée 3 mois. Dans le cas d'un modèle log-Poisson, il est alors naturel de supposer que

$$Y|\boldsymbol{X} \sim \mathcal{P}(\exp[\boldsymbol{X}\boldsymbol{\beta} + \log(e)])$$

où e désigne l'exposition, mesurée en années.

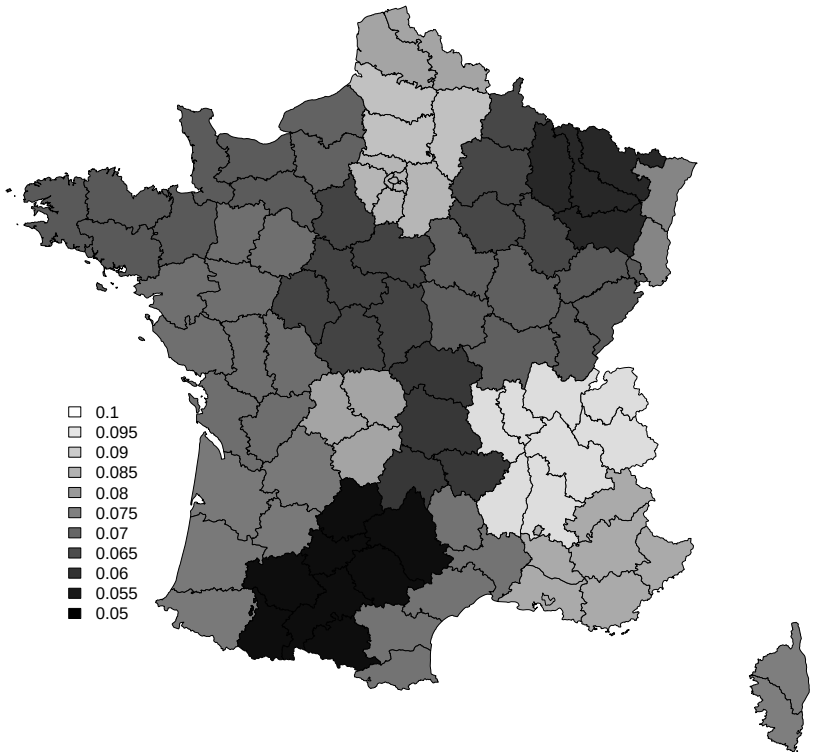


FIGURE 2.16 – Fréquence de sinistres en fonction de la région d'habitation.

Remark 2.4.1. *Plus formellement, on retrouve ici une propriété du processus de Poisson. Si la survenance d'accident pour un individu peut être modélisée par un processus de Poisson homogène de paramètre λ, λ est l'espérance du nombre de sinistre sur un intervalle de longueur 1 (e.g. $[0,1]$). Pour un assuré présent pour une durée t (disons au cours de l'intervalle de temps $[0,t]$) l'espérance du nombre de sinistres est λt, i.e. il est proportionnel à la durée d'exposition rélle au risque.*

Dans le cas des régressions de Poisson, cela peut se faire de la manière suivante

```
> reg <- glm(nombre~0+puissance+region,
+ data=nombre,family=poisson(link="log"),offset=log(exposition))
```

On peut noter que la régression pouvait s'écrire

$$Y|\boldsymbol{X} \sim \mathcal{P}(\exp[\beta_0 + \beta_1 X_1 + \cdots + \beta_k X_k + e])$$

autrement dit, on rajoute l'exposition dans la régression, tout en forçant le coefficient à être égal à 1. Ceci légitime ainsi la seconde écriture possible

```
> reg <- glm(nombre~0+puissance+region+offset(exposition),
+ data=nombre,family=poisson(link="log"))
```

On notera qu'il est possible d'intégrer une variable offset dans la méthode des marges, en notant qu'il convient de faire une moyenne du nombre de sinistres, divisé par la moyenne de l'exposition. Par exemple pour reprendre une régression présentée en introduction

```
> seuils <- c(17,21,25,30,50,80,120)
> reg2 <- glm(nombre~cut(ageconducteur,breaks=seuils),data=sinistres,
+ family=poisson(link="log"),offset=log(exposition))
> predict(reg2,newdata=data.frame(ageconducteur=20,exposition=1),
+ type="response")
[1] 0.2113669
> I <- (sinistres$ageconducteur>=17)&(sinistres$ageconducteur<=21)
> mean(sinistres$nombre[I==TRUE])/mean(sinistres$exposition[I==TRUE])
[1] 0.2113669
```

Une autre manière d'écrire cette grandeur est de faire une moyenne pondérée (par l'exposition) du nombre annualisé de sinistres,

```
> weighted.mean(sinistres$nombre[I==TRUE]/sinistres$exposition[I==TRUE],
+ w=sinistres$exposition[I==TRUE])
[1] 0.2113669
```

2.4.3 Les variables tarifaires continues et la nonlinéarité

Le but de la tarification (et plus généralement de toute prédiction) est d'estimer une espérance conditionnelle,

$$\mathbb{E}(S|\boldsymbol{X}=\boldsymbol{x})=\varphi(\boldsymbol{x}) \text{ ou } S=\varphi(X_1,\cdots,X_k)+\varepsilon$$

où $\varphi:\mathbb{R}^k\to\mathbb{R}$. Supposer un modèle *linéaire* est problement une hypothèse trop forte. Mais on se doute qu'estimer une fonction définie sur $\mathbb{R}^k$ serait trop complexe numériquement. Un bon compromis est proposé par les modèles dit *additifs*.

Pour commencer, on peut récupérer les fréquences empiriques par âge

```
> freq.emp<-tapply(baseFREQ[,"nbre"],as.factor(baseFREQ[,"ageconducteur"]),sum)
+ /tapply(baseFREQ[,"exposition"],as.factor(baseFREQ[,"ageconducteur"]),sum)
```

A titre d'illustration, la Figure 2.17 permet de visualiser l'impact de l'âge du conducteur principal sur la fréquence de sinistre. Les points noirs correspondent à la fréquence moyenne empirique observée,

```
> age <- seq(18,92)
> pred.emp <- freq.emp[as.character(age)]
> reg.splines <- glm(nbre~bs(ageconducteur,10)+offset(log(exposition)),
+ family=poisson(link="log"),data=baseFREQ)
> age <- seq(18,100)
> pred.splines <- predict(reg.splines,newdata=data.frame(ageconducteur=
+ age,exposition=1),type="response",se=TRUE)
> plot(age,pred.splines$fit,lwd=2,type="l",ylab="",xlab="Age du conducteur principal",
+ ylim=c(0,0.25))
> polygon(c(age,rev(age)),
+ c(pred.splines$fit+2*pred.splines$se.fit,
+ rev(pred.splines$fit-2*pred.splines$se.fit)),
+ col="grey",border=NA)
> lines(age,pred.splines$fit,lwd=2)
> lines(age,pred.splines$fit+2*pred.splines$se.fit,lty=2)
> lines(age,pred.splines$fit-2*pred.splines$se.fit,lty=2)
```

```
> abline(h=sum(baseFREQ[,"nbre"])/sum(baseFREQ[,"exposition"])
+ ,lty=2,lwd=.5)
> points(18:92,pred.emp,pch=19,cex=.7,type="b")
```

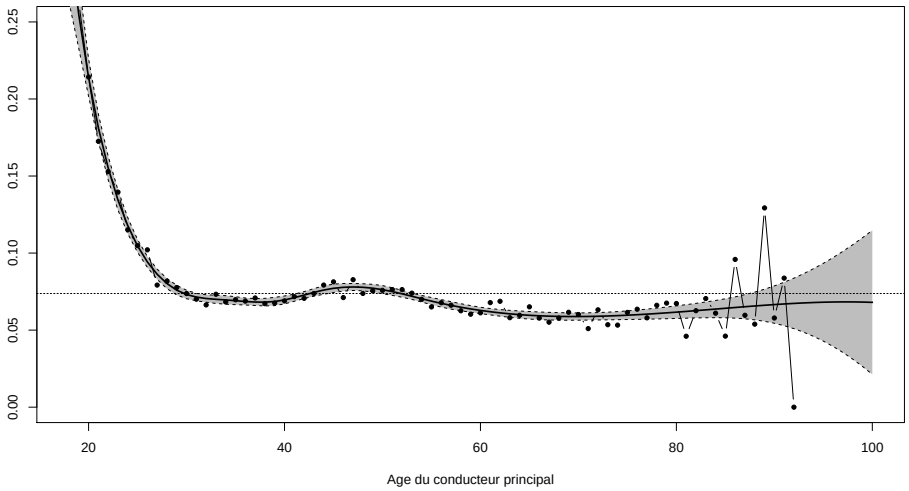


FIGURE 2.17 – Fréquence de sinistres en fonction de l'âge du conducteur principal, régression de Poisson avec splines.

Les modèles GAM

Les modèles additifs ont été introduits par Stone (1985) qui notait qu'estimer une fonction $\varphi : \mathbb{R}^k \to \mathbb{R}$ serait numériquement trop complexe (et probablement peu robuste). Les GAMs ont ètè popularisès ensuite par le livre Hastie & Tibshirani (1990). On cherche ici une décomposition de la forme

$$S = \varphi_1(X_1) + \cdots + \varphi_k(X_k) + \varepsilon$$

où les fonctions $\varphi_j : \mathbb{R} \to \mathbb{R}$ sont supposées suffisament régulières.

On peut reprendre l'exemple traité avec des splines dans un modèle GLM, directement sous forme d'un modèle GAM, que l'on peut visualiser sur la Figure 2.18,

```
> reg.gam <- gam(nbre~s(ageconducteur)+offset(log(exposition)),
+ family=poisson(link="log"),data=baseFREQ)
> pred.gam <- predict(reg.gam,,newdata=data.frame(ageconducteur=age,exposition=1),
+ type="response",se=TRUE)
> plot(age,pred.gam$fit,lwd=2,type="l",ylab="",xlab="Age du conducteur principal",
+ ylim=c(0,0.25))
> polygon(c(age,rev(age)),
+ c(pred.gam $fit+2* pred.gam $se.fit,rev(pred.gam $fit-2*pred.gam $se.fit)),
+ col="grey",border=NA)
> lines(age, pred.gam $fit,lwd=2)
> lines(age, pred.gam $fit+2* pred.gam $se.fit,lty=2)
> lines(age, pred.gam $fit-2* pred.gam $se.fit,lty=2)
```

```
> abline(h=sum(baseFREQ[,"nbre"])/sum(baseFREQ[,"exposition"]),lty=2,lwd=.5)
> points(18:92,pred.emp,pch=19,cex=.7,type="b")
```

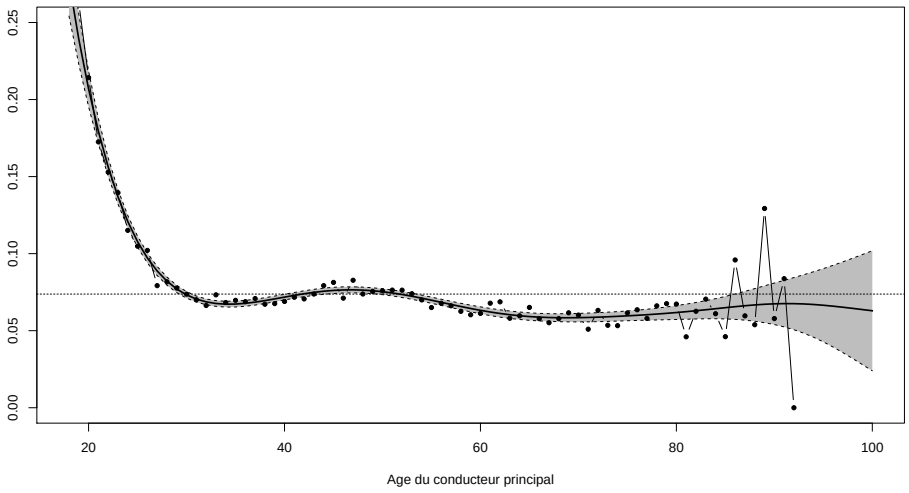


FIGURE 2.18 – Fréquence de sinistres en fonction de l'âge du conducteur principal, régression GAM.

Notons que les modèles sont additifs, aussi, avec une variable continue et un facteur (par exemple le carburant) on aurait

$$S = \varphi_1(X_1) + \beta_2 X_2 + \varepsilon = \varphi_1(X_1) + \beta_2^D \mathbf{1}(X_2 = \mathrm{D}) + \varepsilon,$$

où X_1 est l'âge du conducteur, et X_2 le carburant du véhicule. Notons qu'il serait aussi possible de considérer un modèle de la forme

$$S = \begin{cases} \varphi_{1,E}(X_1) + \varepsilon & \text{si } X_2 = \text{ essence} \\ \varphi_{1,D}(X_1) + \varepsilon & \text{si } X_2 = \text{ diesel} \end{cases}$$

Le premier modèle (additif) est estimé ci-dessous.

```
> regC.gam <- gam(nbre~s(ageconducteur)+carburant+offset(log(exposition)),
+ family=poisson(link="log"),data=baseFREQ)
> predCE.gam <- predict(regC.gam,,newdata=data.frame(ageconducteur=age,
+ exposition=1,carburant="E"),type="response")
> predCD.gam <- predict(regC.gam,,newdata=data.frame(ageconducteur=age,
+ exposition=1,carburant="D"),type="response")
```

On peut visualiser le lien entre la fréquence anuelle de sinistre et l'âge sur la Figure 2.19,

```
> plot(age,predCD2.gam,lwd=2,type="l",ylab="",xlab=
+ "Age du conducteur principal",ylim=c(0,0.25))
> lines(age,predCE2.gam,lwd=2,col="grey")
> lines(age,predCD.gam,lty=2)
> lines(age,predCE.gam,lty=2,col="grey")
> legend(80,.23,c("Diesel","Essence"),col=c("black","grey"),
+ lwd=2,lty=1,bty="n")
```

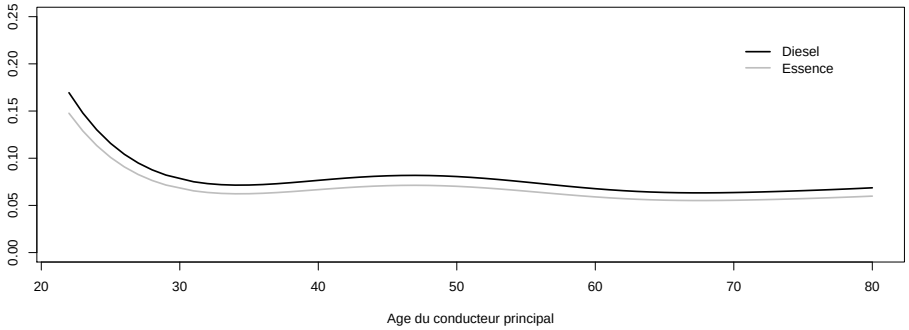


FIGURE 2.19 – Modèle GAM addifif, $S = \varphi_1(X_1) + \beta_2 X_2 + \varepsilon$ où X_2 désigne le type de carburant.

En revanche, pour estimer le second modèle, il convient de faire deux régressions distinctes,

```
> regCE.gam <- gam(nbre~s(ageconducteur)+offset(log(exposition)),
+ family=poisson(link="log"),data=baseFREQ[baseFREQ$carburant=="E",])
> regCD.gam <- gam(nbre~s(ageconducteur)+offset(log(exposition)),
+ family=poisson(link="log"),data=baseFREQ[baseFREQ$carburant=="D",])
> predCE2.gam <- predict(regCE.gam,,newdata=data.frame(ageconducteur=age,exposition=1),
+ type="response")
> predCD2.gam <- predict(regCD.gam,,newdata=data.frame(ageconducteur=age,exposition=1),
+ type="response")
```

On peut visualiser le lien entre la fréquence anuelle de sinistre et l'âge sur la Figure 2.20,

```
> plot(age,predCD2.gam,lwd=2,type="l",ylab="",xlab=
+ "Age du conducteur principal",ylim=c(0,0.25))
> lines(age,predCE2.gam,lwd=2,col="grey")
> lines(age,predCD.gam,lty=2)
> lines(age,predCE.gam,lty=2,col="grey")
> legend(80,.23,c("Diesel","Essence"),col=c("black","grey"),
+ lwd=2,lty=1,bty="n")
```

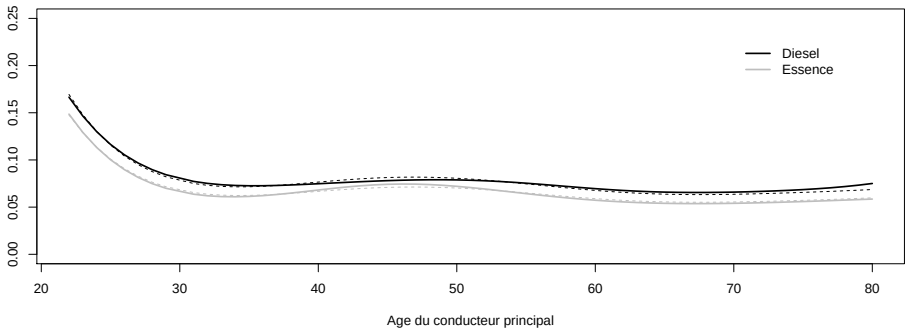


FIGURE 2.20 – Modèle GAM, $S = \varphi_{1,E}(X_1) + \varepsilon$ si $X_2 =$ essence ou $S = \varphi_{1,D}(X_1) + \varepsilon$ si $X_2 =$ diesel où X_2 désigne le type de carburant.

L'estimation de ces modèles peut se faire de plusieurs manières sous $\mathbb{R}$. Il y a tout d'abord la fonction `gam` de `library(gam)`, basé sur l'algorithme proposé par Hastie & Tibshirani (1990). La fonction `gam` de `library(mgcv)` repose sur la méthodologie développée par Wood (2000). Enfin d'autres packages proposent aussi des estimations de ces transformations nonlinéaires, dont `library(gmlss)` ou `library(gss)`.

2.4.4 Les modèles nonlinéaires multivariés

On peut s'autoriser éventuellement encore un peu plus de souplesse en prenant en compte le couple constitué de deux variables continues (comme discuté dans Friedman (1991)),

$$S = \varphi(X_1, X_2) + \varepsilon$$

où $\varphi : \mathbb{R}^2 \to \mathbb{R}$, au lieu d'un modèle GAM classique,

$$S = \varphi_1(X_1) + \varphi_2(X_2) + \varepsilon$$

Cette option est proposée par exemple dans la fonction `gam` de `library(mgcv)`,mais pour des raisons de volume de données, on va se limiter à un échantillon de la base

```
> set.seed(1)
> echantillon=sample(1:nrow(baseFREQ),size=200000)
> reg.gam2 <- gam(nbre~s(ageconducteur,agevehicule)+offset(log(exposition)),
+ family=poisson(link="log"),data=baseFREQ[echantillon,])
```

La Figure 2.21, permet de visualiser la surface de prédiction de la fréquence annuelle de sinistre, en fonction de l'âge du conducteur et de l'ancienneté du véhicule,

```
> pred.gam2 <- predict(reg.gam2,,newdata=data.frame(ageconducteur=
+ C,agevehicule=V,exposition=1),type="response")
> P2 <- matrix(pred.gam2,length(agec),length(agev))
> ZL<-range(P2)
> persp(agec,agev,P2,theta=30,xlab="age conducteur",ylab="age vehicule",zlab="",zlim=ZL)
```

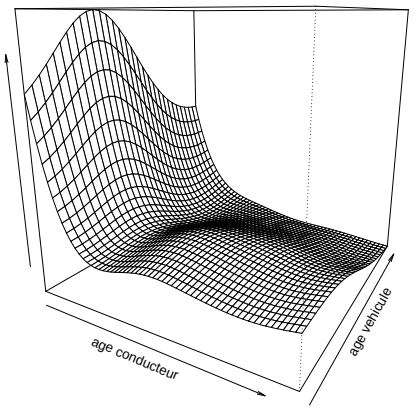


FIGURE 2.21 – Modèle GAM bivariée $s(X_1, X_2)$.

On peut comparer ce modèle joint à un modèle strictement additif, comme sur la Figure 2.22,

```
> reg.gam3 <- gam(nbre~s(ageconducteur)+s(agevehicule)+offset(log(exposition)),
+ family=poisson(link="log"),data=baseFREQ[echantillon,])
> pred.gam3 <- predict(reg.gam3,,newdata=data.frame(ageconducteur=
+ C,agevehicule=V,exposition=1),type="response")
> P3 <- matrix(pred.gam3,length(agec),length(agev))
> persp(agec,agev,P2,theta=30,xlab="age conducteur",ylab="age vehicule",zlab="",zlim=ZL)
```

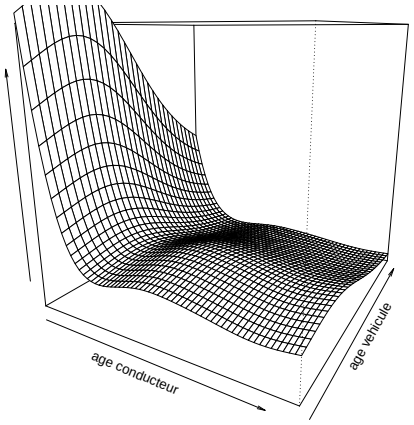


FIGURE 2.22 – Modèle GAM (réellement additif) $s_1(X_1) + s(X_2)$.

Enfin, à titre de comparaison, on peut aussi visualiser sur la Figure 2.23 ce que donne un modèle GLM sans lissage,

```
> reg.glm4 <- glm(nbre~ageconducteur+agevehicule+offset(log(exposition)),
+ family=poisson(link="log"),data=baseFREQ[echantillon,])
> pred.glm4 <- predict(reg.glm4,,newdata=data.frame(ageconducteur=
+ C,agevehicule=V,exposition=1),type="response")
> P4 <- matrix(pred.glm4,length(agec),length(agev))
> persp(agec,agev,P4,theta=30,xlab="age conducteur",ylab="age vehicule",zlab="",zlim=ZL)
```

2.4.5 Prise en compte de la surdispersion

Dans une régression poissonnienne, on suppose que dans une classe de risque (ou conditionnellement aux variables explicatives), la fréquence et l'espérance coïncident, i.e. $\mathbb{V}(Y|\boldsymbol{X}) = \mathbb{E}(Y|\boldsymbol{X})$. Dans l'exemple ci-dessous, on considère le nombre de sinistres RC. On constitue quelques classes tarifaires (les âges des conducteurs croisés avec le carburant)

```
> sumnb = tapply(baseFREQ$nbre , baseFREQ[,c("ageconducteur",
+ "carburant")], sum)
> sumnb2 = tapply(baseFREQ$nbre^2 , baseFREQ[,c("ageconducteur",
+ "carburant")], sum)
> expo = tapply(baseFREQ$exposition , baseFREQ[,c("ageconducteur",
```

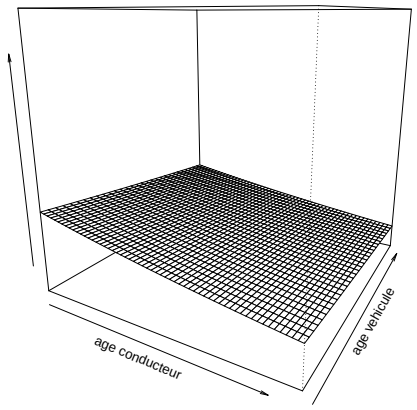


FIGURE 2.23 – Modèle GLM (linéaire) $\beta_1 X_1 + \beta_2 X_2$.

```
+ "carburant")], sum)
> M= sumnb/expo
> V=sumnb2/expo-M^2
> plot(as.vector(M),as.vector(V),xlab="moyenne empirique",
+ ylab="variance empirique")
> abline(a=0,b=1)
> abline(lm(as.vector(V)~as.vector(M)),lty=2)
```

La Figure 2.24 permet de visualiser l'hypothèse d'égalité de la variance et de la moyenne par classe de risque (i.e. conditionnellement à $\boldsymbol{X}$).

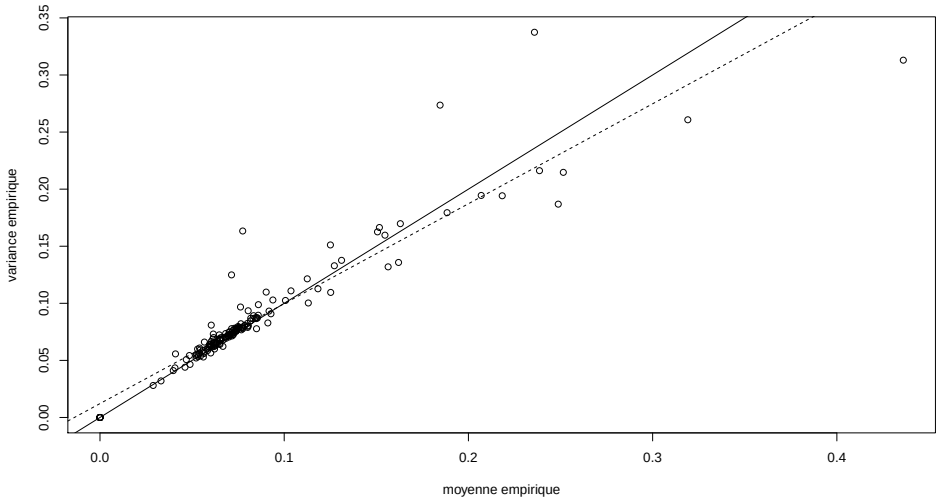


FIGURE 2.24 – Moyenne empirique et variance empirique, par classe de risque.

On peut commencer par faire un premier test, afin de voir si la pente de la régression semble significativement différente de 1

```
> library(AER)
> regression=lm(as.vector(V)~as.vector(M),
+ weight=as.vector(expo))
> linearHypothesis(regression,"as.vector(M)=1")
Linear hypothesis test

Hypothesis:
as.vector(M) = 1

Model 1: restricted model
Model 2: as.vector(V) ~ as.vector(M)

  Res.Df    RSS Df Sum of Sq     F Pr(>F)
1    163 50.786
2    162 50.025  1   0.76148 2.466 0.1183
```

Manifestement, la droite de régression ne semblerait pas significativement différente de la première bissectrice (comme le montrait la Figure 2.24).

Si malgré tout on pense que cette surdispersion est importance, une manière de la quantifier peut être de prendre non une loi de `poisson`, mais une loi `quasipoisson`, telle que $\mathbb{V}(Y|\boldsymbol{X}) = \phi\mathbb{E}(Y|\boldsymbol{X})$, où ϕ devient un paramètre à estimer (tout comme la volatilité des résidus dans une régression linéaire Gaussienne).

```
> regglm <- glm(nbre~bs(ageconducteur)+carburant+ offset(log(exposition)),
+ data=baseFREQ,family=quasipoisson)
> summary(regglm)

Call:
glm(formula = nbre ~ bs(ageconducteur) + carburant + offset(log(exposition)),
    family = quasipoisson, data = baseFREQ)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-0.6249  -0.3542  -0.2589  -0.1419  13.4432

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)
(Intercept)        -1.63342    0.04214 -38.763  < 2e-16 ***
bs(ageconducteur)1 -2.42084    0.14018 -17.270  < 2e-16 ***
bs(ageconducteur)2  0.72919    0.15282   4.772 1.83e-06 ***
bs(ageconducteur)3 -2.70146    0.23513 -11.489  < 2e-16 ***
carburantE         -0.12726    0.01655  -7.690 1.48e-14 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for quasipoisson family taken to be 1.781494)

    Null deviance: 171819  on 678012  degrees of freedom
```

```
Residual deviance: 170731  on 678008  degrees of freedom
AIC: NA

Number of Fisher Scoring iterations: 6

> (summary(regglm)$dispersion)
[1] 1.781494
```

I.e. sur cette régression, $\widehat{\phi} = 1.78$. Pour tester la significativité de cette éventuelle surdispersion, on peut noter que la surdispersion correspond à une hétérogénéité résiduelle, c'est à dire un effet aléatoire. Par exemple on peut supposer que

$$(Y|\boldsymbol{X} = \boldsymbol{X}, \boldsymbol{Z} = \boldsymbol{z}) \sim \mathcal{P}(\exp[\boldsymbol{X}'\boldsymbol{\beta} + \boldsymbol{z}'\boldsymbol{\alpha}])$$

de telle sorte que si $u = \boldsymbol{z}'\boldsymbol{\alpha} - \mathbb{E}(\boldsymbol{Z}'\boldsymbol{\alpha}|\boldsymbol{X} = \boldsymbol{X})$, alors

$$(Y|\boldsymbol{X} = \boldsymbol{X}, \boldsymbol{Z} = \boldsymbol{z}) \sim \mathcal{P}(\exp[\boldsymbol{X}'\boldsymbol{\gamma} + u])$$

On a un modèle dit à effets fixes, au sens où

$$(Y|\boldsymbol{X} = \boldsymbol{X}) \sim \mathcal{P}(\exp[\boldsymbol{X}'\boldsymbol{\gamma} + U]),$$

où $U = \boldsymbol{Z}'\boldsymbol{\alpha} - \mathbb{E}(\boldsymbol{Z}'\boldsymbol{\alpha}|\boldsymbol{X} = \boldsymbol{X})$. Par exemple, si on suppose que $U \sim \gamma(a, a)$, i.e. d'espérance 1 et de variance $\sigma^2 = 1/a$, alors

$$(Y|U = u) \sim \mathcal{P}(\lambda u) \text{ où } \lambda = \exp[\boldsymbol{X}'\boldsymbol{\gamma}],$$

de telle sorte que $\mathbb{E}(Y|U = u) = \mathbb{V}(Y|U = u)$. Mais si on regarde la loi nonconditionnelle, $\mathbb{E}(Y) = \lambda$ alors que

$$\mathbb{V}(Y) = \mathbb{V}(\mathbb{E}[Y|U]) + \mathbb{E}(\mathbb{V}(Y|)) = \lambda + \lambda^2\sigma^2.$$

On peut alors proposer un test de la forme suivante : on suppose que

$$\mathbb{V}(Y|\boldsymbol{X} = \boldsymbol{X}) = \mathbb{E}(Y|\boldsymbol{X} = \boldsymbol{X}) + \tau \cdot \mathbb{E}(Y|\boldsymbol{X} = \boldsymbol{X})^2,$$

on on cherche à tester

$$H_0 : \tau = 0 \text{ contre } \tau > 0.$$

Parmi les statistiques de test classique, on pourra considérer

$$T = \frac{\sum_{i=1}^n [(Y_i - \widehat{\mu}_i)^2 - Y_i]}{\sqrt{2\sum_{i=1}^n \widehat{\mu}_i^2}}$$

qui suit, sous H_0, une loi normale centrée réduite. Sous R, ce test est programmé dans la fonction `dispersiontest()` de `library(MASS)`.

```
> library(AER)
> regglm2 <- glm(nbre~bs(ageconducteur)+carburant+ offset(log(exposition)),
+ data=baseFREQ,family=poisson)
> dispersiontest(regglm2)

        Overdispersion test
```

```
data:  regglm2
z = 3.8802, p-value = 5.218e-05
alternative hypothesis: true dispersion is greater than 1
sample estimates:
dispersion
  1.222467
```

Une autre possibilité est de faire une régression binomiale négative (qui permettra de prendre en compte de la surdispersion). Elle se fait à l'aide de la fonction `glm.nb()` de `library(MASS)`.

```
> library(MASS)
> regbn <- glm.nb(nbre~bs(ageconducteur)+carburant+
+ offset(log(exposition)),data=baseFREQ)
```

Remark 2.4.2. *La loi Binomial Négative est obtenue comme un mélange Poisson-Gamma. Dans* `library(gamlss)` *on parle de loi binomiale négative de type I. Une loi de type II est obtenue en considérant un mélange Poisson-inverse Gaussienne.*

On peut comparer les deux modèles sur la Figure 2.25, avec une représentation des coefficients.

```
> regp  <- glm(nbre~bs(ageconducteur)+carburant+
+ offset(log(exposition)),data=baseFREQ,family=poisson)
> plot(regbn$coefficients,regp$coefficients)
> abline(a=0,b=1,lty=2,col="grey")
> cbind(regbn$coefficients,regp$coefficients)
                          [,1]       [,2]
(Intercept)        -1.6174987 -1.6334197
bs(ageconducteur)1 -2.4311047 -2.4208431
bs(ageconducteur)2  0.7144625  0.7291903
bs(ageconducteur)3 -2.7009294 -2.7014616
carburantE         -0.1260395 -0.1272581
> plot(regbn$coefficients,regp$coefficients,
+ xlab="régression binomiale négative",
+ ylab="régression de Poisson")
```

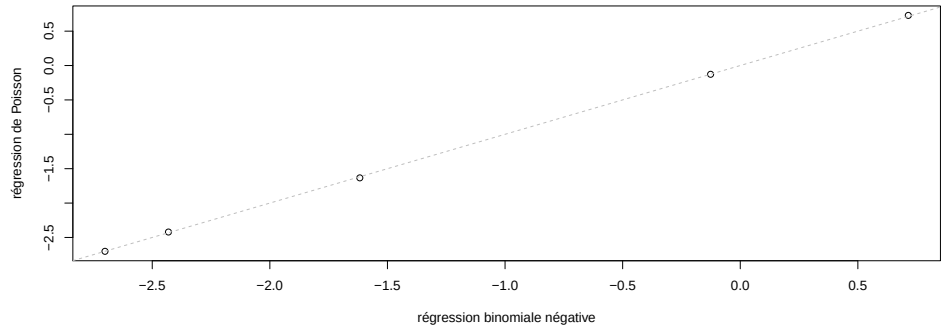


FIGURE 2.25 – Comparaison des coefficients d'une régression binomiale et d'une régression de Poisson.

La surdispersion est ici relativement faible, sauf au sein de quelques classes d'âge, et la meilleure solution serait de continuer à chercher d'autres variables explicatives, permettant de supprimer cette hétérogénéité résiduelle.

2.4.6 Les modèles zèros modifiés, ou à inflation de zéros (*zero-inflated*)

Afin d'éviter l'aléa moral, il n'est pas rare de mettre en place des contrats participatifs. En assurance, l'exemple le plus connu est probablement le mécanisme de bonus-malus. Une personne qui n'a pas d'accident responsable une année a le droit à un rabais l'année suivante (un *bonus*) alors qu'une personne ayant eu un ou plusieurs sinistres subit une majoration de prime (un *malus*). D'un point de vue économétrique, cette solution présente un biais puisqu'elle peut insiter des personnes à ne pas déclarer certains sinistres (dès lors que la majoration excède le coût du sinistre). Il n'est alors pas rare d'observer *trop* de personnes non-sinistrées dans la population totale (par rappport à un modèle Poissonnien).

Un modèle dit zèro modifié (*zero inflated*) est un mélange entre une masse en 0 et un modèle classique de comptage, typiquement un modèle de Poisson, ou binomial négatif. Pour modéliser la probabilité de ne pas déclarer un sinistre (et donc d'avoir un surpoids en 0), considérons un modèle logistique par exemple,

$$\pi_i = \frac{\exp[\boldsymbol{X}_i'\boldsymbol{\beta}]}{1+\exp[\boldsymbol{X}_i'\boldsymbol{\beta}]}$$

Pour le modèle de comptable, on note $p_i(k)$ la probabilité que l'individu i ait k sinistres (correspondant à la loi si la personne décide de déclarer ses sinistres, classiquement modélisé par une loi de Poisson). Aussi,

$$\mathbb{P}(N_i=k) = \begin{cases} \pi_i + [1-\pi_i]\cdot p_i(0) \text{ si } k=0, \\ [1-\pi_i]\cdot p_i(k) \text{ si } k=1,2,\cdots \end{cases}$$

Si p_i correspond à un modèle Poissonnien (de moyenne λ_i), on peut alors montrer facilement que $\mathbb{E}(N_i)=[1-\pi_i]\lambda_i$ et $\mathbb{V}(N_i)=\pi_i\lambda_i+\pi_i\lambda_i^2[1-\pi_i]$.

La `library(gamlss)` propose la fonction `ZIP` (pour *zero inflated Poisson*), mais aussi `ZINBI` (lorsque p_i correspond à une loi binomiale négative), ou `ZIPIG` (pour un mélange Poisson-inverse Gaussien), par exemple. La `library(pscl)` propose également une fonction `zeroinfl` plus simple d'utilisation, proposant aussi bien un modèle de Poisson qu'un modèle binomial négatif.

Il existe aussi des modèles dits *zero adapted*, où l'on suppose que

$$\mathbb{P}(N_i=k) = \begin{cases} \pi_i \text{ si } k=0, \\ [1-\pi_i]\cdot \dfrac{p_i(k)}{1-p_i(0)} \text{ si } k=1,2,\cdots \end{cases}$$

Dans `library(gamlss)` il s'agit du modèle `ZAP`. Et comme auparavant, il existe des fonctions `ZANBI` ou `ZAPIG` .

Ces modèles à inflation de zéros peuvent être particulièrement utiles pour prendre en compte un excès de non-déclarations de sinistres, généralement attribuées à une peur de perdre un niveau intéressant de bonus-malus : la perte financière associée au malus des années suivantes peut excéder l'indemnité versée aujourd'hui. On peut ajuster ici un modèle zero-inflated (logit) avec une loi de Poisson afin d'expliquer la sinistralité en fonction de l'âge du conducteur (en prenant en compte l'âge via une fonction nonlinéaire que l'on estimera à l'aide de splines).

```
> library(pscl)
> regNZI <- glm(nbre~bs(ageconducteur,5)+offset(log(exposition)),
```

```
+ data=baseFREQ,family=poisson(link="log"))
> regZI <- zeroinfl(nbre~bs(ageconducteur) |
+ bs(ageconducteur),offset=log(exposition),
+ data = baseFREQ,dist = "poisson",link="logit")
```

On peut s'intéresser plus particulièrement à l'impact de l'âge sur la probabilité de ne pas déclarer de sinistres (correspondant au paramètre de la loi binomiale), présentée sur la Figure 2.26.

```
> age<-data.frame(ageconducteur=18:90,exposition=1)
> pred0 <- predict(regZI,newdata=age,type="zero")
> plot(age$ageconducteur,pred0,type="l",xlab="",lwd=2,
+ ylim=c(0,1),ylab="Probabilité de ne pas déclarer un sinistre")
```

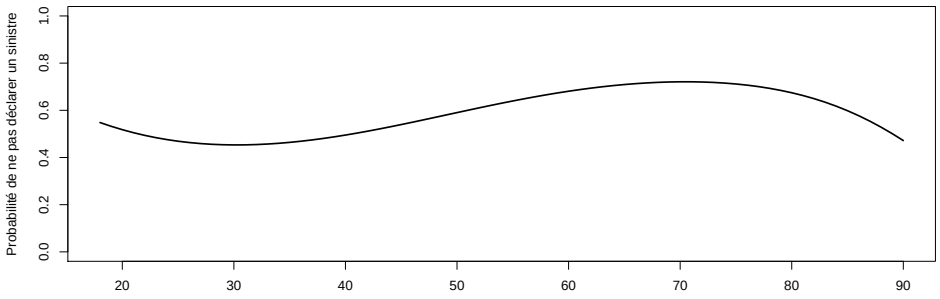


FIGURE 2.26 – Prédiction à l'aide de modèles zero-inflated (logit) avec une loi de Poisson de la sinistralité en fonction du taux de bonus.

La principale explication avancée - en France - pour la non déclaration de sinistre est l'existence du système bonus-malus. Les personnes ayant un très bon taux (proche de 50%) ayant intérêt à ne pas déclarer de sinistre s'ils ne veulent pas voir leur prime s'envoler l'année suivante

```
> regZIbm <- zeroinfl(nbre~1 |
+ bs(bonus),offset=log(exposition),
+ data = baseFREQ,dist = "poisson",link="logit")
> B <- data.frame(bonus=50:200,exposition=1)
> pred0 <- predict(regZIbm,newdata=B,type="zero")
> plot(age$ageconducteur,pred0,type="l",xlab="",lwd=2,
+ ylim=c(0,1),ylab="Probabilité de ne pas déclarer un sinistre")
```

2.5 Modéliser les coûts individuels des sinistres

Les coûts (individuels) de sinistres sont des variables positives.

```
> mean(baseCOUT$cout)
[1] 2265.513
> quantile(baseCOUT$cout,prob=c(.5,.9,.95,.99))
      50%       90%       95%       99%
 1172.000  2767.604  4765.093 16451.224
```

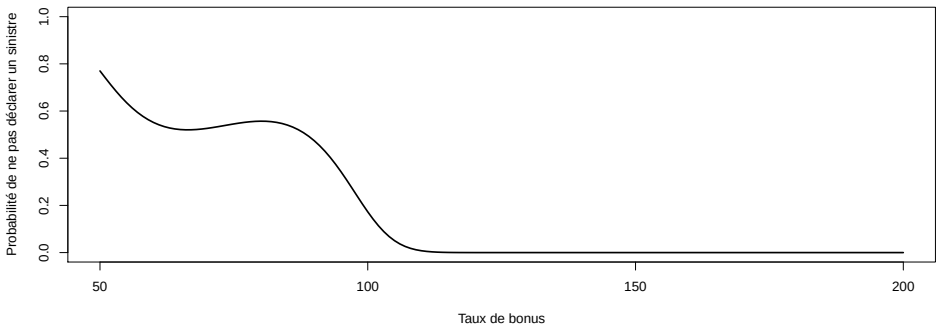


FIGURE 2.27 – Prédiction à l'aide de modèles zero-inflated (logit) avec une loi de Poisson de la sinistralité en fonction de l'âge du conducteur.

Remark 2.5.1. *En présence de coûts fixes (bris de glace par exemple), la loi des coûts de sinistres sera une loi continue, avec des masses de Dirac (là on l'on observe des coûts fixes). La loi est alors*

$$f(y) = (1-p)f_\star(y) + p\mathbf{1}(y = C)$$

où p désigne la probabilité d'avoir un coût qui soit précisément C, et $f_\star$ est la densité des autres coûts de sinistres. Dans notre approche économétrique, on peut envisager un modèle de la forme

$$f(y|\boldsymbol{X} = \boldsymbol{x}) = (1-p(\boldsymbol{x}))f_\star(y|\boldsymbol{X} = \boldsymbol{x}) + p(\boldsymbol{x})\mathbf{1}(y = C)$$

où $p(\boldsymbol{x})$ peut être modélisée par une régression logistique, et où $f_\star(y|\boldsymbol{X} = \boldsymbol{x})$ est une loi positive à densité*. On peut alors chercher à modéliser cette loi continue sur la base où les coûts fixes ont été écartés.*

2.5.1 Modèle Gamma et modèle lognormal

Les deux modèles les plus classiques permettant de modéliser les coûts individuels de sinistre sont

- le modèle Gamma sur les coûts individuels Y_i,
- le modèle log-normal sur les coûts individuels Y_i, ou plutôt un modèle Gaussien sur le logarithme des coûts, $\log(Y_i)$: la loi lognormale n'appartient pas à la famille exponentielle.

Le(s) modèle(s) Gamma

La loi Gamma, de paramètres α et β, de densité

$$f(y) = \frac{\beta^\alpha}{\Gamma(\alpha)} y^{\alpha-1}\exp(-\beta y),\ \text{pour } y \geq 0,$$

vérifie $\mathbb{E}(Y) = \dfrac{\alpha}{\beta}$ et $\mathbb{V}(X) = \dfrac{\alpha}{\beta}^2$. Autrement dit, le coefficient de variation vaut ici

$$\text{CV} = \frac{\sqrt{\mathbb{V}(X)}}{\mathbb{E}(Y)} = \frac{1}{\sqrt{\alpha}},$$

qui peut être analysé comme un cofficient de *dispersion*. En fait, si $\phi = 1/\alpha$, on peut écrire

$$\mathbb{V}(Y) = \frac{1}{\alpha}\frac{\alpha^2}{\beta^2} = \phi \cdot \mathbb{E}(Y)^2,$$

où on retrouve ici une fonction variance de forme quadratique.

Remarque 2.5.1. *Le cas particulier* $\phi = 1$ *corrrespond à la loi* exponentielle.

Bien que le lien canonique de la loi Gamma soit la fonction inverse, il est plus fréquent d'utiliser un lien logarithmique. En effet, la forme multiplicative donne des interprétations simples dans le cas des modèles multiples.

```
> reggamma <- glm(cout~ageconducteur,family=Gamma(link="log"),
+ data=baseCOUT)
> summary(reggamma)

Call:
glm(formula = cout ~ ageconducteur, family = Gamma(link = "log"),
    data = baseCOUT)

Deviance Residuals:
   Min      1Q  Median      3Q     Max
-3.632  -0.977  -0.611  -0.392  52.599

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)    8.180643   0.208009  39.328   <2e-16 ***
ageconducteur -0.010440   0.004383  -2.382   0.0172 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for Gamma family taken to be 109.7107)

    Null deviance: 46482  on 26443  degrees of freedom
Residual deviance: 45637  on 26442  degrees of freedom
AIC: 458704

Number of Fisher Scoring iterations: 9
```

Si on s'intéresse à la valeur prédite pour une personne d'âge `ageconducteur=50`, on obtient

```
> predict(reggamma,newdata=data.frame(ageconducteur=50),
+ type="response")
       1
2118.879
```

Le modèle lognormal

La régression lognormale peut être obtenue en considérant une régression linéaire (Gaussienne) sur le logarithme du coût,

$$\log(Y_i) = \boldsymbol{X}_i'\boldsymbol{\beta} + \varepsilon_i,$$

avec $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$. En effet, par définition de la loi lognormale, $Y \sim LN(\mu, \sigma^2)$ si et seulement si $\log Y \sim \mathcal{N}(\mu, \sigma^2)$. Le principal soucis dans cet écriture est que

$$\begin{cases} \mathbb{E}(Y) = \exp\left(\mu + \dfrac{\sigma^2}{2}\right) \neq \exp(\mu) = \exp[\mathbb{E}(\log Y)] \\ \mathbb{V}(Y) = \exp\left(2\mu + \sigma^2\right)\left[\exp\left(\sigma^2\right) - 1\right] \neq \exp(\sigma^2) = \exp[\mathbb{V}(\log Y)] \end{cases}$$

Autrement dit, pour passer des estimations faites à partir du modèle sur $\log Y$ à des prédictions sur le coût Y, il ne faudra pas oublier de multiplier par $\exp(\sigma^2/2)$. Une régression sur le logarithme des coûts donnerait par exemple,

```
> reglm <- lm(log(cout)~ageconducteur,data=baseCOUT)
> summary(reglm)

Call:
lm(formula = log(cout) ~ ageconducteur, data = baseCOUT)

Residuals:
    Min      1Q  Median      3Q     Max
-6.8699 -0.3110  0.2063  0.2926  8.4297

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)   6.7501521  0.0224328 300.905  < 2e-16 ***
ageconducteur 0.0021392  0.0004727   4.525 6.06e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.13 on 26442 degrees of freedom
Multiple R-squared: 0.0007738,        Adjusted R-squared: 0.0007361
F-statistic: 20.48 on 1 and 26442 DF,  p-value: 6.059e-06

> (sigma <- summary(reglm)$sigma)
[1] 1.129607
```

Si on s'intéresse à la valeur prédite pour une personne d'âge `ageconducteur=50`, on obtient

```
> mu <- predict(reglm,newdata=data.frame(ageconducteur=50))
> exp(mu+sigma^2/2)
       1
1799.239
```

On notera que les deux modèles donnent des résultats *très* sensiblement différents : l'âge semble avoir un impact sur le coût significatif pour les deux modèle, mais en sens inverse ! On peut comparer les prédictions sur la Figure 2.28

```
> reggamma.sp <- glm(cout~bs(ageconducteur,5),family=Gamma(link="log"),
+ data=baseCOUT)
> Pgamma <- predict(reggamma.sp,newdata=data.frame(ageconducteur=
+ age),type="response")
> reglm.sp <- lm(log(cout)~bs(ageconducteur,5),data=baseCOUT)
> sigma <- summary(reglm.sp)$sigma
> mu <- predict(reglm.sp,newdata=data.frame(ageconducteur=age))
> Pln <- exp(mu+sigma^2/2)
```

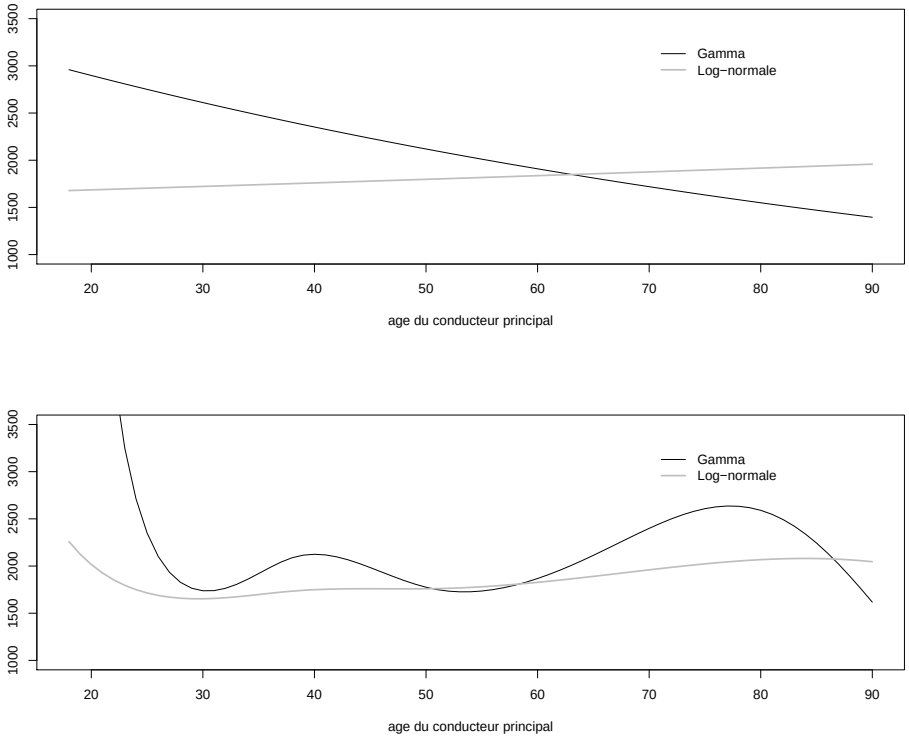


FIGURE 2.28 – Régressions lognormale versus Gamma, où le coût individuel est expliqué par l'âge du conducteur sans lissage (en haut) et avec lissage (en bas).

```
> plot(age,Pgamma,xlab="age du conducteur principal",ylab="",
+ type="l",ylim=c(1000,3500))
> lines(age,Pln,col="grey",lwd=2)
> legend(70,3300,c("Gamma","Log-normale"),col=c("black",
+ "grey"),lwd=c(1,2),lty=1,bty="n")
> Pgamma <- predict(reggamma,newdata=data.frame(ageconducteur=age),
+ type="response")
> mu <- predict(reglm,newdata=data.frame(ageconducteur=age))
> Pln <- exp(mu+sigma^2/2)
> plot(age,Pgamma,xlab="age du conducteur principal",ylab="",
+ type="l",ylim=c(1000,3500))
> lines(age,Pln,col="grey",lwd=2)
> legend(70,3300,c("Gamma","Log-normale"),col=c("black",
+ "grey"),lwd=c(1,2),lty=1,bty="n")
```

La Figure 2.29 montre les mêmes types de modèles si l'on cherche à expliquer le coût par l'ancienneté du véhicule.

```
> age <- 0:25
> reggamma.sp <- glm(cout~bs(agevehicule),family=Gamma(link="log"),
+ data=baseCOUT)
> Pgamma <- predict(reggamma.sp,newdata=data.frame(agevehicule =age),type="response")
```

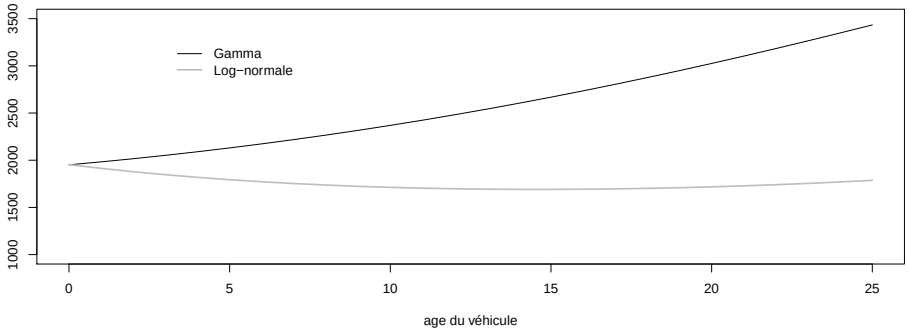


FIGURE 2.29 – Régressions lognormale versus Gamma, où le coût individuel est expliqué par l'âge du véhicule avec lissage.

```
> reglm.sp <- lm(log(cout)~bs(agevehicule),data=baseCOUT)
> sigma <- summary(reglm.sp)$sigma
> mu <- predict(reglm.sp,newdata=data.frame(agevehicule =age))
> Pln <- exp(mu+sigma^2/2)
> plot(age,Pgamma,xlab="age du véhicule",ylab="",type="l",ylim=c(1000,3500))
> lines(age,Pln,col="grey",lwd=2)
> legend(3,3300,c("Gamma","Log-normale"),col=c("black","grey"),lwd=c(1,2),lty=1,bty="n")
```

En fait, la divergence entre les deux modèles vient du fait que le modèle Gamma (quelle que soit la variable explicative) est très sensible aux valeurs extrêmes. Un avantage du modèle lognormal est qu'en prenant le logarithme des coûts, on atténue l'importance des sinistres de coût exceptionnel. En écartant les sinistres tels que `sinistres$cout` est supérieur à 100 000, on obtient des modèles comparables (et proches de ce que donnait la régression lognormale sur l'ensemble de la base)

```
> indice <- baseCOUT$cout<100000
> reggamma.sp <- glm(cout~bs(ageconducteur,5),family=Gamma(link="log"),
+ data=baseCOUT[indice,])
> Pgamma <- predict(reggamma.sp,newdata=data.frame(ageconducteur=
+ age),type="response")
> reglm.sp <- lm(log(cout)~bs(ageconducteur,5),data=baseCOUT[indice,])
> sigma <- summary(reglm.sp)$sigma
> mu <- predict(reglm.sp,newdata=data.frame(ageconducteur=age))
> Pln <- exp(mu+sigma^2/2)
> plot(age,Pgamma,xlab="age du conducteur principal",ylab="",
+ type="l",ylim=c(1000,3500))
> lines(age,Pln,col="grey",lwd=2)
> legend(70,3300,c("Gamma","Log-normale"),col=c("black",
+ "grey"),lwd=c(1,2),lty=1,bty="n")
```

Nous reviendrons plus en détails sur la prise en compte de ces sinistres exceptionnels (qui ici ont simplement été écartés) dans la section suivante. L'idée est de dire que les coûts sinistres de taille modérée peuvent être expliqués par des variables a priori (avec une relative robustesse). Mais pas les sinistres exceptionnels.

