

STATISTIQUE ET ÉCONOMÉTRIE APPLIQUÉES DES
VARIABLES DE DURÉE

Emmanuel DUGUET

Février 2014

Sommaire

| | |
|--|-----------|
| 1 Les variables de durée | 3 |
| 1.1 Fonction de répartition | 3 |
| 1.2 Fonction de densité | 3 |
| 1.3 Fonction de survie | 4 |
| 1.4 Fonction de hasard | 6 |
| 1.5 Relations entre les définitions | 8 |
| 1.6 Les distributions en logarithmes | 9 |
| 2 Distributions usuelles | 11 |
| 2.1 Distribution à un paramètre | 12 |
| 2.1.1 Loi exponentielle | 12 |
| 2.1.2 Loi Gamma à un paramètre | 13 |
| 2.1.3 Loi de Lomax à un paramètre | 14 |
| 2.2 Distributions à deux paramètres | 15 |
| 2.2.1 Loi exponentielle avec état absorbant | 15 |
| 2.2.2 Loi de Weibull | 16 |
| 2.2.3 Loi Gamma à deux paramètres | 18 |
| 2.2.4 Loi log-normale | 19 |
| 2.2.5 Loi log-logistique | 21 |
| 2.2.6 Loi de Pareto | 23 |
| 2.3 Autres distributions | 25 |
| 2.3.1 Mélange discret de lois exponentielles | 25 |
| 2.3.2 Loi de Burr de type 12 | 29 |
| 2.3.3 Loi Gamma généralisée | 31 |
| 3 Estimation paramétrique : théorie | 39 |
| 3.1 Maximum de vraisemblance | 39 |
| 3.2 Ecritures de la vraisemblance | 41 |
| 3.3 Changements de paramètres | 42 |

| | | |
|----------|---|-----------|
| 3.4 | Statistiques utiles | 44 |
| 3.5 | Estimation des fonctions | 44 |
| 4 | Estimation paramétrique : pratique | 47 |
| 4.1 | Loi exponentielle | 47 |
| 4.1.1 | Log-vraisemblance | 47 |
| 4.1.2 | Estimation | 49 |
| 4.1.3 | Statistiques utiles | 49 |
| 4.1.4 | Programmation I | 51 |
| 4.1.5 | Application I | 58 |
| 4.2 | Loi Gamma à un paramètre | 66 |
| 4.2.1 | Log-vraisemblance | 66 |
| 4.2.2 | Estimation | 66 |
| 4.2.3 | Statistiques utiles | 67 |
| 4.2.4 | Programmation II | 68 |
| 4.2.5 | Application II | 75 |
| 4.3 | Loi de Lomax à un paramètre | 80 |
| 4.3.1 | Log-vraisemblance | 80 |
| 4.3.2 | Estimation | 80 |
| 4.3.3 | Statistiques utiles | 81 |
| 4.3.4 | Programmation III | 82 |
| 4.3.5 | Application III | 86 |
| 4.4 | Loi exponentielle avec état absorbant | 90 |
| 4.4.1 | Log-vraisemblance | 90 |
| 4.4.2 | Estimation | 91 |
| 4.4.3 | Statistiques utiles | 92 |
| 4.4.4 | Application IV | 92 |
| 4.5 | Loi de Weibull | 110 |
| 4.5.1 | Log-vraisemblance | 110 |
| 4.5.2 | Estimation | 113 |
| 4.5.3 | Statistiques utiles | 114 |
| 4.5.4 | Programmation V | 116 |
| 4.5.5 | Application V | 122 |
| 4.6 | Loi Gamma à deux paramètres | 127 |
| 4.6.1 | Log-vraisemblance | 127 |
| 4.6.2 | Estimation | 127 |
| 4.6.3 | Statistiques utiles | 128 |
| 4.6.4 | Programmation VI | 129 |
| 4.6.5 | Application VI | 135 |
| 4.7 | Loi log-normale | 139 |

| | | |
|----------|-------------------------------------|------------|
| 4.7.1 | Log-vraisemblance | 139 |
| 4.7.2 | Estimation | 140 |
| 4.7.3 | Statistiques utiles | 141 |
| 4.7.4 | Programmation VII | 142 |
| 4.7.5 | Application VII | 148 |
| 4.8 | Loi log-logistique | 151 |
| 4.8.1 | Log-vraisemblance | 151 |
| 4.8.2 | Estimation | 151 |
| 4.8.3 | Statistiques utiles | 153 |
| 4.8.4 | Programmation VIII | 155 |
| 4.8.5 | Application VIII | 161 |
| 4.9 | Loi de Pareto | 165 |
| 4.9.1 | Log-vraisemblance | 165 |
| 4.9.2 | Estimation | 166 |
| 4.9.3 | Statistiques utiles | 166 |
| 4.9.4 | Programmation IX | 167 |
| 4.9.5 | Application IX | 173 |
| 4.10 | Loi Gamma généralisée | 177 |
| 4.10.1 | Estimation | 177 |
| 4.10.2 | Statistiques utiles | 178 |
| 4.10.3 | Programmation X | 180 |
| 4.10.4 | Application X | 185 |
| 4.11 | Loi de Burr de type 12 | 201 |
| 4.11.1 | Log-vraisemblance | 201 |
| 4.11.2 | Estimation | 202 |
| 4.11.3 | Statistiques utiles | 203 |
| 4.11.4 | Programmation XI | 204 |
| 4.11.5 | Application XI | 209 |
| 5 | Hasard constant par morceaux | 223 |
| 5.1 | Le modèle | 224 |
| 5.2 | Log-vraisemblance | 227 |
| 5.3 | Estimation | 229 |
| 5.4 | Statistiques utiles | 230 |
| 5.5 | Programmation XII | 230 |
| 5.6 | Application XII | 242 |

| | | |
|----------|--|------------|
| 6 | Estimation non paramétrique | 253 |
| 6.1 | Estimateur de Kaplan-Meier | 253 |
| 6.1.1 | Formule de Greenwood | 258 |
| 6.1.2 | Estimateur de Breslow | 259 |
| 6.1.3 | Statistiques utiles | 261 |
| 6.1.4 | Programmation XIII | 262 |
| 6.1.5 | Application XIII | 271 |
| 6.2 | Estimateur actuariel | 280 |
| 6.2.1 | Dérivation | 280 |
| 6.2.2 | Programmation XIV | 283 |
| 6.2.3 | Application XIV | 285 |
| 7 | Introduction des variables explicatives | 291 |
| 7.1 | L'effet des variables explicatives | 291 |
| 7.1.1 | Modèles à hasards proportionnels | 291 |
| 7.1.2 | Modèles à durée accélérée | 294 |
| 7.1.3 | Ecriture en logarithmes | 296 |
| 7.2 | Aspects appliqués | 296 |
| 7.2.1 | Interprétation des coefficients | 296 |
| 7.2.2 | Terme constant et centrage | 297 |
| 7.3 | Estimation du modèle exponentiel | 300 |
| 7.3.1 | Programmation XV | 301 |
| 7.3.2 | Application XV | 302 |
| 7.4 | Estimation du modèle de Weibull | 312 |
| 7.4.1 | Paramétrage à hasards proportionnels | 313 |
| 7.4.2 | Paramétrage à durée accélérée | 313 |
| 7.4.3 | Programmation XVI | 314 |
| 7.4.4 | Application XVI | 316 |
| A | Rappels de statistique | 327 |
| A.1 | Espérance mathématique et fonction de survie | 327 |
| A.2 | Le changement de variable | 327 |
| A.3 | Constante d'Euler | 328 |
| A.4 | Fonctions Gamma, Beta et Polygamma | 329 |
| A.4.1 | Fonction Gamma | 329 |
| A.4.2 | La fonction Beta | 330 |
| A.4.3 | La fonction Polygamma | 330 |
| A.4.4 | La fonction Gamma tronquée | 330 |
| A.5 | Loi Gamma à deux paramètres | 331 |

| | | |
|----------|--|------------|
| B | Calcul des moments | 333 |
| B.1 | Approche par la fonction génératrice des moments | 333 |
| B.1.1 | Moments non centrés | 333 |
| B.1.2 | Moments centrés | 334 |
| B.1.3 | Moments du logarithme | 334 |
| B.1.4 | Loi exponentielle | 335 |
| B.1.5 | Loi de Gumbel | 336 |
| B.1.6 | Loi Gamma à un paramètre | 337 |
| B.1.7 | Loi Gamma à deux paramètres | 338 |
| B.1.8 | Loi de Weibull | 338 |
| B.1.9 | Loi Gamma généralisée | 340 |
| B.1.10 | Loi normale | 341 |
| B.1.11 | Loi log-normale | 342 |
| B.1.12 | Résumé | 343 |
| B.2 | Approche par le calcul direct | 343 |
| B.2.1 | Loi de Pareto | 344 |
| B.2.2 | Loi log-logistique | 346 |
| B.2.3 | Loi de Burr de type 12 | 347 |
| C | Modélisation en logarithmes | 349 |
| C.1 | Distributions à un paramètre | 349 |
| C.1.1 | Modèle exponentiel et loi de Gumbel | 349 |
| C.1.2 | Modèle exponentiel et loi exponentielle | 351 |
| C.1.3 | Modèle Gamma à un paramètre | 352 |
| C.2 | Distributions à deux paramètres | 352 |
| C.2.1 | Modèle de Weibull | 352 |
| C.2.2 | Modèle Gamma à deux paramètres | 353 |
| C.2.3 | Modèle log-normal | 354 |
| C.2.4 | Modèle log-logistique | 355 |
| C.2.5 | Modèle de Pareto | 356 |
| C.3 | Modèle Gamma à trois paramètres | 356 |
| D | Propriétés du modèle exponentiel | 359 |
| D.1 | Propriété utile pour la statistique du score | 359 |
| D.2 | Matrice d'information | 360 |
| E | Les algorithmes d'optimisation | 363 |
| E.1 | Présentation des algorithmes | 363 |
| E.2 | Les méthodes de gradient | 364 |
| E.3 | Algorithme de Newton-Raphson | 365 |

| | | |
|-----------------------------|---|------------|
| E.4 | Algorithme de Berndt-Hall-Hausman | 366 |
| E.5 | Algorithme du score | 367 |
| E.6 | Méthodologie de programmation | 367 |
| Table des Graphiques | | 371 |
| Table des Tableaux | | 373 |

Introduction

ON RENCONTRE DES VARIABLES DE DURÉE dans de nombreux cas. A l'origine, ces modèles ont été développés pour étudier la durée de vie mais d'autres applications ont été mises en œuvre. En économie, on étudie la durée passée au chômage, dans un emploi ou entre deux emplois, la durée d'un trajet de transport, la durée de vie d'une entreprise ou encore la durée d'un crédit de type "revolving". Or les variables de durée ont des caractéristiques particulières : elles sont strictement positives et souffrent souvent de problèmes de censure. En effet, l'arrêt de la collecte à une date fixe ou après une certaine durée fait que des durées commencées n'ont pas eu le temps de se terminer et sont donc censurées. On peut juste affecter une valeur minimale à ces durées observées de manière incomplète. On parle de censure droite. Inversement, il est possible que le fichier disponible commence à une date où le processus observé a déjà commencé pour certains individus, la durée est alors censurée à gauche. Pour obtenir une bonne estimation, il faut tenir compte de toutes les observations, censurées ou non. En effet, plus une durée est longue plus elle a de chances d'être censurée, de sorte qu'enlever les durées censurées revient à causer un biais de sélection. Par exemple, si l'on étudie la durée du chômage, enlever les données censurées reviendrait à réaliser une étude sans les chômeurs de longue durée, ce qui est difficilement envisageable.

Comme pour les variables aléatoires habituelles, on peut définir la loi d'une variable de durée par sa fonction de répartition. Toutefois, on préfère pour des raisons pratiques, utiliser d'autres concepts plus parlants que la fonction de répartition ou la densité. Cette pratique provient de la démographie et utilise donc des concepts spécifiques comme le taux de mortalité, la probabilité de survie ou l'espérance de vie à la naissance. Nous allons montrer que ces concepts sont rigoureusement équivalents à ceux utilisés dans les autres branches de l'économétrie.

