L'Actuariat avec

Arthur Charpentier, Christophe Dutang

Décembre 2012 – Version numérique

rédigé en LaTeX

©Arthur Charpentier, Christophe Dutang
©Arthur Charpentier, Christophe Dutang, Vincent Goulet pour les sections 1.2 et 1.3

Cette création est mise à disposition selon le contrat Paternité-Partage à l'indentique 3.0 France de Creative Commons, cf. http://creativecommons.org/licenses/by-sa/3.0/fr/.

En vertu de ce contrat, vous êtes libre de :
- **partager** – reproduire, distribuer et communiquer l'oeuvre,
- **remixer** – adapter l'œuvre,
- utiliser cette oeuvre à des fins commerciales.

Selon les conditions suivantes

Attribution – Vous devez attribuer l'oeuvre de la manière indiquée par l'auteur de l'oeuvre ou le titulaire des droits (mais pas d'une manière qui suggérerait qu'ils vous approuvent, vous ou votre utilisation de l'oeuvre).

Partage dans les Mêmes Conditions – Si vous modifiez, transformez ou adaptez cette oeuvre, vous n'avez le droit de distribuer votre création que sous une licence identique ou similaire à celle-ci.

comprenant bien que :

Renonciation – N'importe laquelle des conditions ci-dessus peut être levée si vous avez l'autorisation du titulaire de droits.

Domaine Public – Là où l'oeuvre ou un quelconque de ses éléments est dans le domaine public selon le droit applicable, ce statut n'est en aucune façon affecté par la licence.

Autres droits – Les droits suivants ne sont en aucune manière affectés par la licence : (i) Vos prérogatives issues des exceptions et limitations aux droits exclusifs ou fair use ; (ii) Les droits moraux de l'auteur ; (iii) Droits qu'autrui peut avoir soit sur l'oeuvre elle-même soit sur la façon dont elle est utilisée, comme le droit à l'image ou les droits à la vie privée.

Remarque – A chaque réutilisation ou distribution de cette oeuvre, vous devez faire apparaître clairement au public la licence selon laquelle elle est mise à disposition. La meilleure manière de l'indiquer est un lien vers cette page web.

"*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

Avant-propos **iii**

Table des matières **v**

1 Modèles de sinistres sans variables explicatives **1**
1.1 Rappels des lois usuelles en actuariat 1
1.2 Estimation non-paramétrique 16
1.3 Estimation paramétrique 20
1.4 Estimation des copules 27
1.5 Exercices 35

2 La tarification a priori **37**
2.1 Les modèles linéaires généralisés 39
2.2 Régression logistique et arbre de régression 52
2.3 Modéliser la fréquence de sinistralité 62
2.4 Les variables qualitatives ou facteurs 63
2.5 Modéliser les coûts individuels des sinistres 79
2.6 Exercices 87

3 Les provisions pour sinistres à payer **91**
3.1 La problématique du provisionnment 91
3.2 Les cadences de paiements et la méthode Chain Ladder 94
3.3 De Mack à Merz & Wüthrich 97
3.4 Régression Poissonnienne et approches économétriques 108
3.5 Les triangles multivariés 123
3.6 Borhutter-Fergusson, Benktander et les méthodes bayésiennes 126
3.7 Exercices 132

4 Calculs de base en assurance vie et décès **133**
4.1 Quelques notations 133
4.2 Calculs d'annuités 137
4.3 Calculs de provisions mathématiques 140
4.4 Algorithme récursif en assurance-vie 147
4.5 Le package **lifecontingencies** 150
4.6 Exercices 155

5 Les tables prospectives **159**
5.1 Les bases de données prospectives 159
5.2 Le modèle de Lee & Carter 166
5.3 Utilisation du modèle de Lee-Carter projeté 178
5.4 Aller plus loin que le modèle de Lee-Carter 181
5.5 Exercices 182

A Annexes **185**
A.1 Les lois de probabilités 185
A.2 Générateurs aléatoires 187