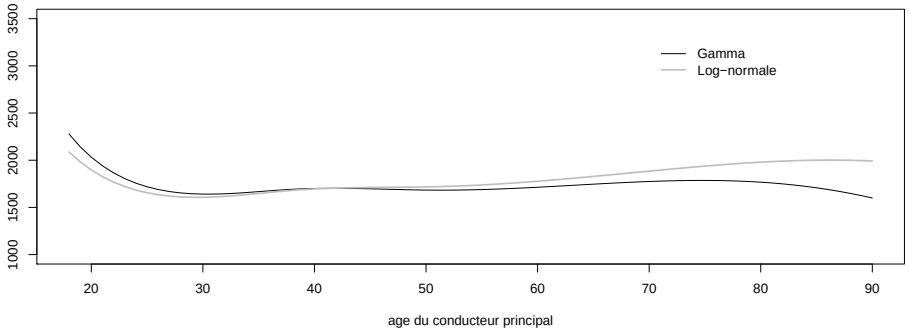


FIGURE 2.30 – Régressions lognormale versus Gamma, où le coût individuel est expliqué par l'âge du véhicule avec lissage, en écartant les sinistres de plus de 100,000.

2.5.2 Modélisation des grands sinistres

Il existe un grand nombre de faÁons de définir les lois à queues épaisses. La plus élégante d'un point de vue actuarielle est probablement la famille des lois *sous exponentielles* (décrites dans Embrechts et al. (1997)). Une loi de fonction de survie $\overline{F}$ sera dite sous-exponentielle si pour tout $n \geq 2$,

$$\lim_{x\to\infty} \frac{\overline{F^{\star n}}(x)}{\overline{F}(x)} = n$$

ou bien, si $X_1, \cdots, X_n, \cdots$ sont des variables i.i.d. de loi F,

$$\mathbb{P}(X_1 + \cdots + X_n > x) \sim \mathbb{P}(\max\{X_1, \cdots, X_n\} > x).$$

Autrement dit, la loi de la charge totale dans un portefeuille a des queues des distributions qui se comportent comme le plus gros sinistres. Ce sont donc des lois qui sont très influencées par ces très gros sinistres. Parmi les lois de la famille sous-exponentielle,

- la loi lognormale, $f(y) \propto \dfrac{1}{y\sigma} \exp\left(-[\log y - \mu]^2/2\sigma^2\right)$
- la loi de Weibull, $f(y) \propto x^{k-1} \exp\left([-x^k]\right)$ si $k < 1$

mais la loi la plus utilisée, en particulier en réassurance, n'est pas dans la famille exponentielle,

- la loi de Pareto, $f(y) \propto [\mu + y]^{-\alpha-1}$

Dans ces familles de lois à queues épaisses, on va ainsi retrouver une autre classe relativement connue, à savoir les lois dite *à variation régulière*. Ces lois sont aussi dite *de type* Pareto, au sens où

$$\mathbb{P}(Y > y) = y^{-\alpha}\mathcal{L}(y),$$

où $\mathcal{L}$ est une fonction à variation lente, i.e.

$$\lim_{x\to\infty} \frac{\mathcal{L}(tx)}{\mathcal{L}(x)} = 1 \text{ pour tout } t > 0.$$

La `library(gamlss)` propose d'autres familles de lois, comme les lois *Reverse Gumbel* ou *Power Exponential*

Il est possible de définir une famille dite *beta généralisée de seconde espèce*, notée GB2. On suppose que

$$\log Y \overset{\mathcal{L}}{=} \mu + \sigma \log \frac{\Gamma_1}{\Gamma_2}$$

où $\Gamma \sim \mathcal{G}(\alpha_i, 1)$ sont indépendantes. Si Γ_2 est une constante ($\alpha_2 \to \infty$) on obtient la *loi gamma généralisée*.

La densité de cette loi s'écrit :

$$f(y) \propto y^{-1} \left[\exp\left(\frac{\log y - \mu}{\sigma}\right)\right]^{\alpha_1} \left[1 + \exp\left(\frac{\log y - \mu}{\sigma}\right)\right]^{-(\alpha_1+\alpha_2)}$$

Supposons que μ soit une fonction linéaire des variables explicatives, $\mu = \boldsymbol{X}'\boldsymbol{\beta}$. Alors

$$\mathbb{E}(Y|\boldsymbol{X}) = C\exp[\mu(\boldsymbol{X})] = C\exp[\boldsymbol{X}'\boldsymbol{\beta}]$$

Ces modèles sont détaillés dans McDonald & Butler (1990).

2.5.3 Ecrêtement des grands sinistres

Si l'on considère des modèles économétriques basés uniquement sur des variables catégorielles (en particulier des classes pour les variables continues) la prime pure est alors généralement la moyenne empirique dans la classe considérée (c'est en tous les cas ce que préconise par exemple la méthode des marges). Mais cette méthode devient alors vite très sensible aux sinistres extrêmes (d'autant plus que les classes sont d'effectif restreint).

Afin d'éviter ce problème, il n'est pas rare d'écrêter les sinistres : on calcule la prime moyenne par groupe tarifaire en écartant les gros sinistres, qui seront répartis sur l'ensemble de la population. On peut bien entendu raffiner cette méthode en considérant des modèles hiérarchiques et en répartissant simplement sur une surclasse.

Supposons que les sinistres extrêmes soient ceux qui dépassent un seuil s (connu, ou au moins fixé *a priori*). Rappelons que la formule des probabilités totales permet d'écrire que (dans le cas discret pour faire simple)

$$\mathbb{P}(A) = \sum_i \mathbb{P}(A \cap B_i) \sum_i \mathbb{P}(A|B_i) \cdot \mathbb{P}(B_i),$$

où (B_i) forme une partition de Ω. En particulier

$$\mathbb{P}(A) = \mathbb{P}(A|B) \cdot \mathbb{P}(B) + \mathbb{P}(A|B^c) \cdot \mathbb{P}(B^c),$$

où B^c désigne le complémentaire de B. En passant à l'espérance, et en travaillant sur des variables aléatoires plutôt que des ensembles, on peut écrire

$$\mathbb{E}(Y) = \mathbb{E}(Y|B) \cdot \mathbb{P}(B) + \mathbb{E}(Y|B^c) \cdot \mathbb{P}(B^c).$$

Si on prend comme cas particulier $B = \{Y \leq s\}$ et $B^c = \{Y > s\}$, alors

$$\mathbb{E}(Y) = \mathbb{E}(Y|Y \leq s) \cdot \mathbb{P}(Y \leq s) + \mathbb{E}(Y|Y > s) \cdot \mathbb{P}(Y > s).$$

finallement, on note que la probabilité $\mathbb{P}$ n'a joué aucun rôle ici, et on peut parfaitement la remplacer par une probabilité conditionnelle, $\mathbb{P}_{\boldsymbol{X}}$, i.e.

$$\mathbb{E}(Y|\boldsymbol{X}) = \mathbb{E}(Y|\boldsymbol{X}, Y \leq s) \cdot \mathbb{P}(Y \leq s|\boldsymbol{X}) + \mathbb{E}(Y|\boldsymbol{X}, Y > s) \cdot \mathbb{P}(Y > s|\boldsymbol{X}),$$

Le premier terme correspond aux sinistres 'normaux' , que l'on pourra modéliser par une loi évoquée précédemment (régression Gamma par exemple). Pour le second terme, on notera que $\mathbb{E}[\mathbb{E}(Y|\boldsymbol{X}, Y > s)] = \mathbb{E}(Y|Y > s)$. Autrement dit, on peut être tenté par ne plus distinguer par classe le coût moyen des très très gros sinistres : on répartira proportionnellement à la fréquence des gros sinistres sinistres.

La prédiction sera donc basée sur trois parties, la première pour les sinistres usuels (plus petits que s), et la seconde pour les grands sinistres (pour les sinistres excédant s), avec comme troisième terme que sera la probabilité, par classe tarifaire, d'avoir un sinistre excédant le seuil s.

```
> seuil=50000
> sinistres.inf = baseCOUT[baseCOUT$cout<=seuil,]
> sinistres.sup = baseCOUT[baseCOUT$cout>seuil,]
> baseCOUT$indic = baseCOUT$cout>seuil
> proba=gam(indic~s(ageconducteur),data= baseCOUT,
+ family=binomial)
> probpred=predict(proba,newdata=data.frame(ageconducteur=age),
+ type="response")
> reg=gam(cout~s(ageconducteur),data= sinistres.inf,
+ family=Gamma(link="log"))
> Y.inf=predict(reg,newdata=data.frame(ageconducteur=
+ age),type="response")
> Y.sup=mean(sinistres.sup$cout)
> Y=Y.inf*(1-probpred)+Y.sup*probpred
> plot(age,Y,type="l",lwd=2,xlab="age du conducteur principal",
+ ylab="",ylim=c(1000,3500))
> lines(age,Pgamma,col="grey")
> legend(70,1800,c("Ecrêté","Brut"),col=c("black","grey"),
+ lwd=c(1,2),lty=1,bty="n")
```

La Figure 2.31 permet de visualiser la différence entre les deux modèles, avec ou sans écrêtement (avec un seuil à 50,000).

La Figure 2.32 permet de visualiser la différence entre les deux modèles, avec ou sans écrêtement avec un seuil beaucoup plus faible (à 5 000). Dans ce cas, la majorité est sinistres sont répartis entre tous les assurés, qui payent la même quantité (l'espérance au delà du seuil d'écrêtement).

2.6 Exercices

Exercise 2.6.1. *Parmi les méthodes proches de celles évoquées dans la section 2.4.1 sur la méthode des marges, il est aussi possible d'utiliser une méthode par moindres carrés. On va chercher à minimiser la somme des carrés des erreurs, i.e.*

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

La condition du premier ordre donne ici

$$\frac{\partial D}{\partial L_i} = -2 \sum_j C_j N_{i,j}(Y_{i.j} - L_i \cdot C_j) = 0$$

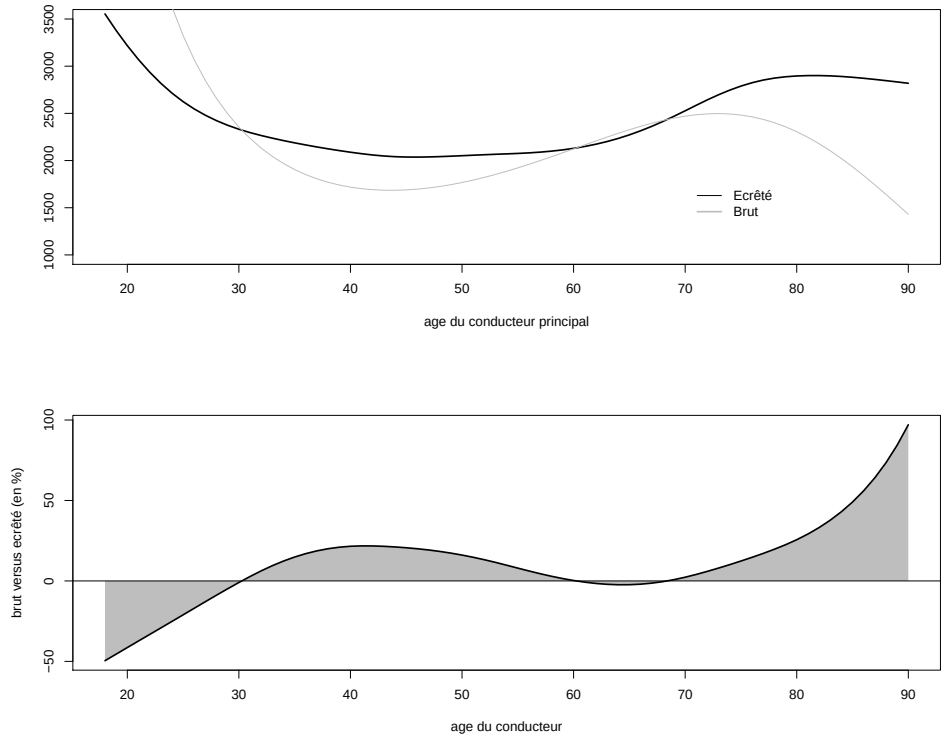


FIGURE 2.31 – Estimation de $\mathbb{E}(Y|\boldsymbol{X})$ avec ou sans écrêment (les sinistres dépassant le seuil fixé sont ici répartis entre les assurés, proportionnellement à leur probabilité d'avoir un gros sinistre), avec un seuil de gros sinistre 50 000. Le graphique du bas compare les prédictions des espérances de coût individuel, avec ou sans écrêment (en variation)

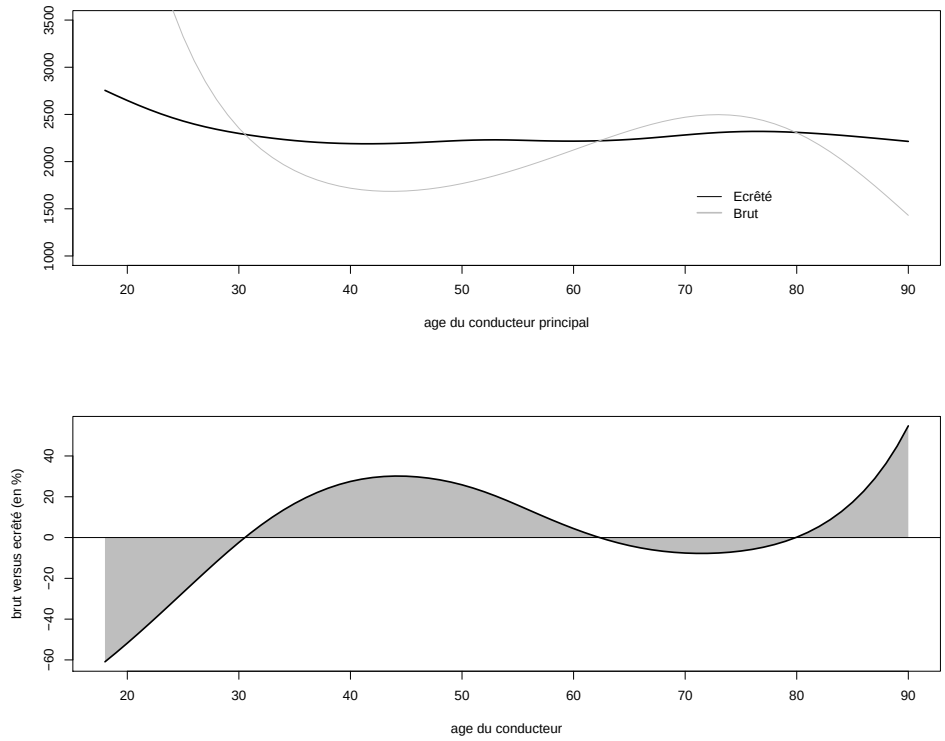


FIGURE 2.32 – Estimation de $\mathbb{E}(Y|\boldsymbol{X})$ avec ou sans écrêment , avec un seuil d'écrêtement à 5 000. Le graphique du bas compare les prédictions des espérances de coût individuel, avec ou sans écrêment

soit

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

L'autre condition du premier ordre donne

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

On résoud alors ce petit systeme de maniere itérative (car il n'y a pas de solution analytique simple). Programmer et comparer avec la méthode des marges.

Exercise 2.6.2. *Parmi les méthodes proches de celles évoquées dans la section 2.4.1 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*

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

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

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

et une expression du même genre pour C_j. Programmer et comparer avec la méthode des marges.

méthode des marges

Chapitre 3

Les provisions pour sinistres à payer

Dans ce chapitre, nous allons étudier les méthodes pour calculer le montant des *provisions pour sinistres à payer*, et plus particulièrement, des méthodes permettant de quantifier la marge d'erreur associée. Comme les définit Simonet (1998), "*les provisions techniques sont les provisions destinées à permettre le règlement intégral des engagements pris envers les assurés et bénéfi-ciaires de contrats. Elles sont liées à la technique même de l'assurance, et imposées par la règlementation*". D'un point de vue plus formel, à la date t, la compagnie d'assurance est tenue de constituer une provision pour les sinistres survenus avant la date t qu'elle sera tenu d'indemniser. Elle doit donc estimer le coût des sinistres survenus, et retrancher les montants déjà versés. Il s'agit donc fondamentalement d'un problème de prévision.

3.1 La problématique du provisionnment

Parmi les méthodes reconnues par les autorités de contrôles, les plus classiques sont basées sur les cadences de paiements. On raisonne pour cela par année de survenance de sinistre, et on suppose une certaine régularité dans la cadence de paiement.

3.1.1 Quelques définitions et notations, aspects règlementaires et comptables

La plupart des méthodes présentées ici sont détaillées dans Denuit & Charpentier (2005), Partrat et al. (2008)ou Wüthrich & Merz (2008). Classiquement, on notera

- i (en ligne) l'année de survenance des sinistres,
- j (en colonne) l'année de développement,
- $i+j$ (en diagonale) l'année calendaire de paiement (pour les incréments),
- $Y_{i,j}$ les *incréments de paiments*, pour l'année de développement j, pour les sinistres survenus l'année i, Table 3.1,
- $C_{i,j}$ les *paiments cumulés*, au sens où $C_{i,j} = Y_{i,0}+Y_{i,1}+\cdots+Y_{i,j}$, pour l'année de survenance j, Table 3.2
- P_i la prime acquise pour l'année i, Table 3.3,
- $N_{i,j}$ le *nombre cumulé de sinistres* pour l'année de survenance i vu au bout de j années, Table 3.4,
- $\Gamma_{i,j}$ la *charge dossier par dossier cumulée* (estimées par les gestionnaires de sinistres sur les $N_{i,j}$ connus, ou partiellement connus), pour l'année de développement j, pour les sinistres survenus l'année i, Table 3.7 (cette matrice ne sera explotée que dans la méthode dite Munich Chain Ladder).

	0	1	2	3	4	5
0	3209	1163	39	17	7	21
1	3367	1292	37	24	10	
2	3871	1474	53	22		
3	4239	1678	103			
4	4929	1865				
5	5217					

TABLE 3.1 – Triangle des incréments de paiements, $\boldsymbol{Y} = (Y_{i,j})$.

	0	1	2	3	4	5
0	3209	4372	4411	4428	4435	4456
1	3367	4659	4696	4720	4730	
2	3871	5345	5398	5420		
3	4239	5917	6020			
4	4929	6794				
5	5217					

TABLE 3.2 – Triangle des paiements cumulés, $\boldsymbol{C} = (C_{i,j})$.

Formellement, toutes ces données sont stockées dans des matrices (ou un vecteur pour la prime), avec des valeurs manquantes `NA` pour les valeurs futures. Ils seront dénommés respectivement `PAID`, `PREMIUM`, `NUMBER` et `INCURRED`

```
> PAID
     [,1] [,2] [,3] [,4] [,5] [,6]
[1,] 3209 4372 4411 4428 4435 4456
[2,] 3367 4659 4696 4720 4730   NA
[3,] 3871 5345 5398 5420   NA   NA
[4,] 4239 5917 6020   NA   NA   NA
[5,] 4929 6794   NA   NA   NA   NA
[6,] 5217   NA   NA   NA   NA   NA
```

Le triangle des incréments se déduit facilement du triangle des cumulés

```
> nc <- ncol(PAID)
> nl <- nrow(PAID)
> INC <- PAID
> INC[,2:nc] <- PAID[,2:nc]-PAID[,1:(nc-1)]
> INC
     [,1] [,2] [,3] [,4] [,5] [,6]
[1,] 3209 1163   39   17    7   21
[2,] 3367 1292   37   24   10   NA
[3,] 3871 1474   53   22   NA   NA
[4,] 4239 1678  103   NA   NA   NA
[5,] 4929 1865   NA   NA   NA   NA
[6,] 5217   NA   NA   NA   NA   NA
```

Dans sa version la plus simple, le but des méthodes de provisionnement est de compléter la partie inférieure des triangles de paiements. Dans la littérature anglo-saxonne, on parlera

Année i	0	1	2	3	4	5
P_i	4591	4672	4863	5175	5673	6431

TABLE 3.3 – Vecteur des primes acquises, $\boldsymbol{P} = (P_i)$.

	0	1	2	3	4	5
0	1043.4	1045.5	1047.5	1047.7	1047.7	1047.7
1	1043.0	1027.1	1028.7	1028.9	1028.7	
2	965.1	967.9	967.8	970.1		
3	977.0	984.7	986.8			
4	1099.0	1118.5				
5	1076.3					

TABLE 3.4 – Triangle des nombres de sinistres, cumulés, en milliers, $\boldsymbol{N} = (N_{i,j})$.

d'IBNR (*incurred but not reported*).

3.1.2 Formalisation du problème du provisionnement

Le provisionnement est un problème de prédiction, conditionnelle à l'information dont on dispose à la date n. On notera $\mathcal{H}_n$ l'information disponible à la date n, soit

$$\mathcal{H}_n = \{(X_{i,j}), i+j \leq n\} = \{(C_{i,j}), i+j \leq n\}.$$

On cherche à étudier, par année de survenance, la loi conditionnelle de $C_{i,\infty}$ (la charge ultime pour une année de survenance donnée) sachant $\mathcal{H}_n$, ou encore, si l'on suppose les sinistres clos au bout de n années la loi de $C_{i,n}$ sachant $\mathcal{H}_n$. Si l'on se focalise sur une année de survenance particulière, on pourra noter

$$\mathcal{F}_{i,n-i} = \{(X_{i,j}), j = 0, \cdots, n-i)\} = \{(C_{i,j}), j = 0, \cdots, n-i)\}.$$

Cette notation permet de prendre en compte que l'information disponible change d'une ligne à l'autre. On cherchera par la suite à prédire le montant des sinistres à payer pour l'année i, i.e.

$$\widehat{C}_{i,n}^{(n-i)} = \mathbb{E}[C_{i,n}|\mathcal{F}_{i,n-i}]$$

et la différence entre ce montant et le montant déjà payé constituera la provision pour sinistres à payer,

$$\widehat{R}_i = \widehat{C}_{i,n}^{(n-i)} - C_{i,n-i}.$$

On essayera ensuite de quantifier l'incertitude associée à cette prédiction. Comme on le verra les méthodes usuelles visaient à calculer

$$\mathbb{V}[C_{i,n}|\mathcal{F}_{i,n-i}] \text{ ou } \mathbb{V}[\widehat{C}_{i,n}^{(n-i)}]$$

ce que l'on appelera "incertitude à horizon ultime". Mais ce n'est pas ce que propose Solvabilité II, demandant plutôt de mesurer une incertitude dite *à un an*. Pour cela, on va s'intéresser à la prédiction qui sera faite dans un an,

$$\widehat{C}_{i,n}^{(n-i+1)} = \mathbb{E}[C_{i,n}|\mathcal{F}_{i,n-i+1}]$$

et plus particulièrement le changement dans l'estimation de la charge ultime

$$\Delta_i^n = \widehat{C}_{i,n}^{(n-i+1)} - \widehat{C}_{i,n}^{(n-i)} = CDR_i(n),$$

parfois noté CDR (*claims development result*). Si cette différence est positive, on parle de *mali* (il faudra gonfler la provision afin de pouvoir payer les sinistres), et si elle est négative, on parle de *boni*. On peut montrer que $\mathbb{E}[\Delta_i^n|\mathcal{F}_{i,n-i}] = 0$, autrement dit, on ne peut espérer faire ni boni, ni mali, en moyenne. Les contraintes règlementaires imposéeés par Solvabilité II demandent de calculer $\mathbb{V}[\Delta_i^n|\mathcal{F}_{i,n-i}]$.

3.2 Les cadences de paiements et la méthode Chain Ladder

L'utilisation des cadences de paiements pour estimer la charge future date des années 1930. On suppose qu'il existe une relation de récurrence de la forme

$$C_{i,j+1} = \lambda_j \cdot C_{i,j} \text{ pour tout } i, j = 1, \cdots, n.$$

Un estimateur naturel pour λ_j, basé sur l'expérience passée est alors le ratio moyen basé sur les $n-j$ années observées :

$$\widehat{\lambda}_j = \frac{\sum_{i=1}^{n-j} C_{i,j+1}}{\sum_{i=1}^{n-j} C_{i,j}} \text{ pour tout } j = 1, \cdots, n-1.$$

De telle sorte que l'on peut alors prédire la charge pour la partie non-observée dans le triangle,

$$\widehat{C}_{i,j} = \left[\widehat{\lambda}_{n+1-i} \cdots \widehat{\lambda}_{j-1}\right] \cdot C_{i,n+1-i}.$$

```
> k <- 1
> weighted.mean(x=PAID[,k+1]/PAID[,k],w=PAID[,k],na.rm=TRUE)
[1] 1.380933
> sum(PAID[1:(nl-k),k+1])/sum(PAID[1:(nl-k),k])
[1] 1.380933
```

On fait alors une boucle pour estimer tous les coefficients de transition

```
> LAMBDA <- rep(NA,nc-1)
> for(k in 1:(nc-1)){
+ LAMBDA[k]=(sum(PAID[1:(nl-k),k+1])/sum(PAID[1:(nl-k),k]))}
> LAMBDA
[1] 1.380933 1.011433 1.004343 1.001858 1.004735
```

Notons qu'au lieu de calculer les facteurs de développement, on peut aussi des taux de développement, cumulés ou non. Autrement dit, au lieu d'écrire $C_{i,j+1} = \lambda_j \cdot C_{i,j}$ pour tout $i, j = 1, \cdots, n$, on suppose que

$$C_{i,j} = \gamma_j \cdot C_{i,n} \text{ ou } Y_{i,j} = \varphi_j \cdot C_{i,n}.$$

On notera que

```
> (GAMMA <- rev(cumprod(rev(1/LAMBDA))))
[1] 0.7081910 0.9779643 0.9891449 0.9934411 0.9952873
> (PHI <- c(GAMMA[1],diff(GAMMA)))
[1] 0.708191033 0.269773306 0.011180591 0.004296183 0.001846141
```

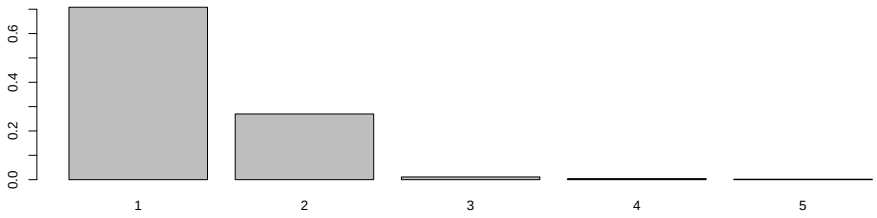


FIGURE 3.1 – Cadence de paiement des sinistres, en fonction de la charge ultime, de la première année (année calendaire de survenance du sinistre) à la cinquième année.

	0	1	2	3	4	n
λ_j	1,38093	1,01143	1,00434	1,00186	1,00474	1,0000
γ_j	70,819%	97,796%	98,914%	99,344%	99,529%	100,000%
φ_j	70,819%	26,977%	1,118%	0,430%	0,185%	0,000%

TABLE 3.5 – Facteurs de développement, $\widehat{\boldsymbol{\lambda}} = (\widehat{\lambda}_i)$, exprimés en cadence de paiements par rapport à la charge utlime, en cumulé (i.e. $\widehat{\boldsymbol{\gamma}}$), puis en incréments (i.e. $\widehat{\boldsymbol{\varphi}}$).

Ce dernier coefficient permet de visualiser la cadence de paiement (en pourcentage de la charge ultime). On peut visualiser ce coefficient sur la Figure 3.1

```
> barplot(PHI,names=1:5)
```

On notera qu'il est possible de voir l'estimateur Chain-Ladder $\widehat{\lambda}_j$ comme une moyenne pondérée des facteurs de transition individuels, i.e.

$$\widehat{\lambda}_j = \sum_{i=1}^{n-j} \omega_{i,j} \cdot \lambda_{i,j} \text{ où } \omega_{i,j} = \frac{C_{i,j}}{\sum_{i=1}^{n-j} C_{i,j}} \text{ et } \lambda_{i,j} = \frac{C_{i,j+1}}{C_{i,j}}.$$

Aussi, on peut obtenir ces coefficients à l'aide de régressions linéaires pondérées sans constantes, en régressant les $C_{\cdot,j+1}$ sur les $C_{\cdot,j}$. Ainsi, pour la première valeur,

```
> lm(PAID[,k+1]~0+PAID[,k],weights=1/PAID[,k])$coefficients
PAID[, k]
 1.380933
```

Le gros avantage numérique de cette méthode est que si des valeurs sont manquantes dans le tables, la fonction reste valide

```
> LAMBDA <- rep(NA,nc-1)
> for(k in 1:(nc-1)){
+ LAMBDA[k]=lm(PAID[,k+1]~0+PAID[,k],
+ weights=1/PAID[,k])$coefficients}
> LAMBDA
[1] 1.380933 1.011433 1.004343 1.001858 1.004735
```

Une fois estimé le facteur de développement, rien de plus simple que de compléter le triangle, toujours en itérant, colonne après colonne :

```
> TRIANGLE <- PAID
> for(i in 1:(nc-1)){
+ TRIANGLE[(nl-i+1):(nl),i+1]=LAMBDA[i]*TRIANGLE[(nl-i+1):(nl),i]}
> TRIANGLE
     [,1]     [,2]     [,3]     [,4]     [,5]     [,6]
[1,] 3209 4372.000 4411.000 4428.000 4435.000 4456.000
[2,] 3367 4659.000 4696.000 4720.000 4730.000 4752.397
[3,] 3871 5345.000 5398.000 5420.000 5430.072 5455.784
[4,] 4239 5917.000 6020.000 6046.147 6057.383 6086.065
[5,] 4929 6794.000 6871.672 6901.518 6914.344 6947.084
[6,] 5217 7204.327 7286.691 7318.339 7331.939 7366.656
```

	0	1	2	3	4	5
0	3209	4372	4411	4428	4435	4456
1	3367	4659	4696	4720	4730	4752.4
2	3871	5345	5398	5420	5430.1	5455.8
3	4239	5917	6020	6046.15	6057.4	6086.1
4	4929	6794	6871.7	6901.5	6914.3	6947.1
5	5217	7204.3	7286.7	7318.3	7331.9	7366.7

TABLE 3.6 – Triangle des paiements cumulés, $\boldsymbol{C} = (C_{i,j})_{i+j\leq n}$ avec leur projection future $\widehat{\boldsymbol{C}} = (\widehat{C}_{i,j})_{i+j>n}$

Le montant de provisions est alors la différence entre ce que l'on pense payer pour chaque année de survenance (la dernière colonne, si on suppose qu'au bout de n années tous les sinistres auront été clôturés, sinon on rajoute un facteur supplémentaire comme nous le verrons dans la section **??**) et que ce l'on a déjà payé (la seconde diagonale)

```
> facteur=1
> chargeultime <- TRIANGLE[,nc]*facteur
> paiements <- diag(TRIANGLE[,nc:1])
> (RESERVES <- chargeultime-paiements)
[1]    0.00000   22.39684   35.78388   66.06466  153.08358 2149.65640
```

On note qu'ici `sum(RESERVES)` vaut `2426.985`, ce qui correspond au montant total de réserves qu'il convient d'allouer. Un algorithme plus rapide est d'utiliser directement la formule basée sur le produit des coefficients de transition. On a alors

```
> DIAG <- diag(TRIANGLE[,nc:1])
> PRODUIT <- c(1,rev(LAMBDA))
> sum((cumprod(PRODUIT)-1)*DIAG)
[1] 2426.985
```

Pour la suite, on pourra développer la function `Chainladder()` qui renverra la charge ultime par ligne et les réserves

```
> Chainladder<-function(TR,f=1){
+ nc <- ncol(TR); nl <- nrow(TR)
+ L <- rep(NA,nc-1)
+ for(k in 1:(nc-1)){
+ L[k]=lm(TR[,k+1]~0+ TR[,k],
+ weights=1/TR[,k])$coefficients}
```

```
+ TRc <- TR
+ for(i in 1:(nc-1)){
+ TRc[(nl-i+1):(nl),i+1]=L[i]* TRc[(nl-i+1):(nl),i]}
+ C <- TRc[,nc]*f
+ R <- (cumprod(c(1,rev(L)))-1)*diag(TR[,nc:1])
+ return(list(charge=C,reserves=R,restot=sum(R)))
+ }
> Chainladder(PAID)
$charge
[1] 4456.000 4752.397 5455.784 6086.065 6947.084 7366.656

$reserves
[1]    0.00000   22.39684   35.78388   66.06466  153.08358 2149.65640

$restot
[1] 2426.985
```

3.3 De Mack à Merz & Wüthrich

La méthode dite *Chain Ladder*, que nous venons de voir, est une méthode dite déterministe, au sens où l'on ne construit pas de modèle probabiliste permettant de mesurer l'incertitude associée à la prédiction du montant des réserves. Différents modèles ont été proposés à partir des années 90, à partir du modèles de Mack, jusqu'à l'approche proposée par Merz & Wüthrich qui introduira la notion d'*incertitude à un an*.

3.3.1 Quantifier l'incertitude dans une prédiction

Nous avons obtenu, par la méthode Chain Ladder, un estimateur du montant de provision, $\widehat{R}$. Classiquement, pour quantifier l'erreur associée à un estimateur, on calcul la *mean squared error* - mse - associée,

$$\mathbb{E}\left([\widehat{R}-R]^2\right)$$

Formellement, comme R est ici une variable aléatoire, on ne parle pas de mse, mais de mse *de prédiction*, notée msep (on ne prédit pas sur les données passées, mais on utilisera les donnéees pour calibrer un modèle qui servira ensuite à faire de la prédiction pour les années futures). Aussi

$$\text{msep}(\widehat{R}) = \mathbb{E}\left([\widehat{R}-R]^2\right).$$

Ce terme peut se décomposer en deux (en faisant une approximation au premier ordre), au sens où

$$\mathbb{E}\left([\widehat{R}-R]^2\right) \approx \underbrace{\mathbb{E}\left([\widehat{R}-\mathbb{E}(R)]^2\right)}_{\text{mse}(\widehat{R})} + \underbrace{\mathbb{E}\left([R-\mathbb{E}(R)]^2\right)}_{\mathbb{V}(R)}$$

où le terme de gauche est l'erreur d'estimation, compte tenu du fait que nous avons dû estimer le montant de provisions à partir de la partie supérieure du triangle, et le terme de droite est l'erreur classique de modèle (tout modèle comportant une partie résiduelle orthogonale aux observations, et donc imprévisible).

En fait, en toute rigueur (et nous en aurons besoin par la suite), on cherche plutôt à calculer un msep conditionnel à l'information dont on dispose au bout de n années,

$$\text{msep}_n(\widehat{R}) = \mathbb{E}([\widehat{R} - R]^2 | \mathcal{H}_n).$$

3.3.2 Le formalisme de Mack

Mack (1993*a*) a proposé un cadre probabiliste afin de justifier l'utilisation de la méthode Chain-Ladder. Pour cela, on suppose que $(C_{i,j})_{j\geq 0}$ est un processus Markovien, et qu'il existe $\boldsymbol{\lambda} = (\lambda_j)$ et $\boldsymbol{\sigma} = (\sigma_j^2)$ tels que

$$\left\{ \begin{array}{l} \mathbb{E}(C_{i,j+1}|\mathcal{H}_{i+j}) = \mathbb{E}(C_{i,j+1}|C_{i,j}) = \lambda_j \cdot C_{i,j} \\ \text{Var}(C_{i,j+1}|\mathcal{H}_{i+j}) = \text{Var}(C_{i,j+1}|C_{i,j}) = \sigma_j^2 \cdot C_{i,j} \end{array} \right.$$

On note que sous ces hypothèses,

$$\mathbb{E}(C_{i,j+k}|\mathcal{H}_{i+j}) = \mathbb{E}(C_{i,j+k}|C_{i,j}) = \lambda_j \cdot \lambda_{j+1} \cdots \lambda_{j+k-1} C_{i,j}$$

Mack (1993*a*) rajoute une hypothèse supplémentaire d'indépendance entre les années de survenance, autrement dit $(C_{i,j})_{j=1,\ldots,n}$ et $\left(C_{i',j}\right)_{j=1,\ldots,n}$ sont indépendant pour tout $i \neq i'$.

Une réécriture du modèle est alors de supposer que

$$C_{i,j+1} = \lambda_j C_{i,j} + \sigma_j \sqrt{C_{i,j}} \cdot \varepsilon_{i,j},$$

où les résidus $(\varepsilon_{i,j})$ sont i.i.d., centrés et de variance unitaire. A partir de cette écriture, il peut paraître légitime d'utiliser les méthodes des moindres carrés pondérés pour estimer ces coefficients, en notant que les poids doivent être inversement proportionnels à la variance, autrement dit aux $C_{i,j}$, i.e. à j donné, on cherche à résoudre

$$\min \left\{ \sum_{i=1}^{n-j} \frac{1}{C_{i,j}} \left(C_{i,j+1} - \lambda_j C_{i,j} \right)^2 \right\}.$$

Pour tester ces deux premières hypothèses, on commence par représenter les $C_{\cdot,j+1}$ en fonction des $C_{\cdot,j}$ à j donné. Si la première hypothèse est vérifiée, les points doivent être alignés suivant une droite passant par l'origine.

```
> par(mfrow = c(1, 2))
> j=1
> plot(PAID[,j],PAID[,j+1],pch=19,cex=1.5)
> abline(lm(PAID[,j+1]~0+PAID[,j],weights=1/PAID[,j]))
> j=2
> plot(PAID[,j],PAID[,j+1],pch=19,cex=1.5)
> abline(lm(PAID[,j+1]~0+PAID[,j],weights=1/PAID[,j]))
> par(mfrow = c(1, 1))
```

La Figure 3.2 permet de visualiser l'hypothèse de cadence de paiements stale dans le temps. La régression est pondérée avec les mêmes poids que ceux utilisés pour estimer les coefficients de transition par régression,

Pour la seconde, on peut étudier les résidus standardisés (Mack (1993*a*) parle de *weighted residuals*), $\epsilon_{i,j} = \dfrac{C_{i,j+1} - \widehat{\lambda}_j C_{i,j}}{\sqrt{C_{i,j}}}$.

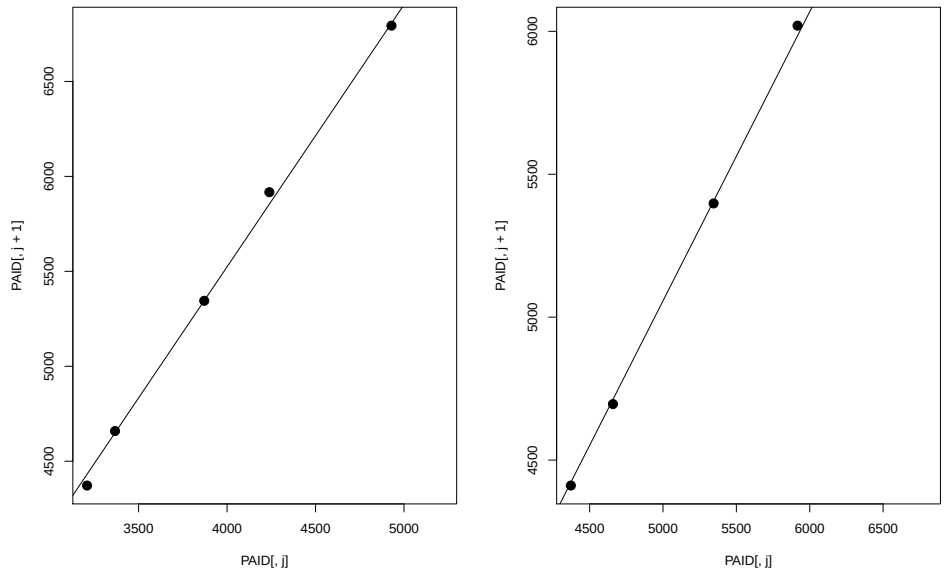


FIGURE 3.2 – Nuage des $C_{\cdot,j+1}$ en fonction des $C_{\cdot,j}$ pour $j = 1$ à gauche, et $j = 2$ à droite. La droite de régression passe par l'origine et les poids sont inversement proportionnels au montant de paiements.

```
> j=1
> RESIDUS <- (PAID[,j+1]-LAMBDA[j]*PAID[,j])/sqrt(PAID[,j])
```

L'utilisation des résidus standardisés nous donnent d'ailleurs une idée simple pour estimer le paramètre de volatilité.

$$\widehat{\sigma}_j^2 = \frac{1}{n-j-1} \sum_{i=0}^{n-j-1} \left(\frac{C_{i,j+1} - \widehat{\lambda}_j C_{i,j}}{\sqrt{C_{i,j}}} \right)^2$$

ce qui peut aussi s'écrire

$$\widehat{\sigma}_j^2 = \frac{1}{n-j-1} \sum_{i=0}^{n-j-1} \left(\frac{C_{i,j+1}}{C_{i,j}} - \widehat{\lambda}_j \right)^2 \cdot C_{i,j}$$

(ce qui est à rapprocher de l'écriture du facteur de transition λ_j comme moyenne pondérée des facteurs de transitions observés).

```
> lambda <- PAID[,2:nc]/PAID[,1:(nc-1)]
> SIGMA <- rep(NA,nc-1)
> for(i in 1:(nc-1)){
+ D <- PAID[,i]*(lambda[,i]-t(rep(LAMBDA[i],nc)))^2
+ SIGMA[i] <- 1/(nc-i-1)*sum(D[,1:(nc-i)])}
>  SIGMA[nc-1] <- min(SIGMA[(nc-3):(nc-2)])
> (SIGMA=sqrt(SIGMA))
[1] 0.72485777 0.32036422 0.04587297 0.02570564 0.02570564
```

Cette méthode permet d'estimer les différents paramètres intervenants dans le modèle de Mack (1993*a*).

3.3.3 La notion de *tail factor*

Pour l'instant, nous supposions que la première ligne de notre triangle est close : il n'y a plus de sinistres ouverts, et donc le montant de provision pour cette année de survenance est nul. Cette ligne servira de base pour tous les développements ultérieurs. Cette hypothèse peut être un peu trop forte pour les branches à déroulement long. Mack (1993*b*) a posé les bases des premiers modèles toujours utilisés permettant de s'affranchir de cette hypothèse. On supposera qu'il existe alors un $\lambda_\infty > 1$ tel que

$$C_{i,\infty} = C_{i,n} \times \lambda_\infty.$$

Une méthode (qui a souvent été utilisée) repose sur l'idée que l'on pouvait projeter les λ_i par une extrapolation exponentielle (ou une extrapolation linéaire des $\log(\lambda_k - 1)$), puis on pose

$$\lambda_\infty = \prod_{k \geq n} \widehat{\lambda}_k.$$

Mais mieux vaut faire attention, en particulier s'il y a des valeurs aberrantes.

```
> logL <- log(LAMBDA-1)
> tps <- 1:(nc-1)
> modele <- lm(logL~tps)
> plot(tps,logL,xlim=c(1,20),ylim=c(-30,0))
> abline(modele)
> tpsP <- seq(6,1000)
> logP <- predict(modele,newdata=data.frame(tps=tpsP))
> points(tpsP,logP ,pch=0)
> (facteur <- prod(exp(logP)+1))
[1] 1.000707
```

Autrement dit, cette méthode prévoit de rajouter 0.07% de charge par rapport à la prédiction faite par les méthodes classiques, en supposant la première année close. Numériquement, cela donnerait pour le montant de provision

```
> chargeultime <- TRIANGLE[,nc]*facteur
> paiements <- diag(TRIANGLE[,nc:1])
> (RESERVES <- chargeultime-paiements)
[1]    3.148948   25.755248   39.639346   70.365538  157.992918 2154.862234
> sum(RESERVES)
[1] 2451.764
```

La Figure 3.3 permet de visualiser le modèle linéaire ajusté sur le logarithme des facteurs de transition

3.3.4 Des estimateurs des paramètres à l'incertitude sur le montant des provisions

A partir de tous ces estimateurs, on peut estimer le msep du montant de provision par année de survenance, $\widehat{R}_i$, mais aussi agrégé, toutes années de survenances confondues. Les formules sont données dans Mack (1993*b*) ou Denuit & Charpentier (2005) ou Mack (1994). Numériquement, on peut utiliser la fonction `MackChainLadder` de `library(ChainLadder)`.

```
> library(ChainLadder)
> MackChainLadder(PAID)
```

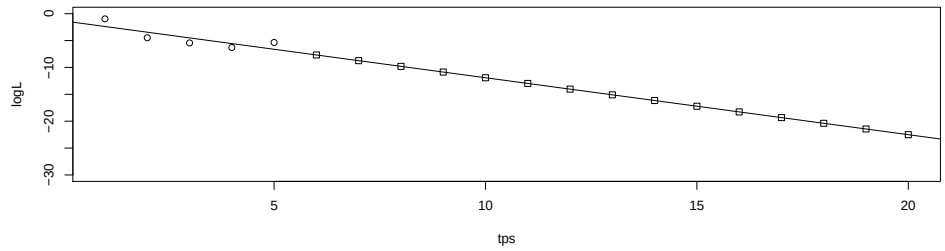


FIGURE 3.3 – Evolution des $\log(\lambda_j - 1)$ et prédiction par un modèle linéaire.

```
MackChainLadder(Triangle = PAID)
```

	Latest	Dev.To.Date	Ultimate	IBNR	Mack.S.E	CV(IBNR)
1	4,456	1.000	4,456	0.0	0.000	NaN
2	4,730	0.995	4,752	22.4	0.639	0.0285
3	5,420	0.993	5,456	35.8	2.503	0.0699
4	6,020	0.989	6,086	66.1	5.046	0.0764
5	6,794	0.978	6,947	153.1	31.332	0.2047
6	5,217	0.708	7,367	2,149.7	68.449	0.0318

```
              Totals
Latest:    32,637.00
Ultimate:  35,063.99
IBNR:       2,426.99
Mack S.E.:     79.30
CV(IBNR):       0.03
```

On retrouve l'estimation du montant total de provisions $\widehat{R}$, `IBNR`, qui vaut `2,426.99`, ainsi que msep$(\widehat{R})$ correspondant au `Mack S.E.` qui vaut ici `79.30`. Les informations par année de survenance i sont indiqués dans la première partie du tableau.