Cet ouvrage vise à présenter toutes les bases nécessaires à la pratique de l'économétrie des variables de durées. Nous commencerons, dans un premier chapitre, par présenter les concepts qui sont utilisés en statistique des variables

de durée et qui s'écartent parfois des pratiques habituelles tout en leur restant équivalentes. Dans le chapitre 2, nous verrons les distributions usuelles et leurs propriétés principales. Ces distributions constituent un ingrédient de base des modèles paramétriques d'économétrie des durées. Nous présenterons ensuite, dans le chapitre 3, une synthèse sur l'estimation des paramètres des distributions usuelles. Le chapitre 4 sera entièrement consacré à l'estimation de ces modèles usuels avant d'aborder les questions d'économétrie appliquée. Ce chapitre constitue une véritable introduction à SAS-IML et propose 11 applications différentes liées à l'estimation des modèles de durée. Nous introduisons progressivement les méthodes d'optimisation non linéaires qui sont nécessaires pour estimer les modèles dont les durées sont censurées. Chaque méthode donne lieu à une application sous SAS-IML. Ce chapitre est essentiel pour bien comprendre l'estimation des modèles de durée et peut aussi être utilisé comme une introduction à l'économétrie non linéaire.

Nous présenterons ensuite, dans le chapitre 5, le cas le plus populaire d'estimation semi paramétrique : le modèle à hasard constant par morceaux. Cette modélisation permet de mieux choisir un modèle de durée car elle fait des hypothèses relativement faibles sur la distribution des durées. Ce chapitre se poursuit assez naturellement par le chapitre 6 sur l'estimation non paramétrique des modèles de durée, dont le contenu est important pour comprendre certaines méthodes d'estimation semi-paramétriques avec variables explicatives. Nous verrons l'estimateur de Kaplan-Meier et l'estimateur actuariel. Les applications de ce chapitre permettront de voir à la fois une programmation directe en SAS-IML et avec la procédure *lifetest* de SAS.

Enfin, nous ne pouvons pas finir cet ouvrage sans proposer une estimation avec variables explicatives. Nous verrons les modèles exponentiel et de Weibull. Nous les avons choisis pour deux raisons. D'une part, leur simplicité permet de se focaliser sur les nouveautés engendrées par l'introduction des variables explicatives dans un modèle de durée. D'autre part, ces deux modèles sont à la fois à hasards proportionnels et à durée accélérée, deux grandes familles de modèles de durée, de sorte que nous n'avons pas besoin de choisir entre ces deux modélisations. Les applications seront présentées à la fois sous SAS-IML et avec la procédure *lifereg* de SAS.

CHAPITRE 1

Les variables de durée

1.1 Fonction de répartition

CONSIDÉRONS une variable aléatoire de durée $T > 0$. La distribution de cette variable de durée est entièrement résumée par sa fonction de répartition. Elle est définie par la probabilité que cette durée soit inférieure à une valeur donnée t :

$$F_T(t) = \Pr[T \leq t], \quad t \in \mathbb{R}_+^*, \quad 0 \leq F_T(t) \leq 1,$$

cette fonction est croissante avec la durée.

Exemple 1.1. Si T désigne la durée du chômage en mois, la fonction de répartition $F_T(t)$ représente la probabilité de rester au plus t mois au chômage à partir de la date d'inscription.

Exemple 1.2. Si T désigne la longueur d'un appel téléphonique en secondes, la fonction de répartition $F_T(t)$ représente la probabilité qu'un appel dure moins de t secondes.

1.2 Fonction de densité

La densité de la durée est donnée par :

$$f_T(t) = \frac{dF_T(t)}{dt} = \lim_{\Delta t \rightarrow 0} \frac{1}{\Delta t} \Pr[t < T \leq t + \Delta t],$$
$$f_T(t) \geq 0,$$

elle représente l'intensité d'occurrence d'une durée située au voisinage de t . Cette intensité d'occurrence peut être supérieure à l'unité car il ne s'agit pas

d'une probabilité mais d'une densité. La probabilité correspondante se calcule sur un petit intervalle de temps $[t, t + \Delta t]$, elle est donnée par :

$$\Pr[t < T \leq t + \Delta t] \simeq f_T(t)\Delta t.$$

et l'on vérifie que les probabilités somment bien à l'unité (quand $\Delta t \rightarrow 0$) :

$$\int_0^{+\infty} f_T(t) dt = 1.$$

La densité permet aussi de caractériser la loi de T car on en déduit la fonction de répartition de la manière suivante :

$$F_T(t) = \int_0^t f_T(x) dx.$$

Exemple 1.3. Si T est la durée du chômage, la quantité $f_T(t)\Delta t$ représente la probabilité que la durée du chômage soit comprise entre t et $t + \Delta t$, c'est-à-dire qu'elle soit à peu près égale à t mois.

Exemple 1.4. Si T est la durée d'un appel téléphonique, la quantité $f_T(t)\Delta t$ représente la probabilité que l'appel dure à peu près t secondes $t \in [t, t + \Delta t]$, $\Delta t \rightarrow 0$.

1.3 Fonction de survie

Les concepts précédents ne sont pas toujours les plus pratiques pour l'interprétation des variables de durée. L'économétrie des durées utilise plus souvent la *fonction de survie* $S_T(t)$ au concept de fonction de répartition. La fonction de survie est définie par la probabilité que la durée soit supérieure à une valeur donnée t :

$$S_T(t) = \Pr[T > t] = 1 - F_T(t),$$

son nom vient de la démographie, elle donne la fraction des individus ayant vécu au moins jusqu'à l'âge t . Si l'on raisonne sur une génération d'individus, il s'agit de la part de cette génération qui a dépassé l'âge t . La fonction de survie est positive, décroissante et admet les limites suivantes dans les cas habituels :

$$\lim_{t \rightarrow 0} S_T(t) = 1, \quad \lim_{t \rightarrow +\infty} S_T(t) = 0,$$

toute la population est en vie en $t = 0$ et plus personne n'est en vie en $t = \{+\infty\}$.

Exemple 1.5. Si T est la durée du chômage, la fonction de survie $S_T(t)$ représente la probabilité qu'un travailleur reste au moins t mois au chômage. Le chômage de longue durée se définit comme une période de chômage de plus d'un an, donc si la durée du chômage est mesurée en mois, $S_T(12)$ est la probabilité du chômage de longue durée.

Exemple 1.6. Si T est la durée d'un appel téléphonique, la fonction de survie $S_T(t)$ représente la probabilité qu'un appel dure plus de t secondes.

La densité est reliée à la fonction de survie par la relation :

$$f_T(t) = \frac{dF_T(t)}{dt} = \frac{d}{dt}(1 - S_T(t)) = -\frac{dS_T(t)}{dt}.$$

Cette modélisation n'est toutefois pas suffisante, car il arrive que, dans certains contextes, certains individus au sens statistique décèdent à la naissance, ce qui implique $\lim_{t \rightarrow 0} S_T(t) < 1$, et que certains individus ne décèdent "jamais", auquel cas on peut avoir $\lim_{t \rightarrow +\infty} S_T(t) > 0$. Les individus statistiques en question peuvent être, par exemple, des objets manufacturés dont une partie présente un défaut dès la phase de production, et dont une autre partie possède une durée de vie trop longue pour être observée par les procédures de collecte disponible, ce qui justifie une approximation par une durée "infinie".

Lorsqu'une proportion $\pi > 0$ d'individus ne sort jamais de l'état étudié, on modifie la fonction de survie de la manière suivante :

$$S_\pi(t) = \pi + (1 - \pi)S_T(t),$$

on a alors :

$$\begin{aligned} \lim_{t \rightarrow 0} S_\pi(t) &= \pi + (1 - \pi) \times 1 = 1 \\ \lim_{t \rightarrow +\infty} S_\pi(t) &= \pi + (1 - \pi) \times 0 = \pi, \end{aligned}$$

et l'on travaille sur $S_\pi(t)$ à la place de $S_T(t)$. L'état dont on ne sort jamais est appelé état absorbant. On peut donc introduire un état absorbant dans un modèle à partir de n'importe quelle distribution de durée classique.

Exemple 1.7. Si T est la durée d'un appel téléphonique, on a $\pi = 0$ car une conversation téléphonique a toujours une fin.

Exemple 1.8. L'enquête Santé et Itinéraires Professionnels permet de dater les handicaps. Ceux-ci peuvent être transitoires (e.g., avoir une jambe cassée) ou permanents (e.g., être aveugle). Dans ce cas π représente la probabilité d'avoir un handicap permanent et $S_T(t)$ représente la distribution des durées des handicaps transitoires. Si on considère l'ensemble des handicaps, on doit donc avoir $\pi > 0$.

Exemple 1.9. On peut introduire un état absorbant dans un modèle de durée du chômage. Dans ce cas π s'interprète comme la probabilité de ne jamais sortir du chômage (ou après une durée très longue, à titre d'approximation). On peut penser que π représente le chômage de découragement, associé à un arrêt de la recherche d'emploi, alors que $S_T(t)$ représente la distribution des durées de chômage des travailleurs qui recherchent toujours activement un emploi.

1.4 Fonction de hasard

La *fonction de hasard* représente l'intensité d'occurrence de la sortie instantanée de l'état étudié, après une durée t passée dans cet état. Elle joue un rôle central en économétrie des durées. Elle est définie comme la probabilité conditionnelle que le phénomène se termine après une durée t sachant que l'on a atteint cette durée. Elle correspond au concept de taux de mortalité, défini comme la probabilité de décéder à l'âge t sachant que l'on a atteint cet âge. En effet, un taux de mortalité à la date t se calcule toujours sur la population survivante à cette date (donc $T > t$), et non sur toute la population d'une même génération (équivalent à un conditionnement $T > 0$), d'où la nécessité du conditionnement. On obtient :

$$\begin{aligned}
 h_T(t) &= \lim_{\Delta t \rightarrow 0} \frac{1}{\Delta t} \Pr[t < T \leq t + \Delta t | T > t] \\
 &= \lim_{\Delta t \rightarrow 0} \frac{1}{\Delta t} \frac{\Pr[(t < T \leq t + \Delta t) \cap (T > t)]}{\Pr[T > t]} \\
 &= \lim_{\Delta t \rightarrow 0} \frac{1}{\Delta t} \frac{\Pr[t < T \leq t + \Delta t]}{\Pr[T > t]} \\
 &= \frac{1}{\Pr[T > t]} \underbrace{\lim_{\Delta t \rightarrow 0} \frac{1}{\Delta t} \Pr[t < T \leq t + \Delta t]}_{f(t)} \\
 &= \frac{f_T(t)}{S_T(t)} \geq 0.
 \end{aligned}$$

On en déduit que :

$$\begin{aligned}
 h_T(0) &= f_T(0), \\
 h_T(t) &\geq f_T(t).
 \end{aligned}$$

Remarque 1.1. Sur un graphique, la fonction de hasard est toujours située au dessus de la densité parce que $0 < S_T(t) \leq 1$.

Ceci permet également de calculer l'équivalent du taux de mortalité instantané en démographie, sur un intervalle de temps $[t, t + \Delta t]$, qui est égal à :

$$\Pr[t < T \leq t + \Delta t | T > t] \simeq h_T(t)\Delta t.$$

Le hasard s'exprime complètement en fonction de la survie :

$$h_T(t) = \frac{f_T(t)}{S_T(t)} = -\frac{1}{S_T(t)} \frac{dS_T(t)}{dt} = -\frac{d \ln S_T(t)}{dt}.$$

Exemple 1.10. Si T est la durée du chômage mesurée en mois, la quantité $h_T(t)\Delta t$ représente la probabilité de sortir du chômage entre les durées t et $t + \Delta t$, pour les travailleurs qui sont au chômage depuis t mois.

Exemple 1.11. Si T est la durée d'un appel téléphonique, la quantité $h_T(t)\Delta t$ mesure la probabilité que la conversation s'interrompe après t secondes.

Exemple 1.12. Si T est la durée d'un handicap, $h_T(t)\Delta t$ représente la probabilité de guérison après un handicap qui a durée t années.

Pour caractériser les distributions de durées, on peut aussi utiliser le hasard cumulé, défini par :

$$H_T(t) = \int_0^t h_T(x) dx,$$

le hasard cumulé est étroitement relié à la fonction de survie et peut donc résumer toute une distribution :

$$\begin{aligned} H_T(t) &= \int_0^t h_T(x) dx \\ &= -\int_0^t \frac{d \ln S_T(x)}{dx} dx \\ &= -[\ln S_T(x)]_0^t \\ &= -\ln S_T(t) + \ln S_T(0) \\ &= -\ln S_T(t), \end{aligned}$$

car pour une variable positive $S_T(0) = \Pr[T > 0] = 1$. On en déduit que :

$$S_T(t) = \exp(-H_T(t)),$$

ce qui implique que l'on peut également écrire la densité en fonction du hasard :

$$f_T(t) = h_T(t)S_T(t) = h_T(t)\exp(-H_T(t)).$$

1.5 Relations entre les définitions

On peut définir une distribution de durée par sa fonction de répartition, sa densité, sa survie ou son hasard. Voici les relations entre ces différents concepts.