Bibliographie **195**

Index **203**

Index des commandes **206**

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)
```

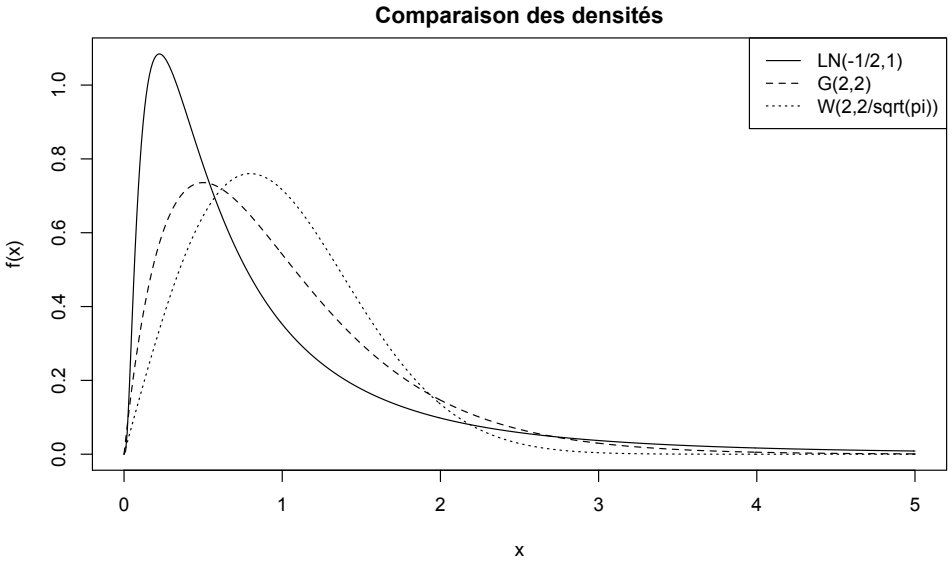


FIGURE 1.1 – Densités de lois usuelles pour des variables positives.

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.

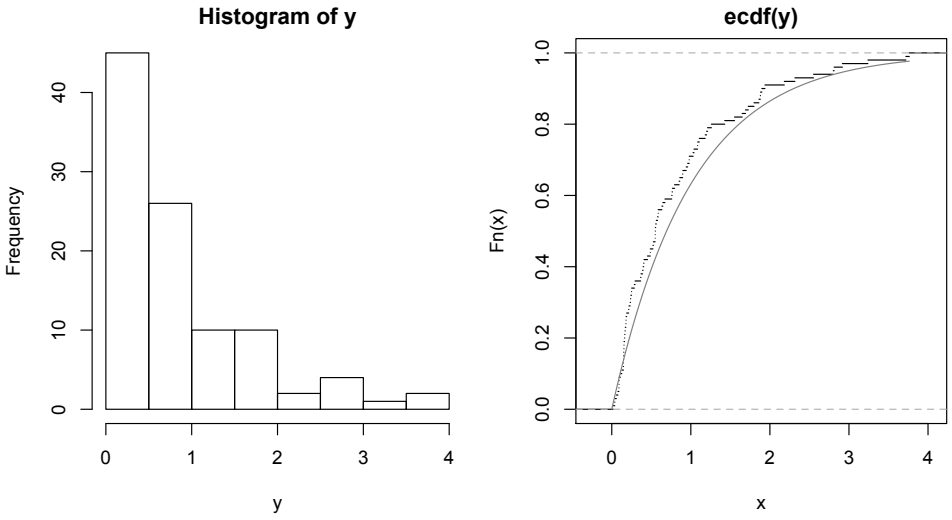


FIGURE 1.2 – Transformée logarithmique.

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

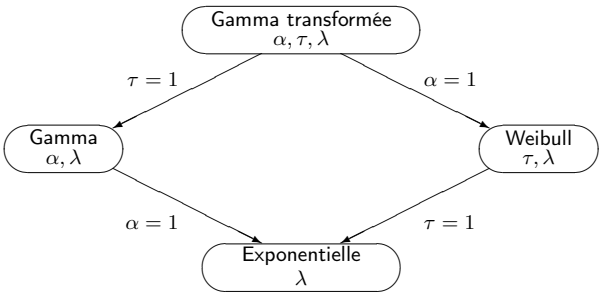


FIGURE 1.3 – Relations entre les membres de la famille gamma transformée

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.

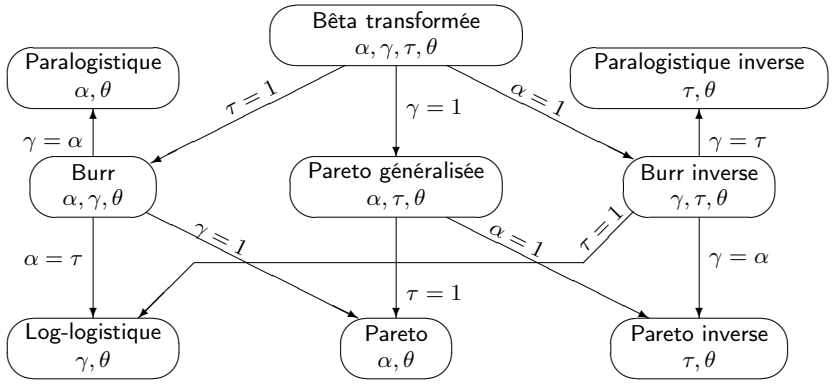
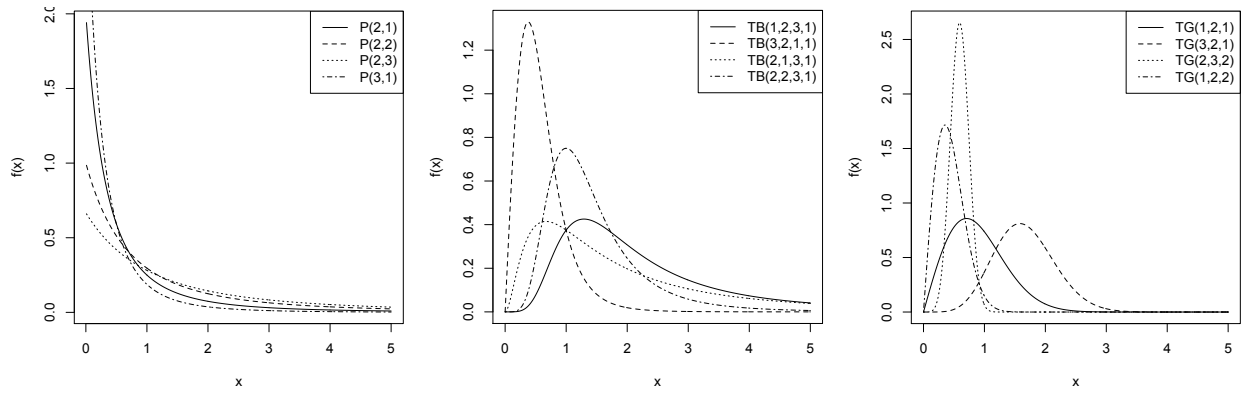


FIGURE 1.4 – Relations entre les membres de la famille béta transformée

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 ($\mathbb{E}(X) = 5$).

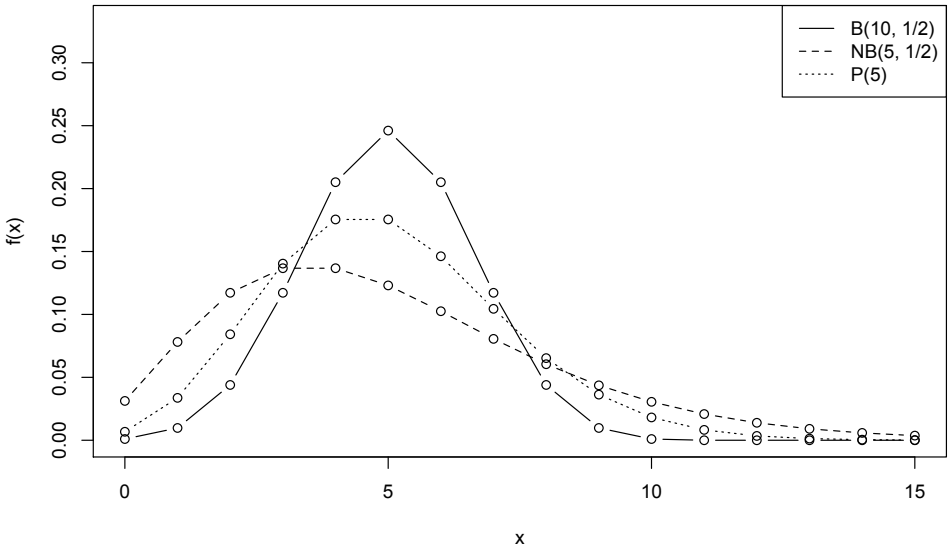


FIGURE 1.6 – Fonctions de masse de probabilité des lois discrètes usuelles

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.

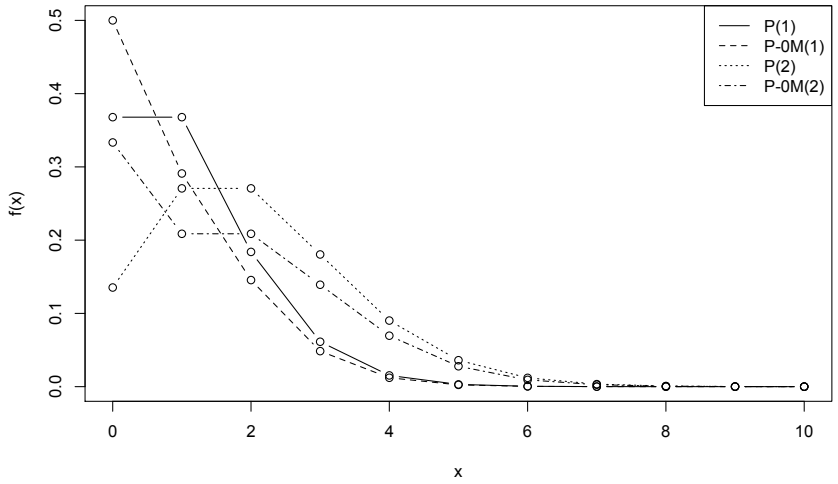


FIGURE 1.7 – La loi de Poisson zéro-modifiée.

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.

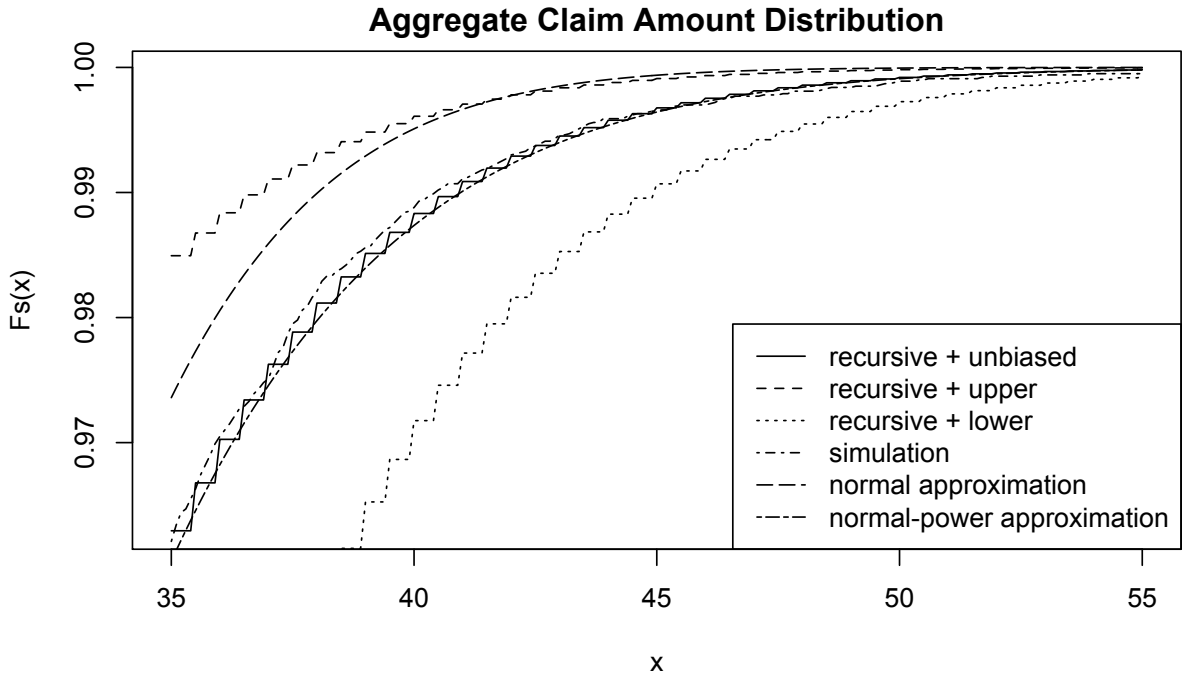