On obtient également plusieurs graphiques en utilisant la fonction `plot()`, correspondant aux Figures 3.4, 3.5 et 3.6

3.3.5 Un mot sur Munich-Chain Ladder

La méthode dite *Munich-Chain-Ladder*, developpée dans Quarg & Mack (2004), propose d'utiliser non seulement les paiements cumulés, mais aussi une autre information disponible : l'estimation des charges des différents sinistres faites par les gestionnaires de sinistres. Les triangles de paiements étaient basés sur des mouvements financiers ; ces triangles de charges sont basées sur des estimations faites par des gestionnaires compte tenu de l'information à leur disposition. Les sinistres tardifs ne sont pas dedans, et certains sinistres seront classés sans suite. Toutefois, il peut paraître légimite d'utiliser cette information.

Dans la méthode *Munich-Chain-Ladder*, on dispose des triangles $(C_{i,j})$ correspond aux paiments cumulés, et $(\Gamma_{i,j})$ les charges dites dossier/dossier. En reprenant les notations de Quarg

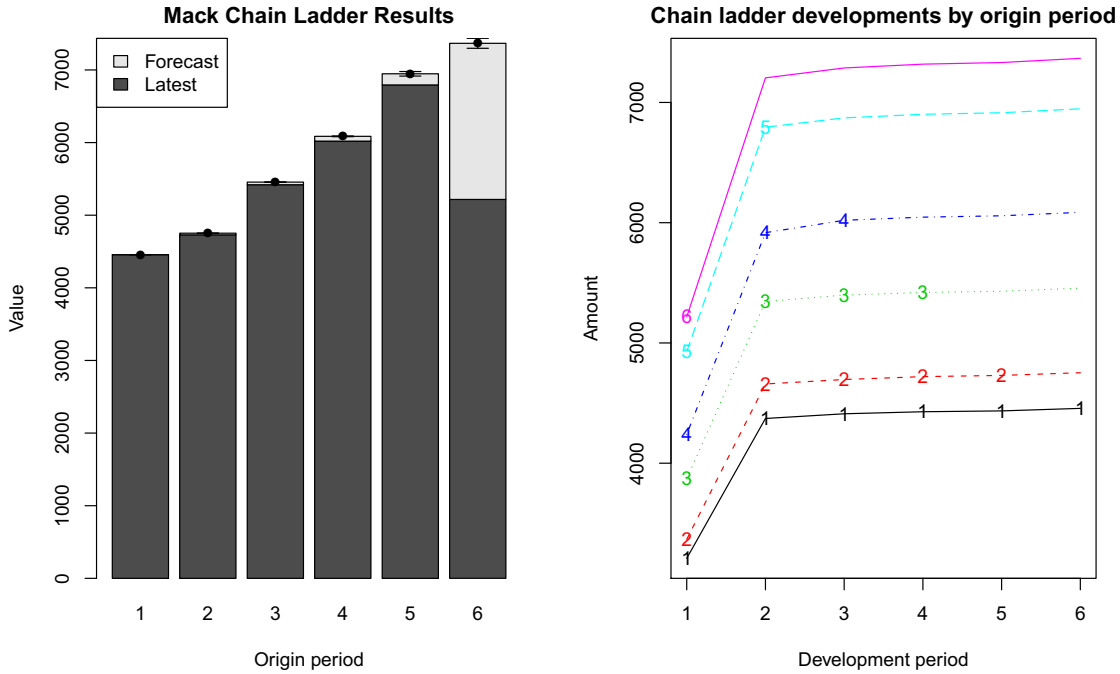


FIGURE 3.4 – Comparaison entre la charge finale estimée et la somme déjà payée, à gauche, et les cadences de paiements prédites par la méthode Chain Ladder.

& Mack (2004) on définie les ratio paiement/charge, et charge/paiement,

$$Q_{i,j} = \frac{C_{i,j}}{\Gamma_{i,j}} \text{ et } Q_{i,j}^{-1} = \frac{\Gamma_{i,j}}{C_{i,j}}$$

Comme dans le modèle Chain-Ladder de base, on suppose que

$$\begin{cases} \mathbb{E}(C_{i,j+1}|\mathcal{F}_{i+j}^{C}) = \lambda_j^C C_{i,j} \text{ et } \mathbb{V}(C_{i,j+1}|\mathcal{F}_{i+j}^{C}) = [\sigma_j^C]^2 C_{i,j} \\ \mathbb{E}(\Gamma_{i,j+1}|\mathcal{F}_{i+j}^{\Gamma}) = \lambda_j^\Gamma \Gamma_{i,j} \text{ et } \mathbb{V}(\Gamma_{i,j+1}|\mathcal{F}_{i+j}^{\Gamma}) = [\sigma_j^\Gamma]^2 \Gamma_{i,j} \end{cases}$$

On rajoute également une information sur les $\lambda_{i,j}$. Si

$$\lambda_{i,j-1}^C = \frac{C_{i,j}}{C_{i,j-1}} \text{ et } \lambda_{i,j-1}^\Gamma = \frac{\Gamma_{i,j}}{\Gamma_{i,j-1}}$$

on suppose que

$$\mathbb{E}(\lambda_{i,j-1}^C|\mathcal{F}_{i+j}) = \lambda_{j-1}^C + \lambda^C \sqrt{\mathbb{V}(\lambda_{i,j-1}^C|\mathcal{F}_{i+j}^C)} \cdot \frac{Q_{i,j-1}^{-1} - \mathbb{E}(Q_{i,j-1}^{-1}|\mathcal{F}_{i+j^C})}{\sqrt{\mathbb{V}(Q_{i,j-1}^{-1}|\mathcal{F}_{i+j^C})}}$$

et

$$\mathbb{E}(\lambda_{i,j-1}^\Gamma|\mathcal{F}_{i+j}) = \lambda_{j-1}^\Gamma + \lambda^\Gamma \sqrt{\mathbb{V}(\lambda_{i,j-1}^\Gamma|\mathcal{F}_{i+j})^\Gamma} \cdot \frac{Q_{i,j-1} - \mathbb{E}(Q_{i,j-1}|\mathcal{F}_{i+j^\Gamma})}{\sqrt{\mathbb{V}(Q_{i,j-1}|\mathcal{F}_{i+j^\Gamma})}}.$$

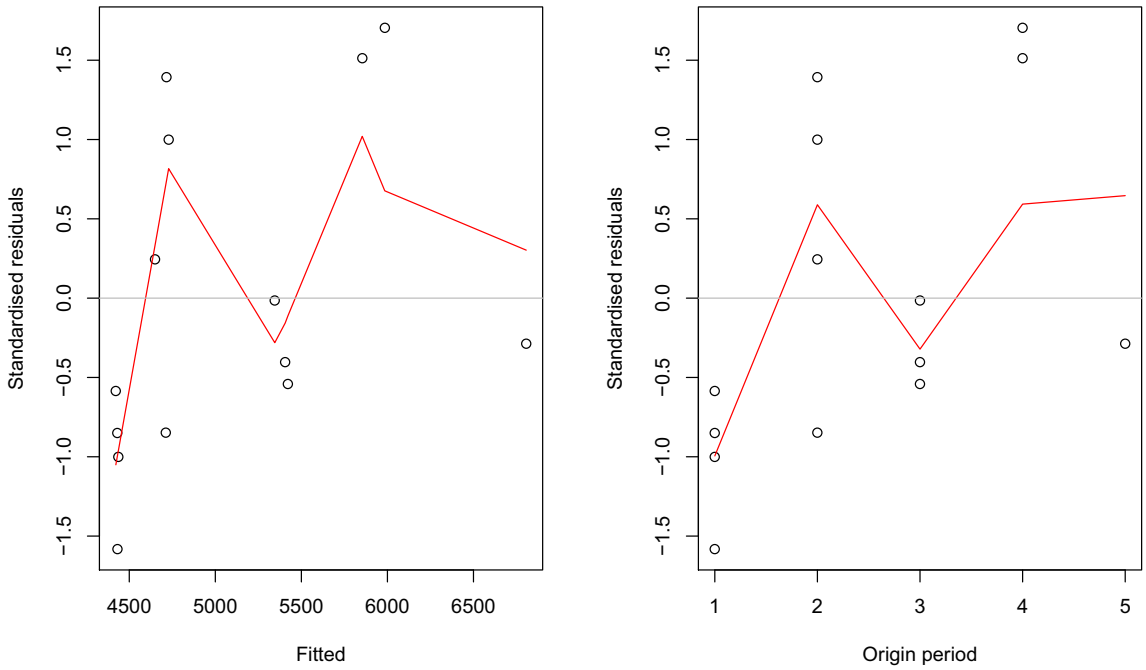


FIGURE 3.5 – Evolution des résidus standardisés en fonction des $\widehat{C}_{i,j}$ et des i.

On notera qu'il s'agit d'une extension du modèle Chain-Ladder, et en particulier

$$\mathbb{E}(\lambda^{\Gamma}_{i,j-1}|\mathcal{F}^{C}_{i+j}) = \mathbb{E}[\mathbb{E}(\lambda^{\Gamma}_{i,j-1}|\mathcal{F}_{i+j})|\mathcal{F}_{i+j})^{C}] = \lambda^{C}_{j-1}.$$

Les termes λ^{C} et λ^{Γ} sont alors simplement des coefficients de corrélation conditionnelle. Plus précisément

$$\lambda^{C} = \mathrm{Cor}(\Gamma_{i,j-1}, C_{i,j}|\mathcal{F}^{C}_{i+j-1})$$

Sous ces hypothèses, il est possible de construire des estimateurs sans biais de $\mathbb{E}(C_{i,j}|C_{i,j-1})$, de $\mathbb{E}(\Gamma_{i,j}|\Gamma_{i,j-1})$, de $\mathbb{E}(Q_{i,j}|\mathcal{F}^{\Gamma}_{i+j})$ et de $\mathbb{E}(Q^{-1}_{i,j}|\mathcal{F}^{C}_{i+j})$.

Pour estimer les deux dernières quantités, posons

$$\widehat{Q}_j = \frac{\sum_{i=0}^{n_j} C_{i,j}}{\sum_{i=0}^{n_j} \Gamma_{i,j}} = \frac{1}{\widehat{Q^{-1}}_j}$$

On peut aussi estimer les variances conditionnelles. Par exemple

$$\widehat{\mathbb{V}}(Q_{i,j}||\mathcal{F}^{\Gamma}_{i+j}) = ()^{-1} \sum_{i=0}^{n-j} \Gamma_{i,j}[Q_{i,j} - \widehat{Q}_j]^2$$

et une expression analogue pour $\widehat{\mathbb{V}}(Q^{-1}_{i,j}|\mathcal{F}^{C}_{i+j})$.

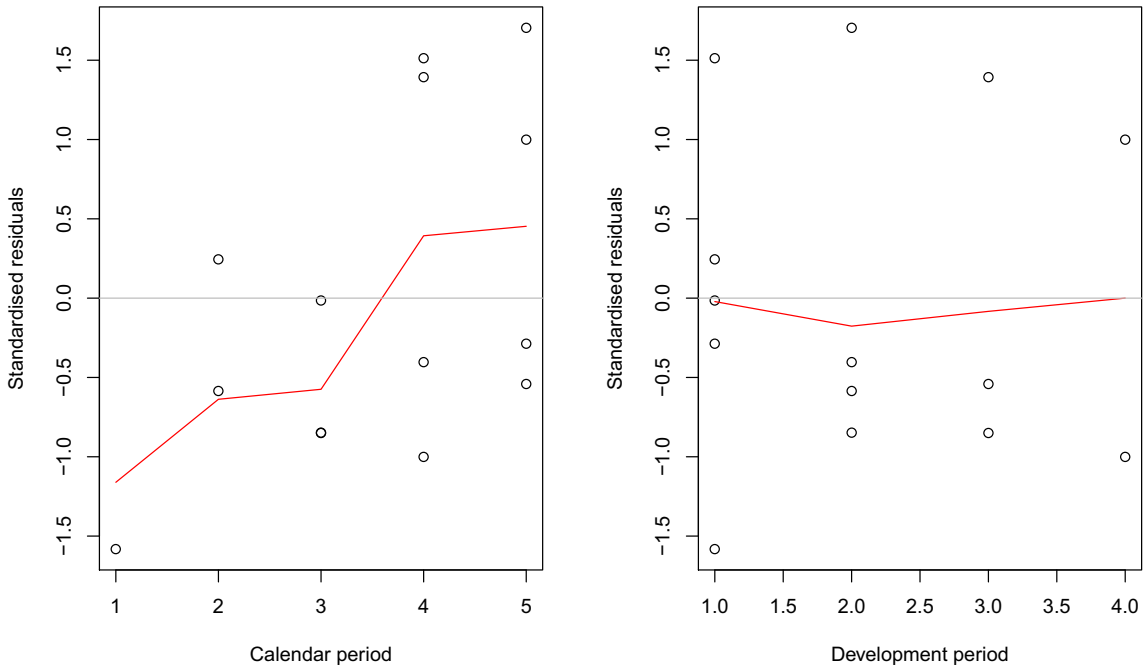


FIGURE 3.6 – Evolution des résidus standardisés en fonction de j et $i+j$.

A partir de ces quantités, posons enfin

$$\tilde{Q}_{i,j} = \frac{Q_{i,j} - \widehat{Q}_j}{\sqrt{\widehat{\mathbb{V}}(Q_{i,j}|\mathcal{F}^{\Gamma}_{i+j})}},$$

$$\tilde{\lambda}^{\Gamma}_{i,j} = \frac{\sqrt{\Gamma_{i,j-1}}}{[\widehat{\sigma}^{I}_{j-1}]^2[\lambda_{i,j-1} - \widehat{\lambda}_{j-1}]}$$

et

$$\widehat{\lambda}^{\Gamma} = \frac{\sum \tilde{Q}_{i,j-1}\tilde{\lambda}^{\Gamma}_{i,j}}{\sum \tilde{Q}^2_{i,j-1}}.$$

L'estimateur Munich-Chain-Ladder est construit de manière intérative. Le détails des formules est donné dans Quarg & Mack (2004) ou Wüthrich & Merz (2008).

```
> (MNCL=MunichChainLadder(Paid=PAID,
+ Incurred=INCURRED))
MunichChainLadder(Paid = PAID, Incurred = INCURRED)

  Latest Paid Latest Incurred Latest P/I Ratio Ult. Paid Ult. Incurred Ult. P/I Ratio
1       4,456           4,456            1.000     4,456         4,456             1
2       4,730           4,750            0.996     4,753         4,750             1
```

	0	1	2	3	4	5
0	4795	4629	4497	4470	4456	4456
1	5135	4949	4783	4760	4750	
2	5681	5631	5492	5470		
3	6272	6198	6131			
4	7326	7087				
5	7353					

TABLE 3.7 – Triangle des estimations de charges dossier/dossier cumulées, $\mathbf{\Gamma} = (\Gamma_{i,j})$

```
3       5,420           5,470           0.991     5,455           5,454           1
4       6,020           6,131           0.982     6,086           6,085           1
5       6,794           7,087           0.959     6,983           6,980           1
6       5,217           7,353           0.710     7,538           7,533           1

Totals
            Paid Incurred P/I Ratio
Latest:   32,637   35,247      0.93
Ultimate: 35,271   35,259      1.00
```

De même que pour la fonction `MackChainLadder`, plusieurs graphiques peuvent être obtenus afin de mieux comprendre les évolutions des paiements, mais aussi de la charge dossier/dossier estimée par les gestionnaires de sinistres, présentés sur les Figures 3.7 et 3.8.

Si on compare les deux triangles, qui ont été complétés en tenant compte des interactions, on obtient des choses relativement proches,

```
> MNCL$MCLPaid
     [,1]     [,2]     [,3]     [,4]     [,5]     [,6]
[1,] 3209 4372.000 4411.000 4428.000 4435.000 4456.000
[2,] 3367 4659.000 4696.000 4720.000 4730.000 4752.569
[3,] 3871 5345.000 5398.000 5420.000 5429.716 5455.324
[4,] 4239 5917.000 6020.000 6046.090 6057.284 6085.875
[5,] 4929 6794.000 6890.045 6932.247 6949.447 6982.539
[6,] 5217 7251.382 7419.621 7478.759 7502.149 7538.194
> sum(MNCL$MCLPaid[,6]-diag(MNCL$MCLPaid[,6:1]))
[1] 2633.502
> MNCL$MCLIncurred
     [,1]     [,2]    [,3]     [,4]     [,5]     [,6]
[1,] 4975 4629.000 4497.00 4470.000 4456.000 4456.000
[2,] 5135 4949.000 4783.00 4760.000 4750.000 4750.415
[3,] 5681 5631.000 5492.00 5470.000 5454.691 5454.445
[4,] 6272 6198.000 6131.00 6100.978 6084.986 6084.770
[5,] 7326 7087.000 6988.37 6984.274 6979.284 6979.732
[6,] 7353 7349.795 7493.64 7522.809 7532.206 7533.461
> sum(MNCL$MCLIncurred[,6]-diag(MNCL$MCLPaid[,6:1]))
[1] 2621.823
```

On peut également utiliser cette technique pour visualiser les cadences de paiement, mais aussi d'estimation de charge dossier par dossier, comme sur la Figure 3.9. On utilise la fonction

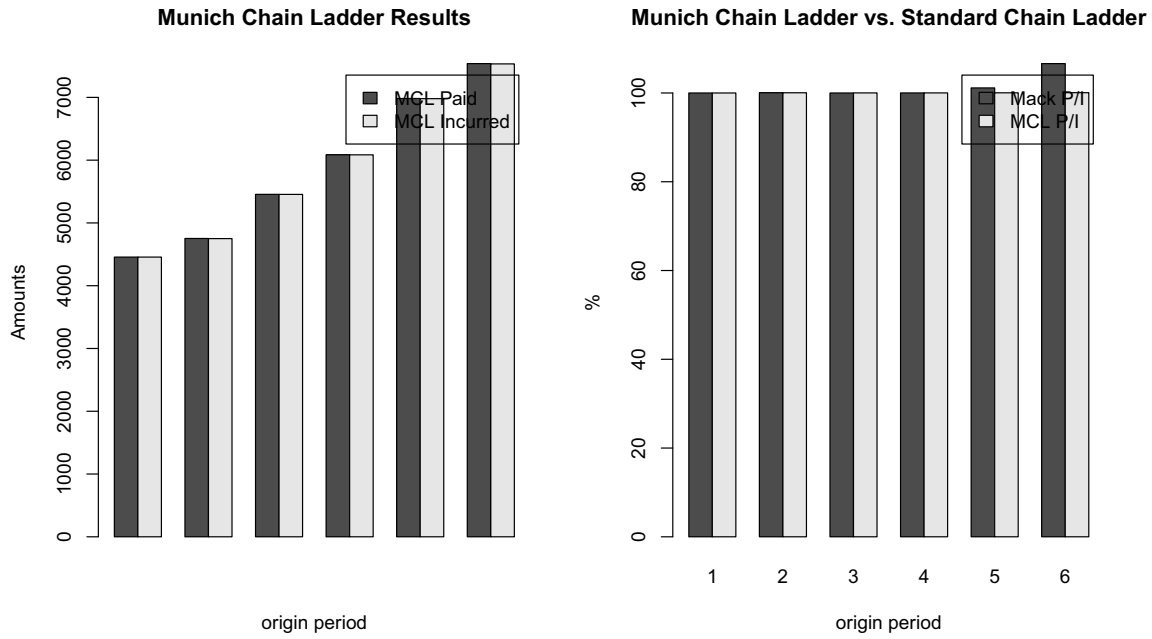


FIGURE 3.7 – Comparaison des méthodes Chain Ladder, et Munich Chain Ladder, en montant à gauche, et en valeurs relatives à droite.

```
> munich=function(k){
+ plot(0:(nc+1-k),c(0,MNCL$MCLPaid[k,1:(nc+1-k)]),pch=19,type="b",
+ ylim=c(0,max(c(MNCL$MCLPaid[k,],MNCL$MCLIncurred[k,]))),xlim=c(0,nc),
+ ylab="",xlab="")
+ lines(0:(nc+1-k),c(0,MNCL$MCLIncurred[k,1:(nc+1-k)]),pch=19,type="b")
+ lines((nc+1-k):nc,MNCL$MCLPaid[k,(nc+1-k):nc],pch=1,type="b")
+ lines((nc+1-k):nc,MNCL$MCLIncurred[k,(nc+1-k):nc],pch=1,type="b")
+ abline(v=nc+1-k,col="grey")
+ }
```

que l'on peut appeler pour deux années de développement différentes, une année close à gauche, et vu après 2 années de développement à droite

3.3.6 L'incertitude à un an de Merz & Wüthrich

Merz & Wüthrich (2008) ont étudier la variation du boni/mali d'une année sur l'autre (appelé $\mathrm{CDR}_i(n)$, *claims development result*), c'est à dire du changement dans la prédiction de la charge totale. Ils ont en particulier montré que

$$\widehat{\mathrm{msepc}}_{n-1}(\mathrm{CDR}_i(n)) = \widehat{C}_{i,\infty}^2 \left(\widehat{\Gamma}_{i,n} + \widehat{\Delta}_{i,n} \right)$$

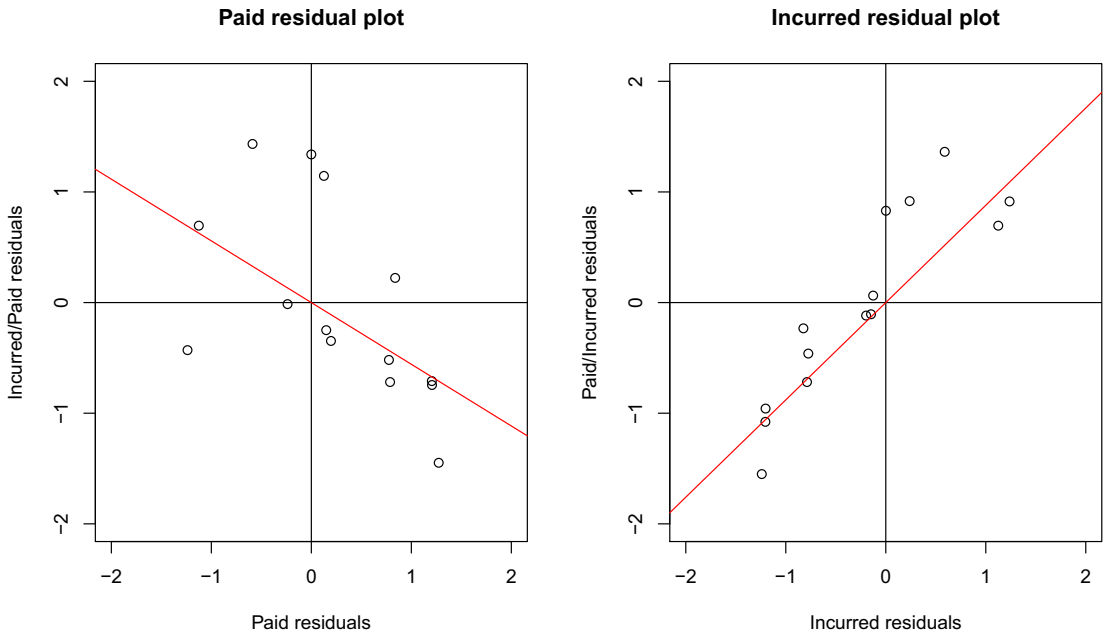


FIGURE 3.8 – Corrélations entre les triangles de développement des paiements, et des charges dossier/dossier.

où

$$\widehat{\Delta}_{i,n} = \frac{\widehat{\sigma}^2_{n-i+1}}{\widehat{\lambda}^2_{n-i+1} S^{n+1}_{n-i+1}} + \sum_{j=n-i+2}^{n-1} \left(\frac{C_{n-j+1,j}}{S^{n+1}_j} \right)^2 \frac{\widehat{\sigma}^2_j}{\widehat{\lambda}^2_j S^n_j}$$

et

$$\widehat{\Gamma}_{i,n} = \left(1 + \frac{\widehat{\sigma}^2_{n-i+1}}{\widehat{\lambda}^2_{n-i+1} C_{i,n-i+1}} \right) \prod_{j=n-i+2}^{n-1} \left(1 + \frac{\widehat{\sigma}^2_j}{\widehat{\lambda}^2_j [S^{n+1}_j]^2} C_{n-j+1,j} \right) - 1$$

Merz & Wüthrich (2008) ont alors approché ce terme par

$$\widehat{\Gamma}_{i,n} \approx \frac{\widehat{\sigma}^2_{n-i+1}}{\widehat{\lambda}^2_{n-i+1} C_{i,n-i+1}} + \sum_{j=n-i+2}^{n-1} \left(\frac{C_{n-j+1,j}}{S^{n+1}_j} \right)^2 \frac{\widehat{\sigma}^2_j}{\widehat{\lambda}^2_j C_{n-j+1,j}}$$

en faisant tout simplement un développement de la forme $\prod(1+u_i) \approx 1 + \sum u_i$, mais qui n'est valide que si u_i est petit, soit ici

$$\frac{\widehat{\sigma}^2_j}{\widehat{\lambda}^2_j} << C_{n-j+1,j}.$$

Ces estimation peuvent être obtenues à l'aide de la fonction `MackMerzWuthrich()` (de Lacoume (2009)), avec le MSEP de Mack, puis les deux de Merz & Wüthrich (avec ou non le terme approché), avec les 6 années de survenance en ligne, et en bas le cumul toutes années confondues,

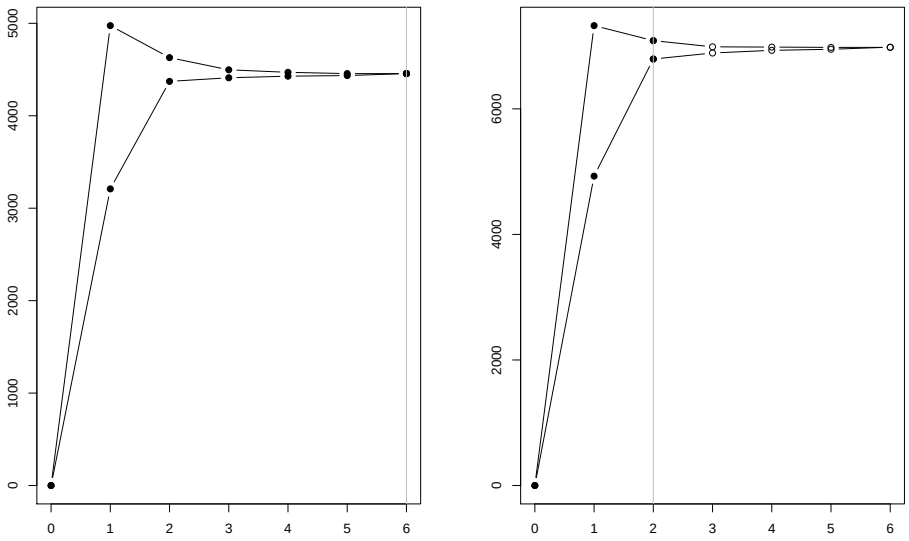


FIGURE 3.9 – Evolution des paiements, et de la charge dossier par dossier.

```
> MackMerzWuthrich(PAID)
     MSEP Mack MSEP MW app. MSEP MW ex.
1    0.0000000     0.000000    0.000000
2    0.6393379     1.424131    1.315292
3    2.5025153     2.543508    2.543508
4    5.0459004     4.476698    4.476698
5   31.3319292    30.915407   30.915407
6   68.4489667    60.832875   60.832898
tot 79.2954414    72.574735   72.572700
```

3.4 Régression Poissonnienne et approches économétriques

Dans cette section, nous nous éloignerons des modèles récursifs inspirés de la méthode Chain Ladder, et nous reviendrons sur des classes de modèles très utilisés dans les années 70, appelés *modèles à facteurs*, remis au goût du jour en proposant une relecture économétrique de ces modèles, permettant ainsi d'obtenir des intervalles de confiance des différentes grandeurs (comme initié par Verrall (2000)).

3.4.1 Les modèles à facteurs, un introduction historique

Avant de présenter l'utilisation des modèles de régression, on peut commencer par évoquer des modèles plus anciens. Par exemple Taylor (1977) supposait que

$$Y_{i,j} = r_j \cdot \mu_{i+j}, \text{ pour tout } i, j,$$

i.e. un effet colonne, de cadence de paiement, et un effet diagonal, que Taylor interprète comme un facteur d'inflation. Ce modèle peut se réécrire, dès lors qu'il n'y a pas d'incrément positif,

$$\log Y_{i,j} = \alpha_i + \gamma_{i+j}$$

qui prend alors une forme linéaire. On montrera par la suite que le cas

$$\log Y_{i,j} = \alpha_i + \beta_j$$

s'apparent à un modèle de type Chain-Ladder. En effet, cela suppose que

$$Y_{i,j} = a_i \times b_j$$

que l'on peut rapprocher du modèle de développement $Y_{i,j} = C_{i,n} \times \varphi_j$. Zehnwirth (1985) avait également proposé d'utiliser une courbe d'Hoerl, c'est à dire

$$\log Y_{i,j} = \alpha_i + \beta_i \cdot \log(j) + \gamma_i \cdot j.$$

3.4.2 Les modèles de de Vylder et de Christophides

De Vylder (22-28) a été un des premiers modèles économétriques de provisionnement. On suppose que

$$Y_{i,j} \sim \mathcal{N}(\alpha_i \cdot \beta_j, \sigma^2), \text{ pour tout } i, j.$$

On peut estimer les coefficients par moindres carrés,

$$(\widehat{\boldsymbol{\alpha}}, \widehat{\boldsymbol{\beta}}) = \text{argmin} \left\{ \sum_{i,j} [Y_{i,j} - \alpha_i \cdot \beta_j]^2 \right\}.$$

Les équations normales s'écrivent ici

$$\widehat{\alpha}_i = \frac{\sum_j Y_{i,j} \cdot \widehat{\beta}_j}{\sum_j \widehat{\beta}_j^2} \text{ et } \widehat{\beta}_j = \frac{\sum_i Y_{i,j} \cdot \widehat{\alpha}_i}{\sum_i \widehat{\alpha}_i^2},$$

ce qui ne résoud pas explicitement. Pour le résoudre, Christofides (1989) a suggéré de le réécrire comme un modèle log-linéaire, i.e.

$$\log Y_{i,j} \sim \mathcal{N}(a_i + b_j, \sigma^2), \text{ pour tout } i, j.$$

```
> ligne <- rep(1:nl, each=nc); colonne <- rep(1:nc, nl)
> INC <- PAID
> INC[,2:6] <- PAID[,2:6]-PAID[,1:5]
> Y <- as.vector(INC)
> lig <- as.factor(ligne)
> col <- as.factor(colonne)
> reg <- lm(log(Y)~col+lig)
> summary(reg)

Call:
lm(formula = log(Y) ~ col + lig)
```

```
Residuals:
     Min       1Q   Median       3Q      Max
-0.26374 -0.05681  0.00000  0.04419  0.33014

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   7.9471     0.1101  72.188 6.35e-15 ***
col2          0.1604     0.1109   1.447  0.17849
col3          0.2718     0.1208   2.250  0.04819 *
col4          0.5904     0.1342   4.399  0.00134 **
col5          0.5535     0.1562   3.543  0.00533 **
col6          0.6126     0.2070   2.959  0.01431 *
lig2         -0.9674     0.1109  -8.726 5.46e-06 ***
lig3         -4.2329     0.1208 -35.038 8.50e-12 ***
lig4         -5.0571     0.1342 -37.684 4.13e-12 ***
lig5         -5.9031     0.1562 -37.783 4.02e-12 ***
lig6         -4.9026     0.2070 -23.685 4.08e-10 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.1753 on 10 degrees of freedom
  (15 observations deleted due to missingness)
Multiple R-squared: 0.9975,        Adjusted R-squared: 0.9949
F-statistic: 391.7 on 10 and 10 DF,  p-value: 1.338e-11
```

On peut alors simplement utiliser cette régression pour construire le triangle de base du modèle. Comme nous l'avions noté dans la Section 2.5.1, on ne peut pas utiliser , $\widehat{Y}_{i,j} = \exp[\widehat{a}_i + \widehat{b}_j]$ car cet estimateur est toutefois biaisé. Si l'on corrige du biais (car $\exp(\mathbb{E}(\log(Y))) \neq \mathbb{E}(Y)$) en posant $\widehat{Y}_{i,j} = \exp[\widehat{a}_i + \widehat{b}_j + \widehat{\sigma}^2/2]$, on obtient alors

```
> sigma <- summary(reg)$sigma
> (INCpred <- matrix(exp(logY+sigma^2/2),nl,nc))
         [,1]     [,2]     [,3]     [,4]     [,5]     [,6]
[1,] 2871.209 1091.278 41.66208 18.27237  7.84125 21.32511
[2,] 3370.826 1281.170 48.91167 21.45193  9.20570 25.03588
[3,] 3767.972 1432.116 54.67438 23.97937 10.29030 27.98557
[4,] 5181.482 1969.357 75.18483 32.97495 14.15059 38.48403
[5,] 4994.082 1898.131 72.46559 31.78233 13.63880 37.09216
[6,] 5297.767 2013.554 76.87216 33.71498 14.46816 39.34771
> sum(exp(logY[is.na(Y)==TRUE]+sigma^2/2))
[1] 2481.857
```

qui est très proche de ce que nous avions eu dans la section précédante.

3.4.3 La régression poissonnienne de Hachemeister & Stanard

Hachemeister & Stanard (1975), Kremer (1982) et enfin Mack (1991) ont montré que dans une régression log-Poisson sur les incréments, la somme des prédictions des paiments à venir correspond à l'estimateur Chain Ladder. On retrouve ici un résultat pouvant être relié à la méthode des marges présentée dans la section 2.4.1.

```
> CL <- glm(Y~lig+col, family=poisson)
> summary(CL)

Call:
glm(formula = Y ~ lig + col, family = poisson)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.3426  -0.4996   0.0000   0.2770   3.9355

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  8.05697    0.01551 519.426  < 2e-16 ***
lig2        -0.96513    0.01359 -70.994  < 2e-16 ***
lig3        -4.14853    0.06613 -62.729  < 2e-16 ***
lig4        -5.10499    0.12632 -40.413  < 2e-16 ***
lig5        -5.94962    0.24279 -24.505  < 2e-16 ***
lig6        -5.01244    0.21877 -22.912  < 2e-16 ***
col2         0.06440    0.02090   3.081  0.00206 **
col3         0.20242    0.02025   9.995  < 2e-16 ***
col4         0.31175    0.01980  15.744  < 2e-16 ***
col5         0.44407    0.01933  22.971  < 2e-16 ***
col6         0.50271    0.02079  24.179  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 46695.269  on 20  degrees of freedom
Residual deviance:    30.214  on 10  degrees of freedom
  (15 observations deleted due to missingness)
AIC: 209.52

Number of Fisher Scoring iterations: 4
```

Notons dès à présent que le modèle de Poisson n'est pas forcément le plus adapté. En effet, il y a une (forte) surdispersion des paiements,

```
> CL <- glm(Y~lig+col, family=quasipoisson)
> summary(CL)

Call:
glm(formula = Y ~ lig + col, family = quasipoisson)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.3426  -0.4996   0.0000   0.2770   3.9355

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

```
(Intercept)  8.05697    0.02769 290.995  < 2e-16 ***
lig2        -0.96513    0.02427 -39.772 2.41e-12 ***
lig3        -4.14853    0.11805 -35.142 8.26e-12 ***
lig4        -5.10499    0.22548 -22.641 6.36e-10 ***
lig5        -5.94962    0.43338 -13.728 8.17e-08 ***
lig6        -5.01244    0.39050 -12.836 1.55e-07 ***
col2         0.06440    0.03731   1.726 0.115054
col3         0.20242    0.03615   5.599 0.000228 ***
col4         0.31175    0.03535   8.820 4.96e-06 ***
col5         0.44407    0.03451  12.869 1.51e-07 ***
col6         0.50271    0.03711  13.546 9.28e-08 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for quasipoisson family taken to be 3.18623)

    Null deviance: 46695.269  on 20  degrees of freedom
Residual deviance:    30.214  on 10  degrees of freedom
  (15 observations deleted due to missingness)
AIC: NA

Number of Fisher Scoring iterations: 4
```

Il y a ici un $2n-1$ paramètres à estimer, γ, $\boldsymbol{c}=(c_1,\cdots,c_{n-1})$ et $\boldsymbol{r}=(r_1,\cdots,r_{n-1})$ (sans compter le paramètre *phi* de surdispersion). Compte tenu du choix des facteurs (ici un facteur ligne r et un facteur colonne c), une fois estimés ces paramètres, il est possible de *prédire* la partie inférieure du triangle très simplement, i.e.

$$\widehat{Y}_{i,j}=\widehat{\mu}_{i,j}=\exp[\widehat{\gamma}+\widehat{r}_i+\widehat{c}_j]$$

```
> Ypred <- predict(CL,newdata=data.frame(lig,col),type="response")
> (INCpred <- matrix(Ypred,nl,nc))
         [,1]     [,2]     [,3]     [,4]      [,5]     [,6]
[1,] 3155.699 1202.110 49.82071 19.14379  8.226405 21.00000
[2,] 3365.605 1282.070 53.13460 20.41717  8.773595 22.39684
[3,] 3863.737 1471.825 60.99889 23.43904 10.072147 25.71173
[4,] 4310.096 1641.858 68.04580 26.14685 11.235734 28.68208
[5,] 4919.862 1874.138 77.67250 29.84594 12.825297 32.73985
[6,] 5217.000 1987.327 82.36357 31.64850 13.599887 34.71719
```

On retrouve ici l'estimateur obtenu par la méthode Chain-Ladder,

```
> sum(Ypred[is.na(Y)==TRUE])
[1] 2426.985
```

La valeur de référence est la valeur dans le coin supérieur gauche. Compte tenu de la forme logarithmique du modèle, on a une interprétation simple de toutes les valeurs, relativement à cette première valeur

$$\mathbb{E}(Y_{i,j}|\mathcal{F}_n)=\mathbb{E}(Y_{0,0}|\mathcal{F}_n)\cdot\exp[r_i+c_j].$$

3.4.4 Incréments négatifs

Dans certains triangles, il n'est pas rare d'avoir des *incréments négatifs*. Considérons par exemple le triangle de paiements suivant,

```
> PAIDneg
     [,1] [,2] [,3] [,4] [,5] [,6]
[1,] 3209 4372 4411 4428 4435 4456
[2,] 3367 4659 4696 4720 4730   NA
[3,] 3871 5345 5338 5420   NA   NA
[4,] 4239 5917 6020   NA   NA   NA
[5,] 4929 6794   NA   NA   NA   NA
[6,] 5217   NA   NA   NA   NA   NA
> INCneg=PAIDneg
> INCneg[,2:6]=PAIDneg[,2:6]-PAIDneg[,1:5]
> INCneg
     [,1] [,2] [,3] [,4] [,5] [,6]
[1,] 3209 1163   39   17    7   21
[2,] 3367 1292   37   24   10   NA
[3,] 3871 1474   -7   82   NA   NA
[4,] 4239 1678  103   NA   NA   NA
[5,] 4929 1865   NA   NA   NA   NA
[6,] 5217   NA   NA   NA   NA   NA
```

Cet incrément négatif de paiement ne perturbe aucunement la méthode Chain Ladder,

```
> LAMBDAneg <- rep(NA,nc-1)
> for(k in 1:(nc-1)){
+ LAMBDAneg[k]=lm(PAIDneg[,k+1]~0+PAIDneg[,k],
+ weights=1/PAIDneg[,k])$coefficients}
> TRIANGLEneg <- PAIDneg
> for(i in 1:(nc-1)){
+ TRIANGLEneg[(nl-i+1):(nl),i+1]=LAMBDAneg[i]*
+ TRIANGLEneg[(nl-i+1):(nl),i]}
> chargeultimeneg <- TRIANGLEneg[,nc]
> paiementsneg <- diag(TRIANGLEneg[,nc:1])
> RESERVESneg <- chargeultimeneg-paiementsneg
> sum(RESERVESneg)
[1] 2469.703
```

En revanche, les deux méthodes de régression que l'on vient de présenter ne peuvent plus être appliquées. Si malgré tout on souhaite utiliser cette technique, une première solution consiste à rebalancer des paiments d'une année sur l'autre. On peut alors prendre à gauche ou à droite de manière à ne plus avoir cet incrément de paiement négatif. Une autre stratégie peut être de faire des translations de certains incréments. Si on translate *toutes* les observations, le code ressemblerait à

```
> predict(glm((Y+k)~X),type="response")-k
```

En effet, dans un modèle linéaire Gaussien, translater les observations Y vers le haut puis translater vers le bas les prédictions $\widehat{Y}$ (d'un même montant) donne exactement les mêmes prédictions. Mais ce n'est pas le cas dans les modèles GLM.

Supposons que l'on translate les incréments de la *colonne* où figure l'incrément négatif, de telle sorte que tous les incréments soient alors positifs. On peut alors faire tourner une régression

de Poisson. En translatant de manière à annuler l'incrément négatif, on obtient le montant de provision suivant

```
> translation<-function(k){
+ Y=as.vector(INCneg)
+ Y[col==3]=Y[col==3]+k
+ base<-data.frame(Y,lig,col)
+ reg<-glm(Y~lig+col,
+ data=base,family=poisson(link="log"))
+ Yp=predict(reg,type="response",
+ newdata=base)
+ Yp[col==3]=Yp[col==3]-k
+ sum(Yp[is.na(Y)==TRUE])}
> translation(7)
[1] 2471.444
```

Une solution peut être alors de translater pour un certain nombre de valeurs, puis d'extrapoler la prédiction pour `k` nul,

```
> K<-7:20
> reserves<-Vectorize(translation)(K)
> (pRes<-predict(lm(reserves~K),newdata=(K=0)))
       1
2469.752
```

On peut d'ailleurs visualiser cette extrapolation sur la Figure 3.10

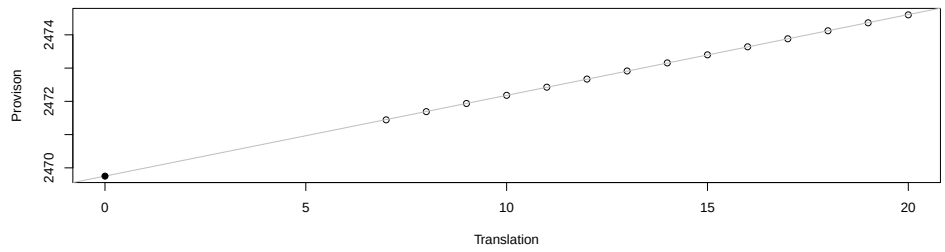


FIGURE 3.10 – Extrapolation du montant de provision sur pour une régression Poissonnienne avec un incrément négatif (avec translations).

3.4.5 Incertitude dans un modèle de régression

Nous avions noté auparavant qu'obtenir une estimation du montant de sinistres restant à payer ne suffisait pas, et qu'il fallait avoir un intervalle de confiance, ou au moins une mesure de la dispersion du vrai montant autour de cette valeur prédite, voire un quantile.

Les formules économétriques fermées

Les modèles de régressions pourraient paraître très intéressants car il existe des formules fermés pour toutes sortes de prédiction. Par exemple, dans une régression GLM avec un lien

logarithmique, rappelons que

$$\mathbb{E}(Y_{i,j}|\mathcal{F}_n) = \mu_{i,j} = \exp[\eta_{i,j}]$$

ou encore

$$\widehat{Y}_{i,j} = \widehat{\mu}_{i,j} = \exp[\widehat{\eta}_{i,j}].$$

La *delta method* nous permet d'écrire que

$$\mathbb{V}(\widehat{Y}_{i,j}) \approx \left|\frac{\partial \mu_{i,j}}{\partial \eta_{i,j}}\right|^2 \cdot \mathbb{V}(\widehat{\eta}_{i,j}),$$

ce qui se simplifie dans le cas où le lien est logarithmique, i.e.

$$\frac{\partial \mu_{i,j}}{\partial \eta_{i,j}} = \mu_{i,j}$$

Aussi, pour une loi de Poisson surdispersée (comme dans Renshaw & Verrall (1998)),

$$\mathbb{E}\left([Y_{i,j} - \widehat{Y}_{i,j}]^2\right) \approx \widehat{\phi} \cdot \widehat{\mu}_{i,j} + \widehat{\mu}_{i,j}^2 \cdot \widehat{\mathbb{V}}(\widehat{\eta}_{i,j})$$

pour la partie inférieure du triangle. De plus, car il sera nécessaire de sommer tous les termes de la partie inférieure du triangle pour déterminer le montant total de provisions,

$$\mathrm{Cov}(\widehat{Y}_{i,j}, \widehat{Y}_{k,l}) \approx \widehat{\mu}_{i,j} \cdot \widehat{\mu}_{k,l} \cdot \widehat{\mathrm{Cov}}(\widehat{\eta}_{i,j}, \widehat{\eta}_{k,l}).$$

Le montant de provision que l'on cherche à estimer étant la somme des prédictions de paiements à venir, $\widehat{R} = \sum_{i+j>n} \widehat{Y}_{i,j}$, alors

$$\mathbb{E}\left([R - \widehat{R}]^2\right) \approx \left(\sum_{i+j>n} \widehat{\phi} \cdot \widehat{\mu}_{i,j}\right) + \widehat{\boldsymbol{\mu}}_F' \cdot \widehat{\mathbb{V}}(\widehat{\boldsymbol{\eta}}_F) \cdot \widehat{\boldsymbol{\mu}}_F,$$

où les vecteurs $\widehat{\boldsymbol{\mu}}_F$ et $\widehat{\boldsymbol{\eta}}_F$ sont des restrictions des vecteurs $\widehat{\boldsymbol{\mu}}$ et $\widehat{\boldsymbol{\eta}}$ aux indices $i + j > n$ (i.e. à la partie inférieure du triangle à prédire).

Remark 3.4.1. *Cette formule est malheureusement asymptotique, ce qui est rarement le cas en provisionnement où l'on dispose de très peu de données (et de beaucoup de facteurs).*

```
> p <- nl+nc-1;
> phi <- sum(residuals(CL,"pearson")^2)/(sum(is.na(Y)==FALSE)-p)
> Sig <- vcov(CL)
> X <- model.matrix(glm(Ypred~lig+col, family=quasipoisson))
> Cov.eta <- X%*%Sig%*%t(X)
> Ypred <-predict(CL,newdata=data.frame(lig,col),type="response")*(is.na(Y)==TRUE)
> se2 <- phi * sum(Ypred) + t(Ypred) %*% Cov.eta %*% Ypred
> sqrt(se2)
         [,1]
[1,] 131.7726
```

Les méthodes de simulations

Les méthodes de simulation sont une bonne alternative si on dispose de trop peu de données pour invoquer des théorèmes asymptotiques. Rappelons, comme le notait Mack (1993*a*) qu'il existe 2 sources d'incertitude,

– l'erreur de modèle (on parle de *process error*)

– l'erreur d'estimation (on parle de *variance error*)

Il sera alors nécessaire d'utiliser deux algorithmes pour quantifier ces deux erreurs.

Afin de quantifier l'erreur d'estimation, il est naturel de simuler des faux triangles (supérieurs), puis de regarder la distribution des estimateurs de montant de provisions obtenus pour chaque triangles (par exemple par la méthode Chain Ladder, à l'aide de la fonction `Chainladder` développée auparavant, ou en refaisant une régression de Poisson). A l'étape b, on génère un pseudo triangle à l'aide des résidus de Pearson. Rappelons que pour une régression de Poisson,

$$\widehat{\varepsilon}_{i,j} = \frac{Y_{i,j} - \widehat{Y}_{i,j}}{\sqrt{\widehat{Y}_{i,j}}}.$$

Toutefois, ces résidus ont besoin d'être ajustés afin d'avoir une variance unitaire. On considère alors classiquement

$$\widehat{\varepsilon}_{i,j} = \sqrt{\frac{n}{n-k}} \cdot \frac{Y_{i,j} - \widehat{Y}_{i,j}}{\sqrt{\widehat{Y}_{i,j}}},$$

où k est le nombre de paramètres estimés dans le modèle.