- Si la distribution est définie par sa fonction de répartition $F_T(t)$, on a :

$$\begin{aligned} f_T(t) &= \frac{dF_T}{dt}(t), \\ S_T(t) &= 1 - F_T(t), \\ h_T(t) &= \frac{1}{1 - F_T(t)} \frac{dF_T}{dt}(t). \end{aligned}$$

- Si la distribution est définie par sa densité, on a :

$$\begin{aligned} F_T(t) &= \int_0^t f_T(x) dx, \\ S_T(t) &= 1 - \int_0^t f_T(x) dx = 1 - F_T(t), \\ h_T(t) &= \frac{f_T(t)}{1 - \int_0^t f_T(x) dx} = \frac{f_T(t)}{1 - F_T(t)}. \end{aligned}$$

- Si la distribution est définie par sa fonction de survie, on a :

$$\begin{aligned} F_T(t) &= 1 - S_T(t), \\ f_T(t) &= -\frac{dS_T}{dt}(t), \\ h_T(t) &= -\frac{1}{S_T(t)} \frac{dS_T}{dt}(t) = -\frac{d \ln S_T(t)}{dt}. \end{aligned}$$

- Si la distribution est définie par sa fonction de hasard, on commence par calculer le hasard cumulé :

$$H_T(t) = \int_0^t h_T(x) dx,$$

et l'on obtient :

$$\begin{aligned} F_T(t) &= 1 - \exp(-H_T(t)), \\ f_T(t) &= h_T(t) \exp(-H_T(t)), \\ S_T(t) &= \exp(-H_T(t)), \\ h_T(t) &= \frac{dH_T}{dt}(t). \end{aligned}$$

1.6 Les distributions en logarithmes

Une pratique courante consiste à modéliser les durées en logarithmes ("log location-scale modeling", en anglais). Les distributions les plus connues sont les distributions log-normale et log-logistique. Dans le cas le plus simple, on dispose d'un paramètre de tendance centrale ou *de position* μ ("location", en anglais), souvent relié à la moyenne, et d'un paramètre *d'échelle* δ ("scale", en anglais), souvent relié à la dispersion de la variable étudiée. Pour définir ces modèles, on utilise une distribution de base, généralement définie sur \mathbb{R} au lieu de \mathbb{R}_+^* , et repérée par l'indice 0. La fonction de survie s'écrit :

$$S_T(t) = S_0\left(\frac{\ln t - \mu}{\delta}\right).$$

Notons ici que la variable :

$$x = \frac{\ln t - \mu}{\delta} \in \mathbb{R},$$

de sorte que la distribution S_0 n'est pas contrainte à posséder un support sur les réels positifs. On en déduit la fonction de répartition :

$$F_T(t) = 1 - S_T(t) = 1 - S_0\left(\frac{\ln t - \mu}{\delta}\right) = F_0\left(\frac{\ln t - \mu}{\delta}\right),$$

avec $F_0(x) = 1 - S_0(x)$. La densité est donnée par :

$$f_T(t) = \frac{1}{\delta t} f_0\left(\frac{\ln t - \mu}{\delta}\right),$$

avec $f_0(x) = F_0'(x)$, et le hasard est donné par :

$$h_T(t) = \frac{f_T(t)}{S_T(t)} = \frac{1}{\delta t} \frac{f_0\left(\frac{\ln t - \mu}{\delta}\right)}{S_0\left(\frac{\ln t - \mu}{\delta}\right)}.$$

En posant :

$$h_0(x) = \frac{f_0(x)}{S_0(x)},$$

on a donc :

$$h_T(t) = \frac{1}{\delta t} h_0\left(\frac{\ln t - \mu}{\delta}\right),$$

ici il est important de noter que $h_0(x)$ n'est pas une fonction de hasard parce qu'elle est définie sur \mathbb{R} et non sur \mathbb{R}_+^* .

On pourrait donc, *a priori*, travailler sur n'importe quelle distribution usuelle définie sur les nombres réels. On remarquera toutefois que les distributions usuelles définies sur les réels positifs peuvent s'écrire sous cette forme. Une généralisation de ce type de distribution est souvent effectuée en introduisant un paramètre *de forme* b ("shape", en anglais), relié aux moments d'ordre 3 et 4 de la distribution de durée. On réécrit alors la distribution en traitant ce paramètre séparément, sous la forme :

$$S_T(t) = S_0\left(\frac{\ln t - \mu}{\delta}, b\right),$$

et l'on déduit les autres fonctions en utilisant les relations usuelles :

$$\begin{aligned} F_T(t) &= F_0\left(\frac{\ln t - \mu}{\delta}, b\right), \\ f_T(t) &= \frac{1}{\delta t} f_0\left(\frac{\ln t - \mu}{\delta}, b\right), \\ h_T(t) &= \frac{1}{\delta t} h_0\left(\frac{\ln t - \mu}{\delta}, b\right), \end{aligned}$$

avec $F_0(x, b) = 1 - S_0(x, b)$, $f_0(x, b) = \partial F_0(x, b) / \partial x$ et $h_0(x, b) = f_0(x, b) / S_0(x, b)$.

CHAPITRE 2

Distributions usuelles

A PRIORI, toutes les lois applicables aux variables réelles positives pourraient être utilisées pour modéliser les variables de durée. Il faut toutefois être conscient des restrictions qu'une hypothèse implique sur les propriétés de la loi de la durée. Une loi usuelle courante, comme la loi log-normale, implique un hasard avec un maximum unique et contraignent donc les sorties à se faire d'abord avec un taux croissant puis avec un taux décroissant. C'est une restriction forte. Pour éviter une restriction de ce type, il est fructueux de fixer d'abord la forme de la fonction de hasard, et d'en déduire l'ensemble de la distribution.

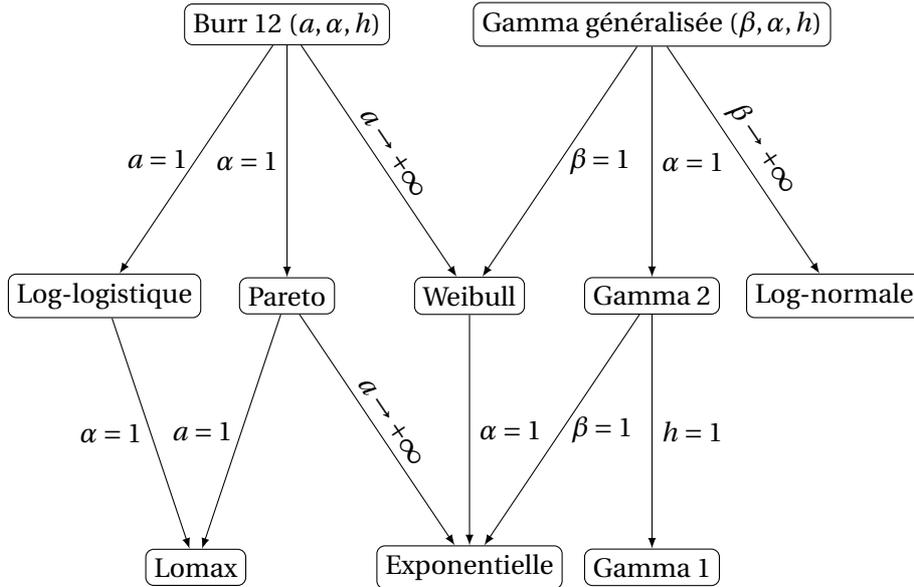
Dans un premier temps, nous verrons les distributions à un paramètre, qui ne sont pas utilisées en économétrie, mais qui permettent d'illustrer certains points importants. Les distributions exponentielle et Gamma sont les ingrédients de base de nombreuses distributions et seront donc présentées dans la première section. Nous avons ajouté la distribution de Lomax à un paramètre pour montrer au lecteur qu'une distribution en apparence anodine peut ne pas admettre de moyenne. Ce cas peut se présenter assez fréquemment sur certains échantillons, de sorte qu'il vaut souvent mieux faire des prévisions sur les quantiles.

La section 2 présente les distributions les plus employées, qui comportent deux paramètres. Elles permettent d'obtenir différentes formes de hasard avec une assez grande flexibilité. Nous y ajoutons la distribution exponentielle avec état absorbant qui s'avère assez intéressante.

Enfin, la section trois présente deux distributions à trois paramètres, qui généralisent la plupart des distributions précédentes. Elle peuvent servir de base à des tests de choix de modélisation.

Dans l'ensemble de ce chapitre un soin particulier a été apporté au paramétrage, afin qu'il soit cohérent avec l'ensemble des distributions présentées.

Par exemple, le paramètre h des lois Gamma généralisée et Burr de type 12 correspond bien au paramètre h de la loi exponentielle quand on se trouve dans ce cas particulier. Les relations entre les distributions sont résumées par le graphique 2.1.



GRAPHIQUE 2.1 – RELATIONS ENTRE LES DISTRIBUTIONS

2.1 Distribution à un paramètre

2.1.1 Loi exponentielle

Cette loi, la plus simple, vérifie la propriété forte d'indépendance temporelle de la fonction de hasard. Le taux de mortalité (i.e. de sortie) est constant dans le temps :

$$h_T(t) = h, \forall t.$$

Cette hypothèse définit ce que l'on appelle un processus de Poisson, qui est également relié à la loi de Poisson dans le cas des données de comptage. En utilisant les propriétés de la section précédente, on retrouve les différentes manières dont on peut caractériser la distribution :

$$H_T(t) = \int_0^t h dx = h[x]_0^t = ht$$

$$\begin{aligned}
 S_T(t) &= \exp(-H_T(t)) = \exp(-ht) \\
 F_T(t) &= 1 - S_T(t) = 1 - \exp(-ht), \\
 f_T(t) &= h_T(t)S_T(t) = h \exp(-ht),
 \end{aligned}
 \tag{2.1}$$

Nous dérivons les moments de cette distribution en annexe (B.1.4) :

$$\begin{aligned}
 E(T^k) &= \frac{\Gamma(1+k)}{h^k}, \quad k \in \mathbb{N}_* \\
 \Rightarrow E(T) &= \frac{1}{h}
 \end{aligned}$$

et le quantile d'ordre ω vaut :¹

$$\begin{aligned}
 Q_\omega &= -\frac{1}{h} \ln(1-\omega) \\
 \Rightarrow \text{Me}(T) &= \frac{\ln 2}{h},
 \end{aligned}$$

donc la moyenne est toujours supérieure à la médiane avec cette distribution.²

La distribution exponentielle peut sembler particulièrement restrictive au premier abord, mais elle est à l'origine de nombreuses modélisations plus réalistes, comme le modèle à hasard constant par morceaux que nous verrons plus loin.

2.1.2 Loi Gamma à un paramètre

Cette loi possède la densité :

$$f_T(t) = \frac{t^{\beta-1} \exp(-t)}{\Gamma(\beta)},$$

sa fonction de répartition est donnée par :

$$F_T(t) = \frac{1}{\Gamma(\beta)} \int_0^t x^{\beta-1} \exp(-x) dx = \gamma(\beta, t),$$

où $\gamma(\beta, t)$ est la fonction gamma tronquée régularisée définie en annexe (A.4.4). D'où la fonction de survie :

$$S_T(t) = 1 - \gamma(\beta, t),$$

1. En utilisant $S_T(Q_\omega) = 1 - \omega$
 2. Rappel : $\ln 2 \approx 0,693$.

le hasard s'écrit donc :

$$h_T(t) = \frac{t^{\beta-1} \exp(-t)}{\Gamma(\beta)[1 - \gamma(\beta, t)]}.$$

Les moments sont égaux à (annexe B.1.6) :

$$\begin{aligned} E(T^k) &= \frac{\Gamma(\beta + k)}{\Gamma(\beta)}, \\ \Rightarrow E(T) &= \beta, \end{aligned}$$

et le quantile d'ordre ω s'obtient par :

$$\begin{aligned} \omega &= \gamma(\beta, Q_\omega) \\ \Leftrightarrow \omega &= F_{\chi^2, 2\beta}(2Q_\omega) \\ \Leftrightarrow Q_\omega &= \frac{1}{2} \times F_{\chi^2, 2\beta}^{-1}(\omega) \end{aligned}$$

où $F_{\chi^2, 2\beta}^{-1}$ est la fonction quantile du χ^2 à 2β degrés de liberté (disponible sous les logiciels usuels). La relation entre la fonction de répartition du χ^2 et la fonction gamma tronquée régularisée est démontrée en annexe (A.4.4). La médiane est donc donnée par :

$$\text{Me}(T) = \frac{1}{2} \times F_{\chi^2, 2\beta}^{-1}\left(\frac{1}{2}\right).$$

2.1.3 Loi de Lomax à un paramètre

Nous présentons ici le cas particulier de cette distribution qui est commun aux distributions log-logistique et de Pareto.³ La fonction de survie de cette loi est donnée par :

$$S_T(t) = \frac{1}{1 + ht},$$

donc sa fonction de répartition est donnée par :

$$F_T(t) = 1 - S_T(t) = \frac{ht}{1 + ht},$$

sa densité par :

$$f_T(t) = -\frac{dS_T}{dt}(t) = \frac{h}{(1 + ht)^2},$$

et le hasard par :

$$h_T(t) = \frac{f_T(t)}{S_T(t)} = \frac{h}{1 + ht}.$$

3. On peut écrire la fonction de survie de la distribution de Lomax (1954) à deux paramètres sous la forme $S(t) = (1 + ht)^{-c}$, ici nous posons $c = 1$.