FIGURE 1.8 – Comparaison de méthodes

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 :

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

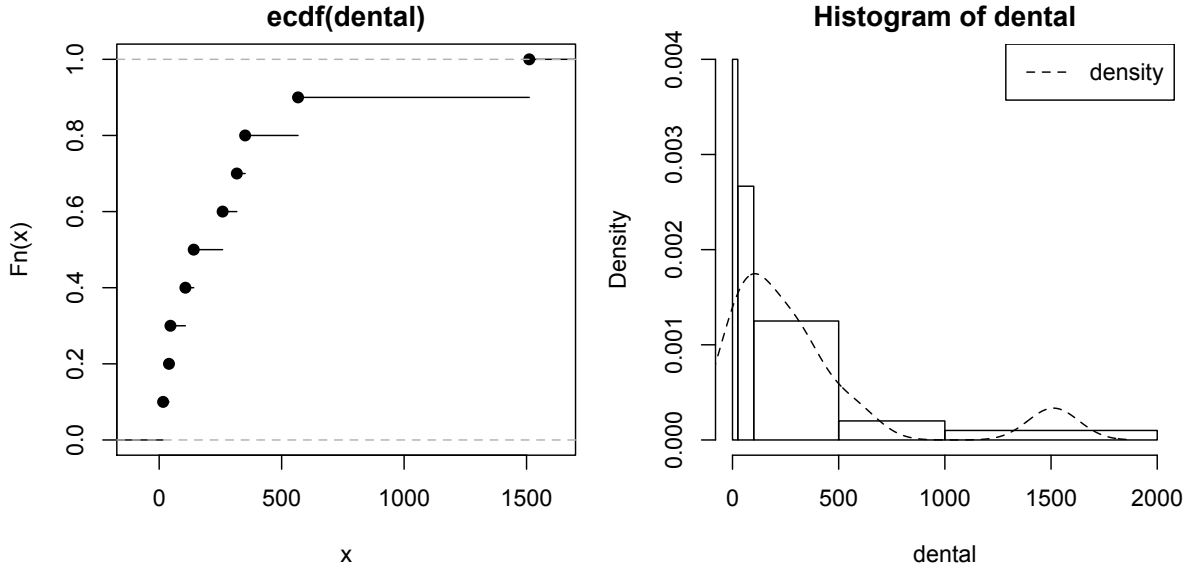


FIGURE 1.9 – Exemple de fonction de répartition empirique (gauche) et d'histogramme (droite) de données individuelles

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 :

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

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.

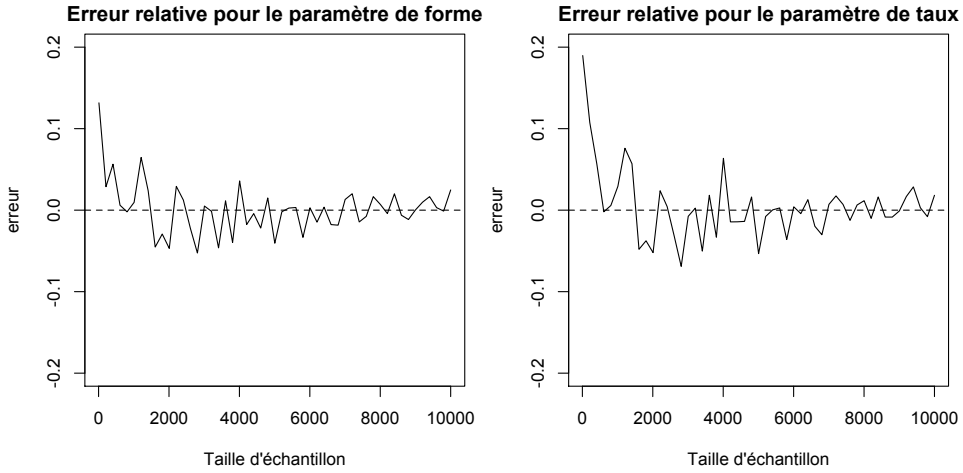


FIGURE 1.10 – Erreur relative sur les paramètres de la loi Gamma

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.

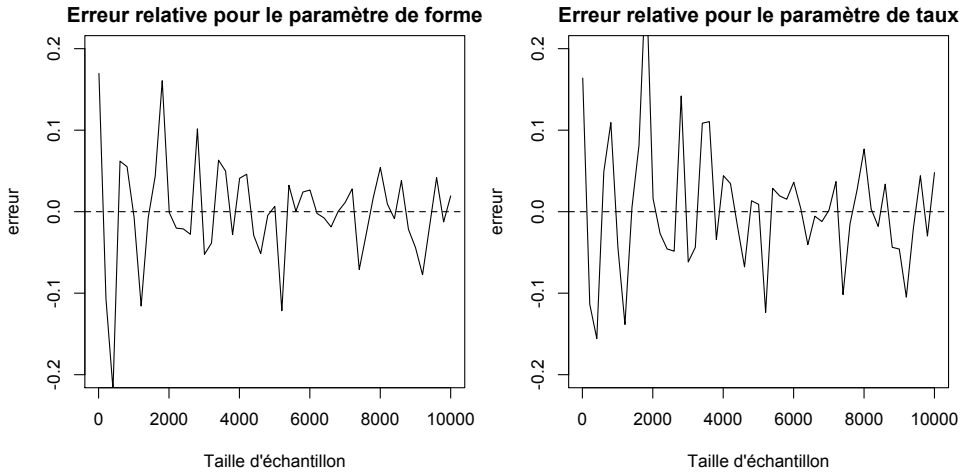


FIGURE 1.11 – Erreur relative sur les paramètres de la loi Pareto

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

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

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 ($k = -1, -2, \dots$).

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.

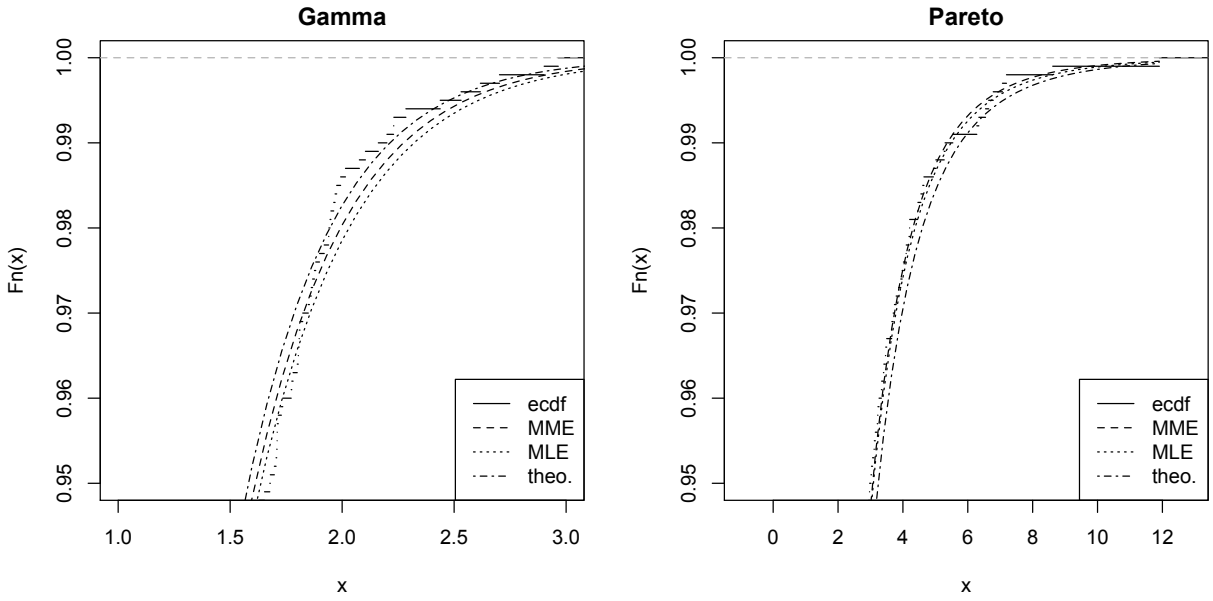
Le package **fitdistrplus** permet d'utiliser la méthode des moments soit directement avec la fonction `mmedist` soit via la fonction `fitdist`. Dans R, cela donne les commandes suivantes :