```
> (residus=residuals(CL,type="pearson"))
       1         2         3         4         5         6         7         8
 9.49e-01  2.40e-02  1.17e-01 -1.08e+00  1.30e-01 -1.01e-13 -1.13e+00  2.77e-01
       9        10        11        13        14        15        16        19
 5.67e-02  8.92e-01 -2.11e-01 -1.53e+00 -2.21e+00 -1.02e+00  4.24e+00 -4.90e-01
      20        21        25        26        31
 7.93e-01 -2.97e-01 -4.28e-01  4.14e-01 -6.20e-15
> n=sum(is.na(Y)==FALSE)
> k=ncol(PAID)+nrow(PAID)-1
> (residus=sqrt(n/(n-k))*residus)
       1         2         3         4         5         6         7         8
 1.37e+00  3.49e-02  1.69e-01 -1.57e+00  1.89e-01 -1.46e-13 -1.63e+00  4.02e-01
       9        10        11        13        14        15        16        19
 8.22e-02  1.29e+00 -3.06e-01 -2.22e+00 -3.21e+00 -1.48e+00  6.14e+00 -7.10e-01
      20        21        25        26        31
 1.15e+00 -4.31e-01 -6.20e-01  6.00e-01 -8.99e-15
> epsilon <- rep(NA,nl*nc)
> epsilon[is.na(Y)==FALSE]=residus
> matrix(epsilon,nl,nc)
          [,1]    [,2]  [,3]   [,4]  [,5]      [,6]
[1,]  1.37e+00 -1.6346 -2.22 -0.710 -0.62 -8.99e-15
[2,]  3.49e-02  0.4019 -3.21  1.149  0.60        NA
[3,]  1.69e-01  0.0822 -1.48 -0.431    NA        NA
[4,] -1.57e+00  1.2926  6.14     NA    NA        NA
[5,]  1.89e-01 -0.3059    NA     NA    NA        NA
[6,] -1.46e-13      NA    NA     NA    NA        NA
```

En simulant des erreurs (qui sont supposées indépendantes et identiquement distribuée), $\tilde{\boldsymbol{\varepsilon}}^b = (\tilde{\varepsilon}^b_{i,j})$, on pose alors

$$Y^b_{i,j} = \widehat{Y}_{i,j} + \sqrt{\widehat{Y}_{i,j}} \cdot \tilde{\varepsilon}^b_{i,j}.$$

Pour générer des erreurs, la méthode la plus usuelle est d'utiliser une simulation nonparamétrique, c'est à dire que l'on va bootstrapper les résidus parmi les pseudo-résidus obtenus. Sinon il est aussi possible d'utiliser un modèle paramétrique (par exemple supposer une loi normale, même si rien théoriquement ne justifie l'utilisation de cette loi). La distribution des résidus peut être obtenue par le code suivant, et visualisé sur la Figure 3.11

```
> par(mfrow = c(1, 2))
> hist(residus,breaks=seq(-3.5,6.5,by=.5),col="grey",proba=TRUE)
> u <- seq(-4,5,by=.01)
> densite <- density(residus)
> lines(densite,lwd=2)
> lines(u,dnorm(u,mean(residus),sd(residus)),lty=2)
> plot(ecdf(residus),xlab="residus",ylab="Fn(residus)")
> lines(u,pnorm(u,mean(residus),sd(residus)),lty=2)
> Femp <- cumsum(densite$y)/sum(densite$y)
> lines(densite$x,Femp,lwd=2)
> par(mfrow = c(1, 1))
```

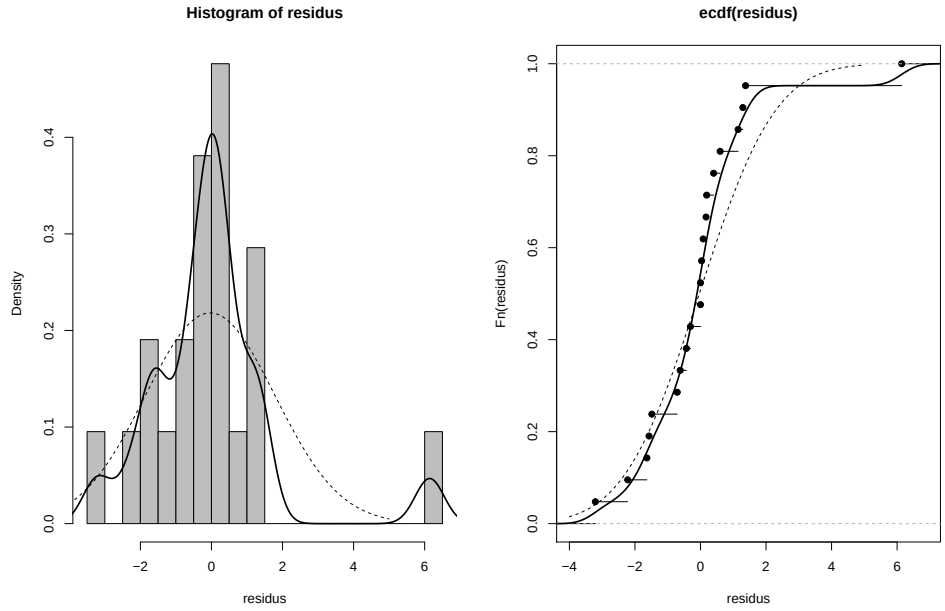


FIGURE 3.11 – Histogramme et densité des résidus (à gauche) et fonctions de répartition (à droite), avec l'ajustement Gaussien en pointillés..

Une fois simulé un pseudo-triangle d'incréments de paiments, on prédit un montant de provision $\widehat{R}^b$ (par exemple via une méthode Chain Ladder). La variance des $\widehat{R}^b$ correspond à l'erreur d'estimation.

Afin de prendre en compte l'erreur de modèle, plusieurs méthodes peuvent être utilisées. La première, et la plus simple, consiste à noter qu'à partir du pseudo triangle $Y^b_{i,j}$, peut obtenir des

prédictions pour la partie inférieure, $\widehat{Y}_{i,j}^b$. Compte tenu du modèle Poissonnien, on peut alors simuler une trajectoire possible d'incréments de paiements en simulant les $Y_{i,j}^b$ à l'aide de loi de Poisson de paramètre $\widehat{Y}_{i,j}^b$. Le code est alors le suivant

```
>  CLsimul1<-function(triangle){
+   rpoisson(length(triangle),lambda=triangle)}
```

Toutefois, simuler des lois de Poisson risque d'être trop conservateur. En effet, comme nous l'avions vu sur la régression quasiPoisson, le paramètre de surdispersion ϕ est significatif,

```
> (summary(CL)$dispersion)
[1] 3.19
```

Il peut alors être pertinent de générer des lois avec davantage de variance (abusivement, on parlera de simulation d'une loi quasiPoisson). La première idée pourra être d'utiliser une loi Gamma. En notant que $\mathbb{E}(Y) = \lambda$ et $\mathbb{V}(Y) = \phi\lambda$, la loi de Gamma $\mathcal{G}(\alpha, \beta)$ vérifiera $\alpha\beta = \lambda$ et $\alpha\beta^2 = \phi\lambda$. Aussi, on utilisera

```
> rqpoisG <- function(n, lambda, phi, roundvalue = TRUE) {
+ b <- phi
+ a <- lambda/phi
+ r <- rgamma(n, shape = a, scale = b)
+ if(roundvalue){r<-round(r)}
+ return(r)}
```

On utilise la fonction `round` afin de renvoyer un entier (ce qui est attendu pour un modèle de Poisson, mais n'a pas grande importance dans un contexte de modélisation de paiements). Une autre idée peut être d'utiliser le lien qui existe entre la loi de Poisson et la loi binomiale négative (qui est un mélange de lois de Poisson, i.e. dont l'hétérogénéité résiduelle n'a pas pu être modélisée par nos facteurs lignes et colonnes). Pour une loi binomiale négative de moyenne μ et de variance $\mu + \mu^2/k$, on pose $\mu = \lambda$ et $k = \lambda(\phi\lambda - 1)^{-1}$, i.e.

```
> rqpoisBN = function(n, lambda, phi) {
+ mu <- lambda
+ k <- mu/(phi * mu - 1)
+ r <- rnbinom(n, mu = mu, size = k)
+ return(r)}
```

On utilise alors le code suivant pour générer des scenarios de paiements,

```
>  CLsimul2<-function(triangle,surdispersion){
+   rqpoissonG(length(triangle),lambda=triangle,phi=surdispersion)}
```

La seconde méthode est d'utiliser une relecture du modèle de Mack (1993*a*), proposée par England & Verrall (1999). A partir du pseudo triangle, on va utiliser les facteurs de développement $\widehat{\lambda}_j$ et les variances associés $\widehat{\sigma}_j^2$ obtenus sur le triangle initial. On prolonge alors le triangle dans la partie inférience via le modèle dynamique

$$\widehat{C}_{i,j+1}^b | \widehat{C}_{i,j}^b \sim \mathcal{N}(\widehat{\lambda}_j \widehat{C}_{i,j}^b, \widehat{\sigma}_j^2 \widehat{C}_{i,j}^b).$$

Le code est alors le suivant, où `triangle` est un triangle de paiements *cumulés* (et non plus des incréments sous forme vectorielle), `l` correspond à un vecteur de facteurs de développement, et `s` à un vecteur de volatilités,

```
>  CLsimul3<-function(triangle,l,s){
+  m<-nrow(triangle)
+  for(i in 2:m){
+   triangle[(m-i+2):m,i]<-rnorm(i-1,
+             mean=triangle[(m-i+2):m,i-1]*l[i-1],
```

```
+            sd=sqrt(triangle[(m-i+2):m,i-1])*s[i-1])
+  }
+  return(triangle) }
```

L'algorithme global pour générer des estimations de charges finales, mais aussi des scenarios de paiements futurs est alors le suivant

```
> ns<-20000
> set.seed(1)
> Yp <- predict(CL,type="response",newdata=base)
> Rs <- R <- rep(NA,ns)
> for(s in 1:ns){
+ serreur <- sample(residus,size=nl*nc,replace=TRUE)
+ E <- matrix(serreur,nl,nc)
+ sY <- matrix(Yp,6,6)+E*sqrt(matrix(Yp,6,6))
+ if(min(sY[is.na(Y)==FALSE])>=0){
+ sbase <- data.frame(sY=as.vector(sY),lig,col)
+ sbase$sY[is.na(Y)==TRUE]=NA
+ sreg <- glm(sY~lig+col,
+ data=sbase,family=poisson(link="log"))
+ sYp <- predict(sreg,type="response",newdata=sbase)
+ R[s]  <- sum(sYp[is.na(Y)==TRUE])
+ sYpscenario <- rqpoisG(36,sYp,phi=3.18623)
+ Rs[s] <- sum(sYpscenario[is.na(Y)==TRUE])
+ }}
```

Lors de la génération de pseudo triangles, des incréments négatifs peuvent apparaître. En effet, pour deux valeurs de $\widehat{Y}_{i,j}$, il est possible que $\widehat{\varepsilon}\sqrt{\widehat{Y}_{i,j}}$ soit négatif (si le résidu est le plus petit résidu obtenu)

```
> sort(residus)[1:2]
   14    13
-3.21 -2.22
> sort(sqrt(Yp[is.na(Y)==FALSE]))[1:4]
  25   26   19   20
2.87 2.96 4.38 4.52
```

La solution retenue est de ne pas prendre en compte les triangles avec des incréments négatifs, ce qui devrait nous faire surestimer les quantiles inférieurs. Toutefois, en provisionnement, les quantiles inférieurs n'ont que peu d'intérêt. Les quantiles d'ordre élévés sont estimés ici par

```
> Rna <- R
> Rna[is.na(R)==TRUE]<-0
> Rsna <- Rs
> Rsna[is.na(Rs)==TRUE]<-0
> quantile(Rna,c(.75,.95,.99,.995))
  75%   95%   99% 99.5%
 2470  2602  2700  2729
> quantile(Rsna,c(.75,.95,.99,.995))
  75%   95%   99% 99.5%
 2496  2645  2759  2800
```

À partir de ces 20 000 triangles simulés, on peut obtenir la distribution des montants de provisions estimés (stockées dans le vecteur `R`) mais aussi des scenarios de paiements (et donc

de provisions nécessaires, dans le vecteur `Rs`). On va pour cela supprimer les 10% des scenarios ayant donné lieu à des valeurs manquantes.

```
> Rnarm <- R[is.na(R)==FALSE]
> Rsnarm <- Rs[is.na(Rs)==FALSE]
```

On notera que les quantiles supérieurs sont biaisés (avec un biais positif), mais faiblement

```
> quantile(Rnarm,c(.75,.95,.99,.995))
  75%   95%   99% 99.5%
 2478  2609  2704  2733
> quantile(Rsnarm,c(.75,.95,.99,.995))
  75%   95%   99% 99.5%
 2507  2653  2764  2805
```

La Figure 3.12 permet d'avoir une estimation de la distribution de $\widehat{R}$ mais aussi de R.

```
> plot(density(Rnarm),col="grey",main="")
> lines(density(Rsnarm),lwd=2)
> boxplot(cbind(Rnarm,Rsnarm),
+ col=c("grey","black"),horizontal=TRUE)
```

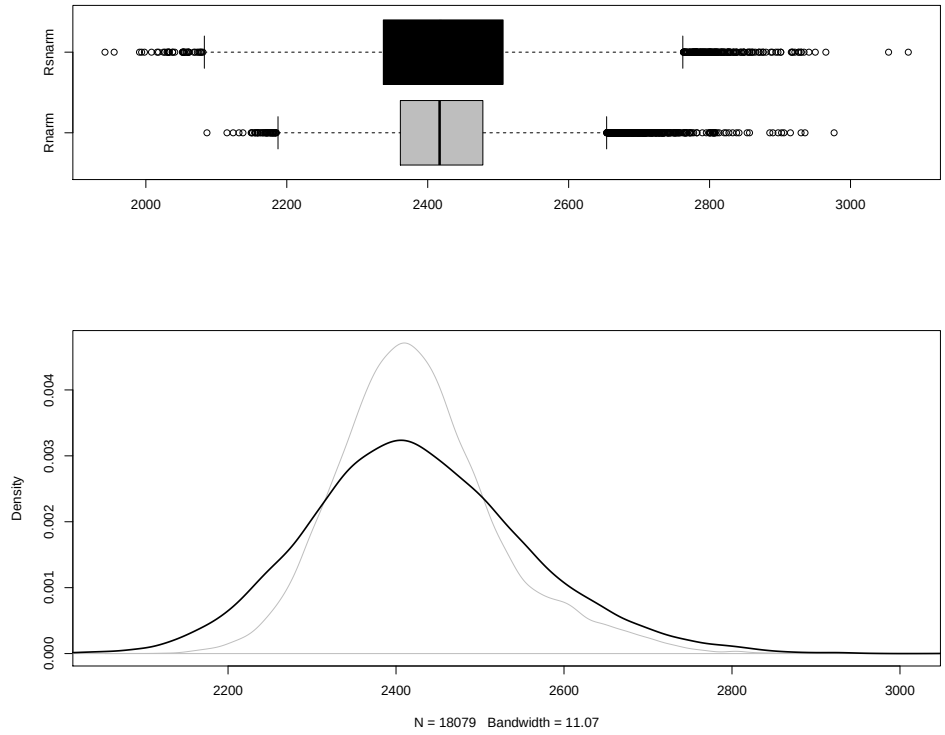


FIGURE 3.12 – Boxplot et densité estimée de $\widehat{R}$ (*estimation error*) et de R (*estimation error* et *process error*) .

Notons que les quantitées obtenues sont très proches de celles obtenues par la fonction `bootChainLadder` de `library(ChainLadder)`,

```
> BootChainLadder(PAID,20000,"od.pois")
BootChainLadder(Triangle = PAID, R = 20000, process.distr = "od.pois")
```

```
  Latest Mean Ultimate Mean IBNR SD IBNR IBNR 75% IBNR 95%
1  4,456         4,456       0.0     0.0        0        0
2  4,730         4,752      22.1    12.0       28       44
3  5,420         5,455      35.3    15.3       44       63
4  6,020         6,085      65.5    19.8       77      101
5  6,794         6,946     152.4    28.6      170      203
6  5,217         7,363   2,146.2   111.6    2,216    2,337

                Totals
Latest:         32,637
Mean Ultimate:  35,059
Mean IBNR:       2,422
SD IBNR:           132
Total IBNR 75%:  2,504
Total IBNR 95%:  2,651
```

3.4.6 Quel modèle de régression ?

Comme nous l'avons mentionné dans le chapitre 2, deux paramètres fondamentaux interviennent dans une régression linéaire généralisée,

- la *fonction lien*, qui lie la prédiction aux facteurs, ici $\widehat{Y}_{i,j} = \mathbb{E}(Y_{i,j}|\mathcal{F}_n) = \exp[\widehat{\gamma} + \widehat{\alpha}_i + \widehat{\beta}_j]$,
- la *loi* ou la *fonction variance*, qui donne la forme de l'intervalle de confiance, ici $\mathbb{V}(Y_{i,j}|\mathcal{F}_n) = \phi \cdot \mathbb{E}(Y_{i,j}|\mathcal{F}_n)$,

L'unique motivation du modèle de Poisson (ou quasi-Poisson) est qu'il permet d'obtenir exactement le même montant que la méthode Chain Ladder. Mais aucun critère statistique n'a été évoqué, pour l'instant, afin de légitimer ce modèle.

Les modèles Tweedie sont une famille de *sur*-modèle, incluant le modèle Poissonnien. On suppose que

- la *fonction lien*, est une fonction puissance, ou plutôt une tranformée de Box-Cox, $\widehat{Y}_{i,j} = g_\lambda^{-1}[\widehat{\gamma} + \widehat{\alpha}_i + \widehat{\beta}_j]$ où $g_\lambda(x) = \lambda^{-1}[x^\lambda - 1]$ si $\lambda > 0$ avec le cas limite $g_0(x) = \log(x)$.
- la *fonction variance*, qui donne la forme de l'intervalle de confiance, ici $\mathbb{V}(Y_{i,j}|\mathcal{F}_n) = \phi \cdot \mathbb{E}(Y_{i,j}|\mathcal{F}_n)^\mu$

où les paramètres λ et μ sont inconnus.

La densité[1] d'une loi Tweedie de paramètre `mu` est ici

```
> ftweedie <- function(y,p,mu,phi){
+ if(p==2){f <- dgamma(y, 1/phi, 1/(phi*mu))} else
+ if(p==1){f <- dpois(y/phi, mu/phi)} else
+ {lambda <- mu^(2-p)/phi /(2-p)
+ if(y==0){ f <- exp(-lambda)} else
+ { alpha <- (2-p)/(p-1)
+ beta <- 1 / (phi * (p-1) * mu^(p-1))
+ k <- max(10, ceiling(lambda + 7*sqrt(lambda)))
+ f  <- sum(dpois(1:k,lambda) * dgamma(y,alpha*(1:k),beta))
+ }}
+ return(f)}
```

1. où le terme *densité* s'entend au sens large, à savoir une probabilité dans le cas discret.

Afin de juger de la pertinance de l'ajustement, on peut calculer la log-vraisemblance du modèle, en gardant un lien logarithmique par exemple (ce qui est parfois plus simple au niveau numérique, mais aussi au niveau de l'interprétation),

```
> pltweedie <- function(puissance){
+ regt <-  glm(Y~lig+col, tweedie(puissance,0))
+ reserve <- sum(predict(regt,type="response",newdata=
+ data.frame(lig,col))[is.na(Y)==TRUE])
+ dev <- deviance(regt)
+ phi <- dev/n
+ mu <- predict(regt,type="response",newdata=data.frame(lig,col))
+ hat.logL <- 0
+ for (k in which(is.na(Y)==FALSE)){
+ hat.logL <- hat.logL + log(ftweedie(Y[k], puissance, mu[k], phi)) }
+ return(list(puissance= puissance,phi=phi,reserve=reserve,logL=hat.logL))
+ }
```

Si on calcule la log-vraisemblance pour 5 valeurs, comprises entre 1 et 2 (correspondant respectivement au cas d'une régression Poisson et une régression Gamma), on obtient

```
> pltweedie(puissance=1.25)
$puissance
[1] 1.25

$phi
[1] 0.466

$reserve
[1] 2427

$logL
[1] -96

> pltweedie(puissance=1.5)
$puissance
[1] 1.5

$phi
[1] 0.155

$reserve
[1] 2428

$logL
[1] -99.2
```

La Figure 3.13 permet de visualiser l'influence du paramètre de la puissance de la fonction variance. Visiblement la vraisemblance est maximal pour une puissance proche de 1 (ce qui correspond au modèle de Poisson) avec un lien logarithmique,

```
> puiss <- seq(1.02,1.98,by=.01)
> plot(puiss,Vectorize(TW)(puiss),type="l",
```

```
+ xlab="Puissance de la loi Tweedie",ylab="log vraisemblance")
```

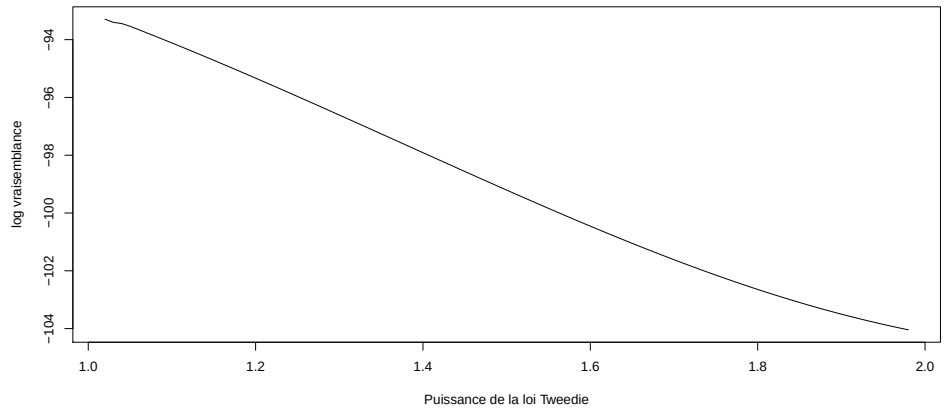


FIGURE 3.13 – Évolution de la log-vraisemblance profilée en fonction de μ .

```
> TW <-  function(p){pltweedie(p)$logL}
> optimize(TW, c(1.01,1.99), tol=1e-4,maximum = TRUE)
$maximum
[1] 1.01

$objective
[1] -92.2
```

3.5 Les triangles multivariés

Comme nous l'avions expliqué dans l'introduction, l'utilisation des triangles, et des méthodes de cadences de paiements, n'est possible que si les triangles sont stables, et homogènes. Or il n'est pas rare qu'un triangle comporte des risques relativement différents dans leur développement. Par exemple en assurance auto, les accidents matériels et corporels sont sensiblement différents.

3.5.1 Hypohtèse d'indépendance entre les triangles, et lois paramétriques

En s'insiprant de l'idée de Mack (1993*a*), on peut supposer que $\widehat{R}_i$ suive une loi $LN(\mu_i, \sigma_i^2)$ pour $i = 1, 2$. Si l'on suppose les risques indépendant, la loi de la somme est simplement la convolée des deux lois. On peut utiliser les familles de distribution au format `S4` et la `library(distr)`. Rappelons que pour si $X \sim LN(\mu, \sigma^2)$,

$$\mu = \log[\mathbb{E}(X)] - \frac{1}{2}\log\left(1 + \frac{\mathbb{V}(X)}{\mathbb{E}(X)^2}\right) \text{ et } \sigma^2 = \log\left(1 + \frac{\mathbb{V}(X)}{\mathbb{E}(X)^2}\right).$$

A partir des moyennes et variances - données par la méthode de Mack (1993*a*) par exemple - on en déduit les lois des deux montants de provision. Si on suppose que les deux triangles sont *indépendants*, alors

```
> library(distr)
> n=nrow(P.mat)
> V=MackChainLadder(P.mat)$Total.Mack.S.E^2
> E=sum(MackChainLadder(P.mat)$FullTriangle[,n]-
+ diag(MackChainLadder(P.mat)$FullTriangle[n:1,]))
> mu = log(E) - .5*log(1+V/E^2)
> sigma2 = log(1+V/E^2)
> LM = Lnorm(meanlog=mu,sdlog=sqrt(sigma2))
> V=MackChainLadder(P.corp)$Total.Mack.S.E^2
> E=sum(MackChainLadder(P.corp)$FullTriangle[,n]-
+ diag(MackChainLadder(P.corp)$FullTriangle[n:1,]))
> mu = log(E) - .5*log(1+V/E^2)
> sigma2 = log(1+V/E^2)
> LC = Lnorm(meanlog=mu,sdlog=sqrt(sigma2))
> LT=LM+LC
```

On peut alors comparer la loi convolée, et la loi lognormale ajustée sur le triangle cumulé,

```
> P.tot = P.mat + P.corp
> library(ChainLadder)
> V=MackChainLadder(P.tot)$Total.Mack.S.E^2
> E=sum(MackChainLadder(P.tot)$FullTriangle[,n]-
+ diag(MackChainLadder(P.tot)$FullTriangle[n:1,]))
> mu = log(E) - .5*log(1+V/E^2)
> sigma2 = log(1+V/E^2)
```

La Figure 3.14 compare la distribution obtenue en convolant deux lois lognormales (supposant une indépendance entre les triangles, et que le montant de provision peut être modélisé par une loi lognormale) et la distribution du montant de provision obtenu en agrégeant les deux triangles de paiements.

```
> u=seq(E-4*sqrt(V),E+4*sqrt(V),length=101)
> vtotal=dlnorm(u,mu,sqrt(sigma2))
> vconvol=d(LT)(u)
> plot(u,vconvol,col="grey",type="l",
+ xlab="montant de provision",ylab="")
> lines(u,vtotal,lwd=2)
> legend(470000,1.2e-05,
+ c("convolution","somme des triangles"),
+ col=c("grey","black"),lwd=c(1,2),bty="n")
```

Les quantiles à 95% sont alors respectivement

```
> q(LT)(.95)
[1] 434616
> qlnorm(.95,mu,sqrt(sigma2))
[1] 467687
```

pour la loi convolée et pour la somme des deux triangles. Deux interprétations sont alors possibles : supposer les triangles comme étant indépendants est probablement une hypothèse trop forte et travailler sur un triangle agrégé (et donc peu homogène) introduit une incertitude supplémentaire.

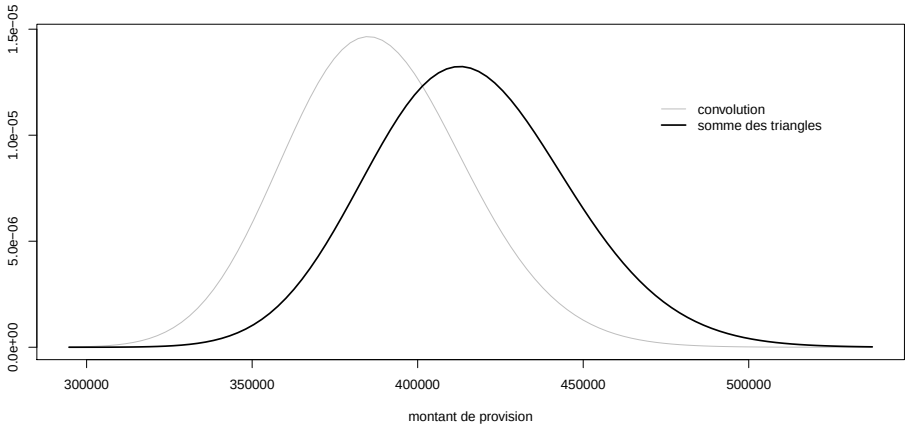


FIGURE 3.14 – Distribution du montant total de provision, en sommant les provisions par triangles - supposés indépendants - et en travaillant sur le triangle agrégé.

3.5.2 Le modèle de Mack bivarié

Pröhl & Schmidt (2005) a proposé une méthode de type Chain-Ladder dans un cadre multivarié. On note

$$\boldsymbol{\lambda}_{i,j} = (\lambda_{i,j}^{(k)}) \text{ où } \lambda_{i,j}^{(k)} = \frac{C_{i,j}^{(k)}}{C_{i,j-1}^{(k)}}$$

et $\boldsymbol{C}_{i,j} = (C_{i,j}^{(k)}) \in \mathbb{R}^K$ On suppose qu'il existe $\boldsymbol{\lambda}_j = \in \mathbb{R}^K$

$$\mathbb{E}[\boldsymbol{C}_{i,j}|\boldsymbol{C}_{i,j-1}] = \text{diag}(\boldsymbol{\lambda}_{j-1}) \cdot \boldsymbol{C}_{i,j-1}$$

et

$$\text{Cov}[\boldsymbol{C}_{i,j}, \boldsymbol{C}_{i,j}|\boldsymbol{C}_{i,j-1}] = \text{diag}(\sqrt{\boldsymbol{C}_{j-1}}) \cdot \boldsymbol{\Sigma}_{j-1} \cdot \text{diag}(\sqrt{\boldsymbol{C}_{j-1}})$$

Alors sous ces hypothèses, comme dans le cas univarié, on peut écrire

$$\mathbb{E}[\boldsymbol{C}_{i,n}|\boldsymbol{C}_{i,n-i}] = \prod_{j=n-i}^{n-1} \text{diag}(\boldsymbol{\lambda}_j)\boldsymbol{C}_{i,n-i}.$$

L'estimateur du facteur de transition est

$$\widehat{\boldsymbol{\lambda}}_j = \left[\sum_{i=0}^{n-j-1} \text{diag}(\sqrt{\boldsymbol{C}_{i,j}}) \cdot \boldsymbol{\Sigma}_j^{-1} \cdot \text{diag}(\sqrt{\boldsymbol{C}_{i,j}})\right]^{-1} \cdot \sum_{i=0}^{n-j-1} \text{diag}(\sqrt{\boldsymbol{C}_{i,j}}) \cdot \boldsymbol{\Sigma}_j^{-1} \cdot \text{diag}(\sqrt{\boldsymbol{C}_{i,j}})\boldsymbol{\lambda}_{i,j+1}$$

L'estimateur Chain-Ladder de la charge ultime est

$$\widehat{\boldsymbol{C}_{i,n}} = \prod_{j=n-i}^{n-1} \text{diag}(\widehat{\boldsymbol{\lambda}}_j)\boldsymbol{C}_{i,n-i}.$$

Cet estimateur vérifie les mêmes propriétés que dans le cas univarié. En particulier, cet estimateur est un estimateur sans biais de $\mathbb{E}[\boldsymbol{C}_{i,n}|\boldsymbol{C}_{i,n-i}]$ mais aussi de $\mathbb{E}[\boldsymbol{C}_{i,n}]$.

Il est aussi possible de calculer les mse de prédiction.

3.5.3 Modèles économétriques pour des risques multiples

L'idée dans les modèles économétriques est de supposer que les *résidus* peuvent être corrélés,

```
> ligne = rep(1:n, each=n); colonne = rep(1:n, n)
> PAID=P.corp; INC=PAID
> INC[,2:n]=PAID[,2:n]-PAID[,1:(n-1)]
> I.corp = INC
> PAID=P.mat; INC=PAID
> INC[,2:n]=PAID[,2:n]-PAID[,1:(n-1)]
> I.mat = INC
> Ym = as.vector(I.mat)
> Yc = as.vector(I.corp)
> lig = as.factor(ligne)
> col = as.factor(colonne)
> base = data.frame(Ym,Yc,col,lig)
> regm=glm(Ym~col+lig,data=base,family="poisson")
> regc=glm(Yc~col+lig,data=base,family="poisson")
> res.corp=residuals(regc,type="pearson")
> res.mat=residuals(regm,type="pearson")
> cor(res.mat,res.corp)
[1] 0.296
```

On notera ainsi que la corrélation n'est pas nulle.

Une fois notée qu'il existe probablement une dépendance entre les deux triangles, il semble légitime de la prendre en compte dans les algorithmes de simulations évoqués dans la partie 3.4.5.

- pour l'erreur d'estimation, quand on tire les résidus, on ne les tire pas indépendement dans les deux triangles. On tire alors les *paires* de résidus $(\widehat{\varepsilon}_{i,j}^{\text{matériel},b}, \widehat{\varepsilon}_{i,j}^{\text{corporel}}, b)$
- pour l'erreur, on peut tirer une loi de Poisson bivariée si on utilise une régression Poissonnienne bivariée (implémentée dans `library(bivpois)` ou un vecteur Gaussien bivarié.

Dans le second cas,

$$\begin{pmatrix} C_{i,j+1}^{\text{matériel}} \\ C_{i,j+1}^{\text{corporel}} \end{pmatrix} \sim \mathcal{N}\left(\begin{pmatrix} \lambda_j^m C_{i,j}^{\text{matériel}} \\ \lambda_j^c C_{i,j}^{\text{corporel}} \end{pmatrix}, \begin{pmatrix} \sigma_j^{m2} C_{i,j}^{\text{matériel}} & \star \\ \star & \sigma_j^{c2} C_{i,j}^{\text{corporel}} \end{pmatrix} \right).$$

3.6 Borhutter-Fergusson, Benktander et les méthodes bayésiennes

Les deux premières méthodes que nous allons voir ont souvent été proposées comme une alternative à la méthode Chain Ladder, car elles introduisent un *a priori* sur la charge ultime.

3.6.1 Le modèle de Borhutter-Ferguson et l'introduction d'un avis d'expert

Classiquement, on continue ici à supposer que

- les années de survenance sont indépendantes les unes des autres
- il existe μ_i et des facteurs de développement $\beta_1, \beta_2, \cdots, \beta_n$ - avec $\beta_n = 1$ - tels que

$$\mathbb{E}(C_{i,1}) = \beta_1 \mu_i$$

$$\mathbb{E}(C_{i,j+k} | C_{i,1}, \cdots, C_{i,j}) = C_{i,j} + [\beta_{j+k} - \beta_j]\mu_i$$

Sous ces hypothèses, pour tout i, j, $\mathbb{E}(C_{i,j}) = \beta_j \mu_i$. Ce qui peut rappeler les modèles à facteurs évoqués auparavant. Sauf qu'ici, seul $\boldsymbol{\beta} = (\beta_1, \beta_2, \cdots, \beta_n)$ sera à estimer statistiquement, $\boldsymbol{\mu} = \widehat{\mu}_i$ étant obtenu par *avis d'expert*, $\widehat{\mu}_i$ étant un estimateur de $\mathbb{E}(C_{i,n})$. Moyennant ces deux estimations, on en déduit l'estimateur de $\mathbb{E}(C_{i,n}|C_{i,1}, \cdots, C_{i,j})$ de la forme

$$\widehat{C}_{i,n} = C_{i,j} + [1 - \widehat{\beta}_{j-i}]\widehat{\mu}_i.$$

L'estimateur proposé par Bornhutter-Ferguson est alors simplement obtenu à partir de la méthode Chain-Ladder, en posant

$$\widehat{\beta}_j = \prod_{k=j+1}^{n} \frac{1}{\widehat{\lambda}_k}$$

Enfin, pour estimer $\widehat{\mu}_i$, on suppose disposer d'un ratio sinistre/prime cible, par exemple de 105%, par année de survenance. Dans ces conditions, on peut alors estimer simplement le montant de provision,

```
> mu <- 1.05*PREMIUM
> beta <- rev(cumprod(rev(1/LAMBDA)))
> Cdiag <- diag(PAID[,nc:1])
> Cultime <- Cdiag+(1-c(1,rev(beta)))*mu
> Cultime-Cdiag
[1]    0.0   23.1   33.5   59.0  131.3 1970.5
> sum(Cultime-Cdiag)
[1] 2217
```

i	0	1	2	3	4	5
prime	4591	4692	4863	5175	5673	6431
$\widehat{\mu}_i$	4821	4927	5106	5434	5957	6753
λ_i	1,380	1,011	1,004	1,002	1,005	
β_i	0,708	0,978	0,989	0,993	0,995	
$\widehat{C}_{i,n}$	4456	4753	5453	6079	6925	7187
$\widehat{R}_i$	0	23	33	59	131	1970

TABLE 3.8 – Estimation du montant de provision par Borhutter-Ferguson, avec un ratio sinistres/primes de 105%.

3.6.2 Benktander

L'estimateur de Benktander (1976), repris quelques années plus tard par Hovinen (1981), repose sur un estimateur *a priori* de la charge ultime $C_{i,n}$, noté μ_i. On suppose également qu'il existe une cadence de paiements $\boldsymbol{\beta} = (\beta_1, \cdots, \beta_n)$, connue, telle que

$$\mathbb{E}(C_{i,j}) = \mu_i \beta_j$$

Sous ces hypothèses, le montant de provision devrait être

$$\widehat{R}_i = \widehat{C}_{i,n} - C_{i,n-i} = [1 - \beta_{n-i}]\mu_i$$

Au lieu de se baser uniquement sur μ_i, Benktander (1976) avait proposé un estimateur crédibilisé de la charge ultime, de la forme

$$\beta_{n-i}\widehat{C}_{i,n}^{\text{CL}} + [1-\beta_{n-i}]\mu_i$$

Il s'agit d'utiliser l'estimateur Chain-Ladder, moyenné avec l'estimation a priori de la charge ultime. Alors

$$\widehat{R}_i^{\text{BH}} = \widehat{C}_{i,n} - C_{i,n-i} = [1-\beta_{n-i}]\left(\beta_{n-i}\widehat{C}_{i,n}^{\text{CL}} + [1-\beta_{n-i}]\mu_i\right)$$

On notera que

$$\widehat{R}_i^{\text{BH}} = (1-\beta_{n-i})\widehat{C}_i^{\text{BF}}$$

si la cadence $\boldsymbol{\beta} = (\beta_1, \cdots, \beta_n)$ est construite à partir des facteurs de développement induits par la méthode Chain-Ladder. Une autre écriture de cette expression est d'écrire la charge ultime (et non plus le montant de provision),

$$\widehat{C}_i^{\text{BH}} = C_{i,n-i} + (1-\beta_{n-i})\widehat{C}_i^{\text{BF}} = \beta_{n-i}\widehat{C}_i^{\text{CL}} + (1-\beta_{n-i})\widehat{C}_i^{\text{BF}}$$

ce qui permet de voir la prédiction de Benktander comme une combinaison convexe des estimateurs Chain-Ladder et de Bornhuetter-Ferguson.

3.6.3 La méthode dite *Cape-Code*

Dans cette approche, on utilise là encore un avis d'expert. L'idée est de réécrire l'expression

$$C_{i,n} = C_{i,n-i} + \left(1 - \frac{C_{i,n-i}}{C_{i,n}}\right) C_{i,n}$$

sous la forme

$$C_{i,n} = C_{i,n-i} + \left(1 - \frac{C_{i,n-i}}{C_{i,n}}\right) LR_i \cdot P_i,$$

où LR_i correspond au *loss ratio* pour l'année i, i.e. $LR_i = C_{i,n}/P_i$. L'idée de la méthode dite *Cape-Code* est d'écrire une forme plus générale,

$$C_{i,n} = C_{i,n-i} + (1-\pi_{n-i})\, LR_i P_i$$

où π_{n-i} correspond à une cadence de paiement, et peut être estimé par la méthode Chain Ladder. Quant aux LR_i il s'agit des *loss ratio* cibles, correspondant à un avis d'expert. On peut aussi proposer un même ratio cible pour plusieurs années de survenance. On posera alors

$$R_i = C_{i,n} - C_{i,n-i} = (1-\pi_{n-i}) LR_{\mathcal{A}} P_i.$$

pour $i \in \mathcal{A}$, où

$$LR_{\mathcal{A}} = \frac{\sum_{k\in\mathcal{A}} C_{n,n-k}}{\sum_{k\in\mathcal{A}} \pi_{n-k} P_k}.$$

Dans un premier temps, on peut calculer les π_i à partir de la méthode Chain Ladder, i.e.

$$\pi_{n-i} = \frac{C_{i,n-i}}{C_{i,n}}$$

où la charge ultime est celle prédite pas la méthode Chain-Ladder.

```
> Cultime=MackChainLadder(PAID)$FullTriangle[,nc]
> (PI <- (1-Cdiag/Cultime))
      1       2       3       4       5       6
0.00000 0.00471 0.00656 0.01086 0.02204 0.29181
> LR <- TRIANGLE[,nc]/PREMIUM
> Cdiag <- diag(PAID[,nc:1])
> (Cultime-Cdiag)/(LR*PREMIUM)
      1       2       3       4       5       6
0.00000 0.00471 0.00656 0.01086 0.02204 0.29181
```

Si on suppose ensuite que $\mathcal{A} = \{1, 2, \cdots, n\}$, alors

```
> LR=sum(TRIANGLE[,6])/sum(PREMIUM)
> PI*LR*PREMIUM
     1      2      3      4      5      6
   0.0   24.6   35.6   62.7  139.6 2095.3
> sum(PI*LR*PREMIUM)
[1] 2358
```

On obtient ici un montant de provision total inférieur à celui obtenu par la méthode Chain Ladder puisque le montant de provisions vaut ici `2357.756`.

3.6.4 Les approches Bayésiennes

Les approches Bayésiennes ont été popularisées en sciences actuarielles par la théorie de la crédibilité, correspondant à une approche Bayésienne dans un cadre linéaire. Mais il est possible d'aller plus loin (plus générallement, sur l'alternative bayésienne en statistique, nous renverrons à Parent & Bernier (2007) ou Robert (2006)). Classiquement, supposons que l'on s'intéresse à $\boldsymbol{Y}$ dont la loi serait $f(\cdot|\boldsymbol{\theta})$, où très généralement, $\boldsymbol{Y} = (Y_{i,j})$ et $\boldsymbol{\theta} = (\theta_{i,j})$. $\boldsymbol{Y}$ peut être ici le triangle des paiements cumulés $\boldsymbol{C}$, le triangle des incréments $\boldsymbol{Y}$, ou le triangle des coefficients de transition des cadences de paiements $\boldsymbol{\lambda} = \boldsymbol{C}_{i,j+1}/\boldsymbol{C}_{i,j}$.

Example 3.6.1.

Dans l'approche de Mack (1993*a*), on cherche à modéliser $\boldsymbol{Y}$ peut être ici le triangle des paiements cumulés $\boldsymbol{C}$, et $\boldsymbol{\theta}_j = (\lambda_j, \sigma_j^2)$.

Application aux cadences de paiements

Ici, on s'intéresse à la loi de $\boldsymbol{\lambda}$, qui dépendra de $\boldsymbol{\theta} = (\boldsymbol{\theta}_j)$ où $\boldsymbol{\theta}_j = (\gamma_j, \sigma_j^2)$, où, pour des simplicités de notations (et éviter de confondre avec les $\lambda_{i,j}$) on note γ_j le facteur de développement sous-jacent.

$$\lambda_{i,j}|(\gamma_j, \sigma_j^2) \sim \mathcal{N}\left(\gamma_j, \frac{\sigma_j^2}{C_{i,j}}\right)$$

Ici, σ^2 ne sont pas les paramètres d'intérêt, et sont supposés estimés séparément (comme nous le faisions déjà dans les modèles linéaires généralisés). Quant aux $C_{i,j}$, ils sont interprétés ici comme des poids, et sont supposés connus. La log-vraisemblance est ici

$$\log \mathcal{L}(\boldsymbol{\lambda}|\boldsymbol{\gamma}) = \sum_{i,j} \frac{1}{2}\left(\log\left[\frac{C_{i,j-1}}{\sigma_j^2}\right] - \frac{C_{i,j-1}}{\sigma_j^2}[\lambda_{i,j} - \gamma_j]^2\right).$$

En utilisant la formule de Bayes, la log-densité de $\boldsymbol{\gamma}$ conditionnelle aux $\boldsymbol{\lambda}$ est simplement

$$\log[g(\boldsymbol{\gamma}|\boldsymbol{\lambda})] = \log[\pi(\boldsymbol{\gamma})] + \log[\mathcal{L}(\boldsymbol{\lambda}|\boldsymbol{\gamma})] + \text{constante},$$

où $\pi(\cdot)$ est une loi *a priori* de $\boldsymbol{\gamma}$ (par exemple un vecteur Gaussien).

L'algorithme de Gibbs et généralisations

On cherche ici à générer un ensemble de vecteurs aléatoires $\boldsymbol{\gamma} = (\gamma_1, \cdots, \gamma_m) \in \mathbb{R}^m$. Contrairement aux méthodes de Monte Carlo où l'on cherche à générer des vecteurs indépendants les uns des autres, on va essayer de construire une suite de manière récurente, vérifiant des propriétés d'ergodicité.

On part d'un vecteur initial $\boldsymbol{\gamma}^{(0)} = (\gamma_1^{(0)}, \cdots, \gamma_m^{(0)})$, par exemple les valeurs obtenues par la méthode Chain Ladder puis on génère, de manière itérée

$$\left\{\begin{array}{l}
\gamma_1^{(k+1)} \sim f(\cdot|\gamma_2^{(k)}, \cdots, \gamma_m^{(k)}, \lambda) \\
\gamma_2^{(k+1)} \sim f(\cdot|\gamma_1^{(k+1)}, \gamma_3^{(k)}, \cdots, \gamma_m^{(k)}, \lambda) \\
\gamma_3^{(k+1)} \sim f(\cdot|\gamma_1^{(k+1)}, \gamma_2^{(k+1)}, \gamma_4^{(k)}, \cdots, \gamma_m^{(k)}, \lambda) \\
\vdots \\
\gamma_{m-1}^{(k+1)} \sim f(\cdot|\gamma_1^{(k+1)}, \gamma_2^{(k+1)}, \gamma_{m-2}^{(k+1)}, \gamma_m^{(k)}, \lambda) \\
\gamma_m^{(k+1)} \sim f(\cdot|\gamma_1^{(k+1)}, \gamma_2^{(k+1)}, \cdots, \gamma_{m-1}^{(k+1)}, \lambda)
\end{array}\right.$$

Ces lois conditionnelles n'ayant pas forcément de forme simple, l'algorithme de *metropolis* (d'acceptation-rejet) peut alors être utiliser pour simuler ces différentes lois conditionnelle.

3.6.5 Approche bayésienne sur les facteurs de développement

En s'inspirant de la relecture du modèle de Mack (1993*a*),

$$\widehat{C}_{i,j+1}^b | \widehat{C}_{i,j}^b \sim \mathcal{N}(\widehat{\lambda}_j \widehat{C}_{i,j}^b, \widehat{\sigma}_j^2 \widehat{C}_{i,j}^b).$$

nous pouvons supposer que les facteurs de développements $\lambda_{i,j}$ suivent une loi lognormale, comme le suggérait Balson (2008). La fonction `bayes-triangle()` donne ici

```
> set.seed(1)
> RESERVES <- bayes-triangle(PAID)$reserves
> res.tot <- RESERVES[,7]
```

On peut visualiser sur la Figure 3.15 montre les 1,000 valeurs générées pour $\widehat{R}$

```
> plot(res.tot,ylab="Montant de provision")
> abline(h=mean(res.tot))
> abline(h=quantile(res.tot,c(.05,.95)),col="grey")
```

La Figure 3.16 montre ainsi la distribution du montant de provision estimé $\widehat{R}$ obtenu par cet algorithme (avec en trait grisé la distribution obtenue par bootstrap des résidus dans le modèle quasiPoisson)

```
> plot(density(res.tot),lwd=2,main="")
> lines(density(Rnarm),col="grey")
> boxplot(cbind(res.tot,Rnarm),
+ col=c("black","grey"),horizontal=TRUE)
```

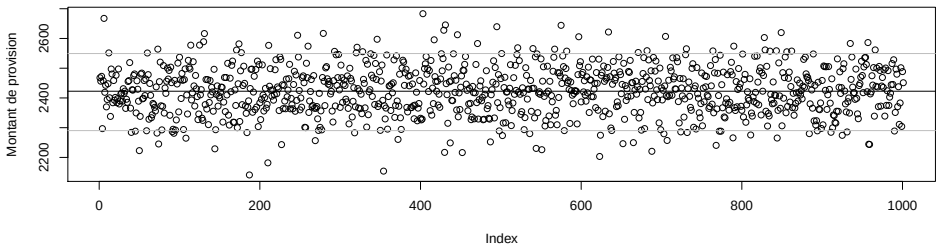


FIGURE 3.15 – Génération d'une suite de montants de provisions $\widehat{R}$.

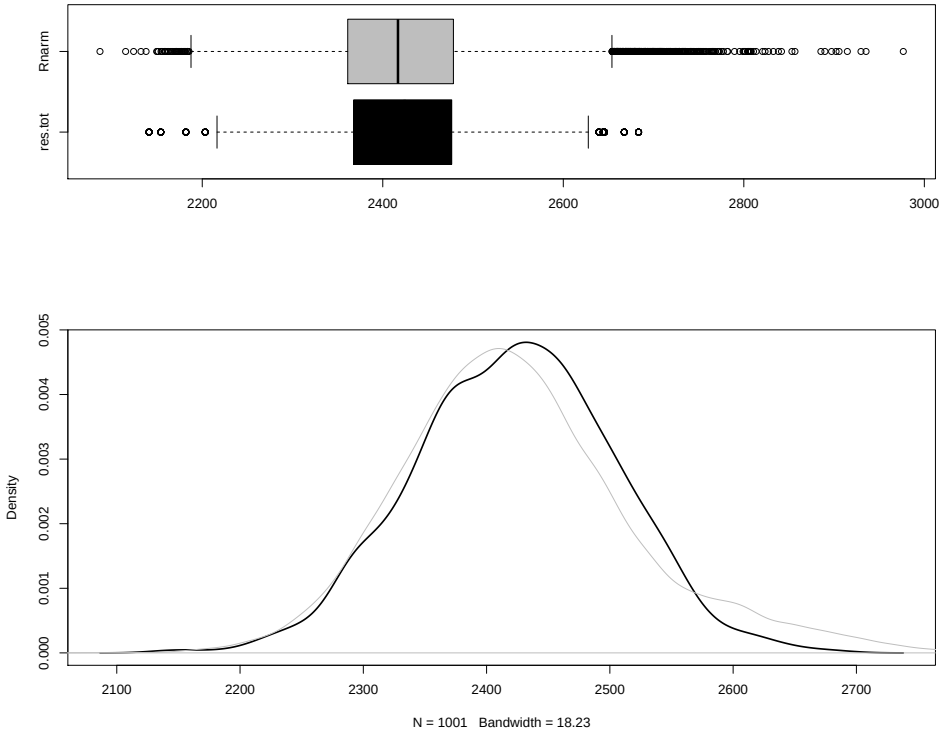


FIGURE 3.16 – Boxplot et densité de la distribution du montant de provisions $\widehat{R}$, en noir, avec en grisé les la distribution obtenue par bootstrap des résidus dans le modèle quasi-Poisson.

Pour conclure, notons que la méthode bayésienne (qui est fondamentalement basée sur un modèle autorégressif) donne une dispersion du montant de provision proche du modèle de Mack.