Le hasard est maximum en $t = 0$ et strictement décroissant ensuite. La moyenne n'existe pas et le quantile d'ordre ω est donné par :⁴

$$Q_\omega = \frac{1}{h} \left(\frac{1-\omega}{\omega} \right),$$

d'où la médiane :

$$\text{Me} = \frac{1}{h}.$$

L'absence d'existence de la moyenne amène à favoriser les quantiles lorsqu'on veut effectuer une prévision de durée. Les quantiles ne nécessitent que l'existence de la fonction de répartition, de sorte qu'ils existent toujours dans les cas de distributions non dégénérées. Ils tendent également à donner des résultats d'estimation plus stables.

2.2 Distributions à deux paramètres

2.2.1 Loi exponentielle avec état absorbant

Il s'agit d'une première généralisation du modèle exponentiel. On suppose maintenant qu'il existe une proportion π d'individus qui ne sortent jamais du processus étudié. Ceci va affecter le hasard, qui ne sera plus constant mais décroissant avec la durée. La fonction de survie est donnée par :

$$S_\pi(t) = \pi + (1 - \pi)S_T(t) \text{ avec } S_T(t) = \exp(-ht),$$

on obtient donc une nouvelle fonction de survie :

$$S_\pi(t) = \pi + (1 - \pi) \exp(-ht).$$

ainsi que la nouvelle fonction de répartition :

$$F_\pi(t) = 1 - S_\pi(t) = (1 - \pi) (1 - \exp(-ht)).$$

La densité est donné par :

$$f_\pi(t) = \frac{dF_\pi}{dt}(t) = (1 - \pi)h \exp(-ht),$$

4. Pour voir que la moyenne n'existe pas, on peut examiner la distribution de Pareto quand $\alpha \rightarrow 1$ ou la distribution log-logistique quand $\alpha \rightarrow 1$.

d'où le hasard :

$$\begin{aligned} h_{\pi}(t) &= \frac{f_{\pi}(t)}{S_{\pi}(t)} \\ &= \frac{h(1-\pi)\exp(-ht)}{\pi + (1-\pi)\exp(-ht)} \\ &= \frac{h}{\frac{\pi}{1-\pi}\exp(ht) + 1}, \end{aligned}$$

et

$$\begin{aligned} \lim_{t \rightarrow 0} h_{\pi}(t) &= (1-\pi)h, \\ \lim_{t \rightarrow +\infty} h_{\pi}(t) &= 0. \end{aligned}$$

La dernière formulation permet de montrer que le hasard est strictement décroissant, qu'il admet un maximum en $t = 0$, et un minimum en $t = \{+\infty\}$: en effet, en $t = \{+\infty\}$, il ne reste plus que des individus qui ne sortent pas dans l'échantillon, d'où la nullité du taux de sortie conditionnel. En $t = 0$, on peut interpréter le hasard comme une moyenne pondérée d'un hasard nul et d'un hasard constant h , puisque :

$$\pi \times 0 + (1-\pi) \times h = h_{\pi}(0).$$

Une légère modification du modèle exponentiel permet donc d'obtenir une distribution intéressante. On pourrait généraliser cette approche à toutes les distributions usuelles qui suivent. Pour ce type de modèle on raisonne en deux parties :

1. On donne la probabilité π que des individus ne sortent jamais, leur durée est infinie par définition, de sorte qu'ils n'ont ni durée moyenne ni durée médiane ;
2. Pour les individus qui sortent, on calcule les moments par la méthode habituelle. Ils admettent donc les mêmes moments que pour la loi exponentielle classique.⁵

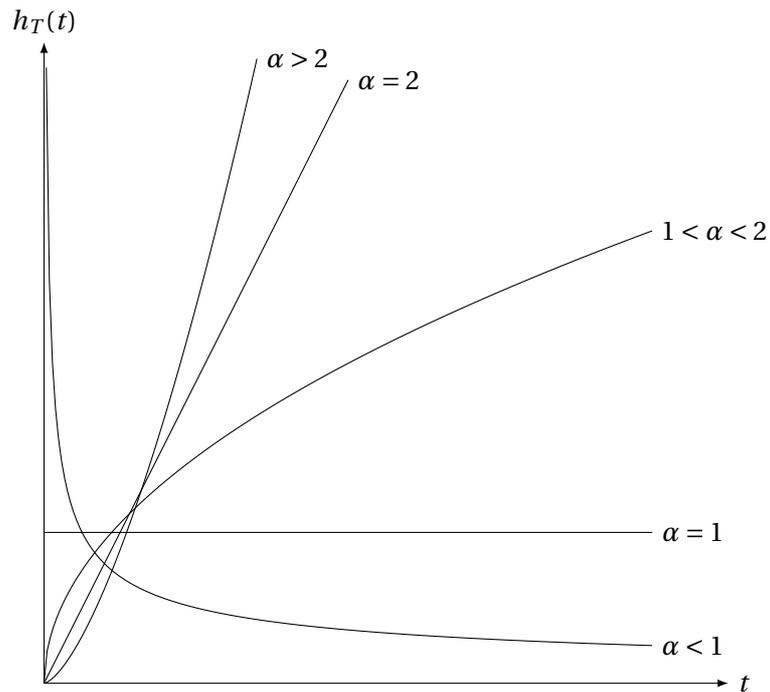
2.2.2 Loi de Weibull

Cette loi est une des plus employées dans les applications économétriques (Weibull, 1961). Elle généralise la loi exponentielle en autorisant plusieurs types

5. Notons que généralement, les estimations donneront des valeurs de h différentes selon que l'on autorise un état absorbant ou non dans le modèle.

d'évolutions de la fonction de hasard avec la durée, résumée dans le graphique 2.2. On remarque que les évolutions du hasard restent toutefois monotones. On a :

$$h_T(t) = h\alpha t^{\alpha-1}.$$



GRAPHIQUE 2.2 – FONCTION DE HASARD DE LA LOI DE WEIBULL ($h = 1$)

Si $\alpha = 1$ on retrouve le modèle exponentiel mais, selon la valeur de α , le hasard peut être aussi bien croissant que décroissant avec la durée. Le hasard cumulé est égal à :

$$\begin{aligned} H_T(t) &= \int_0^t h_T(x) dx \\ &= \int_0^t h\alpha x^{\alpha-1} dx \\ &= h [x^\alpha]_0^t \\ &= ht^\alpha. \end{aligned}$$

On en déduit la fonction de survie :

$$S_T(t) = \exp(-H_T(t)) = \exp(-ht^\alpha),$$

la fonction de répartition est donc donnée par :

$$F_T(t) = 1 - \exp(-ht^\alpha),$$

et la densité par :

$$f_T(t) = h_T(t)S_T(t) = h\alpha t^{\alpha-1} \exp(-ht^\alpha).$$

On montre en annexe (B.1.8) que les moments sont donnés par :

$$\begin{aligned} E(T^k) &= \frac{\Gamma(1+k/\alpha)}{h^{k/\alpha}}, \\ \Rightarrow E(T) &= \frac{\Gamma(1+1/\alpha)}{h^{1/\alpha}}, \end{aligned}$$

et on obtient le quantile d'ordre ω en inversant la fonction de survie (ou de répartition, au choix) :

$$\begin{aligned} Q_\omega &= \left[-\frac{1}{h} \ln(1-\omega) \right]^{1/\alpha} \\ \Rightarrow \text{Me}(T) &= \left[\frac{\ln 2}{h} \right]^{1/\alpha} \end{aligned}$$

2.2.3 Loi Gamma à deux paramètres

Cette loi généralise la loi exponentielle et la loi Gamma à un paramètre. Sa densité est donnée par :

$$f_T(t) = \frac{h^\beta t^{\beta-1} \exp(-ht)}{\Gamma(\beta)},$$

on retrouve la densité de la loi exponentielle en posant $\beta = 1$ et celle de la loi Gamma à un paramètre pour $h = 1$. Sa fonction de répartition est donnée par :

$$\begin{aligned} F_T(t) &= \frac{h}{\Gamma(\beta)} \int_0^t (hx)^{\beta-1} \exp(-hx) dx \\ &= \gamma(\beta, ht), \end{aligned}$$

où $\gamma(.,.)$ est la fonction Gamma tronquée régularisée (annexe A.4.4), d'où la fonction de survie :

$$S_T(t) = 1 - \gamma(\beta, ht),$$

et le hasard :

$$h_T(t) = \frac{h^\beta t^{\beta-1} \exp(-ht)}{\Gamma(\beta) [1 - \gamma(\beta, ht)]}.$$

Les moments sont donnés par (annexe B.1.7) :

$$\begin{aligned} E(T^k) &= \frac{\Gamma(1+k)}{h^k} \\ \Rightarrow E(T) &= \frac{1}{h} \end{aligned}$$

et le quantile d'ordre ω vérifie :

$$\begin{aligned} \omega &= \gamma(\beta, hQ_\omega) \\ \Leftrightarrow \omega &= F_{\chi^2, 2\beta}(2hQ_\omega) \\ \Leftrightarrow Q_\omega &= \frac{1}{2h} \times F_{\chi^2, 2\beta}^{-1}(\omega). \end{aligned}$$

La médiane est donc donnée par :

$$\text{Me}(T) = \frac{1}{2h} \times F_{\chi^2, 2\beta}^{-1}\left(\frac{1}{2}\right).$$

2.2.4 Loi log-normale

Cette loi vient de l'adaptation d'un modèle normal standard aux variables positives. On ne peut pas supposer que T suit une loi normale, puisque la durée ne prend que des valeurs positives, mais on peut supposer que $\ln T$ suit une loi normale $N(m, \sigma^2)$ parce qu'elle est définie sur \mathbb{R} . Ceci revient à dire que T suit une loi log-normale de paramètres (m, σ^2) . Sa densité est donnée par (annexe A.2) :

$$f_T(t) = \frac{1}{\sigma t} \varphi\left(\frac{\ln t - m}{\sigma}\right),$$

où $\varphi(t) = (2\pi)^{-1/2} \exp(-t^2/2)$ est la densité de la loi normale centrée et réduite. Sa fonction de répartition est donc donnée par :

$$F_T(t) = \int_0^t f(x) dx = \int_0^t \frac{1}{\sigma x} \varphi\left(\frac{\ln x - m}{\sigma}\right) dx,$$

on remarque alors que $1/(\sigma x)$ est la dérivée par rapport à x de $(\ln x - m)/\sigma$, de sorte que :

$$F_T(t) = \left[\Phi\left(\frac{\ln x - m}{\sigma}\right) \right]_0^t = \Phi\left(\frac{\ln t - m}{\sigma}\right),$$

car :

$$\lim_{x \rightarrow 0} \Phi\left(\frac{\ln x - m}{\sigma}\right) = 0.$$

On en déduit :

$$S_T(t) = 1 - \Phi\left(\frac{\ln t - m}{\sigma}\right),$$

ainsi que :

$$h_T(t) = \frac{f_T(t)}{S_T(t)} = \frac{1}{\sigma t} \times \frac{\varphi\left(\frac{\ln t - m}{\sigma}\right)}{1 - \Phi\left(\frac{\ln t - m}{\sigma}\right)}.$$