```
> 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.

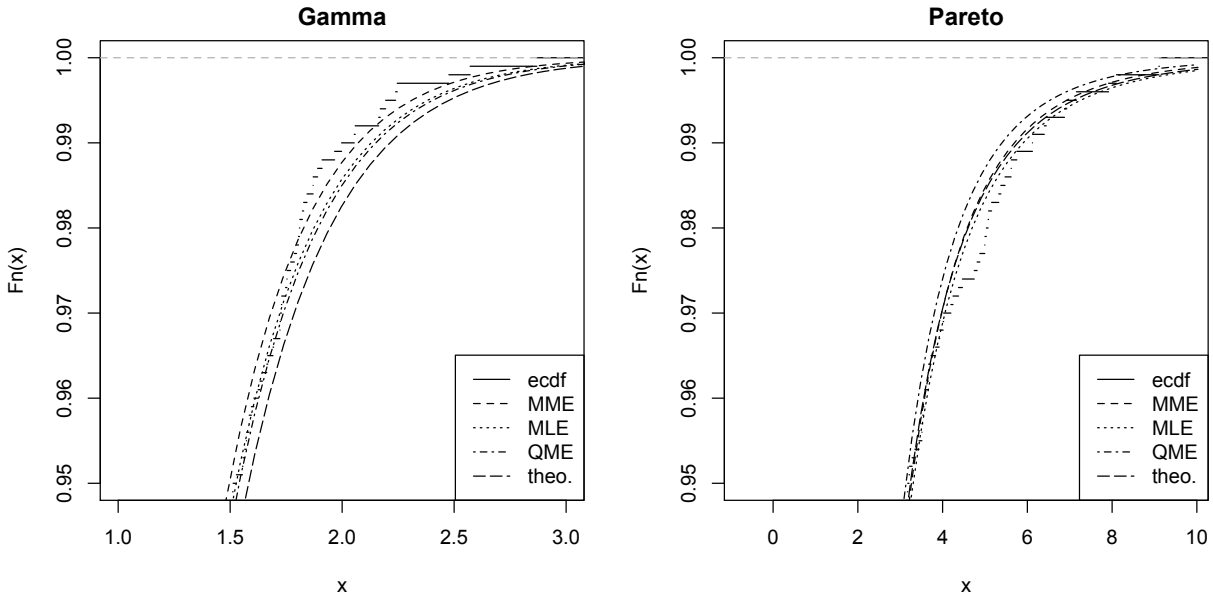
Enfin sur la figure 1.13, on a continué la comparaison des fonctions de répartition entre les trois méthodes paramétriques et la méthode non paramétrique. Notons que l'ordre entre le maximum de vraisemblances et la méthodes des moments semble de nouveau respecté. Par contre, pour le choix de quantiles 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.

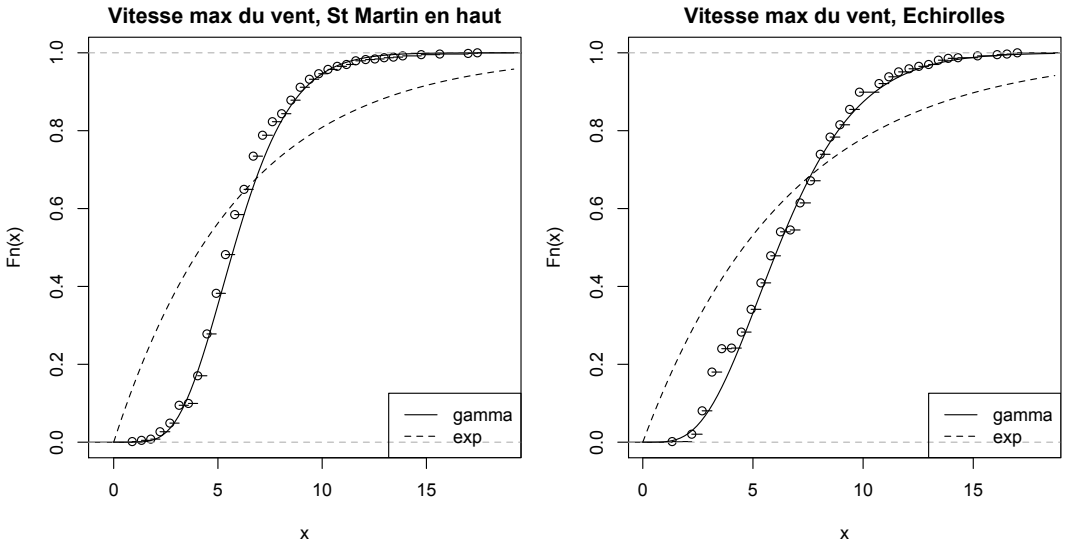


FIGURE 1.14 – Fonctions de répartitions empiriques (marche) et calibrées (courbes)

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.