```
> MackChainLadder(PAID)$Total.Mack.S.E
[1] 79.3
> sd(rest.tot)
[1] 80.7
> sd(Rnarm)
[1] 97.4
```

En revanche, la méthode de bootstrap génère des paiements possibles futurs beaucoup plus grand (compte tenu du résidu très important), et donc la volatilité est plus grande.

3.7 Exercices

Exercise 3.7.1. *Programmer les alogrithmes permettant de modéliser deux triangles supposés non-indépendants.*

Exercise 3.7.2. *Déterminer le quantile à 95% du montant de provision pour le triangle de paiements cumulés* `triangle1`*.*

Exercise 3.7.3. *Déterminer le quantile à 95% du montant de provision pour le triangle de paiements cumulés* `triangle2`*.*

Exercise 3.7.4. *Déterminer le quantile à 95% du montant de provision pour le triangle de paiements cumulés* `triangle3`*.*

Exercise 3.7.5. *Déterminer le quantile à 95% du montant de provision pour le triangle de paiements cumulés* `triangle4`*.*

Chapitre 4

Calculs de base en assurance vie et décès

L'assurance-vie repose essentiellement sur des calculs de valeurs actuelles probables, c'est à dire des calculs d'expressions de la forme $\boldsymbol{c}'\boldsymbol{p} = \sum_j c_j p_j$, où $\boldsymbol{c}$ est un vecteur de flux futurs actualisés, de la forme $(1+i)^{-j}c_j$ où i est le taux d'actualisation (en retenant les notations usuelles) et c_j un flux de paiements qui peut survenir à la date j, et p_j est la probabilité que le $j$ème paiement soit effectué (généralement une probabilité qu'une personne soit en vie pour le calcul des rentes, ou la probabilité qu'une personne décède à cette date pour l'assurance décès).

R est un langage idéal pour les calculs de ces valeurs actuelles probables compte tenu de la forme vectorielle de la plupart des expressions. Nous allons voir dans ce chapitre les bases des calculs actuariels, en présentant quelques calculs d'annuités classiques, ainsi que les valorisations de *provisions mathématiques*. Enfin, nous présenterons un algorithme utilisant des formes récursives de plusieurs grandeurs utilisées en assurance-vie.

Si nous allons définir toutes les grandeurs qui seront calculées, nous renvoyons à Petauton (2004), Denuit & Robert (2007), Hess (2000), Dickson et al. (2009) ou Vylder (2010) pour une présentation plus poussée des notions et des différents concepts.

4.1 Quelques notations

Si l'assurance non-vie repose essentiellement sur des modélisation stochastique des sinistres à venir, l'assurance-vie consiste fondamentalement à actualiser des flux futurs, incluant généralement un part d'incertitude (associée au décès ou à la survie d'un assuré). De la même manière que nous nous étions attachés à calculer des primes à l'aide d'espérance de flux en assurance non-vie (conditionnelles à des variables tarifaires dans le chapitre 2 par exemple), nous allons ici calculer des grandeurs de la forme :

$$\mathbb{E}\left(\sum_{k=1}^{\infty}\frac{C_k}{(1+i)^{T_k}}\cdot\mathbf{1}(\text{paiement à la date } T_k)\right),$$

où l'assureur s'est engagé à verser un capital C_i à des dates T_k (connues), à condition qu'une hypothèse soit vérifiée à la date T_k. Compte-tenu de la linéarité de l'espérance, si l'on suppose le taux d'actualisation non aléatoire, on peut réécrire cette dernière expression sous la forme :

$$\sum_{k=1}^{\infty}\frac{C_k}{(1+i)^{T_k}}\mathbb{P}(\text{paiement à la date } T_k) = \sum_{k=1}^{\infty}C_k\cdot\nu^{T_k}\cdot\mathbb{P}(\text{paiement à la date } T_k),$$

où le facteur d'actualisation $\nu = (1+i)^{-1}$ permettra d'éviter - autant que possible - la notation i, réservée aux taux d'actualisation en assurance-vie, mais désignant un indice de ligne dans les algorithmes.

4.1.1 Les probabilités de décès, ou de survie

Comme le montre la formule précédante, un des points essentiels lors de la valorisation est de disposer de ces probabilités $\boldsymbol{p}$, liées souvent à la survie - ou au décès - d'un assuré (en particulier les taux d'actualisation sont supposés ici connus, et constants).

Considérons un individu d'âge x à la souscription d'un contrat d'assurance (correspondant à la variable `x` sous R), et notons classiquement T_x sa durée de vie résiduelle (qui est aléatoire). On pose ${}_kq_x = \mathbb{P}(T_x \leq k)$ la probabilité de ne plus être en vie à l'âge $x+k$ (c'est à dire k années après la souscription), et ${}_kp_x = \mathbb{P}(T_x > k)$ la probabilité d'être encore en vie à l'âge $x+k$. A x donné, $k \mapsto {}_kp_x = \mathbb{P}(T_x > k)$ est alors la fonction de survie de la variable T_x`qx`.. On peut alors considérer des *vecteurs* `px` et `qx`. Parmi les autres notations, la probabilité de décéder pendant une période particulière, disons entre les âges $x+k$ et $x+k+h$, sera notée

$$ {}_{k|h}q_x = \mathbb{P}(k < T_x \leq k+h) = {}_kp_x - {}_{k+h}p_x.$$

Par abus de notation, on notera parfois p_x la quantité ${}_1p_x$ et q_x la quantité ${}_1q_x$. Et on notera ${}_kd_x = {}_{k|1}q_x$ la probabilité qu'une personne d'âge x décède à l'âge $x+k$ (ce qui n'a toutefois rien d'officiel, mais permettra des simplifications sous R par la suite).

Ces grandeurs sont obtenues numériquement à l'aide des tables de mortalité, c'est à dire un vecteur `L` de L_x pour tous les âges x, correspondant au nombre de survivants ayant atteint l'âge x au sein d'une cohorte de taille L_0 intialement (à la naissance, avec souvent $L_0 = 100000$, par convention). La première valeur du vecteur `L`, i.e. `L[1]` correspondra alors à L_0. Il conviendra d'être particulièrement prudent dans la manipulation des indices. Afin d'illustrer ces calculs, nous utiliseront les anciennes tables françaises (qui présentent l'avantage d'être simples d'utilisation) dites TV88-90 (`TV`, en cas de vie) et TD88-90 (`TD`, en cas de décès) .

Les tables étant un comptage de *survivants*, on en déduit aisément un estimateur des probabilité de survie (et donc aussi de décès, même si nous reviendrons plus longuement sur ce point dans le prochain chapitre). La probabilité pour un individu d'âge $x = 40$ ans d'être encore en vie $k = 10$ ans plus tard (et donc d'atteindre les 50 ans) s'écrit

$$ {}_kp_x = \frac{L_{x+k}}{L_x}, \text{ avec ici } x = 40 \text{ et } k = 10.$$

```
> TD[39:52,]
   Age    Lx
39  38 95237
40  39 94997
41  40 94746
42  41 94476
43  42 94182
44  43 93868
45  44 93515
46  45 93133
47  46 92727
48  47 92295
49  48 91833
```

```
50  49 91332
51  50 90778
52  51 90171
> TD$Lx[TD$Age==50]
[1] 90778
> x <- 40
> h <- 10
> TD$Lx[TD$Age==x+h]/TD$Lx[TD$Age==x]
[1] 0.9581196
> TD$Lx[x+h+1]/TD$Lx[x+1]
[1] 0.9581196
```

Sous cette forme, on retrouve des formules classiques de probabilités conditionnelles (car on conditionne toujours par le fait que l'individu est en vie à l'âge x) par exemple

$$ {}_{k+h}p_x = \frac{L_{x+k+h}}{L_x} = \frac{L_{x+k+h}}{L_{x+k}} \cdot \frac{L_{x+k}}{L_x} = {}_hp_{x+k} \cdot {}_kp_x $$

soit

$$ \mathbb{P}(T_x > k+h) = \mathbb{P}(T > x+k+h | T > x) = \mathbb{P}(T > x+k+h | T > x+k) \cdot \mathbb{P}(T > x+k | T > x). $$

Cette relation sera discutée plus en détails dans le Chapitre 5.

Nous verrons par la suite l'intérêt de toutes ces formules itératives, mais on peut déjà noter que comme il semble intéressant de parfois changer l'âge de l'individu (ici en regardant par exemple un individu d'âge $x+k$), on peut voir ${}_kp_x$ comme le terme générique d'une matrice `p`, dépendant des paramètres x et k (avec toujours $x = 0, 1, 2, 3, \cdots$ ce qui posera des problèmes d'indexation, et $k = 1, 2, 3, \cdots$). Avec cette écriture, nous aurons des soucis pour travailler avec les âges $x = 0$. Toutefois, les produits d'assurance-vie étant souvent destiné à des personnes d'âge plus avancé, nous garderons cette simplification dans la première partie de ce chapitre.

```
>  Lx <- TD$Lx
>  m <- length(Lx)
>  p <- matrix(0,m,m); d <- p
>  for(i in 1:(m-1)){
+  p[1:(m-i),i] <- Lx[1+(i+1):m]/Lx[i+1]
+  d[1:(m-i),i] <- (Lx[(1+i):(m)]-Lx[(1+i):(m)+1])/Lx[i+1]}
>  diag(d[(m-1):1,]) <- 0
>  diag(p[(m-1):1,]) <- 0
>  q <- 1-p
```

La matrice `p` contient les ${}_jp_i$, la matrice `q` contient les ${}_jq_i$, alors que la matrice `d` contient les ${}_jd_i$. On vérifiera sans trop de difficultés que la somme des éléments de `d` par colonne (donc à âge fixé) vaut 1,

```
>  apply(d,2,sum)[1:10]
 [1] 1 1 1 1 1 1 1 1 1 1
```

Aussi, `p[10,40]` correspondra à ${}_{10}p_{40}$:

```
>  p[10,40]
[1] 0.9581196
```

On peut ainsi représenter les fonctions de survie résiduelle, et calculer une espérance de vie résiduelle, en notant que

$$ e_x = \mathbb{E}(T_x) = \sum_{k=1}^{\infty} k \cdot {}_{k|1}q_x = \sum_{k=1}^{\infty} {}_kp_x $$

```
> x <- 45
> S <- p[,45]/p[1,45]
> sum(S)
[1] 30.46237
```

On peut aussi écrire une petite fonction permettant de calculer l'espérance de vie résiduelle à l'âge x, pour $x > 0$ (pour des raisons d'indexation de matrice expliquées auparavant),

```
> esp.vie=function(x){sum(p[1:nrow(p),x])}
> esp.vie(45)
[1] 30.32957
```

On peut aussi utiliser TGH-05 (pour les hommes, base de donnée `TGH` sous R) et TGF-05 (pour les femmes, notée `TGF`) qui ont été construites à partir d'une population de rentiers (et non plus sur l'ensemble de la population française comme les tables TV88-90 et TD88-90).

Ces tables sont différentes au sens où elles intègrent un aspect temporel que nous n'avons pas mentionné jusqu'à présent. Compte-tenu des améliorations des conditions de vie, on imagine que quelqu'un ayant 70 ans en 2010 n'a probablement pas la même fonction de survie résiduelle qu'une personne qui atteindra 70 ans en 2050. Et compte-tenu de la durée des engagements en assurance-vie, il semble légitime d'intégrer cet aspect temporel dans les calculs (ce point fera l'objet du prochain chapitre).

Si on considère une personne d'âge x l'année t, son année de naissance est alors $t-x$, colonne qui va permettre de récupérer les L_{x+k} utiles pour les calculs.

```
> annee <- 2010
> age <- 45
> an    <- annee-age; if(an>2005){an=2005}
> nom   <- paste("X",an,sep="")
> LH    <- TGH[,nom]
> LF    <- TGF[,nom]
```

4.1.2 Calculs de valeurs actuelles probables

La valeur actuelle probable s'écrit, de manière très générale,

$$\sum_{j=1}^{k} \frac{C_j \cdot p_j}{(1+i)^j} = \sum_{j=1}^{k} \nu^j \cdot C_j \cdot p_j$$

où $\boldsymbol{C} = (C_1, \cdots, C_k)$ est l'ensemble des montants à verser (correspondant à un vecteur `C`), i est le taux d'actualisation, et $\boldsymbol{p} = (p_1, \cdots, p_k)$ est le vecteur des probabilités de verser le capital aux différentes dates $\{1, 2, ..., k\}$ (correspondant à un vecteur `P`).

```
>  k <- 20; x <- 40; i <- 0.03
>  C <- rep(100,k)
>  P <- p[1:k,x]
>  sum((1/(1+i)^(1:k))*P*C)
[1] 1417.045
>  sum(cumprod(rep(1/(1+i),k))*P*C)
[1] 1417.045
```

Rappelons que ce calcul peut se faire au sein d'une fonction générique,

```
> LxTD<-TD$Lx
>  VAP <- function(capital=1,m=1,n,Lx=TD$Lx,age,taux=.03)
```

```
+  {
+  proba <- Lx[age+1+m:n]/Lx[age+1]
+  vap <- sum((1/(1+taux)^(m:n))*proba*capital)
+  return(vap)
+  }
>  VAP(capital=100,n=20,age=40)
[1] 1417.045
```

On peut ainsi rapidement changer la table,

```
> VAP(capital=100,n=20,age=40,L=TV$Lx)
[1] 1457.646
> VAP(capital=100,n=20,age=40,L=LH)
[1] 1472.078
> VAP(capital=100,n=20,age=40,L=LF)
[1] 1472.598
```

ou les taux d'actualisation

```
> VAP(capital=100,n=20,age=40,taux=.04)
[1] 1297.245
```

Il est aussi possible de visualiser la sensibilité de ces valeurs actuelles probables en fonction des taux, d'actualisation, ou de l'âge de l'assuré, comme sur la Figure 4.1

```
> VAPtaux <- function(T){VAP(capital=100,n=20,age=40,taux=T)}
> vVAPtaux <- Vectorize(VAPtaux)
> TAUX <- seq(.01,.07,by=.002)
> VAPage <- function(A){VAP(capital=100,n=20,age=A,taux=.035)}
> vVAPage <- Vectorize(VAPage)
> AGE <- seq(20,60)
> par(mfrow = c(1, 2))
> plot(100*TAUX,vVAPtaux(TAUX),xlab="Taux d'actualisation (%)",
+ ylab="Valeur Actuelle Probable")
> plot(AGE,vVAPage(AGE),xlab="Age de l'assuré",ylab="Valeur Actuelle Probable")
> par(mfrow = c(1, 1))
```

4.2 Calculs d'annuités

A partir du moment où nous disposons de toutes les probabilités ${}_kp_x$, il est possible de faire tous les calculs imaginables d'actualisation de flux futurs probables. Nous allons reprendre ici les produits les plus classiques, et notant que tous les produits complexes d'assurance-vie peuvent être vus comme des combinaisons linéaires de ces produits simples. Par linéarité de l'espérance, la valorisation pourra être faite en faisant la même combinaison linéaire de ces valeurs actuelles probables.

4.2.1 Valeurs actuelles probables de capital différé

Le plus simple est probablement la valeur actuelle probable d'un capital différé (*pure endowment*) ${}_kE_x$, correspondant à la valeur actuelle probable d'un capital de 1 dans le cas où une personne actuellement d'âge x soit encore en vie à au bout de k années, i.e.

$$ {}_kE_x = \frac{1}{(1+i)^k} \cdot \mathbb{P}(T > x+k | T > x) = \frac{1}{(1+i)^k} \cdot {}_kp_x $$

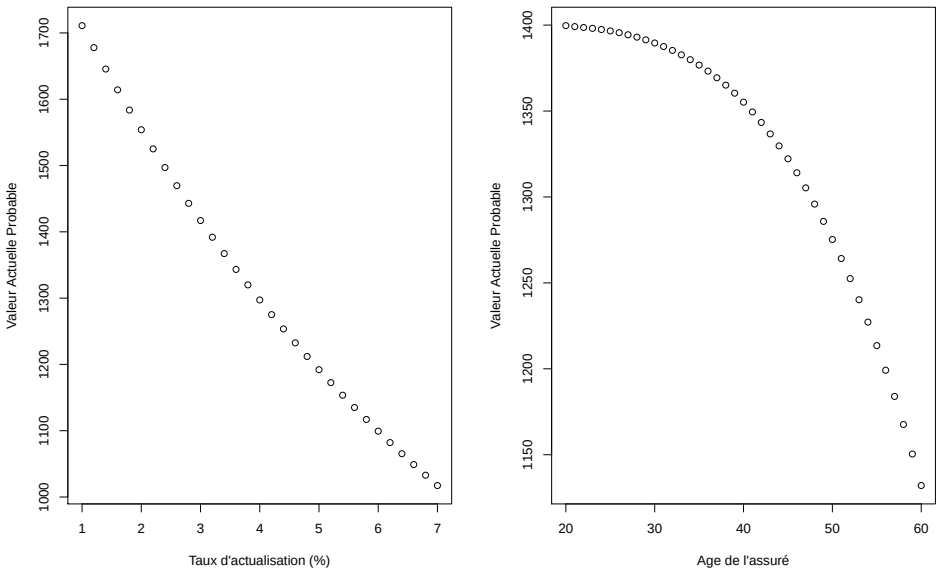


FIGURE 4.1 – Evolution de la valeur actuelle probable de 20 versements de 100 conditionnels à la survie de l'assuré d'âge x au premier versement, en fonction du taux d'actualisation (à gauche), et de l'âge de l'assuré (à droite).

Là encore, ${}_kE_x$ peut être vu comme le terme générique d'une matrice que l'on notera `E`.

```
> E <- matrix(0,m,m)
> i <- .035
> for(j in 1:m){
+ E[,j] <- (1/(1+i)^(1:m))*p[,j]
+ }
> E[10,45]
[1] 0.663491
> p[10,45]/(1+i)^10
[1] 0.663491
```

4.2.2 Exemples d'assurance en cas de vie

Considérons le cas du versement d'une unité monétaire, commençant dès aujourd'hui, et continuant tant que l'assuré sera vivant. On parlera d'annuité "vie entière". On supposera l'annuité payable d'avance. On peut noter que

$$\ddot{a}_x = \sum_{k=0}^{\infty} \frac{1}{(1+i)^k} \cdot {}_kp_x = \sum_{k=0}^{\infty} {}_kE_x$$

Plus généralement, on veut considérer non pas des assurance "vie entière", mais dites "temporaires", d'une durée de n années (avec n versements), i.e.

$${}_n\ddot{a}_x = \sum_{k=0}^{n-1} \frac{1}{(1+i)^k} {}_kp_x = \sum_{k=0}^{n-1} {}_kE_x$$

Le code est alors le suivant :

```
> adot<-matrix(0,m,m)
> for(j in 1:(m-1)){
+ adot[,j]<-cumsum(1/(1+i)^(0:(m-1))*c(1,p[1:(m-1),j]))
+ }
> adot[nrow(adot),1:5]
[1] 26.63507 26.55159 26.45845 26.35828 26.25351
```

Notons que l'on peut également différer de h années,

$$ {}_{h|n}\ddot{a}_x = \sum_{k=h}^{h+n-1} \frac{1}{(1+i)^k} \cdot {}_kp_x = \sum_{k=h}^{h+n-1} {}_kE_x $$

A h fixé, on peut construire la matrice `adot`, contenant les $[{}_{h|n}\ddot{a}_x]$ (indicé ici en n et x),

```
> h <- 1
> adoth <- matrix(0,m,m-h)
> for(j in 1:(m-1-h)){
+ adoth[,j]<-cumsum(1/(1+i)^(h+0:(m-1))*p[h+0:(m-1),j])
+ }
> adoth[nrow(adoth),1:5]
[1] 25.63507 25.55159 25.45845 25.35828 25.25351
```

Dans cet exemple numérique, on décale d'un an, autrement dit, au lieu de considérer des versements payables d'avance, on considère des versements à terme échu. Classiquement, ces ${}_{1|\infty}\ddot{a}_x$ sont notés a_x,

$$ a_x = \sum_{k=1}^{\infty} \frac{1}{(1+i)^k} \cdot {}_kp_x = \sum_{k=1}^{\infty} {}_kE_x $$

```
> a<-matrix(0,m,m)
> for(j in 1:(m-1)){
+ a[,j]<-cumsum(1/(1+i)^(1:(m))*p[1:m,j])
+ }
> a[nrow(a),1:5]
[1] 25.63507 25.55159 25.45845 25.35828 25.25351
```

La dernière ligne de la matrice (présentée ci-dessus) donne les valeurs des annuités "vie entière" en fonction de l'âge de l'assuré. On retrouve ce qu'aurait donné un calcul direct à l'aide des ${}_kE_x$

```
> apply(E,2,sum)[1:5]
[1] 25.63507 25.55159 25.45845 25.35828 25.25351
```

Pour les nouvelles tables, TGH et TGF, il est possible d'utiliser le code suivant, pour calculer la valeur d'une rente de 1 euro, versée pendant une durée (avec une distinction suivant que le versement survient en début ou en fin d'année)

```
> PRIX <- function(annee=2011,age,sex="HOM",taux=0.04,duree,C=1){
+ an <- annee-age; if(an>2005){an=2005}
+ nom <- paste("X",an,sep="")
+ if(sex=="HOM"){L <- TGH[,nom]}
+ if(sex=="FEM"){L <- TGF[,nom]}
+ Q <- L[(age+1):length(L)]/L[(age+1)]
+ actualisation <- (1+taux)^(0:min(duree,120-age))
+ prixsup <- sum(Q[2:(min(duree,120-age)+1)]/
+ actualisation[2:(min(duree,120-age)+1)] )
```

```
+ prixinf <- sum(Q[1:(min(duree,120-age))]/
+ actualisation[1:(min(duree,120-age))] )
+ return(C*c(prixinf,prixsup))}
> PRIX(age=45,duree=20)
[1] 13.95699 13.39479
```

Cette fonction permet d'avoir le prix de la rente versée en début d'année en cas de vie, ou en fin d'année.

4.2.3 Exemples d'assurance en cas de décès

Comme précédemment, le cas le plus simple est probablement l'assuranc décès vie entière, dont la valeur actuelle probable s'écrit, pour un assuré d'âge x qui souhaite le versement d'une unité à la fin de l'année de son décès,

$$A_x = \mathbb{E}\left(\left(\frac{1}{1+i}\right)^{T_x+1}\right) = \sum_{k=0}^{\infty} \mathbb{E}\left(\left(\frac{1}{1+i}\right)^{T_x+1} | T_x = k\right) = \sum_{k=1}^{\infty} \frac{1}{(1+i)^k} \cdot {}_{k-1}p_x \cdot {}_1q_{x+k-1}.$$

Plus généralement, on peut définir une assurance "temporaire décès", où le versement du capital n'a lieu que si le décès survient dans les n années qui suivent la signature du contrat,

$${}_nA_x = \sum_{k=1}^{n} \frac{1}{(1+i)^k} \cdot {}_{k-1}p_x \cdot {}_1q_{x+k-1}.$$

En utilisant la matrice `d` définie auparavant, et $\nu = (1+i)^{-1}$ le facteur d'actualisation, on a alors

```
> A<- matrix(NA,m,m-1)
> for(j in 1:(m-1)){
+  A[,j]<-cumsum(1/(1+i)^(1:m)*d[,j])
+ }
> Ax <- A[nrow(A),1:(m-2)]
```

On peut alors visualiser ces fonctions, et aussi comparer $\mathbb{E}\left(\nu^{1+T_x}\right)$ avec $\left(\nu^{1+\mathbb{E}(T_x)}\right)$ si on considère des versements à terme échu (qui pourraient être vu comme des approximations de ce montant). Afin de faciliter les calculs, on peut utiliser une version vectorisée de la fonction `esp.vie`,

```
> EV <- Vectorize(esp.vie)
```

On peut alors visualiser la différence sur la figure 4.2

```
> plot(0:105,Ax,type="l",xlab="Age",lwd=1.5)
> lines(1:105,v^(1+EV(1:105)),col="grey")
> legend(1,.9,c(expression(E((1+r)^-(Tx+1))),expression((1+r)^-(E(Tx)+1))),
+ lty=1,col=c("black","grey"),lwd=c(1.5,1),bty="n")
```

A partir de ces contrats de base, il est possible de calculer toutes les valeurs actuelles probables de flux futurs aléatoires.

4.3 Calculs de provisions mathématiques

En assurance-vie, les engagements de l'assuré et de l'assureur sont, le plus souvent, répartis dans le temps sur de longues périodes. Pour les rentes par exemple, l'assuré paye ses primes (durant plusieurs années de cotisation), et ensuite seulement l'assureur verse une rente. Il y a alors

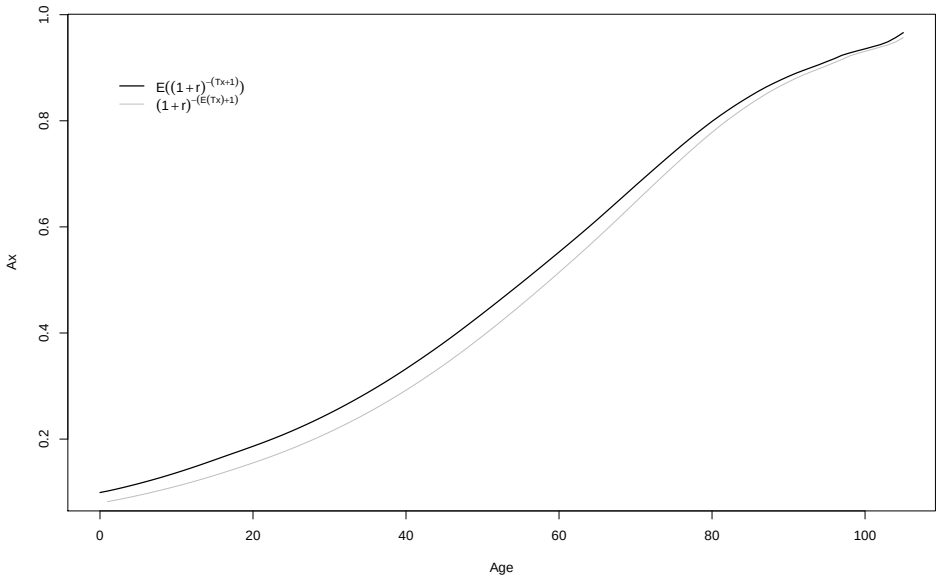


FIGURE 4.2 – Comparison de $x \mapsto A_x = \mathbb{E}\left(\nu^{1+T_x}\right)$ et $\left(\nu^{1+\mathbb{E}(T_x)}\right)$.

un décalage entre les prime payée par l'assurée et la couverture du risque par l'assureur, décalage qui doit être présenté dans les comptes annuels, intégrant les prévisions de dépenses constituées sous forme de "provisions" (dites mathématiques). Pour reprendre la définition de Petauton (2004) et du Code des Assurances, les provisions mathématiques sont "*à l'époque de l'évaluation la différence entre d'une part la valeur actuelle probable des engagements pris par l'assureur* [...] *et d'autre part la valeur actuelle probable des engagements pris par les souscripteurs*".

Notons $VAP^{t_0}_{[t_1,t_2]}$(assuré) la valeur actuelle probable, en t_0, des engagements de l'assuré pour la période $[t_1, t_2]$. Aussi, $VAP^0_{[0,k]}$(assuré) sera la valeur actuelle probable, en 0, des k premières primes annuelles. Et on notera $VAP^0_{[k+1,n]}$(assuré) la valeur actuelle probable, en 0, des engagements de l'assuré pour la période $[k+1, n]$, i.e. la valeur actuelle probable des $n-k$ dernières primes annuelles.

De manière analogue, notons $VAP^{t_0}_{[t_1,t_2]}$(assureur) la valeur actuelle probable, en t_0, des engagements de l'assureur pour la période $[t_1, t_2]$. Compte tenu du principe fondamental de valorisation [1], pour un contrat arrivant à échéance au bout de n années, on doit avoir

$$VAP^0_{[0,n]}(\text{assuré}) = VAP^0_{[0,n]}(\text{assureur})$$

pour un contrat soucrit à la date 0 et tel qu'il n'y a plus d'engagement de part et d'autre part n années. Aussi, pour k compris entre 0 et n,

$$VAP^0_{[0,k]}(\text{assuré}) + VAP^0_{[k+1,n]}(\text{assuré}) = VAP^0_{[0,k]}(\text{assureur}) + VAP^0_{[k+1,n]}(\text{assureur})$$

avec, de manière générale

$$VAP^0_{[0,k]}(\text{assuré}) \geq VAP^0_{[0,k]}(\text{assureur})$$

1. Tous les calculs sont nets, au sens où aucune marge de sécurité n'est considérée, et qu'aucun frais n'est prélevé afin de permettre à la compagnie de fonctionner. À la souscription, la valeur actuelle probable des engagements de l'assuré doit être égale à la valeur actuelle probable des engagements de l'assureur.

et

$$VAP^0_{[k+1,n]}(\text{assuré}) \leq VAP^0_{[k+1,n]}(\text{assureur})$$

(d'où le principe d'inversion du cycle de production de l'assurance). La provision mathématique (pure) de l'année k sera notée ${}_kV_x(t)$ si elle est actualisée à la date t. La référence étant ${}_kV_x = {}_kV_x(k)$ (i.e. on actualise en k). On définie ${}_kV_x(0)$ par

$${}_kV_x(0) = VAP^0_{[0,k]}(\text{assuré}) - VAP^0_{[0,k]}(\text{assureur}).$$

Cette définition sera dite *rétrospective* (car on se place sur la période antérieure à k). On peut aussi écrire, de manière équivalente (compte tenu du principe de valorisation)

$${}_kV_x(0) = VAP^0_{[k+1,n]}(\text{assureur}) - VAP^0_{[k+1,n]}(\text{assuré}).$$

Cette définition sera dite *prospective* (car on se place sur la période postérieure à k). Enfin, il existe une dernière méthode, correspondant à une simple mise à jour, i.e.

$${}_{k-1}V_x(k-1) + VAP^{k-1}_{[k-1,k]}(\text{assuré}) - VAP^{k-1}_{[k-1,k]}(\text{assureur}) = {}_kV_x(k-1).$$

Cette méthode sera dite *itérative*, voire en l'occurence itérative ascendante, car on initialise avec ${}_0V_x(0) = 0$. Mais il sera aussi possible de construire une méthode itérative descendante, commençant à la fin du contrat (ici la récursion est ascendante).

4.3.1 Exemple d'une assurance temporaire décès

Le *principe fondamental de valorisation* nous garantit que

$$VAP^0(\text{assuré}) = VAP^0(\text{assureur})$$

en faisant une valorisation à la date 0, c'est à dire à la date de souscription du contrat.

Plaçons nous du point de vue de l'assuré (d'âge x à la souscription) : il souhaite payer une prime annuelle constante $\pi_{n,x}$, noté plus simplement π, tant qu'il est en vie i.e.

$$VAP^0(\text{assuré}) = \sum_{k=0}^{n-1} \frac{\pi}{(1+i)^k} \cdot \mathbb{P}(T_x > k) = \pi \cdot {}_n\ddot{a}_x,$$

où

$${}_n\ddot{a}_x = \sum_{k=0}^{n-1} \frac{1}{(1+i)^k} \cdot {}_kp_x,$$

(on utilise ici $\ddot{a}$ car le paiement se faisant ici en début de période). De même,

$$VAP^0(\text{assureur}) = \sum_{k=1}^{n} \frac{1}{(1+i)^k} \cdot \mathbb{P}(k-1 < T_x \leq k) = {}_nA_x,$$

où

$${}_nA_x = \sum_{k=1}^{n} \frac{1}{(1+i)^k} \cdot {}_{k-1}p_x \cdot {}_1q_{x+k-1},$$

(l'indemnité étant versée par l'assureur à terme échu). On en déduit que la prime annuelle est alors

$$\pi = \frac{{}_nA_x}{{}_n\ddot{a}_x}.$$

À partir des grandeurs (ou de ces matrices de grandeurs) calculées auparavant, on peut calculer la prime annuelle des contrats décès

```
>  x <-50; n <-30
>  prime <-A[n,x]/adot[n,x]
>  sum(prime/(1+i)^(0:(n-1))*c(1,p[1:(n-1),x]))
[1] 0.3047564
>  sum(1/(1+i)^(1:n)*d[1:n,x])
[1] 0.3047564
```

La méthode prospective

Pour le calcul de la provision mathématique du contrat d'assurance "temporaire décès", la méthode prospective permet d'écrire

$$ {}_kV_x(0) = VAP^0_{[k+1,n]}(\text{assureur}) - VAP^0_{[k+1,n]}(\text{assuré}) $$

Notons que ${}_kV_x(0) = {}_kV_x(k) \cdot {}_kE_x$ où ${}_kE_x$ est la valeur actuelle probable d'un capital différé, relatif au versement d'un euro dans k années, conditionnée par la survie de l'assuré d'âge x à la souscription, i.e.

$$ {}_kE_x = \frac{1}{(1+i)^k} \cdot \mathbb{P}(T_x > k) = \nu^k \cdot {}_kp_x $$

Si l'on se place à la date k (car c'est le plus simple, mais l'assuré a alors l'âge $x+k$), notons que la différence entre les valeurs actuelles probables des engagements des deux parties donne, simplement

$$ {}_kV_x(k) = {}_{n-k}A_{x+k} - \pi \cdot {}_{n-k}\ddot{a}_{x+k} $$

car d'un côté, on a une assurance "temporaire décès" sur les $n-k$ années restantes pour un assuré d'âge $x+k$, et de l'autre, l'assuré a pris l'engagement de verser sa prime (qui reste inchangée) pendant $n-k$ années s'il vit. Aussi,

$$ {}_kV_x(0) = {}_kV_x(k) \cdot {}_kE_x = {}_{k|n-k}A_x - \pi \cdot {}_{k|n-k}\ddot{a}_x $$

où l'on considère des assurances décès différées. On peut aussi écrire

$$ {}_kV_x(k) = \frac{{}_{k|n-k}A_x - \pi \cdot {}_{k|n-k}\ddot{a}_x}{{}_kE_x} $$

```
> VR <- (prime*adot[1:n,x]-A[1:n,x])/E[1:n,x]
> plot(0:n,c(0,VR),xlab="",ylab="Provisions mathématiques",type="b")
```

La méthode retrospective

Pour la méthode rétrospective, on écrit simplement

$$ {}_kV_x(0) = VAP^0_{[0,k]}(\text{assuré}) - VAP^0_{[0,k]}(\text{assureur}) $$

i.e. ${}_kV_x(k) = \pi_k\ddot{a}_x - {}_kA_x$. Or ${}_kV_x(0) = {}_kV_x(k) \cdot {}_kE_x$, et donc

$$ {}_kV_x(k) = \frac{\pi_k\ddot{a}_x - {}_kA_x}{{}_kE_x}. $$

```
> VP <- diag(A[n-(0:(n-1)),x+(0:(n-1))])-
+ primediag(adot[n-(0:(n-1)),x+(0:(n-1))])
> points(0:n,c(VP,0),pch=4)
```

La méthode itérative

Enfin, pour la dernière méthode, l'idée est ici de décrire la variation de la provision mathématique entre deux dates en fonction des variations des engagements de part et d'autre. D'un côté il y a le paiement de la prime (en début de période, donc pas de problème d'actualisation et de non-paiement), et de l'autre, une assurance décès sur un an. Aussi ${}_kV_x(k-1) = {}_{k-1}V_x(k-1) + \pi - {}_1A_{x+k-1}$. Or ${}_kV_x(k-1) = {}_kV_x(k) \cdot {}_1E_{x+k-1}$ ce qui donne, finalement

$$ {}_kV_x(k) = \frac{{}_{k-1}V_x(k-1) + \pi - {}_1A_{x+k-1}}{{}_1E_{x+k-1}} $$

avec la convention que la première provision est nulle (de part notre principe fondamental de valorisation).

```
> VI<-0
>  for(k in 1:n){
+  VI <- c(VI,(VI[k]+prime-A[1,x+k-1])/E[1,x+k-1])
+  }
>  points(0:n,VI,pch=5)
```

Comme le montre la Figure 4.3, ces trois méthodes coïncident (on ne distingue plus les trois points),

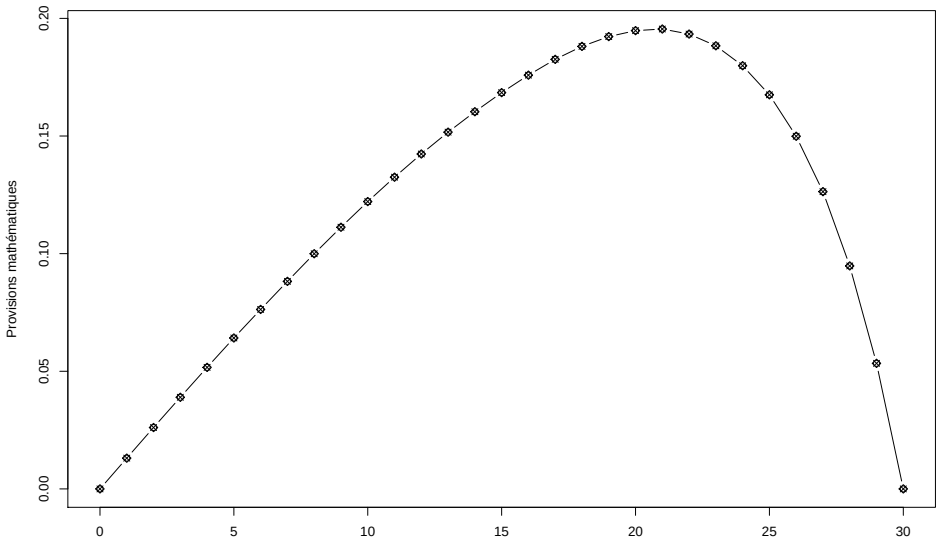


FIGURE 4.3 – Évolution de la provision mathématique pour un contrat d'assurance "temporaire décès", $x = 50$, $n = 30$ et $i = 3.5\%$.

4.3.2 Exemple d'une assurance en cas de vie

On considère ici un assuré d'âge x, cotisant pendant m années pour sa retraite, et touchant au bout de n années de cotisation une rente annuelle d'un montant C, payé tous les ans à

terme échu s'il est en vie, jusqu'à son décès (i.e. une annuité viagère). La prime pure unique (correspondant à la valeur actuelle probable des engagements de l'assureur) serait

$$\Pi = VAP_0 = \sum_{j=n}^{\infty} \frac{C}{(1+i)^j} \Pr(T_x > j),$$

soit, avec les notations actuarielles, $VAP_0 = C \cdot {}_{n|}a_x$ (i.e. la valeur actuelle probable d'une annuité viagère différée de n années). Si l'assuré paye une prime annuelle constante pendant ces n années, en début d'année, alors la prime est

$$\pi = \frac{VAP_0}{{}_n\ddot{a}_x} = C \cdot \frac{{}_{n|}a_x}{{}_n\ddot{a}_x}$$

On peut alors passer au calcul de la provision mathématique, en notant qu'il faudra distinguer les n premières années (période où l'assuré paye sa prime) et les dernières (période où l'assureur verse la rente). Pour le calcul des ${}_{n|}a_x$, on va utiliser la matrice `adiff`

```
> adiff=matrix(0,m,m)
>  for(i in 1:(m-1)){
+  adiff[(1+0:(m-i-1)),i] <- E[(1+0:(m-i-1)),i]*a[m,1+i+(0:(m-i-1))]
+ }
```

La prime annuelle peut être calculée de plusieurs manières pour une personne souscrivant un contrat à $x = 35$ ans.

```
>  x <- 35
>  n <- 30
>  a[n,x]
[1] 17.31146
>  sum(1/(1+i)^(1:n)*c(p[1:n,x]) )
[1] 17.31146
> (prime <- adiff[n,x] / (adot[n,x]))
[1] 0.1661761
> sum(1/(1+i)^((n+1):m)*p[(n+1):m,x] )/sum(1/(1+i)^(1:n)*c(p[1:n,x]) )
[1] 0.17311
```

Une fois obtenue la cotisation à payer (annuellement) pendant n année (notée `prime`), on peut calculer les provisions mathématiques, en distinguant la période de cotisation (où la provision devrait augmenter avec le temps) de la période de retraite (où la provision devrait baisser).

Méthode prospective

On se place ici au bout de k années. Si $k < n$ (l'assuré paye encore sa prime), en faisant la différence entre les engagements restants de l'assureur et ceux de l'assuré, on obtient

$${}_kV_x(0) = C \cdot {}_{n-k|}a_{x+k} - {}_{n-k}\ddot{a}_{x+k}.$$

Si en revanche on suppose que $k \geq n$ (seul l'assureur a encore des engagements) alors

$${}_kV_x(0) = C \cdot a_{x+k}.$$

Tout simplement. En effet, dans le premier cas, l'assuré a vieilli, et il a moins de versements à venir (c'est la partie de droite). Pour l'assureur, il s'agit toujours d'une annuité différée. Dans le second cas, l'assureur doit verser une rente viagère tant que l'assuré est en vie.

```
> VP <- rep(NA,m-x)
> VP[1:(n-1)] <- diag(adiff[n-(1:(n-1)),x+(1:(n-1))] -
+ adot[n-(1:(n-1)),x+(1:(n-1))]*prime)
> VP[n:(m-x)] <- a[m,x+n:(m-x)]
> plot(x:m,c(0,VP),xlab="Age de l'assuré",
+ ylab="Provisions mathématiques")
```

Méthode rétrospective

Là aussi, il faut distinguer suivant la valeur de k. Si $k \leq n$, on obtient simplement que

$$ {}_kV_x(0) = \frac{\pi \cdot {}_k\ddot{a}_x}{{}_kE_x} $$

puisque sur cette période, seul l'assuré a pris des engagements. Pour rappel, ${}_kE_x$ est la valeur actuelle probable du capital diffféré, i.e.

$$ {}_kE_x = \frac{{}_kp_x}{(1+i)^k}. $$

Pour la seconde période, si $k > n$,

$$ {}_kV_x(0) = \frac{\pi \cdot {}_n\ddot{a}_x - C \cdot {}_{n|k}a_x}{{}_kE_x} $$

avec à gauche un terme constant (les engagements de l'assuré étant passés), et à droite les engagements qu'avait pris l'assureur, i.e. les $k-n$ années qui ont suivi l'année n.

Pour les calculs, on utilise le fait que

$$ {}_{n|k}a_x = \sum_{j=n+1}^{n+k} {}_jE_x = {}_{n|}a_x - {}_{n+k|}a_x $$

On peut alors utiliser (comme l'indice x ne change pas) une matrice fonction des deux premiers indices,

```
>  adiff[n,x]
[1] 2.996788
>  adiff[min(which(is.na(adiffx[,n])))-1,n]
[1] 2.996788
> adiff[10,n]
[1] 2.000453
> adiff[n,x]- adiff[n+10,x]
[1] 2.000453
```

A l'aide de ces fonctions, on peut calculer les provisions de manière retrospective,

```
> VR <- rep(NA,m-x)
> VR[1:(n)] <- adot[1:n,x]*prime/E[1:n,x]
> VR[(n+1):(m-x)] <- (adot[n,x]*prime - (adiff[(n),x]-
+ adiff[(n+1):(m-x),x]) )/E[(n+1):(m-x),x]
>  points(x:m,c(0,VR),pch=4)
```

Méthode itérative

Pour la méthode itérative, on notera que si $k \leq n$,

$$ {}_kV_x(0) = \frac{{}_{k-1}V_x(0) + \pi}{{}_1E_{x+k-1}} $$

alors que si $k > n$

$$ {}_kV_x(0) = \frac{{}_{k-1}V_x(0)}{{}_1E_{x+k-1}} - C. $$

Avant la retraite, la provision augmente du montant de la prime, et lorsque l'assuré prend sa retraite, la provision diminue du montant de la rente annuelle versée.

```
>  VI<-0
>  for(k in 1:n){
+  VI<-c(VI,((VI[k]+prime)/E[1,x+k-1]))
+  }
>  for(k in (n+1):(m-x)){
+  VI<-c(VI,((VI[k])/E[1,x+k-1]-1))
+  }
>  points(x:m,VI,pch=5)
```

Comme auparavant, les trois méthodes donnent des résultats identiques, et on peut visualiser l'évolution de la provision mathématique sur la Figure 4.4

```
> provision<-data.frame(k=0:(m-x),
+ retrospective=c(0,VR),prospective=c(0,VP),
+ iterative=VI)
> head(provision)
  k retrospective prospective iterative
1 0     0.0000000   0.0000000 0.0000000
2 1     0.1723554   0.1723554 0.1723554
3 2     0.3511619   0.3511619 0.3511619
4 3     0.5367154   0.5367154 0.5367154
5 4     0.7293306   0.7293306 0.7293306
6 5     0.9293048   0.9293048 0.9293048
> tail(provision)
    k retrospective prospective    iterative
69 68     0.6692860   0.6692860 6.692860e-01
70 69     0.5076651   0.5076651 5.076651e-01
71 70     0.2760524   0.2760524 2.760525e-01
72 71     0.0000000   0.0000000 1.501743e-10
73 72           NaN   0.0000000          Inf
74 73           NaN   0.0000000          Inf
```

4.4 Algorithme récursif en assurance-vie

Giles (1993) a noté que, comme la plupart des quantitées utilisés en assurance vie pouvaient être obtenues de manière récursive, il était possible d'utiliser des algorithmes sur les suites définies par récurrence, pour calculer la plupart des grandeurs usuelles.

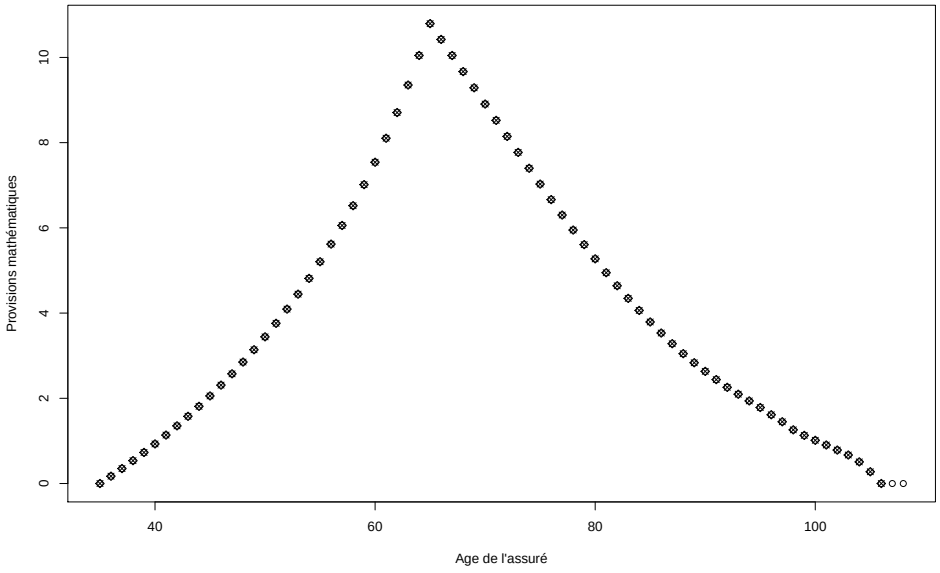


FIGURE 4.4 – Évolution de la provision mathématique pour un contrat d'assurance retraite, avec cotisation annuelle pendant n années puis versement d'une rente viagère, $x = 35$, $n = 30$ et $i = 3.5\%$.

4.4.1 Quelques exemples de relations de récurrence

En notant ${}_{k|}q_x = \mathbb{P}(k < T_x \leq k+1)$, la probabilité de décèder à l'âge $x+k$, la valeur actuelle probable d'un euro payé au décès d'une personne d'âge x aujourd'hui (à terme échu), s'écrit :

$$A_x = \mathbb{E}(\nu^{T_x+1}) = \sum_{k=0}^{\infty} v^{k+1}{}_{k|}q_x.$$

On notera qu'il existe une relation liant A_x et A_{x+1},

$$A_x = \nu q_x + \nu p_x A_{x+1}$$

Considérons maintenant une rente vie entière :

$$\ddot{a}_x = \sum_{k=0}^{\infty} \nu^k{}_k p_x,$$

qui peut se limiter également à n années :

$$\ddot{a}_{x:n\urcorner} = \sum_{k=0}^{n-1} \nu^k{}_k p_x = \frac{1 - A_{x:n\urcorner}}{1-\nu} \text{ o ù } A_{x:n\urcorner} = \nu^n{}_n p_x$$

Si l'on considère des paiements immédiats, et non plus à terme échu, on obtient

$$a_{x:n\urcorner} = \sum_{k=1}^{n} \nu^k{}_k p_x = \ddot{a}_{x:n\urcorner} - 1 + \nu^n{}_n p_x.$$

Dans le cas où on ne limite plus à n années, on a aussi :

$$\ddot{a}_x = 1 + \nu p_x \ddot{a}_{x+1}.$$

4.4.2 Algorithme de calculs itératifs

Les formules obtenues par récurrence sont particulièrement intéressants, car il est facile de les mettre en oeuvre. Supposons que $\boldsymbol{u} = (u_n)$ satisfasse une équation de la forme

$$u_n = a_n + b_n u_{n+1},$$

pour $n = 1, 2, \cdots, m$ de telle sorte que u_{m+1} est connu, où $\boldsymbol{a} = (a_n)$ et $\boldsymbol{b} = (b_n)$ sont connus. La solution générale est alors donnée par

$$u_n = \frac{u_{m+1} \prod_{i=0}^{m} b_i + \sum_{j=n}^{m} a_j \prod_{i=0}^{j-1} b_i}{\prod_{i=0}^{n-1} b_i}$$

avec la convention $b_0 = 1$. On peut utiliser le code générique suivant pour résoudre numériquement de telles relations de récurrence,

```
> recurrence <- function(a,b,ufinal){
+ s <- rev(cumprod(c(1, b)));
+ return(rev(cumsum(s[-1] * rev(a))) + s[1] * ufinal)/rev(s[-1])
+ }
```

Par exemple pour les calculs d'espérance de vie,

$$e_x = p_x + p_x \cdot e_{x+1}$$

Le code est alors tout simplement,

```
> Lx <- TD$Lx
> x <- 45
> kpx <- Lx[(x+2):length(Lx)]/Lx[x+1]
> sum(kpx)
[1] 30.32957
> esp.vie(x)
[1] 30.32957
> px <- Lx[(x+2):length(Lx)]/Lx[(x+1):(length(Lx)-1)]
> e<- recurrence(px,px,0)
> e[1]
[1] 30.32957
```

On retrouve la même espérance de vie restante pour une personne de 45 ans que le calcul direct, sauf qu'ici on a le vecteur des espérances de vie résiduelles à différents âges.

Pour les calculs de valeur actuelle probable, on peut regarder une assurance décès, avec un payement à terme échu, l'année du décés de l'assuré,

$$A_x = \nu q_x + \nu p_x A_{x+1}$$

Là encore, on peut utiliser l'écriture par récurrence,

```
> x <- 20
> qx <- 1-px
> v <- 1/(1+i)
> Ar <- recurrence(a=v*qx,b=v*px,xfinal=v)
```

Si on regarde la valeur de A_x pour $x = 20$,

```
> Ar[1]
[1] 0.1812636
> Ax[20]
[1] 0.1812636
```

Pour les calculs de provisions mathématiques

$$ {}_nV_x = vq_{x+n} - p_x + vp_{x+n}{}_{n+1}V_x $$

```
> x <- 50
> px <- L[(x+2):length(L)]/L[(x+1):(length(L)-1)]
> px <- px[-length(px)]
> qx <- 1-px
> V=recurrence(a=v*qx+px[1],b=v*px,xfinal=0)
```

4.5 Le package lifecontingencies

Toutes ces fonctions - ou presque - ont été programmées dans le package **lifecontingencies**.

4.5.1 Les quantités démographiques

```
> library(lifecontingencies)
```

A partir de `TD$Lx` correspondant au vecteur (L_x), il est possible de calculer à l'aide de la classe `lifetable` une table de mortalité, comportant pour tous les âges x les probabiliés de survie p_x, mais aussi les espérances de vie résiduelles e_x.