Ceci implique que le hasard admet obligatoirement un maximum. En effet,

$$\sigma h'_T(t) = \frac{\varphi(u)}{\sigma t^2 (1 - \Phi(u))^2} (-(\sigma + u)(1 - \Phi(u)) + \varphi(u)),$$

avec :

$$u = \frac{\ln t - m}{\sigma}.$$

On pose :

$$\kappa(u) = \frac{\varphi(u)}{1 - \Phi(u)},$$

cette fonction est définie par analogie avec la fonction de hasard mais sur une loi normale centrée réduite. Donc, il ne s'agit pas à proprement parler d'une fonction de hasard parce que u peut prendre des valeurs négatives. Cette fonction est strictement croissante, comme l'illustre le graphique 2.3. On peut écrire :

$$h'_T(t) = \frac{1}{\sigma^2 t^2} \kappa(u) (\kappa(u) - (\sigma + u)),$$

le hasard atteint son maximum en un point $u^* = (\ln t^* - m)/\sigma$ tel que :

$$\kappa(u^*) = \sigma + u^*.$$

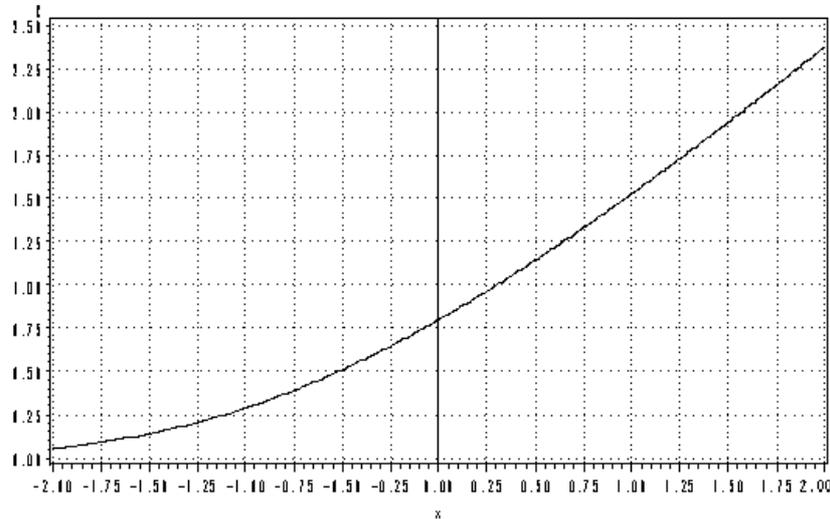
On trouve la valeur exacte de u^* en utilisant un algorithme d'optimisation une fois les paramètres estimés. Ce type de profil du hasard est très particulier, car la présence d'un maximum est imposée, et cette hypothèse ne convient pas forcément à tous les processus de durée. Il faut donc être vigilant quand on l'emploie. Les moments sont donnés par (annexe B.1.11) :

$$\begin{aligned} E(T^k) &= \exp(km + k^2\sigma^2/2) \\ &\Rightarrow E(T) = \exp(m + \sigma^2/2), \end{aligned}$$

et le quantile d'ordre ω est donné par :

$$\begin{aligned} Q_\omega &= \exp(m + \sigma\Phi^{-1}(\omega)) \\ &\Rightarrow \text{Me}(T) = \exp(m), \end{aligned}$$

car $\Phi(0) = \frac{1}{2}$.

GRAPHIQUE 2.3 – LOI NORMALE, $\varphi(x)/(1 - \Phi(x))$

2.2.5 Loi log-logistique

La loi log-logistique possède l'avantage d'admettre une fonction de répartition explicite. Mais elle possède un avantage plus important par rapport à la loi log-normale : elle permet d'obtenir un hasard qui peut être soit décroissant soit avec un maximum. La durée T suit une loi log-logistique de paramètres (m, δ^2) lorsque $(\ln T - m)/\delta$ suit une loi logistique. La fonction de répartition de la loi logistique $(0, 1)$ est donnée par :

$$F_L(z) = \frac{1}{1 + \exp(-z)},$$

avec :

$$z = \frac{\ln t - m}{\delta},$$

et sa densité par :

$$f_L(z) = \frac{\exp(-z)}{(1 + \exp(-z))^2} = F_L(z)(1 - F_L(z)),$$

par analogie on pose :

$$h_L(t) = \frac{f_L(z)}{F_L(z)} = F_L(z),$$

mais il faut faire attention car cette expression n'est pas une fonction de hasard à proprement parler. En effet, z prend ses valeurs dans \mathbb{R} , pas dans \mathbb{R}_+^* . La densité de la loi log-logistique est égale à (annexe A.2) :

$$f_T(t) = \frac{1}{\delta t} f_L\left(\frac{\ln t - m}{\delta}\right).$$

Nous allons maintenant donner un paramétrage plus pratique, qui permet de comparer cette distribution avec la loi de Burr de type 12. On remarque que :

$$\begin{aligned} \exp\left\{-\left(\frac{\ln t - m}{\delta}\right)\right\} &= \frac{1}{ht^\alpha} \\ \text{avec } h &= \exp\left(-\frac{m}{\delta}\right) \\ \text{et } \alpha &= \frac{1}{\delta} \end{aligned}$$

de sorte que :

$$f_T(t) = \frac{\alpha ht^{\alpha-1}}{(1 + ht^\alpha)^2}.$$

La fonction de répartition est égale à :

$$F_T(t) = F_L\left(\frac{\ln t - m}{\delta}\right) = \frac{ht^\alpha}{1 + ht^\alpha},$$

on en déduit la fonction de survie :

$$S_T(t) = 1 - F_L\left(\frac{\ln t - m}{\delta}\right) = \frac{1}{1 + ht^\alpha},$$

et le hasard :

$$h_T(t) = \frac{f_T(t)}{S_T(t)} = \frac{\alpha ht^{\alpha-1}}{1 + ht^\alpha}.$$

La forme du hasard dépend de la valeur du paramètre α . Si $\alpha < 1$, le numérateur est décroissant et le dénominateur croissant. Le hasard est donc décroissant. Si $\alpha = 1$, on obtient un toujours un hasard décroissant :

$$h_T(t) = \frac{h}{1 + ht}$$

avec un maximum en $t = 0$, avec $h(0) = h$. Par contre, si $\alpha > 1$ on obtient un hasard avec un maximum intérieur. La dérivée du hasard est égale à :

$$h'_T(t) = \frac{\alpha ht^{\alpha-2}}{(1 + ht^\alpha)^2} (\alpha - 1 - ht^\alpha),$$

et elle s'annule en :

$$t^* = \left(\frac{\alpha - 1}{h} \right)^{1/\alpha},$$

le hasard est donc croissant avant cette durée, et décroissant après. On remarquera également que l'espérance mathématique n'existe que lorsque $\alpha > 1$, c'est-à-dire quand le hasard admet un maximum intérieur. Plus généralement, les moments sont donnés par (annexe B.2.2) :

$$\begin{aligned} E(T^k) &= \left(\frac{1}{h} \right)^{k/\alpha} B(1 - k/\alpha, 1 + k/\alpha), \quad \alpha > k \\ \Rightarrow E(T) &= \left(\frac{1}{h} \right)^{1/\alpha} B(1 - 1/\alpha, 1 + 1/\alpha), \end{aligned}$$

et le quantile d'ordre ω est donné par :

$$\begin{aligned} Q_\omega &= \left(\frac{\omega}{h(1 - \omega)} \right)^{1/\alpha} \\ \Rightarrow \text{Me}(T) &= \left(\frac{1}{h} \right)^{1/\alpha}. \end{aligned}$$

2.2.6 Loi de Pareto : un mélange continu de lois exponentielles

On peut présenter la loi de Pareto directement ou montrer qu'elle généralise la loi exponentielle. Nous présentons la seconde approche car elle permet de montrer l'effet de l'hétérogénéité inobservée sur la distribution d'une durée exponentielle : le hasard devient décroissant. On suppose maintenant qu'il existe une infinité d'individus ayant chacun un hasard constant. On représente ce hasard par une variable aléatoire $Z > 0$ de réalisations z . Pour représenter cette distribution des taux de hasard on utilise une distribution définie sur les valeurs réelles positives. Pour obtenir une loi de Pareto, il faut prendre une distribution Gamma de densité (annexe A.5) :

$$f_Z(z) = \frac{(a/h)^a z^{a-1} \exp(-az/h)}{\Gamma(a)}, \quad z > 0, \quad a > 0, \quad b > 0 \quad (2.2)$$

qui admet pour espérance $E(Z) = h$ et pour variance $V(Z) = h^2/a$. Nous avons donc réécrit cette distribution pour que sa moyenne soit égale au paramètre h de la distribution exponentielle standard. Quand $a \rightarrow +\infty$, la variance de cette distribution s'annule et on se retrouve dans le cas où tous les individus ont un hasard constant égal à h : le cas exponentiel. On parle de modèle à hasard hétérogène. Pour retrouver la loi exponentielle il suffit donc de prendre la limite

de la loi de Pareto quand $a \rightarrow +\infty$. Dans le cas général, chaque individu de hasard z donné admet une distribution exponentielle de durée dont la densité, conditionnelle à la valeur de z , est exponentielle, égale à :

$$f_T(t|z) = z \exp(-zt),$$

la densité marginale de la durée est donc égale à :

$$f_T(t) = \int_0^{+\infty} f_T(t|z) f_Z(z) dz,$$

on a :

$$f_T(t|z) f_Z(z) = \frac{(a/h)^a z^a \exp(-z(a/h+t))}{\Gamma(a)},$$

ce qui incite à effectuer le changement de variable :

$$u = z \frac{(a+ht)}{h} \Leftrightarrow z = \frac{hu}{a+ht} \Rightarrow dz = \frac{h}{a+ht} du,$$

et les bornes d'intégration sont inchangées car $a > 0$, $h > 0$ et $t > 0$. On obtient donc, en utilisant $\Gamma(a+1) = a\Gamma(a)$:

$$\begin{aligned} f_T(t) &= \frac{1}{\Gamma(a)} \int_0^{+\infty} \left(\frac{a}{h}\right)^a \left(\frac{hu}{a+ht}\right)^a \exp(-u) \frac{h}{a+ht} du \\ &= \frac{1}{\Gamma(a)} \left(\frac{a}{h}\right)^a \left(\frac{h}{a+ht}\right)^{a+1} \int_0^{+\infty} u^a \exp(-u) du \\ &= \frac{\Gamma(a+1)}{\Gamma(a)} \left(\frac{a}{h}\right)^a \left(\frac{h}{a+ht}\right)^{a+1} \\ &= h \left(\frac{a}{a+ht}\right)^{a+1} \\ &= \frac{a(a/h)^a}{(a/h+t)^{a+1}}, \end{aligned}$$

il s'agit de la forme d'une densité de Pareto, exprimée toutefois de manière inhabituelle. La forme la plus usuelle, donnée dans les manuels, est la suivante :

$$f_X(x) = \frac{ax_m^a}{x^{a+1}}, \quad x > x_m,$$

il suffit donc de poser :

$$x_m = \frac{a}{h} \text{ et } x = \frac{a}{h} + t,$$

de sorte que, conformément à la définition de cette distribution, on a toujours $x > x_m$ car $a > 0$, $h > 0$ et $t > 0$. La fonction de répartition marginale est égale à :

$$F_T(t) = 1 - \left(\frac{x_m}{x}\right)^a = 1 - \left(\frac{a}{a+ht}\right)^a,$$

et la fonction de survie marginale à :

$$S_T(t) = \left(\frac{a}{a+ht} \right)^a,$$

ce qui implique le hasard marginal :

$$h_T(t) = \frac{f_T(t)}{S_T(t)} = \frac{ah}{a+ht},$$

il est maximum en $t = 0$ avec $h(0) = h$, puis tend vers 0 avec la durée passée dans l'état étudié. On remarque aussi que :

$$\lim_{a \rightarrow +\infty} h_T(t) = h,$$

de sorte que l'on retrouve le cas exponentiel standard quand la variance de l'hétérogénéité s'annule. Introduire de l'hétérogénéité Gamma dans un modèle à hasard constant mène donc à un hasard décroissant. L'expression des moments est dérivée en annexe (B.2.1) :

$$\begin{aligned} E(T^k) &= \left(\frac{a}{h} \right)^k \frac{\Gamma(a-k)\Gamma(k+1)}{\Gamma(a)}, \quad a > k. \\ \Rightarrow E(T) &= \frac{a}{(a-1)h}, \quad a > 1, \end{aligned}$$

et le quantile d'ordre ω vaut :

$$\begin{aligned} Q_\omega &= \frac{a}{h} \{(1-\omega)^{-1/a} - 1\} \\ \Rightarrow \text{Me}(T) &= \frac{a}{h} (2^{1/a} - 1). \end{aligned}$$

2.3 Autres distributions

2.3.1 Mélange discret de lois exponentielles (modèle mobile-stable)

La propriété d'indépendance temporelle n'est vérifiée que si *tous* les individus ont le même hasard. Lorsqu'une population regroupe des individus hétérogènes, avec chacun un hasard constant dans le temps, le hasard agrégé varie dans le temps. C'est le phénomène "mover-stayer". On dispose d'un groupe d'individus à hasard élevé (groupe "mover") et d'un groupe d'individus à hasard faible (groupe "stayer"). Au fur et à mesure que le temps passe, les individus à hasard élevé sortent les premiers de sorte que la composition de la population totale comporte une part de plus en plus grande d'individus à hasard faible. Le hasard agrégé, celui de la population, est donc de plus en plus faible. Bien que chaque groupe vérifie la propriété d'indépendance temporelle, l'ensemble de la population présente un hasard décroissant avec la durée.