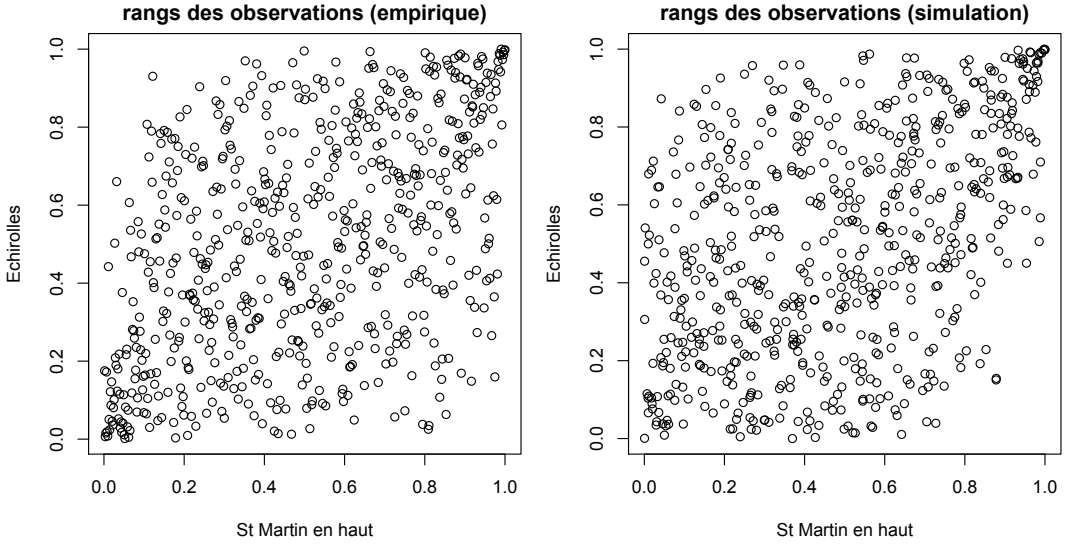


FIGURE 1.15 – qqplot empirique vs simulé

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 payoffs 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 $\hat{\theta}_{cop} = 1,472$. Nous obtenons les résultats donnés dans le tableau 1.4. On constate qu'une augmentation de la dépendance (i.e. $\hat{\theta}_{cop}$ 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.

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

Table 1.4 – Statistiques des payoffs 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

```
> par(mfrow = c(1, 1))
```

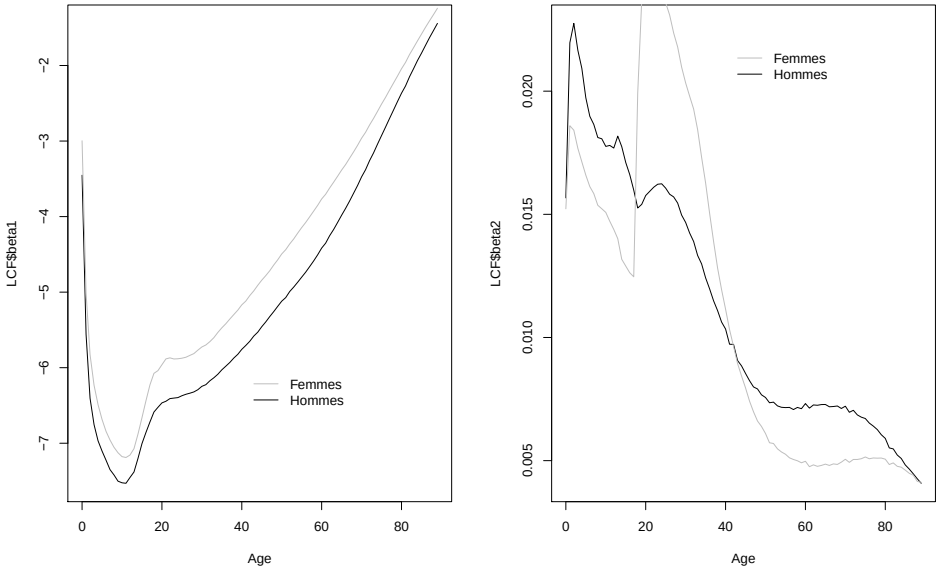


FIGURE 5.13 – Evolution de $x \mapsto \alpha_x$ (à gauche) et $x \mapsto \beta_x$ (à droite), pour les Hommes - en trait sombre - et pour les Femmes - en trait grisé.

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
```

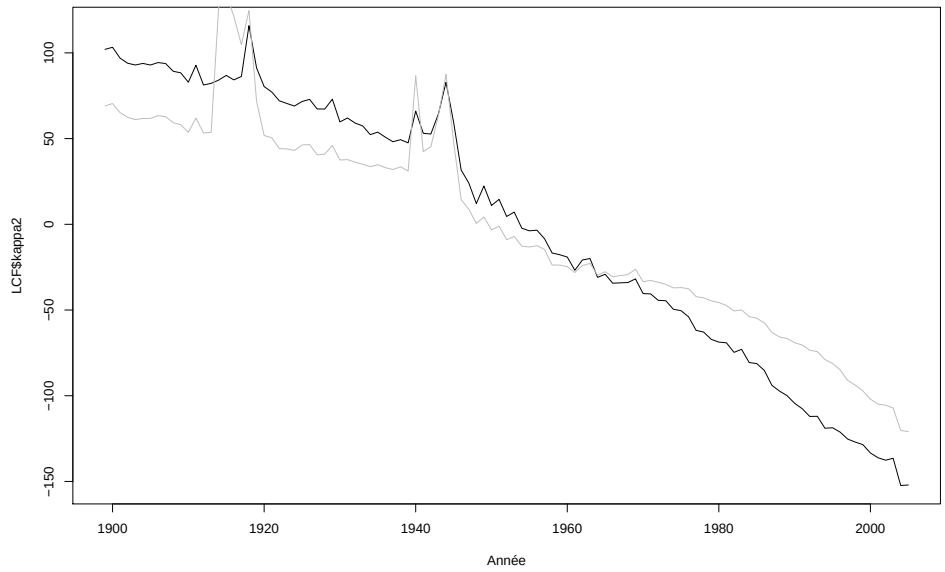


FIGURE 5.14 – Evolution de $t \mapsto \kappa_t$ pour les Hommes - en trait sombre - et pour les Femmes - en trait grisé.

```
>  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))
```

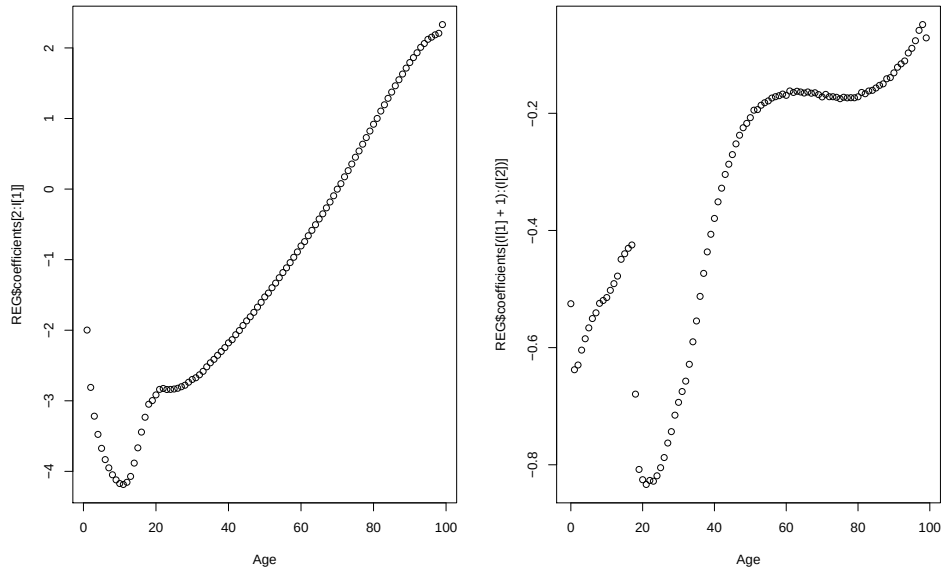


FIGURE 5.15 – Evolution de $x \mapsto \alpha_x$ (à gauche) et $x \mapsto \beta_x$ (à droite) pour les Hommes, en France.

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],
```