```
>  TD8890 <- new("lifetable",x=TD$Age,lx=TD$Lx,name="TD8890")
removing NA and 0s
>  TV8890 <- new("lifetable",x=TV$Age,lx=TV$Lx,name="TV8890")
removing NA and 0s
> TV8890
Life table TV8890

      x     lx        px         ex
1     0 100000 0.9935200 80.2153857
2     1  99352 0.9994162 79.2619494
3     2  99294 0.9996677 78.2881343
4     3  99261 0.9997481 77.3077311
5     4  99236 0.9997783 76.3247626
6     5  99214 0.9997984 75.3400508
7     6  99194 0.9998286 74.3528792
8     7  99177 0.9998387 73.3647956
9     8  99161 0.9998386 72.3765545
10    9  99145 0.9998386 71.3881558
```

Cet objet (de la classe `S4`) peut alors être appelé en utilisant différentes fonctions, comme la probabilité de survie ${}_{10}p_{40}$,

```
> pxt(TD8890,x=40,t=10)
[1] 0.9581196
> p[10,40]
[1] 0.9581196
```

qui correspondent aux calculs effectués auparavant.

Plusieurs autres fonctions peuvent être utilisées pour calculer d'autres quantités, comme ${}_{10}q_{40}$, ou encore $\overset{\circ}{e}_{40:10\urcorner}$,

```
> qxt(TD8890,40,10)
[1] 0.0418804
> exn(TD8890,40,10)
[1] 9.796076
```

Il est aussi possible de calculer des ${}_hp_x$ pour des durées h non entières. Plusieurs interpolations sont proposées, linéaire, avec une force de mortalité constante, ou encore hyperbolique,

```
> pxt(TD8890,90,.5,"linear")
[1] 0.8961018
> pxt(TD8890,90,.5,"constant force")
[1] 0.8900582
> pxt(TD8890,90,.5,"hyperbolic")
[1] 0.8840554
```

On peut visualiser ces trois méthodes d'interpolation sur la Figure 4.5

```
> pxtL <- function(u){pxt(TD8890,90,u,"linear")}
> pxtC <- function(u){pxt(TD8890,90,u,"constant force")}
> pxtH <- function(u){pxt(TD8890,90,u,"hyperbolic")}
> PXTL <- Vectorize(pxtL)
> PXTC <- Vectorize(pxtC)
> PXTH <- Vectorize(pxtH)
> u=seq(0,1,by=.025)
> plot(u,PXTL(u),type="l",xlab="Année",ylab="Probabilité de survie")
> lines(u,PXTC(u),col="grey")
> lines(u,PXTH(u),pch=3,lty=2)
> legend(.45,.99,c("Linéaire","Force de mortalité constante",
+ "Hyperbolique"),lty=c(1,1,2),
+ col=c("black","grey","black"),bty="n")
```

Pour le premier, on utilise tout simplement une interpolation linéaire entre ${}_{[h]}p_x$ et ${}_{[h]+1}p_x$ (en notant $[h]$ la partie entière de $h \geq 0$),

$$ {}_h\tilde{p}_x = (1-h+[h])\ {}_{[h]}p_x + (h-[h])\ {}_{[h]+1}p_x $$

Pour le second, on utilise le fait que

$$ {}_hp_x = \exp\left(-\int_0^h \mu_{x+s}ds\right). $$

Supposons que $h \in [0,1)$, et que $s \mapsto \mu_{x+s}$ est constante sur l'intervalle $[0,1)$, alors la formule précédante devient

$$ {}_hp_x = \exp\left(-\int_0^h \mu_{x+s}ds\right) = \exp[-\mu_x \cdot h] = (p_x)^h. $$

Enfin, la dernière (toujours dans le cas où $h \in [0,1)$), proposée par Baldacci, repose sur l'utilisation d'une relation de la forme

$$ \frac{1}{{}_hp_x} = \frac{1-h+[h]}{{}_{[h]}p_x} + \frac{h-[h]}{{}_{[h]+1}p_x} $$

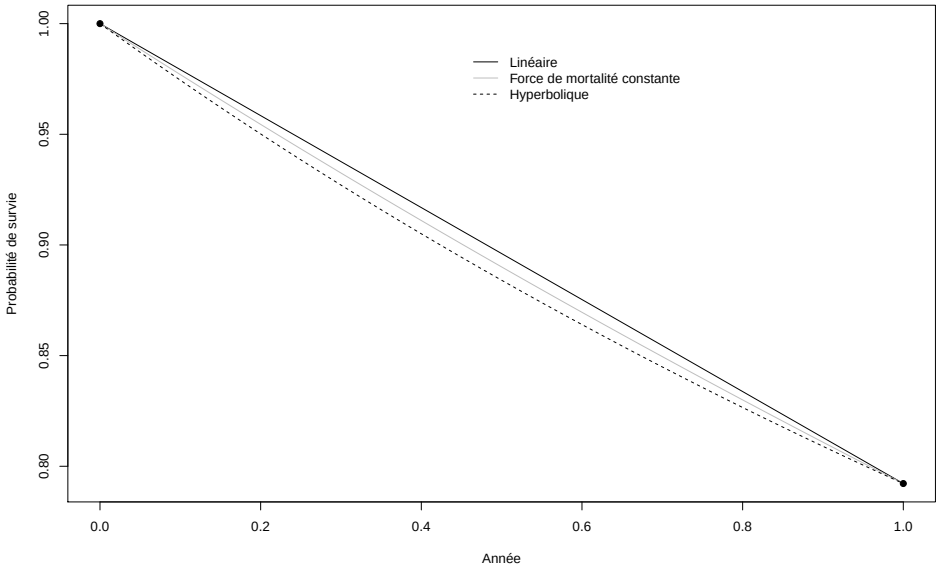


FIGURE 4.5 – Interpolation de ${}_hp_x$ pour $x = 90$ et $h \in [0, 1]$.

Cette relation peut également s'écrire

$$ {}_hp_x = \frac{{}_{[h]+1}p_x}{1 - (1 - h + [h])\ {}_{[h+1]}{}_hq_x} $$

```
> .5*pxt(TD8890,90,1)+.5*1
[1] 0.8961018
> pxt(TD8890,90,1)^.5
[1] 0.8900582
> pxt(TD8890,90,1)/(1-.5*qxt(TD8890,90,1))
[1] 0.8840554
> (.5/1+.5/pxt(TD8890,90,1))^(-1)
[1] 0.8840554
```

On peut aussi travailler sur plusieurs têtes, par exemple un homme (dont la table est TD88-90) et une femme (dont la table est TV88-90). On peut alors calculer des probabilités de survie jointe, ${}_hp_{xy}$, ou 'au contraire' la probabilité qu'au moins une personne soit encore en vie ${}_hp_{\overline{xy}}$,

```
> pxyt(TD8890,TV8890,x=40,y=42,t=10,status="joint")
[1] 0.9376339
> pxyt(TD8890,TV8890,x=40,y=42,t=10,status="last")
[1] 0.9991045
```

On peut aisément retrouver des propriétés classiques, comme

$$ {}_hp_{xy} = {}_hp_x \cdot {}_hp_y, $$

(en supposant les survies indépendantes) mais aussi

$$ {}_hp_{\overline{xy}} = {}_hp_x + {}_hp_y - {}_hp_{xy}. $$

```
> pxt(TD8890,40,10)*pxt(TV8890,42,10)
[1] 0.9376339
> pxt(TD8890,40,10)+pxt(TV8890,42,10)-
+ pxyt(TD8890,TV8890,x=40,y=42,t=10,status="joint")
[1] 0.9991045
```

Pour l'analyse de la survie sur deux têtes, on peut ainsi visualiser les fonctions de survie des durées restantes avant le premier et le dernier décès, sur la Figure 4.6

```
> JOINT=rep(NA,65)
> LAST=rep(NA,65)
> for(t in 1:65){
+ JOINT[t]=pxyt(TD8890,TV8890,x=40,y=42,t-1,status="joint")
+ LAST[t]=pxyt(TD8890,TV8890,x=40,y=42,t-1,status="last") }
> plot(1:65,JOINT,type="l",col="grey",xlab="",ylab="Probabilité de survie")
> lines(1:65,LAST)
> legend(5,.15,c("Dernier survivant","Vie jointe"),lty=1,
+ col=c("black","grey"),bty="n")
```

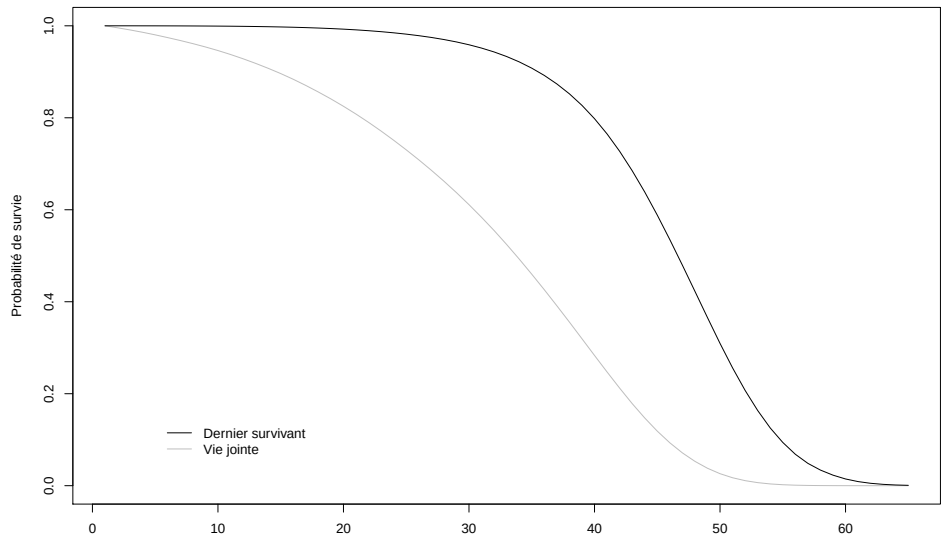


FIGURE 4.6 – Evolution de $h \mapsto {}_hp_{\overline{xy}}$ et $h \mapsto {}_hp_{xy}$ pour $x = 40$ et $y = 42$.

On peut également obtenir les espérances de ces deux lois,

```
> exyt(TD8890,TV8890,x=40,y=42,status="joint")
[1] 30.39645
> exyt(TD8890,TV8890,x=40,y=42,status="last")
[1] 44.21737
```

4.5.2 Les quantités actuarielles classiques

La valeur probable d'un capital différé est ${}_kE_x$, qui peut être calculé par

```
> Exn(TV8890,x=40,n=10,i=.04)
[1] 0.6632212
> pxt(TV8890,x=40,10)/(1+.04)^10
[1] 0.6632212
```

Les calculs d'annuités sont eux aussi relativement simples à obtenir, et à recalculer, par exemple les ${}_n\ddot{a}_x$

```
> Ex <- Vectorize(function(N){Exn(TV8890,x=40,n=N,i=.04)})
> sum(Ex(0:9))
[1] 8.380209
> axn(TV8890,x=40,n=10,i=.04)
[1] 8.380209
```

ou encore les ${}_nA_x$,

```
> Axn(TV8890,40,10,i=.04)
[1] 0.01446302
```

Il est aussi possible d'avoir des flux croissants (*Increasing*) ou décroissants (*Decreasing*) de manière arithmétique, i.e.

$$ {}_nIA_x = \sum_{k=0}^{n-1} \frac{k+1}{(1+i)^k} \cdot {}_{k-1}p_x \cdot {}_1q_{x+k-1}, $$

ou

$$ {}_nDA_x = \sum_{k=0}^{n-1} \frac{n-k}{(1+i)^k} \cdot {}_{k-1}p_x \cdot {}_1q_{x+k-1}, $$

```
> DAxn(TV8890,40,10,i=.04)
[1] 0.07519631
> IAxn(TV8890,40,10,i=.04)
[1] 0.08389692
```

Dans le cas où le capital n'est pas versé en début d'années, mais fractionné (par exemple tous les mois), les calculs sont un peu différents. Par exemple, si on ne verse plus 1 (euro) en début d'année, mais 1/12 tous les mois, la valeur actuelle probable des flux futurs est

```
> sum(Ex(seq(0,5-1/12,by=1/12))*1/12)
[1] 4.532825
```

Ce montant est obtenu directement à l'aide du paramètre `k` dans la fonction `axn`,

```
> axn(TV8890,40,5,i=.04,k=12)
[1] 4.532825
```

4.5.3 Exemple de calculs de primes et de provisions mathématiques

Considérons un contrat d'assurance décès où un capital K est versé aux ayant-droits si le décès d'une personne x survient entre l'âge x et $x+m$. On suppose qu'une prime constante est versée annuellement entre l'âge x et $x+n$ (avec $n \leq m$). La prime π est alors solution de

$$K \cdot A_{x:m\urcorner} = \pi \cdot \ddot{a}_{x:n\urcorner}, \text{ i.e. } \pi = K \cdot \frac{A_{x:m\urcorner}}{\ddot{a}_{x:n\urcorner}}.$$

Ainsi, si un personne de $x = 35$ ans souhaite un contrat assurant le versement d'un capital de $K = 100000$ à son décès s'il survient avant qu'il n'ait 75 ans, et qu'il verse une prime constant jusqu'à ses 75 ans (au plus, il ne verse plus de prime s'il décède), alors la prime est donnée par

```
> (p <- 100000*Axn(TV8890,35,40,i=.04)/axn(TV8890,35,40,i=.04))
[1] 366.3827
```

On parle ici classiquement de *benefit premium*. On peut également calculer la provision mathématique associée à ce contrat, i.e. *benefit reserve*. On se placera dans le cas où $m = n$. La provision est donnée, à la date k, comprise entre 0 et n par

$${}_kV_x = K \cdot A_{x+k:\overline{n-k}|} - \pi \cdot \ddot{a}_{x+k:\overline{n-k}|}$$

(en écriture prospective).

```
> V <- Vectorize(function(k){100000*Axn(TV8890,35+k,40-k,i=.04)-
+ p*axn(TV8890,35+k,40-k,i=.04)})
> V(0:5)
[1]    0.0000  290.5141  590.8095  896.2252 1206.9951 1521.3432
```

La Figure 4.7 permet de visualiser l'évolution de la provision

```
> plot(0:40,c(V(0:39),0),type="b",ylab="provisions mathématiques",xlab="k")
```

4.6 Exercices

Exercise 4.6.1. *Le modèle de Gompertz suppose que la fonction de survie associée à une vie humaine pouvait s'écrire*

$$L_x = \kappa\gamma^{c^x}.$$

A partir des tables TV88-90 et TD88-90, et de ${}_{10}p_{50}$, ${}_{10}p_{60}$ *et* ${}_{10}p_{70}$, *proposer des estimateurs des paramètres* κ, c *et* γ.

Exercise 4.6.2. *On suppose que* $\mu_x = a+bc^x d^{x^2}$. *Construire une fonction permettant de calculer* ${}_kp_x$.

Exercise 4.6.3. *Montrer qu'il existe une relation de récurence sur les* $IA_{x:n\urcorner}$. *En utilisant l'algorithme présenté dans la Section 4.4, les calculer.*

Exercise 4.6.4. *On supposera que les durées de vie résiduelles ne sont plus indépendentes, mais que*

$${}_tp_{xy} = \mathbb{P}(T_x > t, T_y > t) = C({}_tp_x, {}_tp_y)$$

où C *est une copule. Pour les tables TV88-90 et TD88-90, et pour des assurés d'âge* $x = 40$ *et* $y = 45$, *tracer*

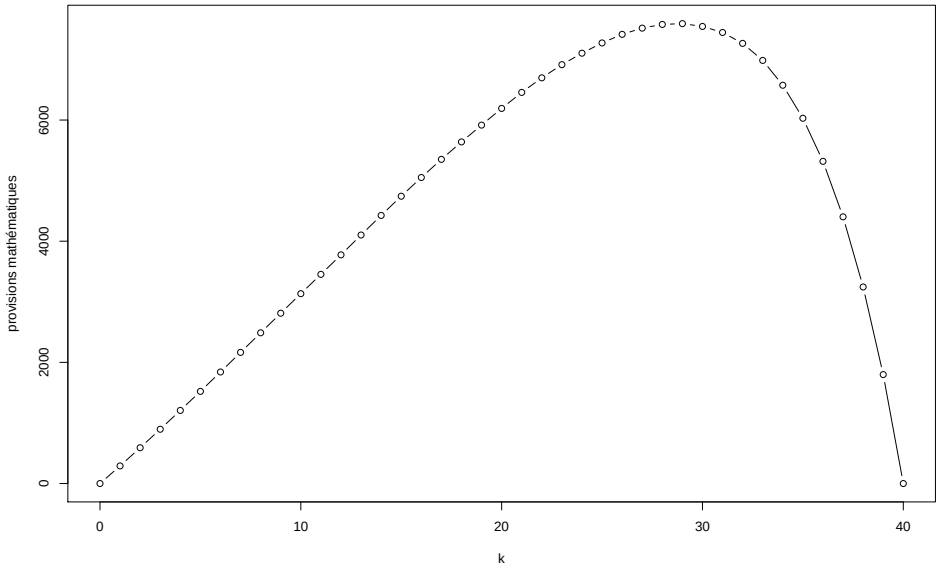


FIGURE 4.7 – Evolution de $h \mapsto {}_hp_{\overline{xy}}$ et $h \mapsto {}_hp_{xy}$ pour $x = 40$ et $y = 42$.

1. *la prime d'une rente de veuvage (versée entre le premier et le dernier décès, à terme échu) en fonction de θ où C_θ est une copule Gaussienne*
2. *la prime d'une rente de veuvage en fonction de θ où C_θ est une copule de Clayton*
3. *la prime d'une rente de veuvage en fonction de θ où C_θ est une copule de Gumbel*

Exercise 4.6.5. *Considérons une assurance de prêt : un individu d'âge x a emprunté un capital d'un montant C et s'est engagé à le rembourser en n annuitiés de montant ρ, payables à terme échu. On suppose qu'à la date de prise d'effet du contrat de prêt, il souscrit une assurance garantissant un remboursement des sommes restant dues si l'assuré décède avant d'avoir atteint l'âge $+n$. On notera t le taux d'intérêt du prêt (qui est a priori différent du taux d'actualisation i).*

1. *Exprimer t en fonction de C, de r et de n. Ecrire la fonction permettant de calculer ce taux.*
2. *On note C_k le capital restant dû à la fin de la $k$ème année, montrer que*

$$C_k = C - (r - tC)\frac{(1+t)^k - 1}{t}$$

Ecrire la fonction renvoyant le vecteur de ces capitaux $(C, C_1, \cdots, C_n)$.

3. *Montrer que la prime pure unique du contrat d'assurance s'écrit*

$$\pi = \sum_{k=1}^{n} {}_{k-1}p_x \cdot {}_1q_{x+k-1} C_{k-1} \frac{1}{(1+i)^k}$$

Écrire une fonction permettant de calculer cette prime en fonction de l'âge de l'assuré x, du taux d'actualisation i, de la table de mortalité retenue L, du capital C, de la durée du prêt n et du taux du prêt t.

4. *En supposant que la prime d'assurance soit payée annuellement (et est constante dans le temps), pendant m années ($1 \leq m \leq n$), et en notant que la prime annuelle s'écrit $\pi/_m\ddot{a}_x$, calculer la provision mathématique par une des trois méthodes (prospective, rétrospective ou recursive).*

5. *En supposant que la prime d'assurance n'est pas plus constante dans le temps, mais proportionnelle au capital restant du (payée aux dates $0, 1, ..., n-1$) montrer que la prime est*

$$\pi_k = \frac{\pi C_k}{\sum_{j=0}^{n-1} {}_kp_x C_k (1+i)^{-k}}.$$

Ecrire une fonction renvoyant le vecteur des primes, et représenter graphique l'évolution de la provision mathématique.

Exercise 4.6.6. *Representer l'évolution des provisions mathématiques pour un contrat avec capital différé (de n années pour un assuré d'âge x) avec contre-assurance, au sens où l'assureur rembourse les primes en cas de décès avant l'échéance.*

Chapitre 5

Les tables prospectives

De même que le provisionnement (évoqué dans le chapitre 3) posait le problème de la dynamique de la vie des sinistres (dont le montant n'est pas connu le jour de la survenance du sinistre), les contrats d'assurance-vie sont liés à des probabilités de décès (ou de survie) dans un futur plus ou moins lointain. Ces calculs doivet donc faire intervenir un aspect temporel. Par exemple, lorsque nous écrivions la formule

$$ {}_{k+h}p_x = {}_hp_{x+k} \cdot {}_kp_x,$$

nous omettons le fait que les probabilités ne devraient pas être calculées à la même date. Si la personne est d'âge x à la date t, elle aura un âge $x+k$ à la date $t+k$. Par exemple, en notant en puissance l'année où la probabilité est calculée, on aurait

$$ {}_{25+25}p_x^{(2010)} = {}_{25}p_{x+25}^{(2035)} \cdot {}_{25}p_x^{(2010)},$$

ou

$$ {}_{35+15}p_x^{(2010)} = {}_{15}p_{x+35}^{(2045)} \cdot {}_{35}p_x^{(2010)}.$$

Si k est elevé, on imagine que les probabilités de survie doivent tenir compte des améliorations de santé, notamment les conditions de vie, les avancées en médecine. Pour des compléments théoriques sur les outils présentés ici, nous renvoyons à Pitacco et al. (2009), Denuit & Robert (2007) ou encore Cairns et al. (2008)

5.1 Les bases de données prospectives

Dans le cadre statique de l'assurance-vie, détaillé dans le Chapitre 4, toutes les grandeurs pouvaient être construites à partir des L_x, ou des ${}_1p_x$, où x était l'âge des individus. Ici, nous allons intégrer la dimension temporelle, en notant qu'une *table* de mortalité est construite à une date t. Aussi, formellement, on notera $L_{x,t}$ le nombre de personnes d'âge x en vie à la date t.

Les données que nous allons utilisées sont tirées du site internet http://www.mortality.org, et il s'agit de données françaises, avec respectivement la mortalité des femmes, des hommes, et de l'ensemble de la population, entre 1899 et 2005. Ici on dispose de $D_{x,t}$ le nombre de personnes décédées à l'âge x l'année t (la base `Deces`), et $E_{x,t}$ l'exposition (la base `Expo`). Un léger travail sur les données du site est nécessaire (car un âge `110+` existe dans la base est rend les âges non numériques),

```
> Deces$Age <- as.numeric(as.character(Deces$Age))
> Deces$Age[is.na(Deces$Age)] <- 110
```

```
> Expo$Age <- as.numeric(as.character(Expo$Age))
> Expo$Age[is.na(Expo$Age)] <- 110
```

Pour commencer, on peut visualiser l'évolution de la surface du taux de mortalité, afin de mieux comprendre la nécessité d'une analyse dynamique de la démographie, où

$$\mu_{x,t} = \frac{D_{x,t}}{E_{x,t}}.$$

L'évolution de cette surface est repésentée sur la Figure 5.1, avec $(x,t) \mapsto \log \mu_{x,t}$.

```
> MU <- Deces[,3:5]/Expo[,3:5]
> Ages  <- unique(Deces$Age)
> Annees  <- unique(Deces$Year)
> matriceMU <- matrix(MU[,3],length(Ages),length(Annees))
> persp(Ages[1:100],Annees,log(matriceMU[1:100,]), theta=-30,
+ xlab="Age",zlab="Taux de décès (log)")
```

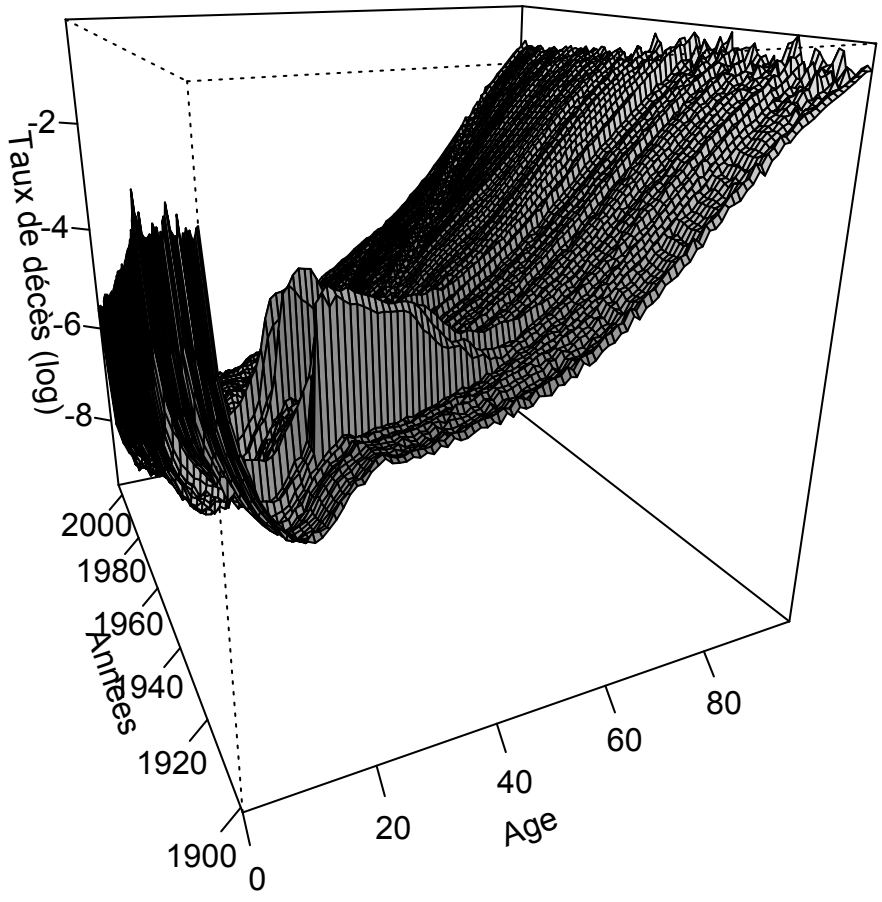


FIGURE 5.1 – Surface de mortalité $(x,t) \mapsto \log \mu_{x,t}$ pour les Hommes, en France, entre 1899 et 2005, et entre 0 et 110 ans.

5.1.1 La lecture longitudinale des tables

Ces données ne sont pas sous le format que nous avions vu dans le chapitre 4. Toutefois, on va pouvoir construire des fonctions proches de celles construites alors. On peut par exemple en placer l'année `an=1900` ou `an=2000` pour décrire la mortalité cette année là.

```
> mu.an <- function(a, pointille=1, cex=1.5){
+ Da <- Deces[Deces$Year==a,]
+ Ea <- Expo[Expo$Year==a,]
+ MUa <- Da[,3:5]/Ea[,3:5]
+ titre <- paste("Taux de mortalit\'e",a,sep=" ")
+ plot(Ages,log(MUa[,1]), type="l", xlab="Age", ylab="Taux de d\'ecès (log)",
+ main=titre, lwd=1.7, ylim=c(-9.8,.5), lty=pointille, cex=cex, cex.axis=cex,
+ cex.lab=cex, cex.main=cex)
+ lines(Ages,log(MUa[,2]),col="grey",lwd=1.7,lty=pointille)
+ legend(75,-6,c("Femmes","Hommes"),lty=pointille,lwd=1.7,
+ col=c("grey","black"),bty="n")
+ }
```

Cette petite fonction permet de tracer $x \mapsto \log \mu_{x,t}$ à t fixé, où $\mu_{x,t} = D_{x,t}/E_{x,t}$. La Figure 5.2, permet de comparer ces deux fonctions, en 1900 et en 2000.

Remark 5.1.1. *Il ne s'agit pas ici du suivi d'une cohorte, mais de l'étude de la mortalité pour des personnes d'âge différents (et nées à des périodes différentes) à une date t bien précise.*

```
> par(mfrow = c(1, 2))
> mu.an(1900)
> mu.an(2000)
> par(mfrow = c(1, 1))
```

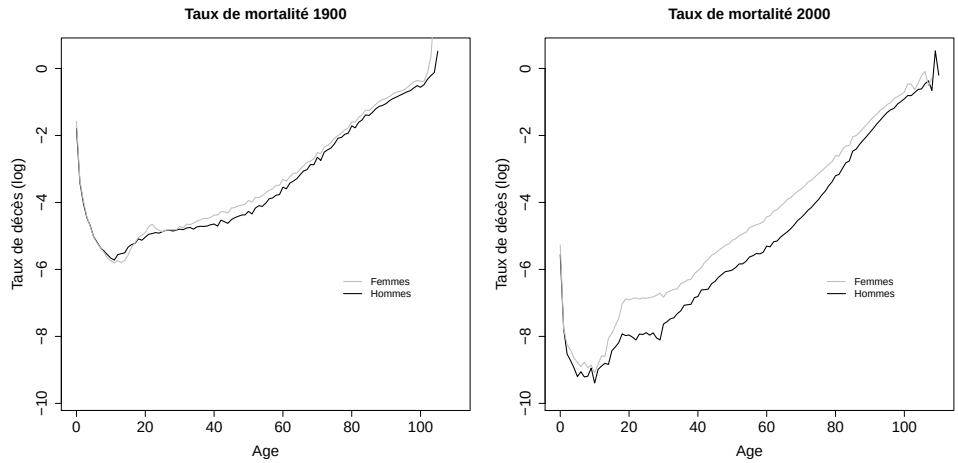


FIGURE 5.2 – Logarithmes des taux de mortalité $x \mapsto \log \mu_{x,t}$ pour les Hommes et les Fennes, en France, entre 0 et 110 ans, en 1900 à gauche, et en 2000 à droite.

Compte tenu du lien entre le taux de hasard et les fonctions de survie, on peut en déduire les fonctions de survie à la naissance (c'est à dire `x=0`). On utilise (comme dans le chapitre

précédant)

$$ {}_hp_{x,t} = \exp(-\int_x^{x+h} \mu_{s,t} ds). $$

Là encore, une fonction générique permettra de comparer des courbes à plusieurs dates.

```
> proba.survie <- function(x, a, cex=1.5){
+ Da <- Deces[Deces$Year==a,]
+ Ea <- Expo[Expo$Year==a,]
+ MUa <- Da[,3:5]/Ea[,3:5]
+ titrey <- paste("Probabilit\'e de survie à l'âge",x,"en",a,sep=" ")
+ titre <- paste("Probabilit\'e de survie en",a,sep=" ")
+ plot(1:length(Ages),exp(-cumsum(MUa[(x+1):length(Ages),2])), type="l", xlab="Age",
+ ylab=titrey, main=titre, lwd=1.7, ylim=c(0,1), cex=cex, cex.axis=cex, cex.lab=cex,
+ cex.main=cex)
+ lines(1:length(Ages),exp(-cumsum(MUa[(x+1):length(Ages),1])),col="grey",lwd=1.7)
+ legend(0,.2,c("Femmes","Hommes"),lty=1,lwd=1.7,col=c("grey","black"),bty="n")
+ }
```

La Figure 5.3, permet de comparer ces deux fonctions, en 1900 et en 2000.

```
> par(mfrow = c(1, 2))
> proba.survie(0,1900)
> proba.survie(0,2000)
> par(mfrow = c(1, 1))
```

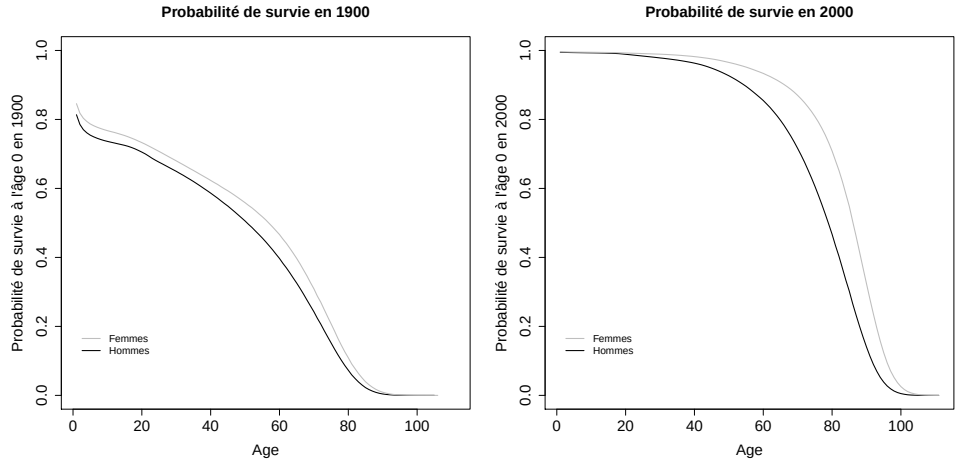


FIGURE 5.3 – Fonctions de survie à la naissance $h \mapsto {}_hp_{0,t}$ pour les Hommes - à gauche - et les Femmes - à droite - en France, entre 0 et 110 ans, entre 1900 (foncé) et 2000 (clair).

Enfin, la figure 5.4, permet de visualiser la *rectangularisation* des fonctions de survie.

```
> cex <- 1.5
>
> par(mfrow = c(1, 2))
> plot(Ages, prob.par.annee(1900, 2), type="l", xlab="Age",
+ ylab="Probabilit\'e de survie à la naissance", main="Mortalit\'e des hommes",
+ ylim=c(0,1), col=gray(1), xlim=c(0,120), cex=cex, cex.axis=cex, cex.lab=cex,
+ cex.main=cex)
```

```
> for(a in 1901:2000){
+ lines(Ages, prob.par.annee(a, 2), col=gray((a-1900)/100))
+ polygon(c(112,112,123,123),(c(a,a-1,a-1,a)-1900)/100, border=NA,
+ col=gray((a-1900)/100))
+ }
> for(a in seq(1900,2000,by=10)){
+ text(104,(a-1900)/100,a)
+ }
>
> plot(Ages, prob.par.annee(1900, 1), type="l", xlab="Age",
+ ylab="Probabilit\'e de survie à la naissance", main="Mortalit\'e des femmes",
+ ylim=c(0,1), col=gray(1), xlim=c(0,120), cex=cex, cex.axis=cex, cex.lab=cex,
+ cex.main=cex)
> for(a in 1901:2000){
+ lines(Ages, prob.par.annee(a, 1),col=gray((a-1900)/100))
+ polygon(c(112,112,123,123),(c(a,a-1,a-1,a)-1900)/100,border=NA,col=gray((a-1900)/100))
+ }
> for(a in seq(1900,2000,by=10)){
+ text(104,(a-1900)/100,a)
+ }
```

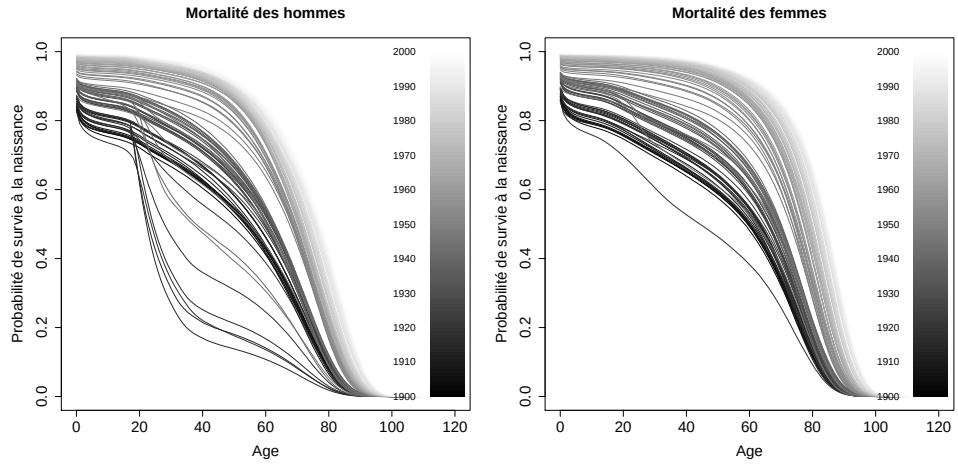


FIGURE 5.4 – Fonctions de survie à la naissance $h \mapsto {}_hp_0$ pour les Hommes et les Fennes, en France, entre 0 et 110 ans, en 1900 à gauche, et en 2000 à droite.

Pour alléger le calcul, on a une petite fonction auxiliaire qui extrait et calcul la probabilité de survie pour un sexe donné.

```
> prob.par.annee <- function(annee, sexe=1)
+ {
+  MUa <- subset(Deces, Year==annee)[, 3:5]/subset(Expo, Year==annee)[, 3:5]
+  exp(-cumsum(MUa[1:length(Ages), sexe]))
+ }
```

5.1.2 La lecture transversale des tables

En fait, cette lecture longitudinale des tables (bien que correspondant à ce que nous avions fait jusqu'à présent, et en particulier dans le chapitre précédant) ne paraît pas forcément très intéressante en assurance-vie, comme nous l'évoquions dans l'introduction. Aussi, afin de lire la fonction de survie pour un individu (ou une cohorte), on ne lit plus la base par année (ou par colonne dans une reprénsation matricielle $L_{x,t}$), mais suivant une diagonale (à $t-x$ constant). Il s'agit en effet de suivre un individu (ou ici une cohorte, par année de naissance) afin de valoriser un produit d'assurance-vie pour un individu (ou des individus de la même génération. Ces trois dimensions x (âge), t (date) et $t-x$ (année de naissance) n'est pas sans rappeler la lecture des triangles de provisionnement j (développement, ou âge d'un sinistre), $i+j$ (année calendaire, ou date de paiement) et i (année de survenance, ou année de naissance du sinistre). Aussi, afin de lire la fonction de survie pour un individu (ou une cohorte), on ne lit plus la base par année, mais suivant une diagonale (comme le suggèrait le diagramme de Lexis).

```
> Nannee <- max(Deces$Year)
> deces.trans <- function(naissance){
+ taille <- Nannee - naissance
+ Vage <- seq(0,length=taille+1)
+ Vnaissance <- seq(naissance,length=taille+1)
+ Cagreg <- Deces$Year*1000+ Deces$Age
+ Vagreg <- Vnaissance*1000+Vage
+ indice <- Cagreg %in% Vagreg
+ return(list(DecesT=Deces[indice,],ExpoT=Expo[indice,]))
+ }
> head(deces.trans(1950)$DecesT)
     Year Age   Female     Male    Total
5662 1950   0 18943.05 25912.38 44855.43
5774 1951   1  2078.41  2500.70  4579.11
5886 1952   2   693.20   810.32  1503.52
5998 1953   3   375.08   467.12   842.20
6110 1954   4   287.04   329.09   616.13
6222 1955   5   205.03   246.07   451.10
> tail(deces.trans(1950)$DecesT)
      Year Age Female Male Total
11262 2000  50   1051 2532  3583
11374 2001  51   1047 2702  3749
11486 2002  52   1246 2801  4047
11598 2003  53   1361 2985  4346
11710 2004  54   1371 3042  4413
11822 2005  55   1396 3217  4613
```

C'est à partir de cette extraction que l'on peut construire les mêmes types de graphiques qu'auparavant. Sur la Figure 5.5, on peut ainsi comparer l'impact sur le taux de mortalité d'une lecture tranversale. La fonction générique est ici

```
> mu.an.transv <- function(a,add=TRUE){
+ Da <-  deces.trans(a)$DecesT
+ Ea <-  deces.trans(a)$ExpoT
+ MUa  <-  Da[,3:5]/Ea[,3:5]
+ titre <- paste("Taux de mortalit\'e",a,sep=" ")
```

```
+ if(add==FALSE){plot(0:(nrow(MUa)-1),log(MUa[,1]),type="l",
+ xlab="Age",ylab="Taux de d\'ecès (log)",main=titre,lwd=1.7,
+ ylim=c(-9.8,.5))}
+ lines(0:(nrow(MUa)-1),log(MUa[,1]),type="l",lwd=1.7,ylim=c(-9.8,.5),lty=1)
+ lines(0:(nrow(MUa)-1),log(MUa[,2]),col="grey",lwd=1.7,lty=1)
+ legend(75,-7.5,c("Femmes","Hommes"),lty=1,lwd=1.7,
+ col=c("grey","black"),bty="n")
+ if(add==TRUE){text(90,-7.45,"Transversale");text(90,-6,"Longitudinale")}}
```

On peut alors comparer les taux de mortalité pour les personnes née en 1900 et celles née en 1950 (comme on travaille ici par cohorte, celle n'est en 1950 n'aura été observée que partiellement),

```
> par(mfrow = c(1, 2))
> mu.an(1900,pointille=2)
> mu.an.transv(1900,add=TRUE)
> mu.an(1950,pointille=2)
> mu.an.transv(1950,add=TRUE)
> par(mfrow = c(1, 1))
```

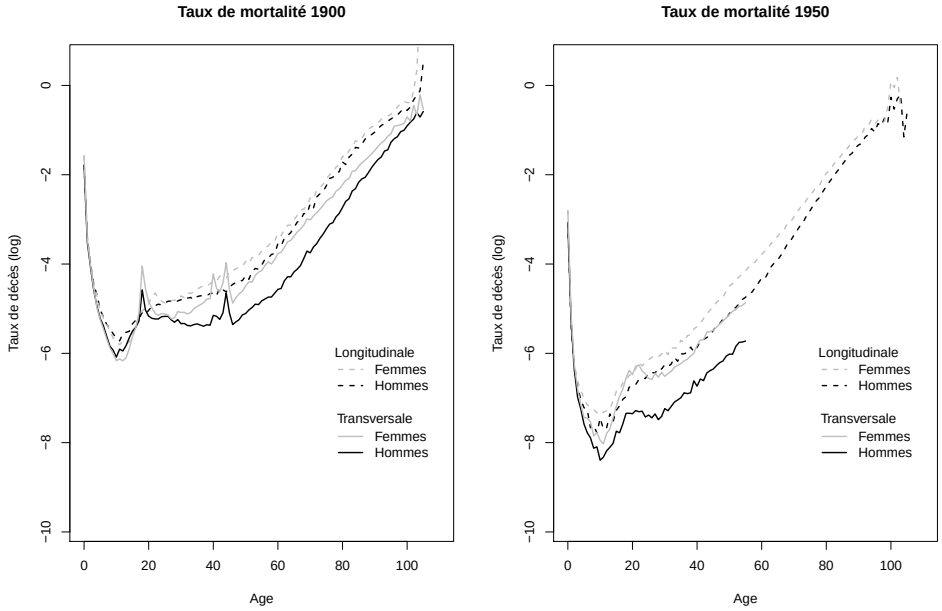


FIGURE 5.5 – Logarithmes des taux de décès : lecture transversale versus lecture longitudinale, pour une personne née en 1900 (à gauche) ou en 1950 (à droite).

Si la lecture transversale semble plus pertinante pour suivre une individu ou une cohorte, on est limité par le fait qu'il faudra prévoir les taux de mortalité pour les personnes les plus jeunes pour les années à venir. Les sections suivantes vont présenter la mise en oeuvre de plusieurs modèles permettant de prédire le taux de mortalité.

5.2 Le modèle de Lee & Carter

La modélisation retenue par Lee & Carter (1992) pour le taux instantané de mortalité est la suivante :

$$\log \mu_{xt} = \alpha_x + \beta_x k_t + \varepsilon_{xt},$$

avec les variables aléatoires ε_{xt} i.i.d. L'idée du modèle est donc d'ajuster à la série (doublement indicée par x et t) des logarithmes des taux instantanés de décès une structure paramétrique (déterministe) à laquelle s'ajoute un phénomène aléatoire ; le critère d'optimisation retenu va consister à maximiser la variance expliquée par le modèle, ce qui revient à minimiser la variance des erreurs. On retient en général les deux contraintes d'identifiabilité suivantes :

$$\sum_{x=x_m}^{x_M} \beta_x = 1 \quad \text{et} \quad \sum_{t=t_m}^{t_M} k_t = 0.$$

L'estimation des paramètres s'effectue en résolvant un problème de type "moindres carrés" :

$$\left(\hat{\alpha}_x, \hat{\beta}_x, k_t\right) = \arg\min \sum_{x,t} \left(\log \mu_{xt} - \alpha_x - \beta_x k_t\right)^2 .$$

5.2.1 La `library(demography)`

Le package `demography` propose une implémentation de Lee-Carter, avec en plus des fonctions permettant de projeter les taux de mortalité dans le futur. Dans un premier temps on prépare les données en vue de leur utilisation avec la fonction `lca`.

```
> library(forecast)
> library(demography)
> YEAR <- unique(Deces$Year);nC=length(Annees)
> AGE  <- unique(Deces$Age);nL=length(Ages)
> MUF  <- matrix(Deces$Female/Expo$Female,nL,nC)
> MUH  <- matrix(Deces$Male/Expo$Male,nL,nC)
> POPF <- matrix(Expo$Female,nL,nC)
> POPH <- matrix(Expo$Male,nL,nC)
```

On a alors les données prêtes à être transformées dans des données de `demography`,

```
> BASEH <- demogdata(data=MUH, pop=POPH, ages=AGE,
+ years=YEAR, type="mortality",
+ label="France", name="Hommes", lambda=1)
> BASEF <- demogdata(data=MUF, pop=POPF,ages=AGE,
+ years=YEAR, type="mortality",
+ label="France", name="Femmes", lambda=1)
```

Estimation des coefficients α_x, β_x et κ_t

On peut alors utiliser les fonctions de démographie, dont la fonction permettant d'estimer les paramètres du modèle de Lee-Carter. La Figure 5.6 permet ainsi de visualiser l'évolution de $x \mapsto \alpha_x$ et $x \mapsto \beta_x$

```
> par(mfrow = c(1, 2))
> LCH <- lca(BASEH)
> plot(LCH$age,LCH$ax)
```

```
> plot(LCH$age,LCH$bx)
> par(mfrow = c(1, 1))
```

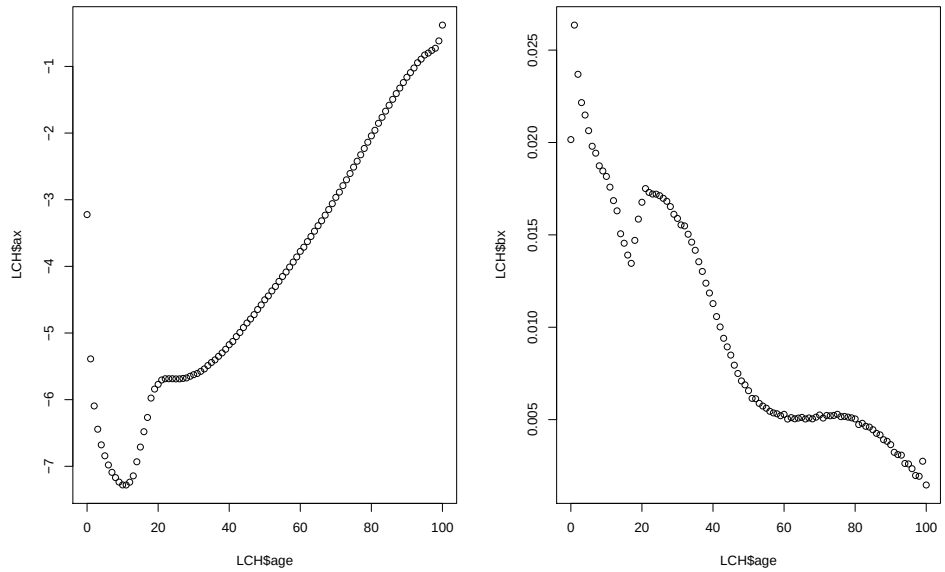


FIGURE 5.6 – Evolution de $x \mapsto \alpha_x$ (à gauche) et $x \mapsto \beta_x$ (à droite) .

Projection des κ_t

Une fois l'ajustement réalisé sur les données disponibles, on peut réaliser des projections de la mortalité future. En particulier, `library(forecast)` propose de nombreuses fonctions possibles pour prédire les valeurs κ_t futures.

Par exemple, les méthodes de lissage exponentiel,

```
> Y <- LCH$kt
> (ETS <- ets(Y))
ETS(A,N,N)

Call:
 ets(y = Y)

  Smoothing parameters:
    alpha = 0.8923

  Initial states:
    l = 71.5007

  sigma:  12.3592

     AIC     AICc      BIC
1042.074 1042.190 1047.420
```

```
> (ARIMA <- auto.arima(Y,allowdrift=TRUE))
Series: Y
ARIMA(0,1,0) with drift

Coefficients:
        drift
      -1.9346
s.e.   1.1972

sigma^2 estimated as 151.9:  log likelihood=-416.64
AIC=837.29   AICc=837.41   BIC=842.62
```

Graphiquement, il est alors possible de visualiser les prédictions obtenues pour ces deux modèles, avec respectivement un lissage exponentiel, et une marche aléatoire (ARIMA(0,1,0)) avec une tendance linéaire, comme le montre la Figure 5.7

```
> par(mfrow = c(1, 2))
> plot(forecast(ETS,h=100),type="p",ylim=c(-560,120))
> plot(forecast(ARIMA,h=100),type="p",ylim=c(-560,120))
> par(mfrow = c(1, 1))
```

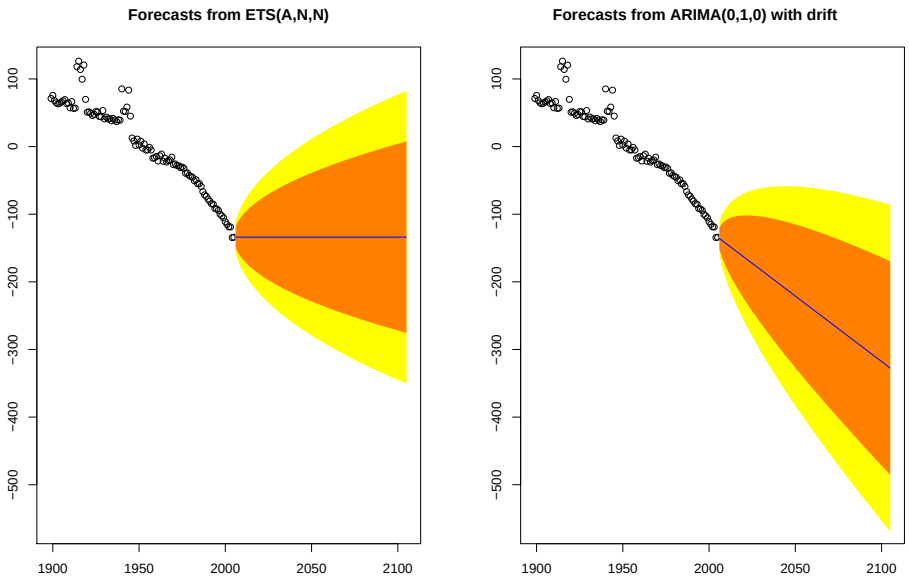


FIGURE 5.7 – Projection des κ_t du modèle de Lee-Carter par un modèle de lissage exponentiel (à gauche) et une marche aléatoire avec une tendance linéaire (à droite) .

Le modèle initial de Lee-Carter proposait de considérer un processus ARMA(1,1) sur la série différenciée (une fois),

$$\Delta\kappa_t = \phi\Delta\kappa_{t-1} + \delta + u_t - \theta u_{t-1}$$

où $\Delta\kappa_t = \kappa_t - \kappa_{t-1}$, i.e. un processus ARIMA(1,1,1). Mais il est aussi possible (et c'est ce qui avait été retenu ici) d'utiliser un processus ARIMA autour d'une tendance linéaire

$$\kappa_t = \alpha + \beta t + \phi\kappa_{t-1} + u_t - \theta u_{t-1}.$$

Restriction des données à la période après guerre

La volatilité de la prédiction semble venir de la prise en compte des deux séries de sauts des coefficients κ_t correspondant à la surmortalité pendant les deux guerres mondiales, 1914-1918 et 1939-1945 (avec également l'épisode de grippe espagnole en 1918).