Exemple 2.1. *On dispose de deux échantillons de demandeurs d'emploi. Dans le premier échantillon, les chômeurs ont un diplôme équivalent ou supérieur au baccalauréat, dans le second échantillon, ils ont un diplôme de niveau strictement inférieur au baccalauréat. Même si les taux de sortie de ces deux groupes sont constants dans le temps, le taux de sortie agrégé sera décroissant dans le temps, car les demandeurs d'emploi les plus diplômés sortiront du chômage en premier. Au fur et à mesure que la durée s'allonge, il reste une plus forte proportion de chômeurs dont le niveau de formation est inférieur au baccalauréat dans l'échantillon, donc le taux de sortie du chômage dans l'échantillon diminue au fur et à mesure que la durée passée au chômage s'allonge.*

On utilise le modèle exposé dans cette section quand on ne peut pas identifier séparément les deux groupes d'individus. Si on le pouvait, on procéderait à deux estimations séparées. Pour fixer les idées, prenons une population où une fraction α_1 de la population admet un hasard constant égal à h_1 et où une fraction α_2 admet un hasard constant égal à $h_2 > h_1$, avec $\alpha_1 + \alpha_2 = 1$.

$$F_1(t) = 1 - \exp(-h_1 t) \text{ et } F_2(t) = 1 - \exp(-h_2 t).$$

La fonction de répartition agrégée est donnée par (en utilisant $\alpha_1 + \alpha_2 = 1$) :

$$\begin{aligned} F(t) &= \alpha_1 F_1(t) + \alpha_2 F_2(t) \\ &= 1 - \alpha_1 \exp(-h_1 t) - \alpha_2 \exp(-h_2 t), \end{aligned}$$

donc la survie est donnée par :

$$\begin{aligned} S(t) &= 1 - F(t) \\ &= \alpha_1 \exp(-h_1 t) + \alpha_2 \exp(-h_2 t), \end{aligned}$$

et la densité est donnée par :

$$f(t) = -\frac{dS}{dt}(t) = \alpha_1 h_1 \exp(-h_1 t) + \alpha_2 h_2 \exp(-h_2 t),$$

on en déduit le hasard :

$$\begin{aligned} h(t) &= \frac{f(t)}{S(t)} \\ &= \frac{\alpha_1 h_1 \exp(-h_1 t) + \alpha_2 h_2 \exp(-h_2 t)}{\alpha_1 \exp(-h_1 t) + \alpha_2 \exp(-h_2 t)} \\ &= \frac{\alpha_1 h_1 + \alpha_2 h_2 \exp(-(h_2 - h_1)t)}{\alpha_1 + \alpha_2 \exp(-(h_2 - h_1)t)}, \end{aligned}$$

et l'on ne retrouve l'indépendance temporelle que lorsque $h_1 = h_2$. Dans le cas étudié, $h_2 > h_1$ de sorte que la fonction de hasard agrégée est décroissante. En effet, les individus dont la durée moyenne est la plus courte ($1/h_2$) sortent les premiers, ce qui implique que la population comporte, à chaque date, une proportion plus importante d'individus de durée moyenne longue ($1/h_1$). Le hasard est maximum en $t = 0$ où :

$$h(0) = \alpha_1 h_1 + \alpha_2 h_2,$$

et il décroît jusqu'en :

$$\lim_{t \rightarrow +\infty} h(t) = h_1,$$

de sorte que le hasard admet une borne inférieure. La durée moyenne peut également être calculée directement :

$$\begin{aligned} E(T) &= \int_0^{+\infty} S(x) dx \\ &= \int_0^{+\infty} [\alpha_1 \exp(-h_1 x) + \alpha_2 \exp(-h_2 x)] dx \\ &= \alpha_1 \left[-\frac{1}{h_1} \exp(-h_1 x) \right]_0^{+\infty} + \alpha_2 \left[-\frac{1}{h_2} \exp(-h_2 x) \right]_0^{+\infty} \\ &= \frac{\alpha_1}{h_1} + \frac{\alpha_2}{h_2}, \end{aligned}$$

elle est simplement égale à la moyenne pondérée des moyennes des deux groupes ($1/h_1$ et $1/h_2$ respectivement). On trouve le quantile d'ordre ω en résolvant numériquement la relation suivante en Q_ω pour une valeur donnée de (α_1, h_1, h_2) :

$$\begin{aligned} 1 - \omega &= \alpha_1 \exp(-h_1 Q_\omega) + (1 - \alpha_1) \exp(-h_2 Q_\omega) \\ \Rightarrow \frac{1}{2} &= \alpha_1 \exp(-h_1 \text{Me}(T)) + (1 - \alpha_1) \exp(-h_2 \text{Me}(T)). \end{aligned}$$

On généralise facilement à J groupes, de pondérations respectives $\alpha_1, \alpha_2, \dots, \alpha_J$ avec $\sum_{j=1}^J \alpha_j = 1$, et de hasards constants respectifs h_1, \dots, h_J . Pour simplifier l'exposé, on redéfinit les indices de manière à avoir $h_1 < h_2 < \dots < h_J$. La fonction de répartition de la durée agrégée est donnée par :

$$\begin{aligned} F(t) &= \sum_{j=1}^J \alpha_j (1 - \exp(-h_j t)) \\ &= 1 - \sum_{j=1}^J \alpha_j \exp(-h_j t), \end{aligned}$$

en utilisant $\sum_{j=1}^J \alpha_j = 1$. La fonction de survie est donc égale à :

$$\begin{aligned} S(t) &= 1 - F(t) \\ &= \sum_{j=1}^J \alpha_j \exp(-h_j t), \end{aligned}$$

et la densité à :

$$f(t) = -\frac{dS}{dt}(t) = \sum_{j=1}^J \alpha_j h_j \exp(-h_j t),$$

d'où le hasard :

$$\begin{aligned} h(t) &= \frac{f(t)}{S(t)} \\ &= \frac{\sum_{j=1}^J \alpha_j h_j \exp(-h_j t)}{\sum_{j=1}^J \alpha_j \exp(-h_j t)}. \end{aligned}$$

On peut réécrire cette expression en utilisant $h_1 < h_2 < \dots < h_J$, et en multipliant le numérateur et le dénominateur par $\exp(h_1 t)$, de sorte que :

$$\begin{aligned} h(t) &= \frac{\sum_{j=1}^J \alpha_j h_j \exp(-h_j t)}{\sum_{j=1}^J \alpha_j \exp(-h_j t)} \\ &= \frac{\alpha_1 h_1 + \sum_{j=2}^J \alpha_j h_j \exp(-(h_j - h_1) t)}{\alpha_1 + \sum_{j=2}^J \alpha_j \exp(-(h_j - h_1) t)}, \end{aligned}$$

ce hasard est maximum en $t = 0$:

$$h(0) = \sum_{j=1}^J \alpha_j h_j,$$

la moyenne pondérée des hasards des J groupes, et atteint son minimum quand $t \rightarrow +\infty$:

$$\lim_{t \rightarrow +\infty} h(t) = h_1,$$

le hasard le plus faible. Les individus avec les plus grands hasards sortent les premiers, donc le stock d'individus restant possède un hasard moyen plus faible. Les sorties continuent jusqu'à ce qu'il ne reste que des individus du groupe qui sort le plus lentement. On montre que la durée moyenne est donnée par la moyenne pondérée des durées moyennes ($1/h_j$) des J groupes :

$$E(T) = \sum_{j=1}^J \frac{\alpha_j}{h_j}.$$

On trouve le quantile d'ordre ω en résolvant numériquement la relation suivante en Q_ω pour une valeur donnée des (α_j, h_j) :

$$\begin{aligned} 1 - \omega &= \sum_{j=1}^J \alpha_j \exp(-h_j Q_\omega) \\ \Rightarrow \frac{1}{2} &= \sum_{j=1}^J \alpha_j \exp(-h_j \text{Me}(T)). \end{aligned}$$

2.3.2 Loi de Burr de type 12 : un mélange continu de lois de Weibull

Comme pour la loi de Pareto, on suppose qu'il existe une infinité de groupes ayant chacun un hasard spécifique. Mais ici, l'hétérogénéité porte sur un paramètre de la fonction de hasard et non sur le hasard lui-même. Ce paramètre correspond au paramètre h dans la fonction de hasard du modèle de Weibull. On considère que ce paramètre est une variable aléatoire Z de réalisations z . Le hasard est défini par :

$$h_T(t|z) = z\alpha t^{\alpha-1},$$

l'hétérogénéité porte donc sur une translation vers le haut ou vers le bas de toute la fonction de hasard d'une distribution de Weibull. Tous les individus partagent le même paramètre α . Pour représenter la distribution du paramètre intervenant dans la fonction de hasard, on utilise une distribution définie sur les valeurs réelles positives. Pour obtenir une loi de Burr de type 12, on prend une fonction Gamma de densité (2.2). On rappelle que $E(Z) = h$ et $V(Z) = h/a^2$ de sorte que l'on retrouve le cas particulier de la distribution de Weibull quand $a \rightarrow +\infty$. Le hasard cumulé conditionnel à l'hétérogénéité est donnée par :

$$H_T(t|z) = \int_0^{+\infty} h(x|z) dx = zt^\alpha,$$

d'où la survie conditionnelle :

$$S_T(t|z) = \exp(-H_T(t|z)) = \exp(-zt^\alpha).$$

La densité conditionnelle à l'hétérogénéité est donnée par :

$$f_T(t|z) = h_T(t|z) S_T(t|z) = z\alpha t^{\alpha-1} \exp(-zt^\alpha),$$

et la densité jointe par :

$$\begin{aligned} f_{T,Z}(t, z) &= f_T(t|z) f_Z(z) \\ &= z\alpha t^{\alpha-1} \exp(-zt^\alpha) \times \frac{(a/h)^a z^{a-1} \exp(-az/h)}{\Gamma(a)} \\ &= \left(\frac{a}{h}\right)^a \alpha t^{\alpha-1} \frac{z^a \exp(-z(t^\alpha + a/h))}{\Gamma(a)}. \end{aligned}$$

Pour trouver la distribution de la nouvelle loi, on cherche l'intégrale :

$$I = \int_0^{+\infty} \frac{z^a \exp(-z(t^\alpha + a/h))}{\Gamma(a)} dz,$$

ce qui incite au changement de variable :

$$u = z \left(\frac{ht^\alpha + a}{h} \right) \Leftrightarrow z = \frac{h}{ht^\alpha + a} \times u,$$

avec :

$$\lim_{z \rightarrow 0} u = 0, \quad \lim_{z \rightarrow +\infty} u = +\infty, \quad dz = \frac{h}{ht^\alpha + a} \times du,$$

d'où :

$$\begin{aligned} I &= \frac{1}{\Gamma(a)} \left(\frac{h}{ht^\alpha + a} \right)^{a+1} \int_0^{+\infty} u^a \exp(-u) du \\ &= \frac{\Gamma(a+1)}{\Gamma(a)} \left(\frac{h}{ht^\alpha + a} \right)^{a+1} \\ &= a \left(\frac{h}{ht^\alpha + a} \right)^{a+1}, \end{aligned}$$

ce qui permet d'écrire :

$$\begin{aligned} f_T(t) &= \int_0^{+\infty} f_{T,Z}(t, z) dz \\ &= \int_0^{+\infty} f_T(t|z) f_Z(z) dz \\ &= \left(\frac{a}{h} \right)^a \alpha t^{\alpha-1} \times a \left(\frac{h}{ht^\alpha + a} \right)^{a+1} \\ &= \alpha t^{\alpha-1} h \left(\frac{a}{h} \right)^{a+1} \left(\frac{h}{ht^\alpha + a} \right)^{a+1} \\ &= h \alpha t^{\alpha-1} \left(\frac{a}{ht^\alpha + a} \right)^{a+1}, \end{aligned}$$

il s'agit de la densité d'une loi de Burr de type 12. La fonction de répartition est donnée par :

$$\begin{aligned} F_T(t) &= \int_0^t h \alpha x^{\alpha-1} \left(\frac{a}{hx^\alpha + a} \right)^{a+1} dx \\ &= 1 - \left(\frac{a}{ht^\alpha + a} \right)^a \end{aligned}$$

ce qui implique la fonction de survie :

$$S_T(t) = 1 - F(t) = \left(\frac{a}{ht^\alpha + a} \right)^a.$$

Si $\alpha = 1$, on retrouve la loi de Pareto. Le hasard se déduit des fonctions précédentes :

$$h_T(t) = \frac{f_T(t)}{S(t)} = \frac{ah\alpha t^{\alpha-1}}{a + ht^\alpha},$$

cette fonction de hasard admet pour dérivée :

$$h'_T(t) = \frac{ah\alpha t^{\alpha-2}}{(a + ht^\alpha)^2} ((\alpha - 1)a - ht^\alpha),$$

le hasard est donc décroissant si $\alpha \leq 1$ et admet un maximum si $\alpha > 1$. Dans ce dernier cas le maximum est atteint en :

$$t^* = \left(\frac{(\alpha - 1)a}{h} \right)^{1/\alpha}.$$

On remarque également que :

$$\lim_{a \rightarrow +\infty} h_T(t) = h\alpha t^{\alpha-1},$$

le cas particulier du hasard de la distribution de Weibull. L'expression des moments est dérivée en annexe (B.2.3) :

$$\begin{aligned} E(T^k) &= a \left(\frac{a}{h} \right)^{k/\alpha} B(a - k/\alpha, 1 + k/\alpha), \quad a\alpha > k, \\ \Rightarrow E(T) &= a \left(\frac{a}{h} \right)^{1/\alpha} B(a - 1/\alpha, 1 + 1/\alpha), \end{aligned}$$

et le quantile d'ordre ω est donné par :

$$\begin{aligned} Q_\omega &= \left(\frac{a}{h} \right)^{1/\alpha} [(1 - \omega)^{-1/\alpha} - 1]^{1/\alpha} \\ \Rightarrow \text{Me}(T) &= \left(\frac{a}{h} \right)^{1/\alpha} [2^{1/\alpha} - 1]^{1/\alpha}. \end{aligned}$$