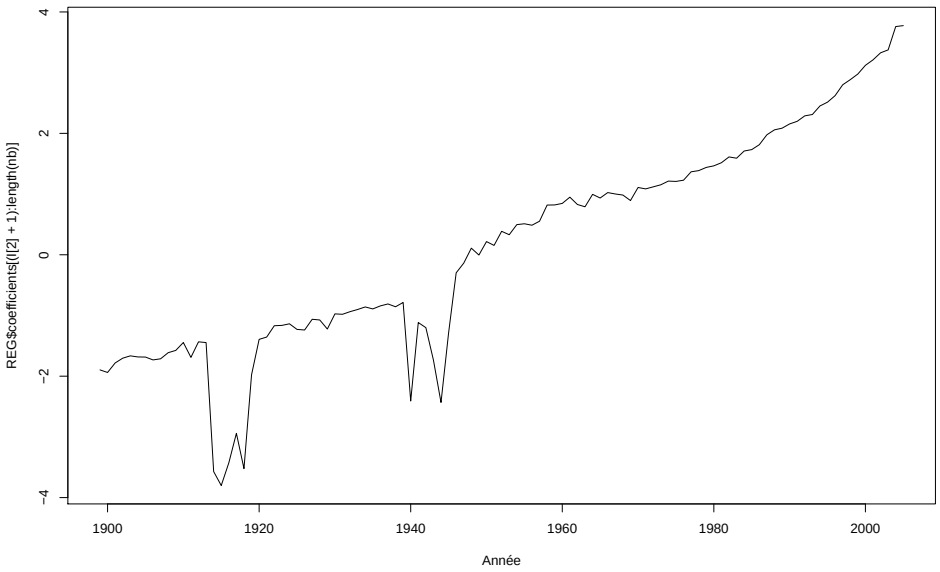


FIGURE 5.16 – Evolution de $t \mapsto \kappa_t$ pour les Hommes, en France.

```
+ 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,

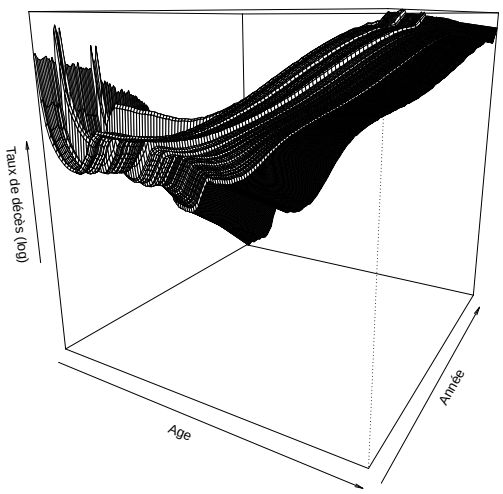


FIGURE 5.17 – Evolution de $(x,t) \mapsto \log \widehat{\mu}_{x,t}$ pour les Hommes, en France.

```
> 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

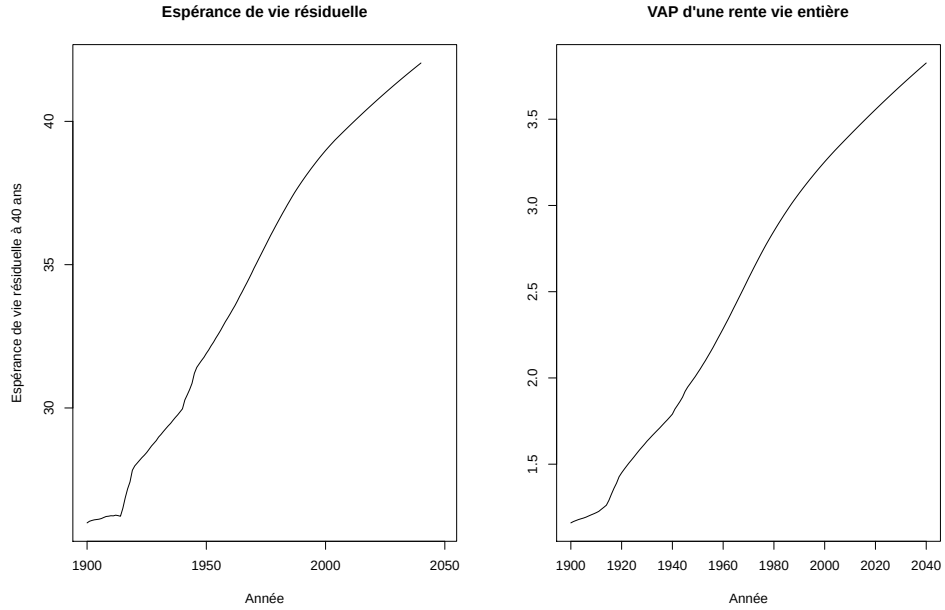


FIGURE 5.18 – Evolution de l'espérance de vie résiduelle pour les Hommes de 40 ans, en France, à gauche, et évolution de la valeur actuelle probable d'une rente vie entière différée achetée l'année t par un assuré de 40 ans.