```
> LCH0=lca(BASEH,years=1948:2005)
> Y0 <- LCH0$kt
> Ys <- Y[((length(Y)-length(Y0)):length(Y))]
> Y0s <- (Y0-mean(Y0))/sd(Y0)*sd(Ys)+mean(Ys)
> (ARIMA0 <- auto.arima(Y0s,allowdrift=TRUE))
Series: Y0s
ARIMA(1,1,0) with drift

Coefficients:
          ar1    drift
      -0.5417  -2.4717
s.e.   0.1180   0.3834

sigma^2 estimated as 19.64:  log likelihood=-165.92
AIC=337.84   AICc=338.29   BIC=343.96
```

En se restraignant à la période après guerre, le meilleur modèle ARIMA - autour de la tendance linéaire - continu à être intégré ($d = 1$), mais la volatilité du bruit blanc est ici beaucoup plus faible que sur le jeu de données incluant les deux guerres. Graphiquement, les prédictions peuvent se comparer sur la Figure 5.8

```
> par(mfrow = c(1, 2))
> plot(forecast(ARIMA,h=100),type="p",ylim=c(-560,120),xlim=c(1900,2100))
> plot(forecast(ARIMA0,h=100),type="p",ylim=c(-560,120),xlim=c(1900,2100))
> abline(v=1948,lty=2)
> par(mfrow = c(1, 1))
```

On peut également comparer les estimateurs des coefficients α et β sur les deux jeux de données, comme sur la Figure 5.9, avec en trait plein les estimations sur les données après guerre et en grisé les coefficients précédants,

```
> par(mfrow = c(1, 2))
> plot(LCH$age,LCH$ax,col="grey",ylim=range(LCH0$ax))
> lines(LCH0$age,LCH0$ax,lwd=2)
> plot(LCH$age,LCH$bx,col="grey")
> lines(LCH0$age,LCH0$bx,lwd=2)
> par(mfrow = c(1, 1))
```

Projection de différentes quantités actuarielles

Pour commencer, le plus simple est de regarder l'évolution de l'espérance de vie en 2005 pour une personne d'âge x, que l'on peut visualiser sur la Figure 5.10

```
> LCHf<-forecast(LCH,h=100)
> LCHT<-lifetable(LCHf)
> LCHTu<-lifetable(LCHf,"upper")
> LCHTl<-lifetable(LCHf,"lower")
```

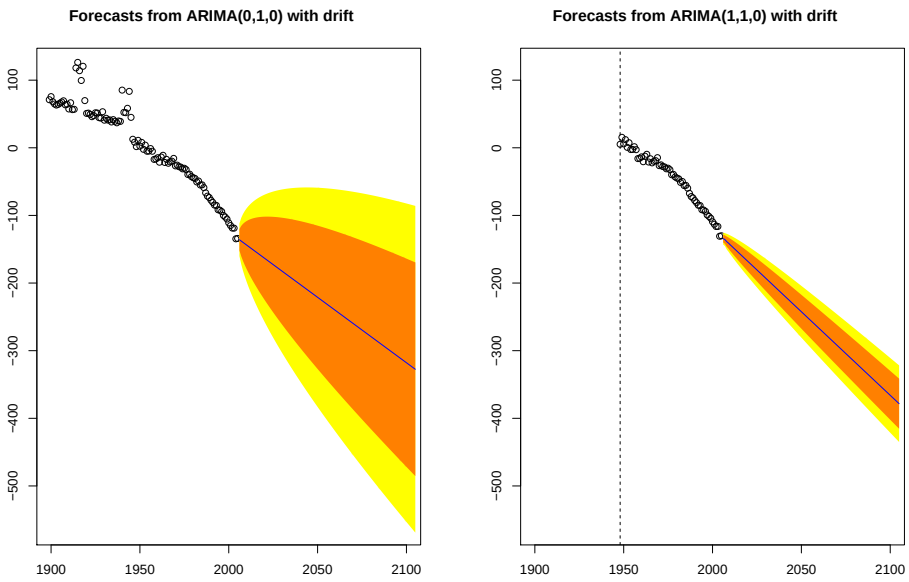


FIGURE 5.8 – Projection des κ_t du modèle de Lee-Carter par un modèle de marche aléatoire avec une tendance linéaire avec les données complètes (à gauche) et les données après guerre (à droite).

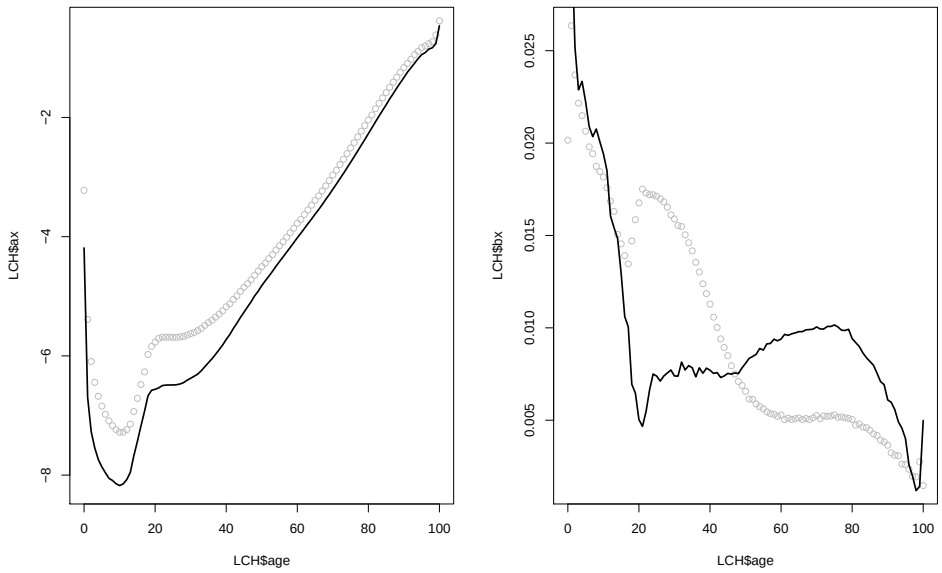


FIGURE 5.9 – Evolution de $x \mapsto \alpha_x$ (à gauche) et $x \mapsto \beta_x$ (à droite), avec l'estimation sur les données après guerre en noir, et sur le XXème siècle en grisé.

```
> plot(0:100,LCHT$ex[,5],type="l",lwd=2,main="Esp\'erance de vie en 2005",
+ ylab="Esp\'erance de vie r\'esiduelle",xlab="Age")
```

```
> polygon(c(0:100,100:0),c(LCHTu$ex[,5],rev(LCHTl$ex[,5])),
+ border=NA,col="grey")
> lines(0:100,LCHT$ex[,5],type="l",lwd=2)
```

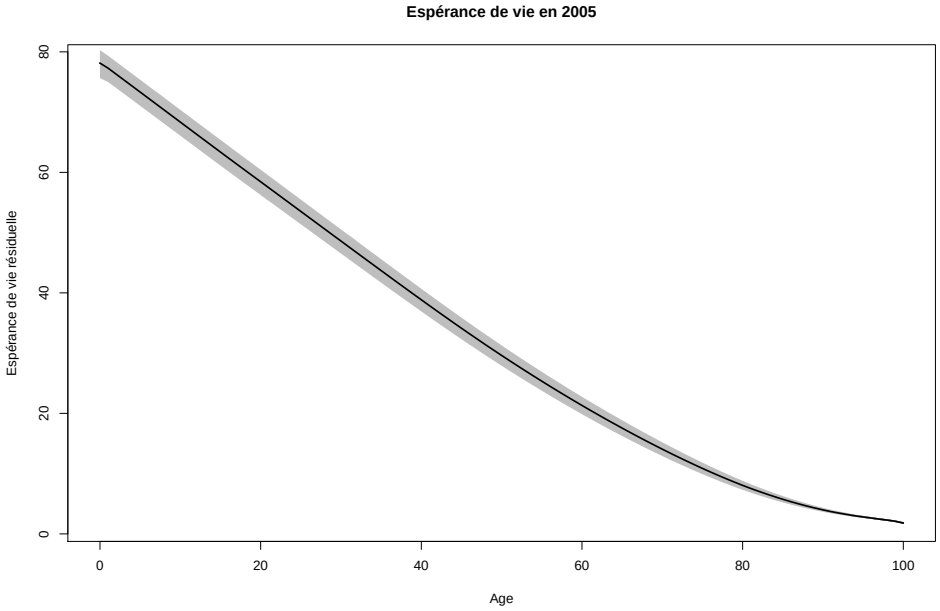


FIGURE 5.10 – Espérance de vie résiduelle à l'âge x, en 2005.

Les résidus du modèle

Dans le modèle de Lee-Carter, nous avions

$$\log \mu_{x,t} = \alpha_x + \beta_x \cdot \kappa_t + \varepsilon_{x,t},$$

où les résidus $\varepsilon_{x,t}$ sont supposés i.i.d. Notons $\widehat{\varepsilon}_{x,t}$ les pseudo-résidus obtenus lors de l'estimation, i.e.

$$\widehat{\varepsilon}_{x,t} = \log \mu_{x,t} - \left(\widehat{\alpha}_x + \widehat{\beta}_x \cdot \widehat{\kappa}_t\right).$$

Il est important de vérifier que les résidus peuvent être considérés comme i.i.d. On peut visualiser les erreurs $\widehat{\varepsilon}_{x,t}$ en fonction de x sur la Figure 5.11 et de t sur la Figure 5.12.

```
> RES<-residuals(LCH)
> couleur<-gray(seq(0,1,by=1/length(RES$x)))
> plot(rep(RES$y,length(RES$x)),RES$z,col=
+ couleur[rep(RES$x-RES$x[1]+1,each=length(RES$y))],
+ xlim=c(0,120),ylim=c(-1.62,1.62),
+ xlab="Age",ylab="")
> for(a in 1901:2000){
+ polygon(c(112,112,123,123),(c(a,a-1,a-1,a)-1900)/
+ 100*3-1.5,border=NA,col=gray((a-1900)/100))}
> for(a in seq(1900,2000,by=10)){
+ text(106,(a-1900)/100*3-1.5,a)}
```

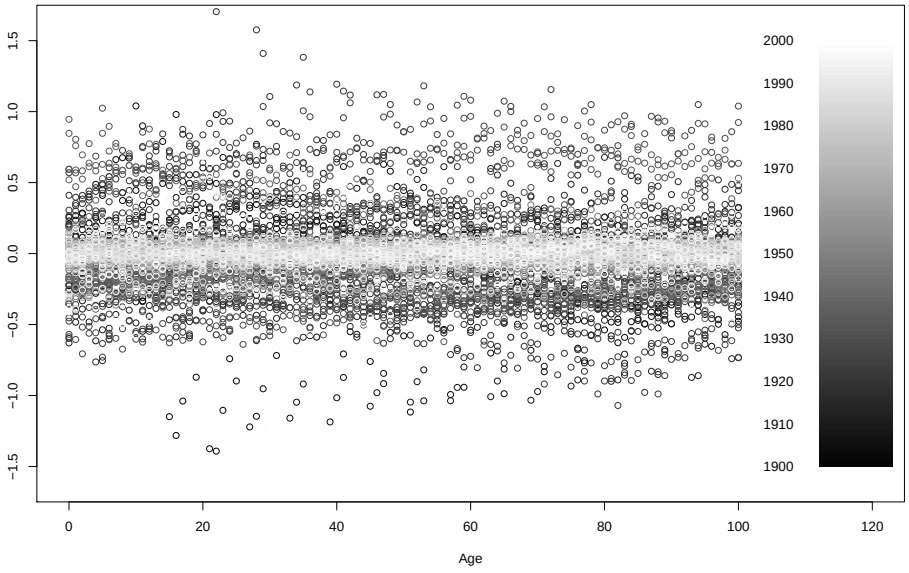


FIGURE 5.11 – Visualisation des pseudo-résidus, $x \mapsto \widehat{\varepsilon}_{x,t}$.

Pour l'évolution des résidus en fonction de t, le code est :

```
> couleur=gray(seq(0,1,by=1/length(RES$y)))
> plot(rep(RES$x,each=length(RES$y)),RES$z,col=
+ couleur[rep(RES$y-RES$y[1]+1,length(RES$x))],
+ xlim=c(1899,2020),ylim=c(-1.62,1.62),
+ xlab="Ann\'ee",ylab="")
> for(a in 1:110){
+ polygon(c(2012,2012,2023,2023),(c(a,a-1,a-1,a))/
+ 110*3-1.5,border=NA,col=gray(a/110))}
> for(a in seq(0,110,by=10)){
+ text(2009,a/100*3-1.5,a)}
```

5.2.2 Les fonctions de LifeMetrics

Le package LifeMetrics [1] proposé par JP Morgan propose une implémentation simple à mettre en oeuvre du modèle de Lee-Carter et de certaines variantes (notamment avec la prise en compte de cohortes).

Une fois le script chargé (via l'instruction `source("fitModels.r")`), il suffit de passer en paramètres deux tableau `etx` et `dtx` de dimensions (nombre d'années) $\times$ (nombre d'âges) contenant respectivement les expositions au risque l'année `t` à l'âge `x` et le nombre de décès. L'ajustement s'effectue par l'appel :

```
> res=fit701(x, y, etx, dtx, wa)
```

où `x` est une vecteur contenant les âges, `y` les années et `wa` est une matrice de poids (non utilisée dans le modèle standard, il suffit de la passer avec `wa=1`. On reprend ici l'exemple utilisé à la

1. Les codes sont en ligne sur http://www.jpmorgan.com/pages/jpmorgan/investbk/solutions/lifemetrics/software.

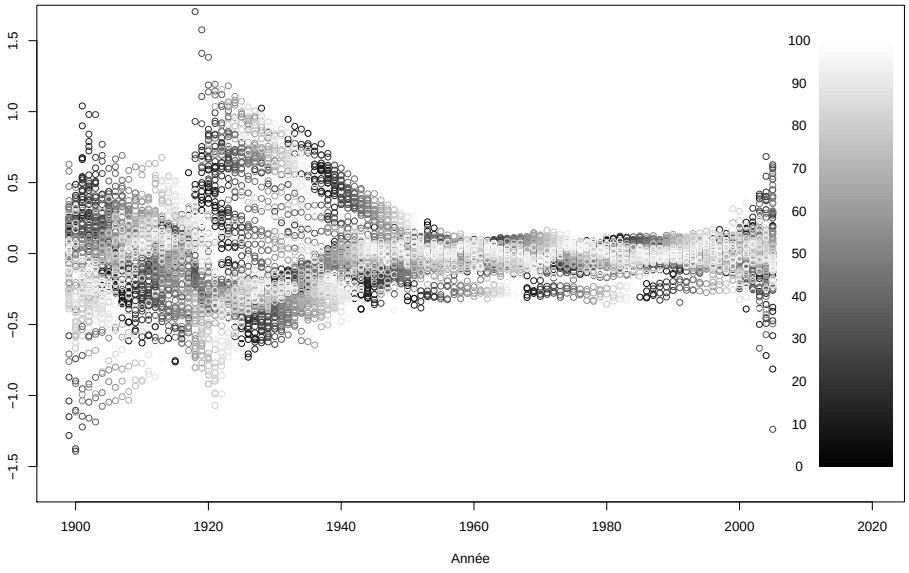


FIGURE 5.12 – Visualisation des pseudo-résidus, $t \mapsto \widehat{\varepsilon}_{x,t}$.

section précédente, pour lesquels on calcule les logarithmes des taux de décès instantanés (pour l'année `an=1986`) :

```
> source("fitModels.r")
> Deces <- Deces[Deces$Age<90,]
> Expo <- Expo[EXPOSURE$Age<90,]
> XV <- unique(Deces$Age)
> YV <- unique(Deces$Year)
> ETF <- t(matrix(Expo[,3],length(XV),length(YV)))
> DTF <- t(matrix(Deces[,3],length(XV),length(YV)))
> ETH <- t(matrix(Expo[,4],length(XV),length(YV)))
> DTH <- t(matrix(Deces[,4],length(XV),length(YV)))
> WA <- matrix(1,length(YV),length(XV))
> LCF  <-  fit701(xv=XV,yv=YV,etx=ETF,dtx=DTF,wa=WA)
> LCH  <-  fit701(xv=XV,yv=YV,etx=ETH,dtx=DTH,wa=WA)
```

On peut ainsi comparer les coefficients α_x et β_x entre les hommes et les femmes, comme sur la Figure 5.13

```
>  par(mfrow = c(1, 2))
>  plot(LCF$x,LCF$beta1,type="l",xlab="Age")
>  lines(LCH$x,LCH$beta1,col="grey")
>  legend(40,-6,c("Femmes","Hommes"),lty=1,
+  lwd=1,col=c("grey","black"),bty="n")
>  plot(LCF$x,LCF$beta2,type="l",xlab="Age")
>  lines(LCH$x,LCH$beta2,col="grey")
>  legend(40,.022,c("Femmes","Hommes"),lty=1,
+ lwd=1,col=c("grey","black"),bty="n")
```

```
> par(mfrow = c(1, 1))
```

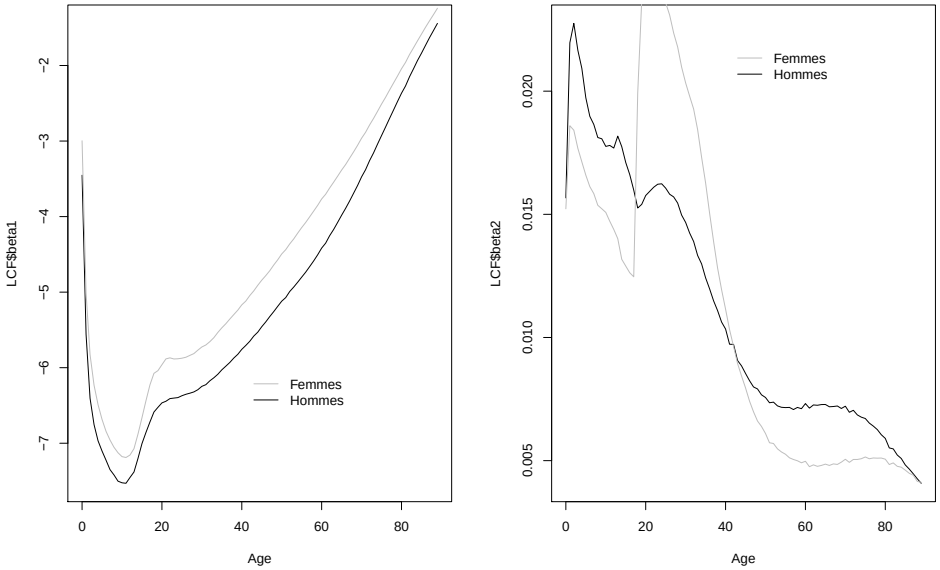


FIGURE 5.13 – Evolution de $x \mapsto \alpha_x$ (à gauche) et $x \mapsto \beta_x$ (à droite), pour les Hommes - en trait sombre - et pour les Femmes - en trait grisé.

Il est aussi possible d'estimer les coefficients κ_t sur la période passée, que l'on peut visualiser sur la Figure 5.14

```
> plot(LCF$y,LCF$kappa2,type="l",xlab="Ann\'ee")
> lines(LCH$y,LCH$kappa2,col="grey")
```

Notons que plusieurs fonctions sont proposées ici, correspondant soit à des

- $\log \mu(x,t) = \beta_x^{(1)} + \beta_x^{(2)}\kappa_t^{(2)}$,
- $\log \mu(x,t) = \beta_x^{(1)} + \beta_x^{(2)}\kappa_t^{(2)} + \beta_x^{(3)}\gamma_{t-x}^{(3)}$,
- $\log \mu(x,t) = \beta_x^{(1)} + \kappa_t^{(2)} + \gamma_{t-x}^{(3)}$,
- $\text{logit} q(x,t) = \text{logit}(1 - e^{-\mu(x,t)}) = \kappa_t^{(1)} + (x-\alpha)\kappa_t^{(2)}$,
- $\text{logit} q(x,t) = \text{logit}(1 - e^{-\mu(x,t)}) = \kappa_t^{(1)} + (x-\alpha)\kappa_t^{(2)} + \gamma_{t-x}^{(3)}$.

5.2.3 La `library(gnm)`

Les deux exemples ci-dessus s'appuyaient sur des implémentations (directes) du modèle de Lee-Carter. Avec des algorithmes optimisés pour estimer les coefficients α_x, β_x et κ_t. Mais on peut effectuer l'estimation des paramètres du modèle en s'appuyant sur sa variante log-Poisson, qui conduit formellement à mettre en oeuvre un modèle linéaire généralisé. Ou plutôt *non*linéaire car les facteurs interviennent sous la forme $\alpha_x + \beta_x \cdot \kappa_t$, que ne peut pas se mettre sous une forme linéaire. On peut donc utiliser la `library(gnm)`, et lancer une régression à l'aide d'un outil plus général.

```
> library(gnm)
> Y <- Deces$Male
> E <- Expo$Male
```

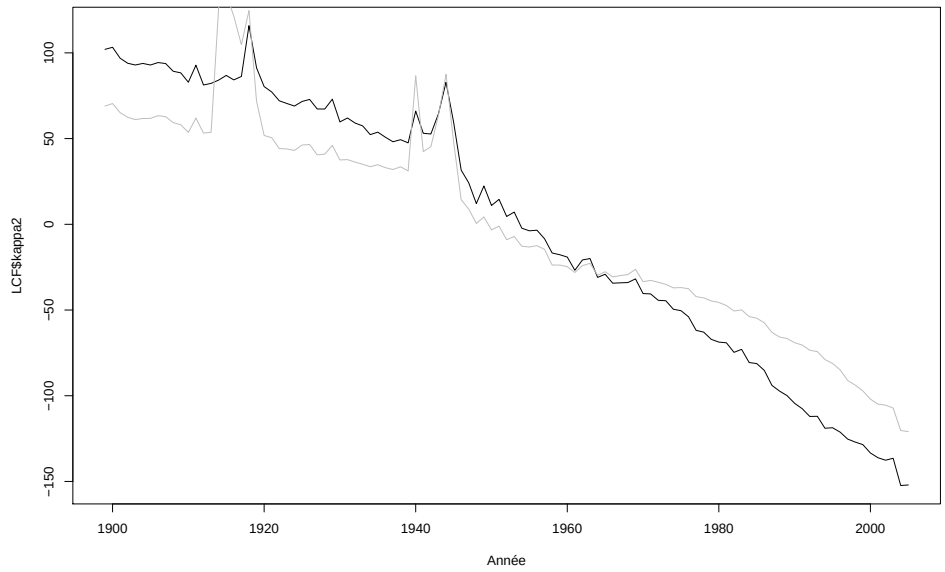


FIGURE 5.14 – Evolution de $t \mapsto \kappa_t$ pour les Hommes - en trait sombre - et pour les Femmes - en trait grisé.

```
>  Age <- Deces$Age
>  Year <- Deces$Year
>  I <- (Deces$Age<100)
>  base <- data.frame(Y=Y[I],E=E[I],Age=Age[I],Year=Year[I])
>  REG <- gnm(Y~factor(Age)+Mult((factor(Age)),factor(Year)),
+ data=base,offset=log(E),family=quasipoisson)
Initialising
Running start-up iterations..
Running main iterations.....................
Done
```

Comme il y a plus de 300 coefficients estimés, il convient d'aller chercher les α_x, les β_x et les κ_t au bon endroit.

```
> names(REG$coefficients[c(1:5,93:103)])
> nomvar <- names(REG$coefficients)
> nb3 <- substr(nomvar,nchar(nomvar)-3,nchar(nomvar))
> nb2 <- substr(nomvar,nchar(nomvar)-1,nchar(nomvar))
> nb1 <- substr(nomvar,nchar(nomvar),nchar(nomvar))
> nb <- nb3
> nb[substr(nb,1,1)=="g"]<- nb1[substr(nb,1,1)=="g"]
> nb[substr(nb,1,1)=="e"]<- nb2[substr(nb,1,1)=="e"]
> nb <- as.numeric(nb)
> I <- which(abs(diff(nb))>1)
```

Par exemple pour les coefficients α_x et β, le code R est le suivant, et les coefficients peut être visualisés sur la Figure 5.15

```
> par(mfrow = c(1, 2))
> plot(nb[2:I[1]],REG$coefficients[2:I[1]],xlab="Age")
> plot(nb[(I[1]+1):(I[2])],REG$coefficients[(I[1]+1):(I[2])],xlab="Age")
> par(mfrow = c(1, 1))
```

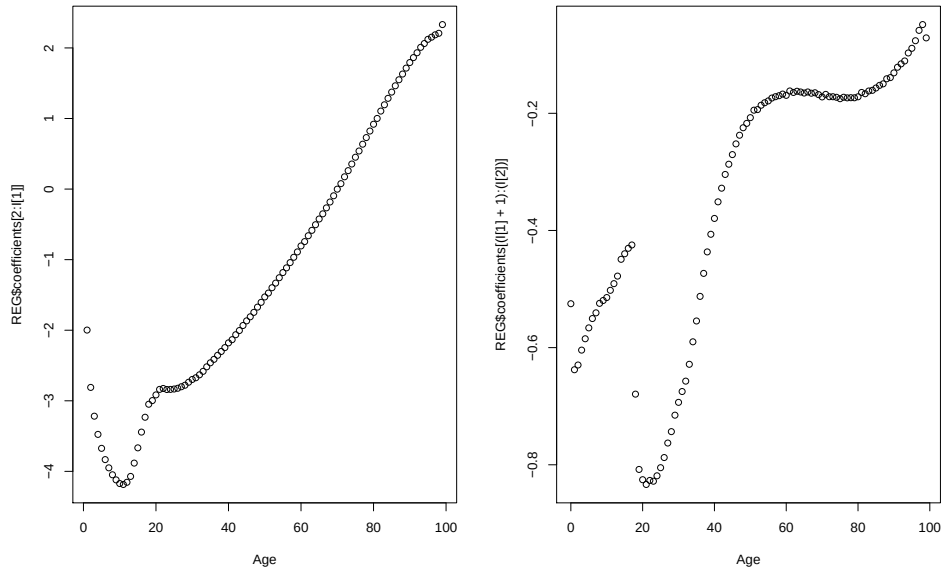


FIGURE 5.15 – Evolution de $x \mapsto \alpha_x$ (à gauche) et $x \mapsto \beta_x$ (à droite) pour les Hommes, en France.

On peut aussi visualiser les coefficients κ_t, comme sur la Figure 5.16

```
> plot(nb[(I[2]+1):length(nb)],REG$coefficients[(I[2]+1):length(nb)],
+ xlab="Ann\'ee",type="l")
```

Le code peut être un peu long à faire tourner, mais ce code permet d'implémenter n'importe quel modèle de démographie (nous présenterons une application dans la dernière section en introduisant un effet *cohorte*). De plus, cette fonction ne permet pas de prendre en compte les contraintes d'identifiabilité imposées avec les deux autres fonctions. D'où une estimation des κ opposée à celle obtenue avec les deux autres fonctions

5.2.4 Comparaison des trois algorithmes

Afin de faire une comparaison rapide, plaçons nous en un point particulier de la surface de mortalité, e.g. $x = 40$ et $t = 1980$. Les trois jeux d'estimateurs des coefficients sont les suivants

```
> x <- 40
> t <- 1980
> param <- matrix(NA,3,3)
> param[1,] <- c(LCH.lca$ax[LCH.lca$age==x],
+ LCH.lca$bx[LCH.lca$age==x],
+ LCH.lca$kt[LCH.lca$year==t])
> param[2,] <- c(LCH.fit701$beta1[LCH.fit701$x==x],
```

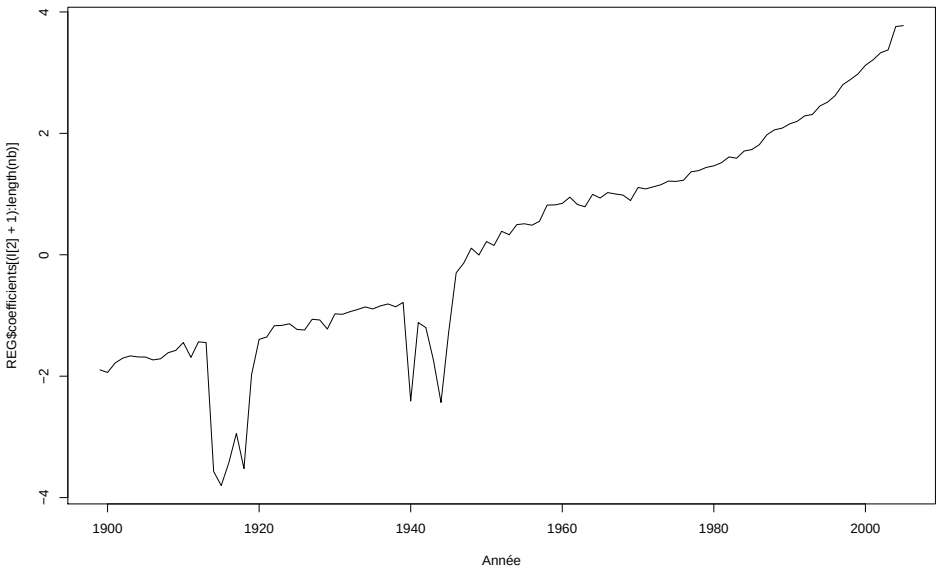


FIGURE 5.16 – Evolution de $t \mapsto \kappa_t$ pour les Hommes, en France.

```
+ LCH.fit701$beta2[LCH.fit701$x==x],
+ LCH.fit701$kappa2[LCH.fit701$y==t])
> param[3,] <- c(REG$coefficients[41]+
+ REG$coefficients[1],REG$coefficients[141],
+ REG$coefficients[282])
> param
          [,1]       [,2]        [,3]
[1,] -5.175210 0.01128062 -44.2225390
[2,] -5.168065 0.01114861 -45.5798063
[3,] -5.604863 0.55793271  -0.1244905
```

avec en ligne respectivement la fonction `lca`, la fonction `fit701` et la fonction `gnm`, et en colonne α_x, β_x et κ_t. Les deux premières fonctions utilisent la même containte sur les β_x, il est donc rassurant d'avoir les mêmes ordres de grandeurs :

```
> sum(LCH.lca$bx)
[1] 1
> sum(LCH.fit701$beta2)
[1] 1
```

Toutefois, si on compare les prédictions faites sur les taux de mortalité, les ordres de grandeurs sont comparables,

```
> exp(param[,1]+param[,2]*param[,3])
[1] 0.003433870 0.003426497 0.003433001
```

pour les trois modèles.

5.3 Utilisation du modèle de Lee-Carter projeté

A l'aide des techniques présentées auparavant, c'est à dire l'estimation des α_x, β_x, κ_t, et des κ_t projetśur le futur, il est possible de calculer d'autres quantités, dans un contexte de valorisation de produits d'assurance-vie.

5.3.1 Calcul des espérances de vie

Utilisons par exemple les sorties de la fonction `lca` de `library(demography)` pour calculer des estimations des taux de mortalité, ainsi que des projections pour le futur,

```
> LCH <- lca(BASEH)
> LCHf<-forecast(LCH,h=100)
> A <- LCH$ax
> B <- LCH$bx
> K1 <- LCH$kt
> K2 <- K1[length(K1)]+LCHf$kt.f$mean
> K <- c(K1,K2)
> MU <- matrix(NA,length(A),length(K))
> for(i in 1:length(A)){
+ for(j in 1:length(K)){
+ MU[i,j] <- exp(A[i]+B[i]*K[j])
+ }}
```

Au début du chapitre, nous avions visualisé la surface du taux de mortalité $\log \mu_{x,t}$ entre 1900 et 2005. Il est alors possible de visualiser en plus $\log \widehat{\mu}_{x,t}$ entre 2005 et 2105, comme sur la Figure 5.17

```
> persp(LCH$age,c(LCH$year,LCHf$year),log(MU),
+ xlab="Age",ylab="Ann\'ee",
+ zlab="Taux de d\'ecès (log)",theta=30)
```

On peut alors en déduire l'analogue dynamique des ${}_kp_x$, en $t = 2000$, en fonction de k (i.e. la fonction de survie de la durée de vie résiduelle)

```
> t <- 2000
> x <- 40
> s <- seq(0,99-x-1)
> MUd <- MU[x+1+s,t+s-1898]
> (Pxt <- cumprod(exp(-diag(MUd))))
 [1] 0.99838440 0.99663098 0.99469369 0.99248602 0.99030804 0.98782725 0.98504242
 [8] 0.98193715 0.97845243 0.97467199 0.97047250 0.96582764 0.96088832 0.95550220
[15] 0.94965857 0.94336539 0.93658314 0.92930373 0.92154725 0.91319233 0.90438349
[22] 0.89480210 0.88472880 0.87396961 0.86265381 0.85073003 0.83801863 0.82466285
[29] 0.81038237 0.79546804 0.77988277 0.76302933 0.74551160 0.72697144 0.70739380
[36] 0.68689788 0.66487519 0.64171557 0.61723877 0.59149492 0.56434547 0.53479039
[43] 0.50445361 0.47249581 0.43977367 0.40591799 0.37078337 0.33562397 0.29958914
[50] 0.26442814 0.22994614 0.19533606 0.16340038 0.13465331 0.10752312 0.08461961
[57] 0.06521622 0.04858994 0.03578809
```

On peut ainsi calculer les espérances de vie résiduelles pour des individus âgés de $x = 40$ ans, à différentes dates,

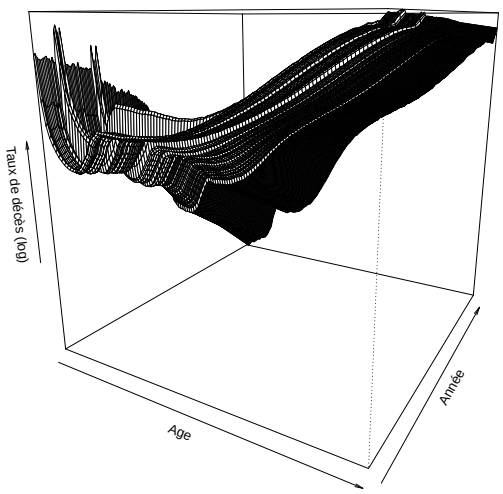


FIGURE 5.17 – Evolution de $(x,t) \mapsto \log \widehat{\mu}_{x,t}$ pour les Hommes, en France.

```
> x <- 40
> E <- rep(NA,150)
> for(t in 1900:2040){
+ s <- seq(0,90-x-1)
+ MUd <- MU[x+1+s,t+s-1898]
+ Pxt <- cumprod(exp(-diag(MUd)))
+ ext <- sum(Pxt)
+ E[t-1899] <- ext}
```

La Figure 5.18 (à gauche) permet de visualiser l'espérance de vie résiduelle à 40 ans, et son évolution au cours du temps (entre 1900 et 2050)

```
> plot(1900:2049,E,xlab="Ann\'ee",ylab="Esp\'erance de vie r\'esiduelle
+ (à 40 ans)",main="Esp\'erance de vie r\'esiduelle (à 40 ans)",type="l")
```

5.3.2 Valorisation de contrats d'assurance

On peut aussi valoriser des contrats d'assurance-vie. Considérons ainsi un individu qui souhaite une rente vie entière différée. On cherche alors la valeur actuelle probable du contrat achet'e par un assuré d'âge $x = 40$, qui souhaite toucher 1 (à terme échu) jusqu''a sa mort, à partir de $x + n = 70$ ans (i.e. différées de $n = 30$ ans).

```
> x <- 40
> r <- .035
> m <- 70
> VV <- rep(NA,141)
> for(t in 1900:2040){
+ s <- seq(0,90-x-1)
+ MUd <- MU[x+1+s,t+s-1898]
```

```
+ Pxt <- cumprod(exp(-diag(MUd)))
+ h <- seq(0,30)
+ V <- 1/(1+r)^(m-x+h)*Pxt[m-x+h]
+ VV[t-1899] <- sum(V,na.rm=TRUE)}
> plot(1900:2040,VV,xlab="Ann\'ee",ylab="",
+ main="VAP d'une rente vie entière",type="l")
>  par(mfrow = c(1, 1))
```

L'évolution du prix d'un tel contrat peut être visualisé sur la Figure 5.18

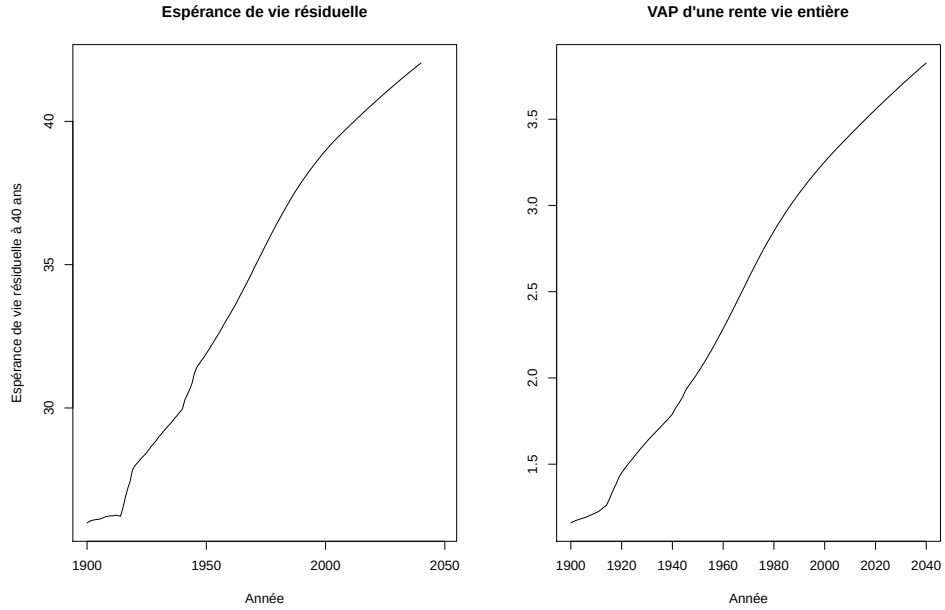


FIGURE 5.18 – Evolution de l'espérance de vie résiduelle pour les Hommes de 40 ans, en France, à gauche, et évolution de la valeur actuelle probable d'une rente vie entière différée achetée l'année t par un assuré de 40 ans.

Approche fonctionnelle des taux de mortalité

Les taux de mortalitéau peuvent être vues comme des fonctions.

```
> library(fts)
> rownames(MUH)=AGE
> colnames(MUH)=YEAR
> rownames(MUF)=AGE
> colnames(MUF)=YEAR
> MUH=MUH[1:90,]
> MUF=MUF[1:90,]
> MUHF=fts(x = AGE[1:90], y = log(MUH), xname = "Age",
+ yname = "Log Mortality Rate")
> MUFF=fts(x = AGE[1:90], y = log(MUF), xname = "Age",
+ yname = "Log Mortality Rate")
```

On peut aussi projeter les fonctions sur les deux deux premiers axes d'une analyse en composantes principales,

```
> par(mfrow = c(1, 2))
> fboxplot(data = MUHF, plot.type = "functional", type = "bag")
> fboxplot(data = MUHF, plot.type = "bivariate", type = "bag")
> par(mfrow = c(1, 1))
```

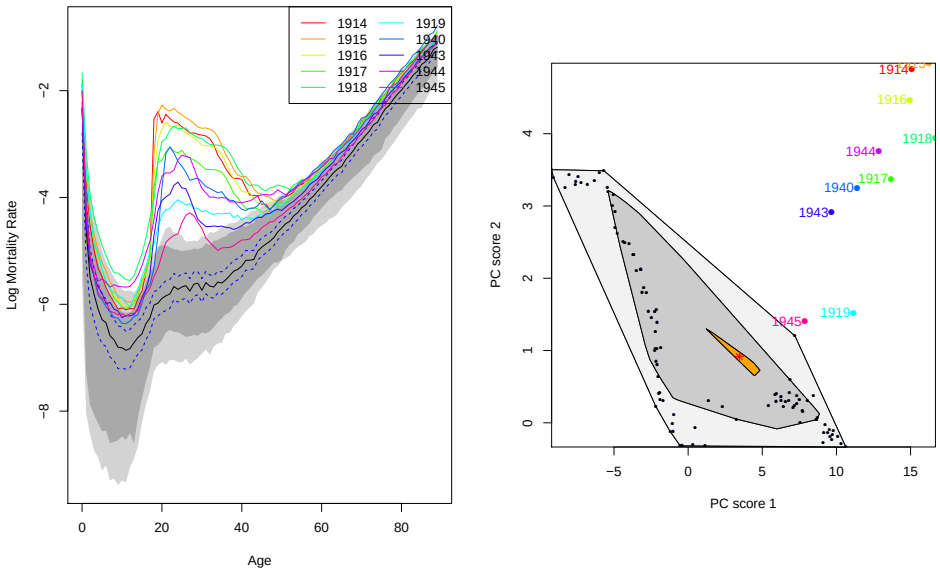


FIGURE 5.19 – Détection d'années 'aberrantes' dans le modèle de Lee-Carter.

5.4 Aller plus loin que le modèle de Lee-Carter

5.4.1 Prise en compte d'un effet cohorte

L'idée est ici de rajouter un nouveau terme dans le modèle de Lee-Carter, intégrant un effet cohorte, c'est à dire un terme dépendant de l'année de naissance $t - x$. On a ainsi

$$\log \mu_{x,t} = \alpha_x + \beta_x \cdot \kappa_t + \gamma_x \cdot \delta_{t-x} + \eta_{x,t},$$

en reprenant la modélisation proposée dans Renshaw & Haberman (2006).

A l'aide de la fonction `gnm` il est facile de rajouter autant de terme que l'on veut dans le modèle (à condition que le modèle soit identifiable, moyennant souvent quelques contraintes supplémentaires). Ici, on va donc créer un troisième facteur, en plus de l'âge x et de la date t,

```
> library(gnm)
>  Y <- Deces$Male
>  E <- Expo$Male
>  Age <- Deces$Age
>  Year <- Deces$Year
>  Cohorte <- Year-Age
```

```
>  I <- (Deces$Age<100)
>  base <- data.frame(Y=Y[I],E=E[I],Age=Age[I],Year=Year[I], Cohorte = Cohorte[I])
>  REG <- gnm(Y~factor(Age)+Mult((factor(Age)),factor(Year))+
+  Mult((factor(Age)),factor(Cohorte)),
+  data=base,offset=log(E),family=quasipoisson)
Initialising
Running start-up iterations..
Running main iterations.....................................................
Done
```

L'avantage est qu'il n'est pas nécessaire de projeter le coefficient de cohorte puisque l'on considère uniquement des projections pour des personnes qui pourraient acheter des contrats *aujourd'hui*, et dont la cohorte a pu être observée. Comme auparavant, il faut aller chercher les coefficients dans la sortie de la régression,

```
> nomvar <- names(REG$coefficients)
> nb3 <- substr(nomvar,nchar(nomvar)-3,nchar(nomvar))
> nb2 <- substr(nomvar,nchar(nomvar)-1,nchar(nomvar))
> nb1 <- substr(nomvar,nchar(nomvar),nchar(nomvar))
> nb <- nb3
> nb[substr(nb,1,1)=="g"]<- nb1[substr(nb,1,1)=="g"]
> nb[substr(nb,1,1)=="e"]<- nb2[substr(nb,1,1)=="e"]
> nb <- as.numeric(nb)
> I <- which(abs(diff(nb))>1)
```

On peut alors représenter l'ensemble des coefficients. Le coefficient α a la même allure qu'auparavant (ce qui est normal car il représente la mortalité *moyenne* par âge). En revanche, pour les coefficients liés au temps ou à la cohorte, on a les résultats suivants. La Figure 5.20 représente l'évolution des β_x et κ_t (respectivement à gauche et à droite),

```
> par(mfrow = c(1, 2))
> #plot(nb[2:I[1]],REG$coefficients[2:I[1]],xlab="Age")
> plot(nb[(I[1]+1):(I[2])],REG$coefficients[(I[1]+1):(I[2])],xlab="Age")
> plot(nb[(I[2]+1):(I[3])],REG$coefficients[(I[2]+1):(I[3])],xlab="Ann\'ee")
> par(mfrow = c(1, 1))
```

La Figure 5.21 représente l'évolution des coefficients γ_x et δ_{t-x} (respectivement à gauche et à droite),