2.3.3 Loi Gamma généralisée

On la définit par sa densité (Stacy, 1968) :

$$f_T(t) = \frac{\alpha h^\beta t^{\alpha\beta-1} \exp(-ht^\alpha)}{\Gamma(\beta)}.$$

On retrouve la densité de la loi de Weibull pour $\beta = 1$, celle de la loi exponentielle pour $\alpha = 1$ et $\beta = 1$, celle de la loi Gamma à deux paramètres pour $\alpha = 1$, celle de la loi Gamma à un paramètre pour $\alpha = 1$ et $h = 1$ et celle de la loi log-normale pour $\beta \rightarrow +\infty$. Il s'agit donc d'une loi suffisamment générale pour qu'on l'utilise comme un critère de choix de modèle. On en déduit la fonction de répartition de la variable de durée de la manière suivante :

$$F_T(t) = \frac{\alpha h^\beta}{\Gamma(\beta)} \int_0^t u^{\alpha\beta-1} \exp(-hu^\alpha) du,$$

on effectue le changement de variable $v = hu^\alpha$, ce qui implique $u = h^{-1/\alpha} v^{1/\alpha}$ et $du = h^{-1/\alpha} \alpha^{-1} v^{1/\alpha-1} dv$, et les bornes d'intégration deviennent 0 et ht^α :

$$\begin{aligned} F_T(t) &= \frac{\alpha h^\beta}{\Gamma(\beta)} \int_0^{ht^\alpha} v^{\beta-1/\alpha} h^{1/\alpha-\beta} \exp(-v) h^{-1/\alpha} \alpha^{-1} v^{1/\alpha-1} dv \\ &= \frac{1}{\Gamma(\beta)} \int_0^{ht^\alpha} v^{\beta-1} \exp(-v) dv \\ &= \gamma(\beta, ht^\alpha), \end{aligned}$$

où $\gamma(\cdot, \cdot)$ est la fonction Gamma tronquée régularisée (annexe A.4.4). On vérifie que lorsque $t \rightarrow 0$, $\gamma(\beta, 0) = 0$ de sorte que $F_T(0) = 0$, et que $\lim_{t \rightarrow +\infty} \gamma(\beta, t) = 1$, donc $F_T(t) \rightarrow 1$. La fonction de survie est donc donnée par :

$$S_T(t) = 1 - \gamma(\beta, ht^\alpha),$$

et le hasard est égal à :

$$h_T(t) = \frac{f_T(t)}{S_T(t)} = \frac{\alpha h^\beta t^{\alpha\beta-1} \exp(-ht^\alpha)}{\Gamma(\beta) [1 - \gamma(\beta, ht^\alpha)]}.$$

Le nom de cette distribution vient du fait qu'elle généralise la loi Gamma à deux paramètres, qui correspond au cas $\alpha = 1$. Les moments sont données par (annexe B.1.9) :

$$\begin{aligned} E(T^k) &= \frac{\Gamma(k/\alpha + \beta)}{h^{k/\alpha} \Gamma(\beta)}, \\ \Rightarrow E(T) &= \frac{\Gamma(1/\alpha + \beta)}{h^{1/\alpha} \Gamma(\beta)} = \frac{\Gamma(1/\alpha)}{h^{1/\alpha} B(1/\alpha, \beta)}, \end{aligned}$$

où la seconde formule évite les problèmes de dépassement ("overflow", en anglais) posés lors de la programmation par les grandes valeurs de β . Le quantile

d'ordre ω vérifie :

$$\begin{aligned}\omega &= \gamma(\beta, h(Q_\omega)^\alpha) \\ \Leftrightarrow \omega &= F_{\chi^2, 2\beta}(2h(Q_\omega)^\alpha) \\ \Leftrightarrow Q_\omega &= \left[\frac{1}{2h} \times F_{\chi^2, 2\beta}^{-1}(\omega) \right]^{1/\alpha},\end{aligned}$$

la médiane est donc donnée par :

$$\text{Me}(T) = \left[\frac{1}{2h} \times F_{\chi^2, 2\beta}^{-1}\left(\frac{1}{2}\right) \right]^{1/\alpha}.$$

Le paramétrage précédent n'est toutefois pas très pratique pour l'optimisation. C'est la raison pour laquelle, on utilise la réécriture de cette loi par Prentice (1974).⁶ Nous reprenons la démonstration de cet article. La variable de durée peut s'écrire en logarithme sous la forme :

$$W = g(T) = \alpha (\ln T - \ln h^{-1/\alpha}) = \alpha \ln T + \ln h,$$

donc :

$$T = g^{-1}(W) = \exp\left(\frac{W - \ln h}{\alpha}\right),$$

et

$$\frac{dg^{-1}(w)}{dw} = \frac{1}{\alpha} \exp\left(\frac{w - \ln h}{\alpha}\right),$$

la densité de W est alors donnée par (annexe A.2) :

$$\begin{aligned}f_W(w) &= \frac{1}{\alpha} \exp\left(\frac{w - \ln h}{\alpha}\right) f_T\left(\exp\left(\frac{w - \ln h}{\alpha}\right)\right) \\ &= \frac{1}{\Gamma(\beta)} \exp(\beta w) \exp(-\exp(w)).\end{aligned}$$

La variable aléatoire W suit alors une distribution log-gamma généralisée qui tend vers la distribution normale quand $\beta \rightarrow +\infty$. Or, comme il est démontré en annexe (C.3) :

$$\begin{aligned}E(\ln T) &= \frac{1}{\alpha} (-\ln h + \psi^{(0)}(\beta)), \\ V(\ln T) &= \frac{1}{\alpha^2} \psi^{(1)}(\beta),\end{aligned}$$

6. Par rapport à l'article de Prentice, cet ouvrage utilise les notations suivantes pour l'équation (1) p. 539 :

$$k = \beta, b = \alpha \text{ et } a = h^{-1/\alpha}.$$

Nous remplaçons également le paramètre α de cet article par le paramètre m .

où $\psi^{(0)}$ est la fonction digamma et $\psi^{(1)}$ la fonction trigamma. Pour alléger les notations, on pose :

$$\psi_0 \triangleq \psi^{(0)}(\beta), \quad \psi_1 \triangleq \psi^{(1)}(\beta).$$

On en déduit que :

$$\begin{aligned} E(W) &= \alpha E(\ln T) + \ln h = \psi_0, \\ V(W) &= \alpha^2 V(\ln T) = \psi_1. \end{aligned}$$

Pour la suite, nous posons :

$$q = \beta^{-1/2} \quad (\text{donc } \beta = q^{-2})$$

et nous utiliserons la variable centrée :

$$E = \frac{W - \psi_0}{q}$$

dont la variance approche l'unité quand $q \rightarrow 0$. Sa densité est obtenue par la méthode du changement de variable :

$$\begin{aligned} E = g(W) &= \frac{W - \psi_0}{q} \\ W = g^{-1}(E) &= \psi_0 + qE \\ \frac{dg^{-1}(\varepsilon)}{d\varepsilon} &= q \end{aligned}$$

on en déduit :

$$f_E(\varepsilon) = \frac{|q|}{\Gamma(q^{-2})} \exp(\beta(\psi_0 + q\varepsilon)) \exp(-\exp(\psi_0 + q\varepsilon)).$$

Enfin, on peut écrire :

$$\ln T = m + \sigma E,$$

avec :

$$m = \frac{\psi_0 - \ln h}{\alpha}, \quad \sigma = \frac{q}{\alpha}.$$

Le point important est que $X \triangleq \ln T$ suit une loi normale (m, σ^2) quand $q \rightarrow 0$. La densité de la loi du logarithme de la durée peut s'écrire à partir du changement de variable :

$$\begin{aligned} X = g(E) &= m + \sigma E \\ E = g^{-1}(X) &= \frac{X - m}{\sigma} \\ \frac{dg^{-1}(x)}{dx} &= \frac{1}{\sigma} \end{aligned}$$

d'où la densité du logarithme de la loi de la durée :

$$f_X(x) = \left| \frac{1}{\sigma} \right| \frac{|q|}{\Gamma(q^{-2})} \exp\left(\beta\left(\psi_0 + \frac{q}{\sigma}(x-m)\right)\right) \exp\left(-\exp\left(\psi_0 + \frac{q}{\sigma}(x-m)\right)\right).$$

Arrivé à ce stade, on remarque que :

$$\begin{aligned} \psi_0 + \frac{q}{\sigma}(x-m) &= \psi_0 + \frac{q}{\sigma}(m + \sigma\varepsilon - m) \\ &= \psi_0 + q\varepsilon \\ &= w, \end{aligned}$$

d'où l'écriture que l'on rencontre habituellement (en imposant $\sigma > 0$) :

$$\begin{aligned} f_X(x) &= \frac{|q|}{\sigma\Gamma(q^{-2})} \exp(\beta w) \exp(-\exp(w)) \\ &= \frac{|q|}{\sigma\Gamma(q^{-2})} \exp(\beta w - \exp(w)), \end{aligned}$$

en utilisant la limite de cette densité quand $q \rightarrow 0$, on obtient la formulation de Prentice (1974) :

$$f_X(x) = \begin{cases} |q| \{\sigma\Gamma(q^{-2})\}^{-1} \exp(q^{-2}w - \exp(w)) & \text{si } q \neq 0 \\ (2\pi\sigma^2)^{-1/2} \exp\{-\frac{1}{2}(x-m)^2/\sigma^2\} & \text{si } q = 0 \end{cases}$$

Prentice (1974) précise que l'on peut utiliser les valeurs négatives de q . Pour retrouver la forme classique des paramètres, avec $\alpha > 0$, $\beta > 0$ et $h > 0$ (donc $q > 0$), il suffit d'inverser leur définition en fonction des paramètres d'intérêt. On trouve :

$$\beta = q^{-2}, \alpha = \frac{q}{\sigma} \text{ et } h = \exp\left(\psi^{(0)}(q^{-2}) - \frac{q}{\sigma}m\right).$$