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))
```

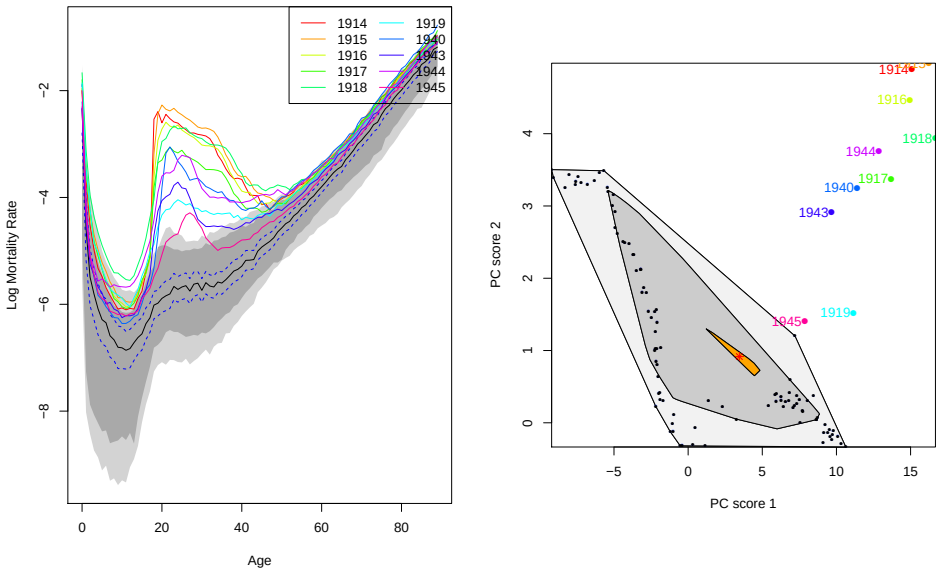


FIGURE 5.19 – Détection d'années 'aberrantes' dans le modèle de Lee-Carter.

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.*

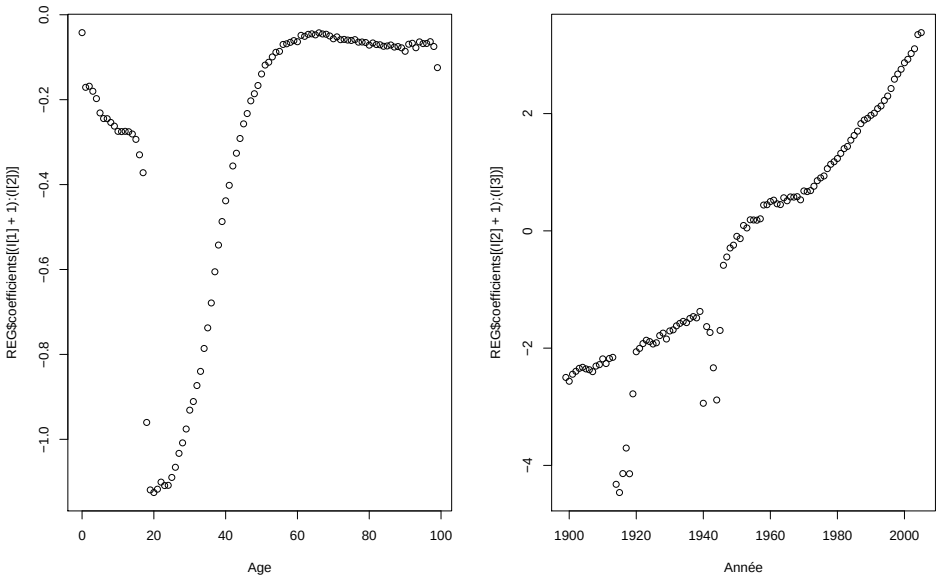


FIGURE 5.20 – Evolution des coefficients β_x et κ_t pour les Hommes en France dans le modèle avec un effet cohorte.

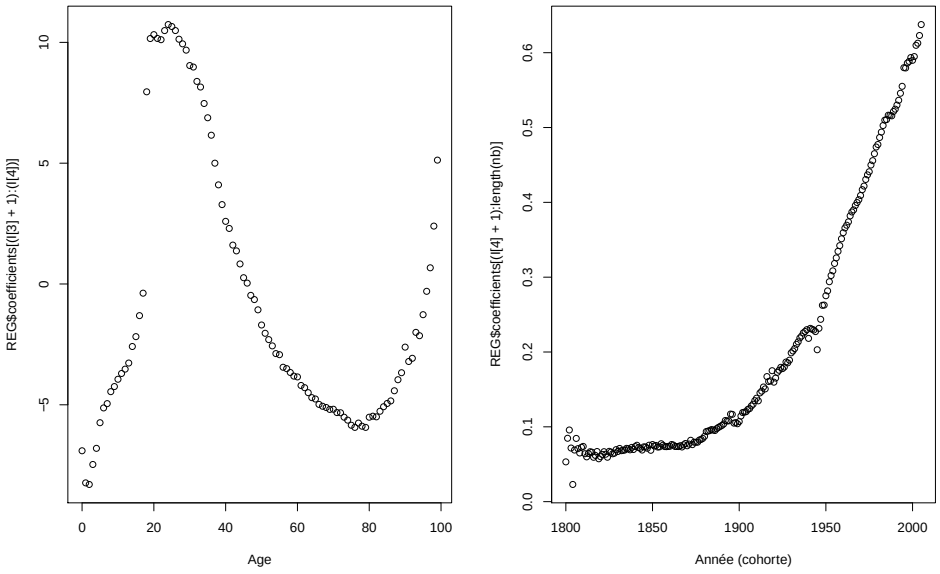


FIGURE 5.21 – Evolution des coefficients γ_x et δ_{t-x} pour les Hommes en France dans le modèle avec un effet cohorte.

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)))
```

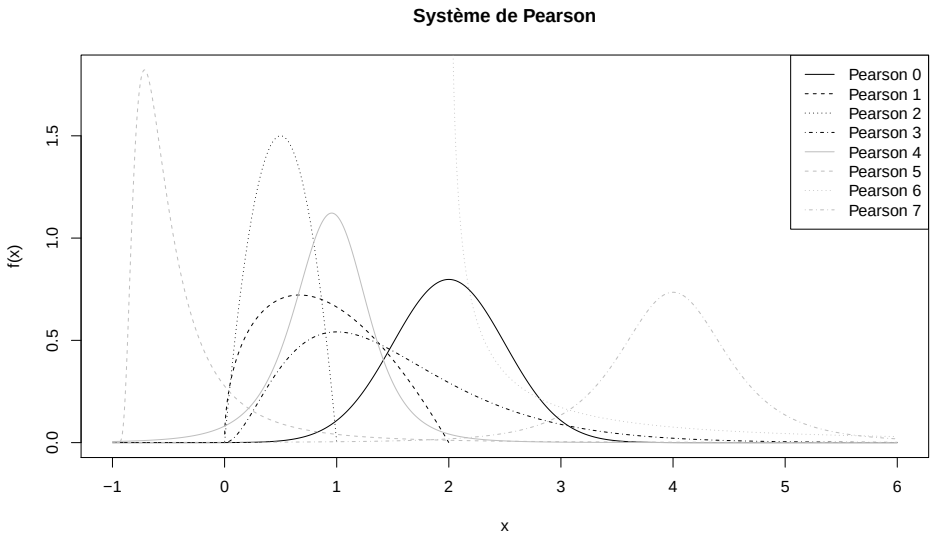


FIGURE A.1 – Système de Pearson et formes de principales densités.

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

additif, 68
AIC, 52
algorithme récursif, 147
année, 164
annuité vie entière, 138
approximation, 12
arbre, 53
ARIMA, 168
A_x, 140, 148, 149
$\ddot{a}_x$, 138, 148
${}_nDA_x$, 154
${}_nIA_x$, 154
$a_{x:n\urcorner}$, 148

Bailey, 63
Baldacci, 151
bayésien
    provisions, 131
benefit premium, 155
Benktander, 127
BIC, 52
binomiale, 52
binomiale négative, 7, 8, 118
boni-mali, 94, 106
bonus-malus, 78
bootstrap, 116
Borhutter-Ferguson, 126
Box-Cox, 121

calendaire, 164
Cape Code, 128
capital différé, 137
carte, 66
Chain Ladder, 94, 96
charge ultime, 95, 128
chi-deux, 55, 90
Cholesky, 192
claims development result, 94, 106
classification and regression tree, 53
cohorte, 181
colonne, 112
convolution, 124
copules, 14, 27, 30, 155
    Archimédiennes, 15
    elliptiques, 15
    extrêmes, 15
    mélange, 29
corrélation, 123
cumuls
    nombres, 91
    paiements, 91

décés, 140
développement, 94
déviance, 51
diagonale, 164
dispersion, 41
dossier-dossier, 91, 101

ecrêtement, 86
entropie, 53
Epanechnikov, 17
error
    process, 116
    variance, 116
espérance, 19
espérance de vie, 178
espérance limitée, 19
esprérance de vie, 149
estimation
    méthode des moments, 24
    méthode des quantiles, 26
    maximum de vraisemblance, 21
    non-paramétrique, 16
    paramétrique, 20, 43
${}_kE_x$, 137
e_x, 135, 149, 178
expert, 126
exposition, 39, 63, 66, 159, 172

facteur, 63, 108, 174
facteurs de transition, 94
Fast Fourier Transform, 12
F_n, 16
f_n, 16

gamma, 2, 5, 118, 122
generalized additive models, 69
generalized linear models, 39
Gibbs, 130
Gini, 53
Glivenko-Cantelli, 17
Gompertz, 155

hétérogénéité, 37
histogramme, 17, 35

IA_x, 155
IBNR, 93
incréments
    négatifs, 113
    paiements, 91
inflation de zéros, 10, 78
interpolation, 151