```
> par(mfrow = c(1, 2))
> plot(nb[(I[3]+1):(I[4])],REG$coefficients[(I[3]+1):(I[4])],xlab="Age")
> plot(nb[(I[4]+1):length(nb)],REG$coefficients[(I[4]+1):length(nb)],
+ xlab="Ann\'ee (cohorte)",ylim=c(-5,3))
> par(mfrow = c(1, 1))
```

5.5 Exercices

Exercise 5.5.1. *A l'aide des modèles ajustés sur les données françaises, commentez l'affirmation "*tous les ans, on gagne un trimestre d'espérance de vie*".*

Exercise 5.5.2. *A l'aide des tables de mortalités Canadiennes* `CAN.Deces` *et* `CAN.Expo`*, calibrer un modèle de Lee-Carter, et comparer les espérances de vie à la naissance entre les Canadiens et les Français.*

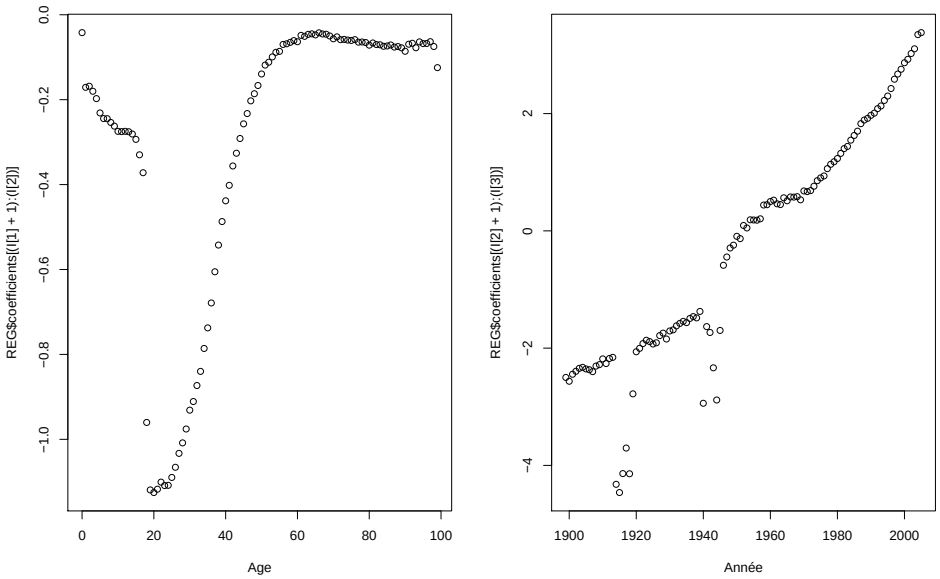


FIGURE 5.20 – Evolution des coefficients β_x et κ_t pour les Hommes en France dans le modèle avec un effet cohorte.

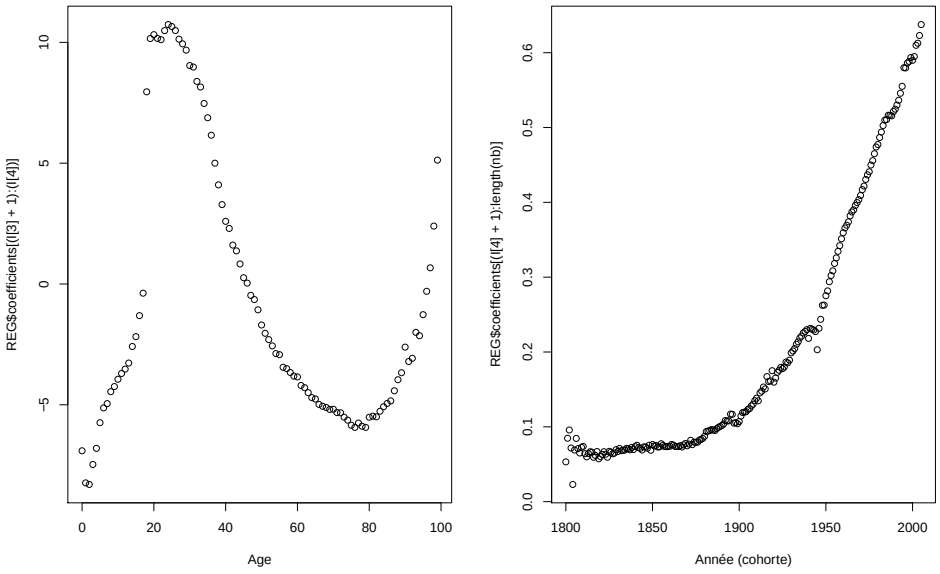


FIGURE 5.21 – Evolution des coefficients γ_x et δ_{t-x} pour les Hommes en France dans le modèle avec un effet cohorte.

Exercise 5.5.3. *A l'aide des tables de mortalités Japonaises* `JAP.Deces` *et* `JAP.Expo`*, calibrer un modèle de Lee-Carter, et comparer les espérances de vie à la naissance entre les Japonais et les Français. Comparer les probabilités d'atteindre 100 ans dans les deux pays.*

Exercise 5.5.4. *A l'aide des tables de mortalités Suisses* `CH.Deces` *et* `CH.Expo`*, calibrer un modèle de Lee-Carter, et comparer les espérances de vie à la naissance entre les Suisses et les Français.*

Exercise 5.5.5. *A l'aide des tables de mortalités Belges* `BEL.Deces` *et* `BEL.Expo`*, calibrer un modèle de Lee-Carter, et comparer les espérances de vie à la naissance entre les Belges et les Français.*

Exercise 5.5.6. *A l'aide des tables de mortalités Néo-Zélandaises* `NZM.Deces`*,* `NZM.Expo`*,* `NZNM.Deces` *et* `NZNM.Expo`*, calibrer deux modèles de Lee-Carter, sur la population Maori (*`NZM`*) et non-Maori (*`NZNM`*), et comparer les espérances de vie à la naissance.*

Annexe A

Annexes

A.1 Les lois de probabilités

A.1.1 Les lois continues

Traitons le cas où il existe une dérivée à la fonction de répartition appelée fonction de densité ou plus simplement densité. Il y a une infinité de fonctions qui peuvent et pourraient servir de densités à une variable aléatoire.

Le système de Pearson

Pearson (1895) a étudié ce sujet et a proposé une approche globale et unifiée à partir d'une équation différentielle. Une densité f serait solution de l'équation différentielle :

$$\frac{1}{f(x)}\frac{df(x)}{dx} = -\frac{a+x}{c_0 + c_1 x + c_2 x^2}. \tag{A.1}$$

Comme f doit représenter une densité, il faut que f soit positive sur D et normalisée $\int_D f(x)dx = 1$. Ceci impose des contraintes sur les coefficients a, c_0, c_1, c_2.
L'équation A.1 possède les cas particuliers suivants :
- type 0 : les coefficients c_1, c_2 sont nuls, alors on la solution de A.1 est

$$f(x) = Ke^{-\frac{(2a+x)x}{2c_0}}.$$

  On reconnait la loi normale.
- type I : le polynome $c_0 + c_1 x + c_2 x^2$ possède des racines réelles a_1, a_2 de signes opposées $a_1 < 0 < a_2$. Donc f a pour expression

$$f(x) = K(x - a_1)^{m_1}(a_2 - x)^{m_2},$$

  où $m_1 = \frac{a+a_1}{c_2(a_2-a_1)}$, $m_2 = -\frac{a+a_2}{c_2(a_2-a_1)}$ pour $x \in]-a_1, a_1[\cap]-a_2, a_2[$. On reconnait la loi Béta de première espèce. Si m_1 et m_2 sont du même signes alors f a une forme en U, sinon une forme en cloche.
- type II : Le type II correspond au cas où $m_1 = m_2 = m$.
- type III : si $c_2 = 0$ et $c_0, c_1 \neq 0$ alors le polynome $c_0 + c_1 x + c_2 x^2$ devient de premier degré. Par conséquent, f devient

$$f(x) = K(c_0 + c_1 x)^m e^{-x/c_1},$$

  pour $x \geq -\frac{c_0}{c_1}$ ou $x \leq -\frac{c_0}{c_1}$. On reconnaitra les lois gamma (incluant donc la loi exponentielle).

- type IV : le polynome $c_0 + c_1 x + c_2 x^2$ n'a pas de solutions réelles [1]. On peut néanmoins en déduire une expression pour f :

$$f(x) = K\left(C_0 + c_2(x + C_1)^2\right)^{-(2c_2)^{-1}} e^{-\frac{a-c_1}{\sqrt{c_2 c_0}} \tan^{-1}\left(\frac{x+c_1}{\sqrt{c_0/c_2}}\right)}.$$

  Barndoff-Nielsen utilise une approximation de l'expression supra pour obtenir la loi inverse Gaussienne généralisée.
- type V : si le polynome $c_0 + c_1 x + c_2 x^2$ est un carré parfait, alors l'expression de la densité est la suivante

$$f(x) = K(x + C_1)^{-1/c_2} e^{\frac{a-C_1}{c_2(x+C_1)}},$$

  pour $x \geq -C_1$ ou $x \leq -C_1$. Si le terme exponentiel s'annule alors on a le particulier $f(x) = K(x + C_1)^{-1/c_2}$, où $c_2 > 0$ ($c_2 < 0$) corresponds au type VIII (type IX respectivement).
- type VI : si le polynome $c_0 + c_1 x + c_2 x^2$ possède des racines réelles a_1, a_2 de même signe alors on obtient

$$f(x) = K(x - a_1)^{m_1}(x - a_2)^{m_2},$$

  pour $x \geq \max(a_1, a_2)$. Ceci corresponds à la loi Béta généralisée.
- type VII : enfin le type VII corresponds au cas "dégénéré" lorsque $c_1 = a = 0$. Ainsi la solution est

$$f(x) = K(c_0 + c_2 x^2)^{-(2c_2)^{-1}}.$$

  Le type VII corresponds à la loi Student et la loi de Cauchy.

Du système de Pearson, on peut construire toutes les autres lois continunes à l'aide de transformations "simples" : transformation linéaire, transformation puissance, transformation exponentielle ou logarithme (e.g. la loi log-normale).

Le package **PearsonDS** implémente les lois de probabilité selon le système de Pearson. Le code ci-dessous est un exemple très succint de graphiques. Sur la figure A.1, on observe des lois à supports bornés (Pearson I, II et VI), d'autres à supports positifs (Pearson III, V) ou sur $\mathbb{R}$ tout entier (Pearson 0, IV).

```
> library(PearsonDS)
> x <- seq(-1, 6, 0.001)
> y0 <- dpearson0(x, 2, 1/2)
> y1 <- dpearsonI(x, 1.5, 2, 0, 2)
> y2 <- dpearsonII(x, 2, 0, 1)
> y3 <- dpearsonIII(x, 3, 0, 1/2)
> y4 <- dpearsonIV(x, 2.5, 1/3, 1, 2/3)
> y5 <- dpearsonV(x, 2.5, -1, 1)
> y6 <- dpearsonVI(x, 1/2, 2/3, 2, 1)
> y7 <- dpearsonVII(x, 3, 4, 1/2)
> plot(x, y0, type="l", ylim=range(y0, y1, y2, y3, y4, y5, y7), ylab="f(x)",
> main="Système de Pearson",lty=1)
> lines(x[y1 != 0], y1[y1 != 0], lty=2)
> lines(x[y2 != 0], y2[y2 != 0], lty=3)
> lines(x[y3 != 0], y3[y3 != 0], lty=4)
> lines(x, y4, col="grey",lty=1)
> lines(x, y5, col="grey",lty=2)
> lines(x[y6 != 0], y6[y6 != 0], col="grey",lty=3)
```

1. il est toujours strictement positif et peut se réécrire $C_0 + c_2(x + C_1)^2$.

```
> lines(x[y7 != 0], y7[y7 != 0], col="grey",lty=4)
> legend("topright", leg=paste("Pearson", 0:7), lty=c(1:4,1:4),
+  col=c(rep("black",4),rep("grey",4)))
```

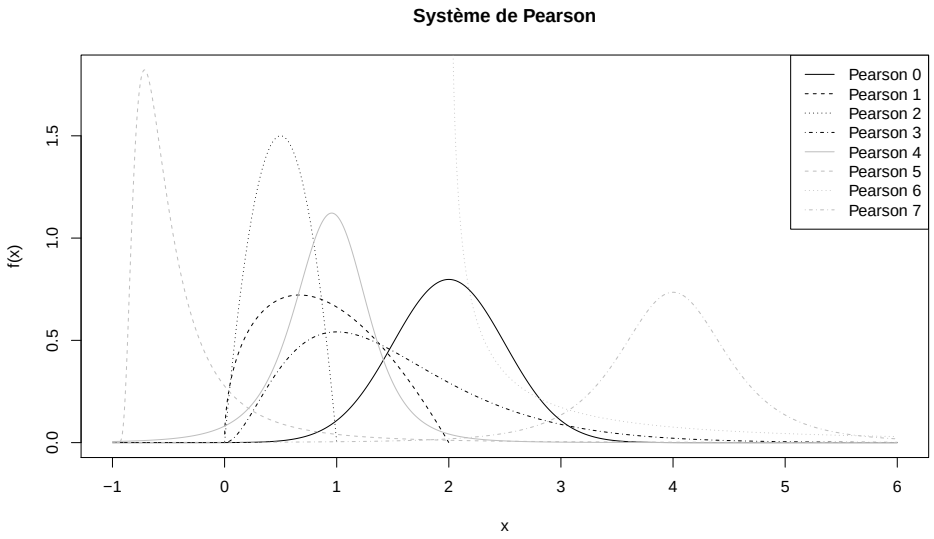


FIGURE A.1 – Système de Pearson et formes de principales densités.

A.2 Générateurs aléatoires

Les générateurs aléatoires ont montré un intérêt croissant de la part des scientifiques avec le développement des méthodes de Monte-Carlo, méthodes consistant à simuler n fois un modèle, un problème et d'en prendre la quantité empirique désirée (moyenne, quantile, etc...). Dans un premier temps, nous présentons la génération de nombres aléatoires de loi uniforme sur $[0, 1]$ et dans un second temps leur utilisation pour générer n'importe quelles lois.

A.2.1 Loi uniforme

A ses débuts, la génération de nombre aléatoire se faisait par une mesure de phénomènes physiques aléatoires, telles que le taux de radioactivité de sources nucléaires ou le bruit thermique de semi-conducteurs. Ces méthodes avaient un gros avantage à savoir générer des nombres parfaitement aléatoires mais souffraient d'un défaut majeur : leur cout en temps et en prix.

Avec le développement de l'ordinateur, les chercheurs mirent au point des algorithmes complètement déterministes pour générer une suite de nombres à partir d'un nombre initial (appelée graine, seed en anglais). Les nombres générés sur un ordinateur nous paraissent aléatoires seulement par ce que la graine est calculée à partir du temps machine (secondes et micro-secondes).

Dans la littérature, trois notions d'aléatoire sont à distinguer : les générateurs vraiment aléatoire (true randomness en anglais) liés à des mesures de phénomènes physiques, les générateurs pseudo-aléatoires (pseudo randomness) et les générateurs quasi-aléaoires (quasi randomness) qui sont des algorithmes déterministes.

Générateurs pseudo-aléatoires

Comme précisé dans L'Ecuyer (1990), un générateur aléatoire se caractérise par un ensemble d'états S, une loi de probabilité initiale μ sur S, une fonction de transition $f : S \mapsto S$, d'un ensemble de sortie $U \subset \mathbb{R}$ et d'une fonction de sortie $g : S \mapsto U$. D'un état initial s_0 donné par μ, on génére la suite d'états $s_n = f(s_{n-1})$ et de nombres réels $u_n = g(s_n)$.
Jusqu'au début des années 90, f était la fonction congruentielle $f(x) = (ax+c) \mod m$ et $S = \mathbb{N}$ et g la fonction proportion $g(x) = x/m$. Ainsi pour certains a, c, m bien choisis [2], on pouvait générer des entiers aléatoires sur entre 0 et 2^{32} et des réels sur 32 bits avec une période dépendant des paramètres a, c, m. Tout l'enjeu résidait dans le choix de ses paramètres de manière à maximiser la période [3].
Cette approche comporte des défauts à savoir un temps de calcul élevé [4] et une période courte (nombres d'états entre deux états identiques). Heureusement pour la science, Matsumoto & Nishimura (1998) publièrent le très célèbre générateur Mersenne-Twister, révolutionnaire sur deux points : son temps de calcul et sa période.
Les deux auteurs exploitèrent la structure binaire des ordinateurs à savoir que n'importe quel entier est representé par ω bits (e.g. $\omega = 32$) et que les opérations élémentaires sont extrêment peu couteuses.
La récurrence du $n + i$ème terme de MT est la suivante :

$$x_{i+n} = x_{i+m} \oplus (x_i^{upp} | x_{i+1}^{low}) A,$$

où $n > m$ sont des entiers constants, x_i^{upp} (respectivement x_i^{low}) désigne la partie supérieure (inférieure) $\omega - r$ (r) bits du terme x_i et A [5], une $\omega \times \omega$ matrice de $\{0, 1\}$. | est l'opérateur de concaténation, donc $x_i^{upp} | x_{i+1}^{low}$ concatène les $\omega - r$ bits supérieurs de x_i avec les r bits inférieurs de x_{i+1}.
Matsumoto & Nishimura (1998) ajoute une étape d'ajustement après chaque récurrence pour augementer l'équidistribution dans l'hypercube unité (voir l'article). Les auteurs fournissent un jeu de paramètres sélectionné de manière à maximiser la période et assurer une bonne équidistribution :
– $(\omega, n, m, r) = (32, 624, 397, 31)$,
– $a =$ 0x9908B0DF, $b =$ 0x9D2C5680, $c =$ 0xEFC60000,
– $u = 11$, $l = 18$, $s = 7$ et $t = 15$.
La période est de $2^{n\omega - r} - 1 = 2^{19937} - 1$, d'où le nom du générateur MT19937.
L'implémentation de MT19937 en C, disponible sur la page des auteurs [6], est très rapide du fait de l'utilisation d'opérations systématique bit à bit. D'autres générateurs ont depuis été inventé utilisant ce formalisme, notamment les générateurs WELL de L'Ecuyer et SFMT de Matsumoto.
MT19937 et ses extensions rentrent dans la catégorie des générateurs pseudo-aléatoires et sont utilisés dans les méthodes de Monte-Carlo. Par la loi des grands nombres, la moyenne empirique de l'échantillon $(X_1, \ldots, X_n)$ converge presque surement vers la moyenne théorique $E(X)$. Le théorème centrale limite nous donne la vitesse de convergence : $\frac{1}{\sqrt{n}}$.

2. Pour $m = 2^{31} - 1$, $a = 16807$ et $c = 0$, on obtient le générateur de Park-Miller d'une période de 2^{31}.
3. Voir le théorème de Knuth
4. L'opération modulo nécessite un grand nombre de opérations arithmétiques élementaires.
5. La matrice A est égale à $\begin{pmatrix} 0 & I_{\omega-1} \\ a & \end{pmatrix}$ où la multiplication à droite est faite par un décalage de bit à bit et une addition avec un entier a.
6. Téléchargeable à l'adresse http://www.math.sci.hiroshima-u.ac.jp/~m-mat/MT/emt.html

Par conséquent, on construit l'intervalle de confiance suivant :

$$\left[\overline{X}_n - \frac{1}{\sqrt{n}} S_n t_{n-1,\alpha}; \overline{X}_n + \frac{1}{\sqrt{n}} S_n t_{n-1,\alpha}\right],$$

où S_n la variance empirique débiaisée et $t_{n-1,\alpha}$ le quantile de la loi de Student à $n-1$ degré de liberté [7].
Dans R, le générateur aléatoire utilisé est MT19937 via la fonction `runif`. D'autres générateurs sont disponibles notamment Wichman-Hill, Knuth-TAOCP,...via la fonction `RNGkind`. De plus, le package **randtoolbox** implémentent des générateurs pseudo-aléatoires plus récents et **random** propose des variables vraiment aléatoires via le site http://www.random.org.

Générateurs quasi-aléatoires

Les méthodes de Monte-Carlo présentent un défaut : une convergence lente. Pour combler ce problème, deux approches sont envisagées soit par réduction de la variance soit par des méthodes quasi-aléatoires. Nous détaillerons dans cette section, les méthodes dites quasi-aléatoires.
Soient I^d l'hypercube unité de dimension d et f une fonction multivariée bornée et intégrable sur I^d. Les méthodes de Monte-Carlo consiste à approximer l'intégrale de f par

$$\int_{I^d} f(x)dx \approx \frac{1}{n}\sum_{i=1}^{n} f(X_i),$$

où $(X_i)_{1\leq i\leq n}$ sont des variables aléatoires independantes sur I^d. La loi des grands nombres nous assurent la convergence presque sûre de l'estimateur de Monte-Carlo. Et le théorème centrale limite nous précise que la vitesse de convergence est en $O(\frac{1}{\sqrt{n}})$.
La gross différence entre les méthodes pseudo Monte-Carlo et quasi Monte-Carlo (QMC) est de ne plus considérer les points $(x_i)_{1\leq i\leq n}$ comme réalisations de variables aléatoires mais comme points déterministes. Contrairement au tests statistiques, l'intégration numérique ne dépends pas sur le caractère aléatoire. Les méthodes QMC datent des années 50 pour des problèmes d'interpolation et de résolution d'intégrales.
Dans la suite, nous considérons les points $(u_i)_{1\leq i\leq n}$ de I^d comme déterministes. La condition de convergence de $\frac{1}{n}\sum_{i=1}^{n} f(u_i)$ vers $\int_{I^d} f(x)dx$ repose sur la bonne répartition des points dans l'hypercube I^d.
On dit que les points sont uniformément distribués si

$$\forall J \subset I^d, \lim_{n\to+\infty} \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}_J(u_i) = \lambda_d(J),$$

où λ_d désigne le volume en dimension d. Le problème est que ce critère est trop restrictif puisqu'il y aura toujours un sous ensemble de l'hypercube avec aucun points à l'intérieur.
Par conséquent, on définit une définition plus flexible de l'uniformité à l'aide des cardinaux $\mathrm{Card}_E(u_1,\ldots,u_n) = \sum_{i=1}^{n} \mathbb{1}_E(u_i)$. La discrépance d'une suite $(u_i)_{1\leq i\leq n}$ de I^d est

$$D_n(u) = \sup_{J\in\mathcal{J}} \left| \frac{\mathrm{Card}_J(u_1,\ldots,u_n)}{n} - \lambda_d(J) \right|$$

où $\mathcal{J}$ corresponds à la famille de tous les sous-intervalles du type $\prod_{i=1}^{d}[a_i, b_i]$.

7. i.e. $P(|Y| > t_{n-1,\alpha}) = \alpha$ où Y est une variable aléatoire Student.

La discrépance $D_n(u)$ d'une suite nous permet de borner l'erreur de la manière suivante

$$\left| \frac{1}{n} \sum_{i=1}^{n} f(u_i) - \int_{I^d} f(x)dx \right| \leq V_d(f) D_n(u),$$

où $V_d(f)$ est la variation d-dimensionelle au sens de Hardy et Krause (cf. Niederreiter (1992)). D'où l'intérêt pour les suites à discrépance faible. Les plus connues sont les suites de Van Der Corput, de Halton et de Sobol.
Dans R, le package **randtoolbox** implémentent plusieurs suites à discrépance faible, tandis que le package **lhs** propose la méthode "Latin Hypercube Sampling", une méthode hybride quasi et pseudo aléatoire.

A.2.2 Loi quelconque

En pratique, on ne simule pas des lois uniformes par une loi discrète ou continue particulière. D'une suite de nombres aléatoires uniformes $U_1, \ldots, U_n$, on va donc générer une suite $X_1, \ldots, X_n$ de fonction de répartition F.
On notera que sous R, la plupart des lois usuelles peuvent être simulées directement via des algorithmes optimisés. La fonction `rpois` permettra de générer des suites indépendantes suivant une loi de Poisson, alors que `rnorm` permettra de générer des suites indépendantes suivant une loi normale.
Pour simuler suivant une loi composée (e.g. Poisson-exponentielles), on peut utiliser tout simplement

```
> sum(rexp(rpois(1,lambda),mu)
```

On peut utiliser ce code pour comparer les résultats de la Figure **??** par la méthode de Panjer, pour calculer la probabilité que la loi composée dépasse 25,

```
>  nsim <- 100000
>  set.seed(1)
>  N <- rpois(nsim,lambda)
>  X <- rexp(sum(N))
>  I <- rep(1:nsim,N)
>  S <- as.vector(tapply(X,I,sum))
>  sum(S>25)/nsim
[1] 0.00361
```

Méthode de la transformée inverse

Notons F^{-1} l'inverse de la fonction de répartition

$$F^{-1}(u) = \inf_x F(x) \geq u,$$

pour $u \in [0,1]$. Il est facile de voir que la variable $F^{-1}(U_1)$ a la même fonction de répartition que X_1. La méthode de la transformée inverse utilise cette propriété pour donner l'algorithme suivant
– générer $U_1, \ldots, U_n \overset{iid}{\sim} \mathcal{U}(0,1)$,
– calculer $X_i = F^{-1}(U_i)$.
Notons que si X est une variable discrète, F est une fonction en escalier et l'inverse se calcule par une suite de if-else. Au contraire si X est une variable continue, l'inverse de F peut être une formule exacte comme pour la loi exponentielle $F^{-1}(u) = -\frac{\log(1-u)}{\lambda}$. Dans ce cas, la génération est très rapide.

Méthode Alias

La méthode Alias permet de générer des variables aléatoires discrètes décrites par les probabilités élémentaires $P(X = x_k)$ pour $k = 1, \ldots, n$. Toutes variables discrètes avec au plus n valeurs peut être représenté par un mélange équiprobable de $n-1$ variables discrètes bimodales (i.e. à 2 valeurs). On a

$$P(X = x) = \frac{1}{n-1} \sum_{i=1}^{n-1} q_i(x),$$

où $q_i(x)$ sont des fonctions de masse de probabilité non nulles pour deux valeurs x_i et y_i.
L'algorithme devient
– générer U, V de loi uniforme $\mathcal{U}(0,1)$,
– $k = \lceil (n-1)U \rceil$,
– si $V < q_k$ alors retourner x_k sinon retourner y_k.
Voir Walker (1977).

Inversion numérique

Dans le cas d'une variable continue X, il n'existe pas forcément d'expression explicite pour F^{-1}. Une inversion numérique est néanmoins possible. Leydold et Hormann propose une interpolation polynomiale nécessitant à partir du calcul de p points $(u_i = F(x_i), x_i, f_i = f(x_i))$. Ensuite $F^{-1}(u)$ est interpolé par un polynome d'Hermite d'ordre 3 ou 5 [8] en utilisant les points $(u_i, x_i, f_i)_i$.
L'erreur de ces méthodes d'inversion numérique est evidemment controlable. En pratique (Leydold et Hormann), le temps de calcul de ces méthodes est tout à fait acceptable car p (envirion 300 pour une précision de 10^{-6}) est relativement faible comparitivement au nombre de réalisations voulues n. Il existe même des versions pour n'utilisant que la densité $f(x_i)$ et pas la fonction de répartition. Ceci est particulièrement apprécié pour la loi normale et ses extensions par exemple Ces méthodes sont disponibles dans le package **Runuran** écrit par Leydold & Hörmann (2011).

Algorithme du rejet

Si X possède une densité f, l'algorithme du rejet-acceptation consiste à tirer dans des variables aléatoires d'un loi proche de f (mais plus facile à simuler) et de ne garder que celle qui répondent à une certaine contrainte.
Notons Y une variable aléatoire de densité et fonction de répartition g et U une variable aléatoire uniforme. S'il existe une constante $C \geq 1$ telle que on a la majoration $\forall x, f(x) \leq cg(x)$, alors la loi conditionnelle de Y sachant que $cUg(Y) < f(Y)$ égale celle de X.
Pour générer X_i, l'algorithme est le suivant
Répéter :
– générer $U \sim \mathcal{U}(0,1)$,
– générer Y selon g,
tant que $cUg(Y) < f(Y)$.
affecter $X_i = Y$.
Le nombre de rejet suit une loi géométrique de paramètre $1/C$. Par conséquent plus l'approximation est bonne (C proche de 1), plus le nombre de rejets est faible.

8. l'interpolation linéaire (d'ordre 1) n'est pas efficace car le nombre p de points est trop élevé.

A.2.3 Processus aléatoires et Variables multivariées

Des applications d'actuariat nécessiteront la simulation de processus aléatoires et pas seulement de variables indépendantes. Dans ce cas, l'équation différentielle stochastique doit être discrétisée de manière à simuler la $i$ème trajectoire "complète" $(X_{t_0,i},\dots X_{t_T,i})$ sur $[t_0,t_T]$. Par conséquent le nombre de points $n(T+1)$ grandit rapidement. Il faut donc bien réfléchir si toute la trajectoire du processus est nécessaire ou si seule la valeur terminale où le supremum nous intéresse.
Par exemple, considérons la simulation d'un processus de Poisson. Si on s'intéresse à un processus de Poisson homogène, d'intensité λ, on va générer les durées entre sauts, qui sont exponentielles. Pour générer un vecteur de dates de sauts sur un intervalle de temps $[0,T]$ on considère le code suivant

```
> nmax <- 10000
> ST <- cumsum(rexp(nmax,lambda))
> ST <- ST[ST<=T]
```

On peut alors construire la fonction $t\mapsto N_t$ sous la forme

```
> Nt <- function(t) sum(ST<=t)
```

Si le processus de Poisson est non-homogène, d'intensité $\lambda(t)$ (que l'on supposera bornée par $\overline{\lambda}$), il est possible d'utiliser l'algorithme suivant pour générer un processus : on va générer un processus de Poisson d'intensité $\overline{\lambda}$, et on utilise une méthode de type acceptation-rejet pour savoir si on garde un saut.

- poser $T_0=0$ et $T_\star=0$,
- générer E exponentielle de moyenne 1λ et poser $T_\star=T_\star+E$,
- générer U uniforme sur $[0,1]$: si $U>\lambda(T_\star)/\overline{\lambda}$ on retourne à la seconde étape, et on tire un nouveau E, sinon on pose $T_i=T_\star$.

Une autre possibilité est de noter que pour un processus de Poisson homogène, on partait de $T_0=0$, et on utilisait

$$T_i=T_{i-1}+F^{-1}(U),$$

où F est la fonction de répartition de la loi exponentielle de moyenne $1/\lambda$. Ici, on va utiliser

$$T_i=T_{i-1}+F^{-1}_{T_{i-1}}(U),$$

où F_s est la fonction de répartition du temps d'attente entre le $N_s$ième saut, et le suivant, i.e.

$$F_s(t)=1-\mathbb{P}(N_{s+t}-N_s=0)=1-\exp\left(\int_s^{s+t}\lambda(u)du\right).$$

Ces fonctions sont programmée dans le package **PtProcess**.
La simulation multivariée nécessite aussi du doigté, car en dehors d'une loi à composante indépendante, la $i$ème réalisation du vecteur $(U_{1,i},\dots,U_{d,i})$ n'est pas triviale à calculer. Par exemple, l'algorithme de rejet/acceptation sur la suite $(V_{1,i}=1-2U_{1,i},\dots,V_{d,i}=1-2U_{d,i})_i$ avec la condition $\sum_j V_{j,i}^2\leq 1$ simule une loi uniforme dans la sphère unité d-dimensionnelle.
La génération d'une loi normale multivariée $\mathcal{N}(\mu,\Sigma)$ est un peu plus complexe :

- générer d variables indépendantes $X_i\sim\mathcal{N}(0,1)$,
- calculer la décomposition de Cholesky $\Sigma=C'C$,
- calculer $Y=\mu+C'X$.

Notons que si l'on veut simuler une variable multivariée sur l'hyperellypse définie par $\{x,x^T\Sigma x\leq r\}$, il suffit de remplacer la première étape par la génération de d variables uniformément distribuées dans la sphère unité.

```
>  set.seed(1)
>  rmultinormal <- function(n,S){
+  Z <- matrix(NA,n,ncol(S))
+  C <- chol(S)
+  for(i in 1:n){Z[i,] <- t(C) %*% rnorm(3)}
+ return(Z)}
>  Sigma <- matrix(c(1,.7,.3,.7,1,-.3,.3,-.3,1),3,3)
>  rmultinormal(1,Sigma)
           [,1]       [,2]       [,3]
[1,] -0.6264538 -0.3073701 -0.8475816
>  cor(rmultinormal(10000,Sigma))
          [,1]       [,2]       [,3]
[1,] 1.0000000  0.7034906  0.2989346
[2,] 0.7034906  1.0000000 -0.2918081
[3,] 0.2989346 -0.2918081  1.0000000
```

Bibliographie

Amiot, E. (1999), *Introduction aux probabilités et à la statistique*, Gaetan Morin. 1

Arnold, B. C. (1983), *Pareto Distributions*, International Co-operative Publishing House. 14

Arnold, B. C. (2008), Pareto distributions, *in* 'Encyclopedia of Statistical Sciences', Wiley Interscience. 14

Bailey, A. L. (1950), 'Credibility procedures, Laplace's generalization of Bayes' rule and the combination of collateral knowledge with observed data', *Proceedings of the Casualty Actuarial Society* **37**, 7–23.

Bailey, R. (1963), 'Insurance rates with minimum bias', *Proceedings of the Society of Actuaries* **50**, 4–11. 63, 64

Balson, N. (2008), *Mesure d'incertitude sur l'estimation des provisions de sinistres en Assurance Non Vie*, Institut des Actuaires - ENSAE. 130

Belhadj, H., Goulet, V. & Ouellet, T. (2009), 'On parameter estimation in hierarchical credibility', *ASTIN Bulletin* **39**(2).

Benktander, G. (1976), 'An approach to credibility in calculating ibnr for casualty excess reinsurance', *Actuarial Review* **3**, 7–31. 127, 128

Bernegger, S. (1997), 'The swiss re exposure curves and the mbbefd distribution class', *Astin Bull.* **27**(1), 99–111. 10, 11

Bowers, N. L., Jones, D. A., Gerber, H. U., Nesbitt, C. J. & Hickman, J. C. (1997), *Actuarial Mathematics, 2nd Edition*, SOA. iii, iv

Bühlmann, H. (1967), 'Experience rating and credibility', *ASTIN Bulletin* **4**, 199–207.

Bühlmann, H. (1969), 'Experience rating and credibility', *ASTIN Bulletin* **5**, 157–165.

Bühlmann, H. & Gisler, A. (1997), 'Credibility in the regression case revisited', *ASTIN Bulletin* **27**, 83–98.

Bühlmann, H. & Gisler, A. (2005), *A Course in Credibility Theory and its Applications*, Springer.

Bühlmann, H. & Jewell, W. S. (1987), 'Hierarchical credibility revisited', *Bulletin of the Swiss Association of Actuaries* **87**, 35–54.

Bühlmann, H. & Straub, E. (1970), 'Glaubwürdigkeit fur schadensätze', *Bulletin of the Swiss Association of Actuaries* **70**, 111–133.

Cairns, A. J. G., Blake, D., Dowd, K., Coughlan, G. D. & Epstein, D. (2008), 'A Quantitative Comparison of Stochastic Mortality Models using Data from England and Wales and the United States', *North American Actuarial Journal* **13**(1), 1–35. 159

Chambers, J. (2009), *Software for Data Analysis : Programming with R*, Springer Verlag. iii

Christofides, S. (1989), Regression models based on log-incremental payments, *in* I. of Actuaries, ed., 'Claims Reserving Manual'. 109

Cormen, T. H., Leiserson, C. E., Rivest, R. L. & Stein, C. (2009), *Introduction to Algorithms*, The MIT Press. iii

Dagnelie, P. (2007), *Statistique théorique et appliquée*, De Boeck Université. 21

Dalgaard, P. (2008), *Introductory Statistics with R*, Springer. 21

Dalgaard, P. (2009), *Introductory Statistics with R*, Springer Verlag. iii

Davison, A. & Snell, E. (1991), Residuals and diagnostics, *in* N. R. D.V. Hinkley & E. Snell, eds, 'Statistical Theory and Modelling', Chapman and Hall. 51

Daykin, C. D., Pentikainen, T. & Pesonen, M. (n.d.), *Practical Risk Theory for Actuaries*, Chapman and Hall. 12

de Jong, P. & Zeller, G. (2008), *Generalized Linear Models for Insurance Data*, Cambridge University Press. iii, 39

De Vylder, F. (1981), 'Practical credibility theory with emphasis on parameter estimation', *ASTIN Bulletin* **12**, 115–131.

De Vylder, F. (22-28), Estimation of ibnr claims by least squares, *in* 'Proc. First Meeting Contact-group Actuarial Sciencesl'. 109

Delmas, J.-F. (2012), *Introduction aux probabilités et à la statistique*, Ensta. 1

Denuit, M. & Charpentier, A. (2004), *Mathématiques de l'assurance non-vie : principes fondamentaux de théorie du risque. Tome 1.*, Economica. iii, 37

Denuit, M. & Charpentier, A. (2005), *Mathématiques de l'assurance non-vie : Tarification et provisionnement. Tome 2.*, Economica. iii, 37, 39, 91, 100

Denuit, M. & Robert, C. (2007), *Actuariat des Assurances de Personnes : Modélisation, Tarification et Provisionnement*, Economica. 133, 159

Dickson, D. C., Hardy, M. R. & Waters, R. H. (2009), *Actuarial Mathematics for Life Contingent Risks*, Cambridge University Press. iii, iv, 133

Dubey, A. & Gisler, A. (1981), 'On parameter estimation in credibility', *Bulletin of the Swiss Association of Actuaries* **81**, 187–211.

Dubreuil, E. & Vendé, P. (2005), Les couvertures indicielles en réassurance catastrophe. Prise en compte de la dépendance spatiale dans la tarification. 30

Dutang, C., Goulet, V. & Pigeon, M. (2008), '**actuar** : An R package for actuarial science', *Journal of Statistical Software* **25**(7). 7

Embrechts, P., Klüppelberg, C. & Mikosch, T. (1997), *Modelling Extremal Events*, Springer. 85

Embrechts, P., Lindskog, F. & McNeil, A. (2001), Modelling dependence with copulas and applications to risk management, Technical report, ETH Zurich. 13, 15

England, P. D. & Verrall, R. J. (1999), 'Analytic and bootstrap estimates of prediction errors in claims reserving', *Insurance : Mathematics and Economics* **25**, 281–293. 118

Frees, E. (2009), *Regression modeling with actuarial and financial applications*, Cambridge University Press. iii, 39

Frees, E. W. & Valdez, E. (1998), 'Understanding Relationships Using Copulas', *North American Actuarial Journal* **2**(1). 13

Frees, E. W. & Wang, P. (2006), 'Copula credibility for aggregate loss models', *Insurance : Mathematics and Economics* **38**, 360–373. 13

Friedman, J. (1991), 'Multivariate additive regression splines', *Annals of Statistics* **19**(1), 1–67. 72

Genest, C., Kojadinovic, I., Nešlehová, J. & Yan, J. (2011), 'A goodness-of-fit test for bivariate extreme-value copulas', *Bernoulli* **17**(1), 253–275. 33

Gentle, J. (2009), *Computational Statistics*, Springer Verlag. iii

Gerber, H. & Shiu, E. (1994), 'Option pricing by esscher transforms', *Transactions of the Society of Actuaries Society of Actuaries* **46**, 99–191. 49

Giles, T. L. (1993), 'Life insurance application of recursive formulas', *Journal of Actuarial Practice* **1**(2), 141–151. 147

Gilks, W. & Wild, P. (2004), 'Adaptive rejection sampling from log-concave density', *Applied Statistics* **42**, 701–709.

Goovaerts, M. J. & Hoogstad, W. J. (1987), *Credibility Theory*, number 4 *in* 'Surveys of actuarial studies', Nationale-Nederlanden N.V., Netherlands.

Goulet, V. (2008), Credibility, *in* E. Melnick & B. Everitt, eds, 'Encyclopedia of Quantitative Risk Analysis and Assessment', Wiley.

Hachemeister, C. A. (1975), Credibility for regression models with application to trend, *in* 'Credibility, theory and applications', Proceedings of the Berkeley actuarial research conference on credibility, Academic Press, New York, pp. 129–163.

Hachemeister, C. A. & Stanard, J. N. (1975), Ibnr claims count estimation with static lag functions, *in* '12th ASTIN Colloquium', Portimao, Portugal. 110

Hastie, T. & Tibshirani, R. (1990), *Generalized Additive Models*, Chapman and Hall. 69, 72

Hess, C. (2000), *Méthodes actuarielles de l'assurance vie*, Economica. 133

Hogg, R. V., Craig, A. T. & McKean, J. W. (2005), *Introduction to Mathematical Statistics*, 6 edn, Prentice Hall, Upper Saddle River, NJ. 21

Hogg, R. V. & Klugman, S. A. (1984), *Loss Distributions*, Wiley, New York.

Hovinen, E. (1981), Additive and continuous ibnr, *in* 'ASTIN Colloquium', Loen, Norway. 127

Hsiao, C., Kim, C. & Taylor, G. (1990), 'A statistical perspective on insurance rate-making', *Journal of Econometrics* **44**(1-2), 5 – 24.

Hurvich, C. M. & Tsai, C.-L. (1995), 'Model selection for extended quasi-likelihood models in small samples', *Biometrics* **51**, 1077–1084. 52

Hyndman, R. J. & Fan, Y. (1996), 'Sample quantiles in statistical packages', *American Statistician* **50**, 361–365. 17

Ihaka, R. & Gentleman, R. (1996), 'R : A language for data analysis and graphics', *Journal of Computational and Graphical Statistics* **5**(3), 299–314.

Jeffrey, A. & Dai, H.-H. (2008), *Handbook of mathematical formulas and integrals*, Academic Press.

Jewell, W. S. (1974), 'Credible means are exact bayesian for exponential families', *Astin Bull.* **8**, 77–90.

Jewell, W. S. (1975), 'The use of collateral data in credibility theory : A hierarchical model', *Giornale dell'Istituto Italiano degli Attuari* **38**, 1–16.

Joe, H. (1997), Multivariate dependence measure and data analysis, *in* 'Monographs on Statistics and Applied Probability', Vol. 73, Chapman & Hall. 28, 33

Johnson, N. L., Kotz, S. & Balakrishnan, N. (1997), *Discrete Multivariate Distributions*, Wiley Interscience. 14

Johnson, N. L., Kotz, S. & Kemp, A. W. (2005), *Univariate discrete distributions*, 3rd edn, Wiley Interscience. 9

Jung, J. (1968), 'On automobile insurance ratemaking', *ASTIN Bulletin* **5**, 41–48. 63

Kaas, R., Goovaerts, M., Dhaene, J. & Denuit, M. (2009), *Modern Actuarial Risk Theory*, Springer Verlag. iii, 39

Klugman, S. A., Panjer, H. H. & Willmot, G. (1998), *Loss Models : From data to Decisions*, Wiley, New York.

Klugman, S. A., Panjer, H. H. & Willmot, G. E. (2009), *Loss Models : From Data to Decisions*, Wiley Series in Proability and Statistics. iii, iv

Knuth, D. E. (1997*a*), *The Art of Computer Programming, volume 1 : Fundamental algorithms*, Massachusetts : Addison-Wesley. iii

Knuth, D. E. (1997*b*), *The Art of Computer Programming, volume 2 : Seminumerical Algorithms*, Massachusetts : Addison-Wesley. iii

Knuth, D. E. (1998), *The Art of Computer Programming, volume 3 : Sorting and Searching*, Massachusetts : Addison-Wesley. iii

Kotz, S., Balakrishnan, N. & Johnson, N. L. (1994*a*), *Continuous Multivariate Distributions*, Vol. 2, Wiley Interscience. 2, 14

Kotz, S., Balakrishnan, N. & Johnson, N. L. (1994*b*), *Continuous Multivariate Distributions*, Vol. 1, Wiley Interscience. 2, 14

Kotz, S., Balakrishnan, N. & Johnson, N. L. (2002), *Continuous Multivariate Distributions*, Vol. 1, Wiley Interscience.

Krause, A. (2009), *The Basics of S-PLUS*, Springer Verlag. iii

Kremer, E. (1982), 'Ibnr claims and the two-way model of anova', *Scandinavian Actuarial Journal* pp. 47–55. 110

Lacoume, A. (2009), *Mesure du risque de réserve sur un horizon de un an*, Institut des Actuaires - ISFA. 107

L'Ecuyer, P. (1990), 'Random numbers for simulation', *Communications of the ACM* **33**, 85–98. 188

Lee, R. & Carter, L. (1992), 'Modeling and forecasting u.s. mortality', *Journal of the American Statistical Association* **87**(419), 659–671. 166

Leydold, J. & Hörmann, W. (2011), *Runuran : R interface to the UNU.RAN random variate generators*. 191

Mack, T. (1991), 'A simple parametric model for rating automobile insurance or estimating ibnr claims reserves', *ASTIN Bulletin* **21**, 93–109. 110

Mack, T. (1993*a*), 'Distribution-free calculation of the standard error of chain-ladder reserve estimates', *ASTIN Bulletin* **15**, 133–138. 98, 99, 116, 118, 123, 129, 130

Mack, T. (1993*b*), 'The standard error of chain-ladder reserve estimates : Recursive calculation and inclusion of a tail factor', *ASTIN Bulletin* **29**, 361–366. 100

Mack, T. (1994), 'Which stochastic model is underlying the chain-ladder method ?', *Insurance : Mathematics and Economics* **23**, 213–225. 100

Maindonald, J. & Braun, W. J. (2007), *Data Analysis and Graphics Using R : An Example-Based Approach*, Cambridge University Press. iii

Marceau, E. (2012), *Modélisation et évaluation des risques en actuariat*, Springer. iii

Marshall, A. W. & Olkin, I. (1988), 'Families of multivariate distributions', *Journal of the American Statistical Association* **83**(403), 834–841. 15

Matsumoto, M. & Nishimura, T. (1998), 'Mersenne twister : A 623-dimensionnally equidistributed uniform pseudorandom number generator', *ACM Trans. on Modelling and Computer Simulation* **8**(1), 3–30. 188

Mayerson, A. L. (1964), 'A bayesian view of credibility', *Proceedings of the Casualty Actuarial Society* **51**, 85–104.

McCullagh, P. & Nelder, J. (1991), *Generalized Linear Models*, CRC Press. 39

McDonald, J. & Butler, R. (1990), 'Regression models for positive random variables', *Journal of Econometrics* **43**, 227–251. 86

Merz, M. & Wüthrich, M. V. (2008), 'Modelling the claims development result for solvency purposes', *CAS E-Forum* pp. 542–568. 106, 107

Moral, P. D., Remillard, B. & Rubenthaler, S. (2006), *Introduction aux probabilités*, Ellipses. 1

Mori, Y. (2009), *Handbook of Computational Statistics*, Springer Verlag. iii

Mowbray, A. H. (1914), 'How extensive a payroll exposure is necessary to give a dependable pure premium ?', *Proceedings of the Casualty Actuarial Society* **1**, 25–30.

Nelsen, R. B. (1999), *An introduction to copulas*, Springer. 15

Nelsen, R. B. (2006), *An introduction to copulas*, Springer. 15

Niederreiter, H. (1992), *Random Number Generation and Quasi-Monte Carlo Methods*, SIAM, Philadelphia. 190

Ohlsson, E. & Johansson, B. (2010), *Non-Life Insurance Pricing with Generalized Linear Models*, Springer Verlag.

Ohlsson, E. & Johansson, B. (2010), *Non-life insurance pricing with Generalized Linear Models*, Springer Verlag. iii, 39

Olver, F. W. J., Lozier, D. W., Boisvert, R. F. & Clark, C. W., eds (2010), *NIST Handbook of Mathematical Functions*, Cambridge University Press.
URL: *http ://dlmf.nist.gov/* 2

Panjer, H. H. (1981), 'Recursive evaluation of a family of compound distributions', *Astin Bull.* **12**(1), 22–26. 11

Parent, E. & Bernier, J. (2007), *Le raisonnement bayésien*, Springer Verlag. 129

Partrat, C., Lecoeur, E., Nessi, J., Nisipasu, E. & Reiz, O. (2008), *Provisionnement technique en Assurance non vie*, Economica. 91

Pearson, K. (1895), 'Contributions to the mathematical theory of evolution, ii : Skew variation in homogeneous material', *Philosophical Transactions of the Royal Society of London* . 185

Petauton, P. (2004), *Théorie et pratique de l'assurance vie*, Dunod. 133, 141

Pitacco, E., Denuit, M., Haberman, S. & Olivieri, A. (2009), *Modelling Longevity Dynamics for Pensions and Annuity Business*, Oxford University Press. 159

Planchet F., T. P. (2006), *Modèles de durée & applications actuarielles*, Economica.

Pröhl, C. & Schmidt, K. D. (2005), Multivariate chain-ladder, *in* 'ASTIN Colloquium', Zurich. 125

Quarg, G. & Mack, T. (2004), 'Munich chain-ladder and a reserving method that reduces the gap between ibnr projections based on paid losses and ibnr projections based on incurred losses', *Variances* **2**, 267–299. 101, 104

Renshaw, A. E. & Haberman, S. (2006), 'A cohort-based extension to the lee-carter model for mortality reduction factors', *Insurance : Mathematics and Economics* **58**, 556–570. 181

Renshaw, A. E. & Verrall, R. J. (1998), 'A stochastic model underlying the chain-ladder technique', *British Actuarial Journal* **4**, 903–923. 115

Robert, C. (2006), *Le choix bayésien, Principes et pratique*, Springer Verlag. 129

Saporta, G. (2006), *Probabilités, analyse de donnés et statistique*, Technip. 21, 29

Simonet, G. (1998), *Comptabilité des entreprises d'assurance*, L'Argus de l'Assurance. 91

Sklar, A. (1959), 'Fonctions de répartition à n dimensions et leurs marges', *Publications de l'ISUP de Paris 8* **8**, 229–231. 14

Stone, C. (1985), 'Additive regression and other nonparametric models', *Annals of Statistics* **13**(2), 689–705. 69

T., M. & T., S. (2006), *Dynamic Regression Models for Survival Data*, Springer Verlag.

Tanner, M. A. & Wong, W. H. (1983), 'The estimation of the hazard function from randomly censored data by the kernel method', *The Annals of Statistics* .

Taylor, G. (1977), 'Separation of inflation and other effects from the distribution of non-life insurance claim delays', *ASTIN Bulletin* **9**, 217–230. 108

Therneau, T. (2009), ***survival*** *: Survival Analysis, Including Penalised Likelihood.* R package version 2.35-4. Original R port by Thomas Lumley.

Venables, W. N. & Ripley, B. D. (2002*a*), *Modern Applied Statistics with S*, 4th edn, Springer. iii

Venables, W. N. & Ripley, B. D. (2002*b*), *Modern Applied Statistics with S*, 4 edn, Springer, New York.

Verrall, R. J. (2000), 'An investigation into stochastic claims reserving models and the chain-ladder technique', *Insurance : Mathematics and Economics* **26**, 91–99. 108

Vylder, E. D. (2010), *Life Insurance Theory : Actuarial Perspectives*, Springer Verlag. 133

Walker, A. J. (1977), 'An efficient method for generating discrete random variables with general distributions', *ACM Transactions on Mathematical Software* **3**, 253–256. 191

Wheeler, B. (2006), ***SuppDists*** *: Supplementary Distributions.* R package version 1.1-0. **URL:** *http ://www.bobwheeler.com/stat*

Whitney, A. W. (1918), 'The theory of experience rating', *Proceedings of the Casualty Actuarial Society* **4**, 275–293.

Wood, S. (2000), 'Additive regression and other nonparametric models', *Annals of Statistics* **62**(2), 413–428. 72

Wüthrich, M. V. & Merz, M. (2008), *Stochastic Claims Reserving Methods in Insurance*, Wiley Interscience. 91, 104

Zehnwirth, B. (1985), *Interactive claims reserving forecasting system (ICRFS)*, Benhar Nominees Pty Ltd. Tarramurra N.S.W., Australia. 109

Zuur, A. F., Ieno, E. N. & Meesters, E. (2009), *A Beginner's Guide to R*, Springer Verlag. iii

Index

Index des commandes