$\mathcal{L}$, 20
λ_j, 94
λ_∞, 100
Lee-Carter, 166
    α_x, 166, 172, 174, 177
    β_x, 166, 172, 174, 177
    κ_t, 166, 168, 172, 174
    résidus, 171
Lexis, 164
lien, 41
LifeMetrics, 172
ligne, 112
lissage, 16, 72
log-linéaire, 109
log-normale, 2, 123
logit, 52, 174
loi
    Beta, 185
    beta, 5
    binomiale, 7, 40, 52
    binomiale négative, 7, 73, 77, 118
    Cauchy, 186
    chi-deux, 2
    composée, 11, 37
    continue, 3
    discrete, 7
    Erlang, 2
    exp, 31
    exponentielle, 23, 25, 26, 81, 185
    expontielle, 35
    famille exponentielle, 40, 121
    gamma, 2, 5, 6, 23, 25, 31, 41, 42, 80, 118, 121, 185
    inverse Gaussienne, 186
    log-normale, 2, 35, 80, 109, 123, 186
    MBBEFD, 10
    normale, 14, 40, 42, 80, 185
    Pareto, 6, 14, 23, 25, 35, 85
    Poisson, 7, 9, 40, 42, 62, 73, 108, 110, 118, 121
    Poisson composée, 121
    quasi-Poisson, 118
    simulations, 12
    Student, 186
    tronquée, 9
    Tweedie, 121
    Weibull, 3, 5
    zéro-modifiée, 10
longitudinale, 164
loss ratio, 128
L_x, 134

méthode des marges, 63, 110
Mack, 98
Markov, 98, 135
maximisation, 21, 43
mean squared error, 97, 115
Merz & Wüthrich, 106
moindres carrés, 87, 166
moment, 19
μ, 19
$\mu_{x,t}$, 160, 174

Newton-Raphson, 23, 43
normale, 14
noyau, 16

offset, 66

paiements
    cumulés, 91
    incréments, 91
Panjer, 11

Pareto, 14
Pearson, 185
Poisson, 7, 9, 110, 122
    processus, 192
`predict`, 119
prime
    acquise, 91
    pure, 37, 38, 141, 142, 145
probabilité
    décés, 134, 159
    survie, 134
probit, 52
provisions mathématiques, 141, 155, 157
    itérative, 142, 144, 147
    prospective, 142, 143, 145
    retrospective, 142, 143, 146
provisions pour sinistres à payer, 91
pseudo triangle, 118
${}_kp_x$, 134, 150, 159
${}_kp_{xy}$, 152, 155
${}_hp_{\overline{xy}}$, 152

quantile, 120, 124
quasi-Poisson, 118
${}_kq_x$, 134, 151

réassurance, 11, 30, 85
résidus, 50, 98, 116, 126, 171
Renshaw, 115
rente, 145
    vie entière, 179
reserves, 96

`S4`, 150
σ_j^2, 98
simulations, 187
splines, 57, 68
surdispersion, 73, 111, 118

table
    prospective, 159
    rectangularisation, 162
    TD88-90, 134, 150
    TGF-05, 136
    TGH-05, 136
    TV88-90, 134, 150
tail factor, 100
Taylor, 108
tempête, 30
temporaire décès, 142
test, 48
transervale, 164
triangle, 91
    automobile, 91
    corporel, 123
    matériel, 123
Tweedie, 122

valeur actuelle probable, 133, 141
variance, 41
vraisemblance, 20, 28, 43, 51
${}_kV_x$, 142–145, 147, 150
de Vylder, 109

Weibull, 3

Index des commandes

actuar, 11, 17
AER, 75
ageconducteur, 38
agevehicule, 38
aggregateDist, 11
AIC, 29, 52
aod, 52
arima, 168
as.factor, 109, 174, 181
auto.arima, 167, 169
Axn, 154
axn, 154

baseCOUT, 38
baseFREQ, 38
bayes-triangle, 130
BIC, 29, 52
binomiale, 40
BMA, 52
BootChainLadder, 120
boxplot, 130
bs, 57, 68, 77, 82

car, 48
carburant, 38
ChainLadder, 100
Chainladder, 96, 116
chol, 192
contrat, 38
convolution, 11
convolve, 12
cut, 46, 63

danish, 35
DAxn, 154
Deces, 159
Expo, 174
demogdata, 166
demography, 166
density, 17, 130
dental, 17
deviance, 50
dispersiontest, 76
dispomod, 49
distr, 123
dlnorm, 124
dtx, 172

ecdf, 17
esp.vie, 136
ets, 167
etx, 172
evir, 35
Exn, 153
Expo, 159
ext, 151
extractAIC, 52
exyt, 153

factor, 109, 174, 181
fboxplot, 181
fit701, 172
fitdist, 26
fitdistrplus, 23, 26
forecast, 166, 167
fts, 181
ftweedie, 121

gam, 69, 70
gamlss, 78, 85
gamlss, 49
gamma, 41, 80, 122
gaussian, 40, 80
gini, 57, 60
glm, 40–42, 44, 66, 67, 110, 111, 122, 126
glm.nb, 77
gmlss, 72
gnm, 174, 181
goodfit, 62
gss, 72

IAxn, 154
INCURRED, 92
is.na, 119

knots, 17

lca, 166
lifecontingencies, 150
lifetable, 169
linearHypothesis, 48, 75
lm, 82, 95, 109
lm.disp, 49
log-normal, 80
logLik, 51
loi
    mixte, 10

MackChainLadder, 100
MunichChainLadder, 104
MackMerzWuthrich, 107
maps, 66
maptools, 66
MASS, 76, 77
mean, 79
merge, 38
mgcv, 72
mincut, 56
mledist, 23
model.matrix, 115

NA, 92
NUMBER, 92

offset, 66, 159, 174, 181
offset**offset**, 45
optimize, 23, 122

PAID, 92
pearson, 50, 126
PearsonDS, 186
persp, 160
poisson, 40, 44, 66, 110, 122, 174
predict, 42, 52, 56, 68, 70, 72, 82, 112, 113
PREMIUM, 92, 127
PtProcess, 192
pxt, 150
pxyt, 152

qmedist, 26
quantile, 120
quasipoisson, 73, 111, 118, 174, 181
qxt, 151

readShapeSpatial, 66
region, 38
residuals, 50, 115, 171
rqpoisBN, 118
rqpoisG, 118

S4, 124
sample, 116
set.seed, 119, 190
sigma, 110
sinistre, 38
splines, 73
summary, 17, 82, 109

tapply, 63, 68
TD, 134, 150
TGF, 136
TGH, 136
tree, 56, 57, 60
TV, 134, 150
tweedie, 122

vcd, 62
Vectorize, 114, 137

weights, 95

ZAP, 78
zeroinfl, 78
ZINBI, 78
ZIP, 78