Pour estimer les modèles avec censure, il faut utiliser la fonction de répartition du logarithme de la durée X . On commence par déterminer la loi de $T = \exp(X)$. Pour $q \neq 0$, on a (avec $\beta = q^{-2}$) :

$$\begin{aligned} f_T(t) &= \frac{|q|}{\sigma t\Gamma(\beta)} \exp(\beta w - \exp(w)) \\ &= \frac{|q|}{\sigma t\Gamma(\beta)} \exp(w)^\beta \exp(-\exp(w)) \end{aligned}$$

soit :

$$f_T(t) = \begin{cases} \frac{|q|}{\sigma t \Gamma(q^{-2})} \exp(\psi_0 + qu)^{q^{-2}} \exp(-\exp(\psi_0 + qu)) & \text{si } q \neq 0 \\ \frac{1}{\sigma t} \varphi(u) & \text{si } q = 0 \end{cases}$$

avec :

$$u = \frac{\ln t - m}{\sigma},$$

la fonction de répartition est donc égale à :

$$F_T(t) = \int_0^t \frac{|q|}{\sigma \tau \Gamma(\beta)} \exp(w(\tau))^\beta \exp(-\exp(w(\tau))) d\tau,$$

avec

$$w(\tau) = \psi_0 + \frac{q}{\sigma} (\ln \tau - m)$$

et on fait le changement de variable :

$$z = \exp(w(\tau)) = \exp\left(\psi_0 + \frac{q}{\sigma} (\ln \tau - m)\right)$$

donc :

$$\tau = \exp\left(m - \frac{\sigma}{q} \psi_0\right) z^{\sigma/q}$$

et

$$d\tau = \exp\left(m - \frac{\sigma}{q} \psi_0\right) \frac{\sigma}{q} z^{\sigma/q-1} dz,$$

la fonction à intégrer s'écrit donc :

$$\begin{aligned} & \frac{1}{\tau} \frac{|q|}{\sigma \Gamma(\beta)} \exp(w(\tau))^\beta \exp(-\exp(w(\tau))) d\tau = \\ & \exp\left(-m + \frac{\sigma}{q} \psi_0\right) z^{-\sigma/q} \frac{|q|}{\sigma \Gamma(\beta)} z^\beta \exp(-z) \exp\left(m - \frac{\sigma}{q} \psi_0\right) \frac{\sigma}{q} z^{\sigma/q-1} dz \end{aligned}$$

ce qui donne :

$$\frac{|q|}{q} \frac{1}{\Gamma(\beta)} z^{\beta-1} \exp(-z) dz$$

après simplification. Les bornes sont données par :

$$\lim_{\tau \rightarrow 0} z = 0, \lim_{\tau \rightarrow t} z = \exp\left(\psi_0 + \frac{q}{\sigma} (\ln t - m)\right) = \exp(w(t)),$$

d'où

$$\begin{aligned} F_T(t) &= \frac{|q|}{q} \frac{1}{\Gamma(\beta)} \int_0^{\exp(w(t))} z^{\beta-1} \exp(-z) dz \\ &= \frac{|q|}{q} \gamma(\beta, \exp(w(t))) \end{aligned}$$

en posant $q > 0$ (donc $\beta > 0$), $|q| = q$, on obtient :

$$F_T(t) = \gamma(\beta, \exp(w(t))), \quad w(t) = \psi_0 + \frac{q}{\sigma} (\ln t - m)$$

de sorte que la survie s'écrit :

$$S_T(t) = 1 - \gamma(\beta, \exp(w(t))).$$

Globalement, si on raisonne sur la variable centrée et réduite $u = (\ln t - m)/\sigma$, on a :

$$S_T(t) = \begin{cases} 1 - \gamma(q^{-2}, \exp(\psi_0 + qu)) & \text{si } q \neq 0 \\ 1 - \Phi(u) & \text{si } q = 0 \end{cases}$$

et l'on estimera le jeu de paramètres (q, m, σ) .

CHAPITRE 3

Estimation paramétrique : aspects théoriques

CE CHAPITRE DONNE LES ÉLÉMENTS qui structurent la totalité du chapitre sur les applications. Il faut le lire avant d'entrer dans les détails de chaque application. Dans ce chapitre, nous verrons essentiellement l'approche par le maximum de vraisemblance. Nous ne discuterons que de l'estimation des modèles censurés, puisque le cas des données complètes s'en déduit immédiatement. On considère dans ce chapitre un N -échantillon de variables de durées indépendamment et identiquement distribuées. Nous allons d'abord voir l'écriture des vraisemblances, puis les changements de paramètres qu'il faut effectuer pour éviter de recourir à l'optimisation sous contraintes. Nous verrons ensuite les statistiques intéressantes à présenter en complément des paramètres du modèle, puis la représentation graphique des fonctions qui caractérisent les lois de la variable de durée.

3.1 Maximum de vraisemblance

On dispose d'un N -échantillon de réalisations d'une variable aléatoire Y . Ses réalisations sont indépendamment et identiquement distribuées selon une loi de densité $f_Y(y; \theta)$, où θ est le vecteur $(p, 1)$ de paramètres de la distribution. La vraisemblance de l'échantillon représente l'intensité d'occurrence de l'échantillon $y = (y_1, \dots, y_N)$, elle est définie par :

$$L(y; \theta) = \prod_{i=1}^N f_Y(y_i; \theta),$$

l'estimateur du maximum de vraisemblance de θ , noté $\hat{\theta}$, est défini par :

$$L(y; \hat{\theta}) = \operatorname{argmax}_{\theta} L(y; \theta),$$

en supposant que le maximum est unique. On trouve ce maximum par les conditions du premier ordre et comme il est plus facile de dériver une somme qu'un produit, on utilise la log-vraisemblance de l'échantillon plutôt que la vraisemblance :

$$\ell(y; \theta) = \ln L(y; \theta).$$

On voit que :

$$\hat{\theta} : L(y; \hat{\theta}) \geq L(y; \theta) \Leftrightarrow \ln L(y; \hat{\theta}) \geq \ln L(y; \theta),$$

donc $\hat{\theta}$ maximise la log-vraisemblance. On peut donc écrire :

$$\ell(y; \theta) = \sum_{i=1}^N \ln f_Y(y_i; \theta),$$

et on résout les conditions du premier ordre :

$$\frac{\partial \ell}{\partial \theta}(y; \hat{\theta}) = \sum_{i=1}^N \frac{1}{f_Y(y_i; \hat{\theta})} \frac{\partial f_Y}{\partial \theta}(y_i; \hat{\theta}) = 0,$$

sachant que la condition du second ordre pour un maximum doit également être vérifiée :

$$\frac{\partial^2 \ell}{\partial \theta \partial \theta'}(y; \hat{\theta}) \ll 0.$$

Sous les hypothèses usuelles :

$$\sqrt{N}(\hat{\theta} - \theta) \overset{a}{\rightsquigarrow} N(0, \Sigma_1)$$

où $\overset{a}{\rightsquigarrow}$ désigne la distribution asymptotique, avec :

$$\begin{aligned} \Sigma_1 &= I_1(\theta)^{-1} = J_1(\theta)^{-1}, \\ I_1(\theta) &= E_Y \left[\frac{\partial \ln f_Y(y_i; \theta)}{\partial \theta} \frac{\partial \ln f_Y(y_i; \theta)}{\partial \theta'} \right], \\ J_1(\theta) &= E_Y \left[-\frac{\partial^2 \ln f_Y(y_i; \theta)}{\partial \theta \partial \theta'} \right]. \end{aligned}$$

Dans la pratique, ce qui nous intéresse n'est toutefois pas $\operatorname{Vas}(\sqrt{N}\hat{\theta})$ mais $\operatorname{Vas}(\hat{\theta})$, on utilisera donc l'approximation suivante :

$$\hat{\theta} \overset{a}{\rightsquigarrow} N(\theta, \Sigma_N),$$

avec :

$$\begin{aligned}\Sigma_N &= \mathbf{I}_N(\theta)^{-1} = \mathbf{J}_N(\theta)^{-1}, \\ \mathbf{I}_N(\theta) &= \mathbb{E}_Y \left[\sum_{i=1}^N \frac{\partial \ln f_Y(y_i; \theta)}{\partial \theta} \frac{\partial \ln f_Y(y_i; \theta)}{\partial \theta'} \right] \\ \mathbf{J}_N(\theta) &= \mathbb{E}_Y \left[- \sum_{i=1}^N \frac{\partial^2 \ln f_Y(y_i; \theta)}{\partial \theta \partial \theta'} \right] = \mathbb{E}_Y \left[- \frac{\partial^2 \ell}{\partial \theta \partial \theta'}(y; \theta) \right],\end{aligned}$$

que l'on peut estimer par :

$$\begin{aligned}\mathbf{I}_N(\hat{\theta}) &= \sum_{i=1}^N \frac{\partial \ln f_Y(y_i; \hat{\theta})}{\partial \theta} \frac{\partial \ln f_Y(y_i; \hat{\theta})}{\partial \theta'}, \\ \mathbf{J}_N(\hat{\theta}) &= - \frac{\partial^2 \ell}{\partial \theta \partial \theta'}(y; \hat{\theta}).\end{aligned}$$

Nous aurions pu prendre l'approximation par les produits croisés des dérivées premières mais les logiciels actuels permettent de calculer facilement le hessien numérique donc nous favoriserons les dérivées secondes. Nous estimerons la variance asymptotique de $\hat{\theta}$ par :

$$\hat{\Sigma}_N = \left(- \frac{\partial^2 \ell}{\partial \theta \partial \theta'}(y; \hat{\theta}) \right)^{-1}.$$

3.2 Ecritures de la vraisemblance

On définit d_i une indicatrice d'observation complète de la variable de durée, définie par :

$$d_i = \begin{cases} 1 & \text{si } t_i \leq \bar{t}_i \\ 0 & \text{si } t_i > \bar{t}_i \end{cases}$$

où t_i est la vraie durée passée dans l'état étudié par l'individu i et \bar{t}_i la durée maximale observable pour l'individu i du fait de la censure. On observe donc une durée :

$$y_i = \min(t_i, \bar{t}_i)$$

et l'on observe $d_i = 1$ quand $y_i = t_i$ et $d_i = 0$ quand $y_i = \bar{t}_i$. Quand la durée est observée complètement son intensité d'occurrence est donnée par sa densité $f_T(y_i) = f_T(t_i)$ alors que lorsque elle est censurée, l'intensité de son occurrence est donnée par la fonction de survie $S_T(y_i) = \Pr(T > y_i) = \Pr(T > \bar{t}_i)$. On peut donc écrire la log-vraisemblance d'une observation ℓ_i sous la forme :

$$\ell_i = d_i \ln f_T(y_i) + (1 - d_i) \ln S_T(y_i),$$

mais on peut également écrire cette définition en utilisant :

$$h_T(y_i) = \frac{f_T(y_i)}{S_T(y_i)} \Leftrightarrow \ln f_T(y_i) = \ln h_T(y_i) + \ln S_T(y_i),$$

de sorte que :

$$\ell_i = d_i \ln h_T(y_i) + \ln S_T(y_i),$$

et que

$$\ln S_T(y_i) = -H_T(y_i)$$

de sorte que l'on peut écrire la log-vraisemblance des trois manières équivalentes suivantes :

$$\begin{aligned} \ell &= \sum_{i=1}^N d_i \ln f_T(y_i) + (1 - d_i) \ln S_T(y_i) \\ &= \sum_{i=1}^N d_i \ln h_T(y_i) + \ln S_T(y_i) \\ &= \sum_{i=1}^N d_i \ln h_T(y_i) - H_T(y_i), \end{aligned}$$

on choisit l'expression la plus pratique selon la distribution.

3.3 Changements de paramètres

Les paramètres des distributions de durée doivent parfois être strictement positifs. Afin d'éviter de recourir à une estimation sous contrainte, on effectue un changement de paramètres qui garantit cette positivité. On supposera ici que le paramètre d'intérêt est ζ et que le paramètre d'optimisation est θ . La transformation la plus utilisée est simplement :

$$\zeta = \exp(\theta), \theta \in \mathbb{R}$$

où ζ est le paramètre d'intérêt (celui de la distribution originelle) et θ le paramètre le plus pratique pour effectuer l'optimisation, appelé paramètre d'optimisation. Une fois l'estimation de θ réalisée, on revient à l'estimation de ζ par la transformation :

$$\hat{\zeta} = \exp(\hat{\theta}) \in \mathbb{R}_+^*.$$

Cette façon de procéder repose sur la propriété d'invariance fonctionnelle du maximum de vraisemblance : il est équivalent d'optimiser la vraisemblance par rapport à ζ ou par rapport à θ , on peut donc choisir la méthode la plus pratique

librement. Plus précisément, on voit que, si $\hat{\theta}$ est l'estimateur du maximum de vraisemblance et si $\zeta = g(\theta)$:

$$\begin{aligned}\ell(g(\hat{\theta})) &\geq \ell(g(\theta)), \forall \theta \\ \Leftrightarrow \ell(\hat{\zeta}) &\geq \ell(\zeta), \forall \zeta,\end{aligned}$$

donc $\hat{\zeta}$ est l'estimateur du maximum de vraisemblance de ζ . La seule difficulté consiste à retrouver la variance asymptotique de $\hat{\zeta}$ à partir de celle de $\hat{\theta}$. Pour cela on utilise le théorème de Slutsky ("delta method", en anglais) :

$$\widehat{\text{Vas}}(g(\hat{\theta})) = \frac{\partial g}{\partial \theta'}(\hat{\theta}) \widehat{\text{Vas}}(\hat{\theta}) \frac{\partial g}{\partial \theta'}(\hat{\theta})',$$

où θ est un vecteur colonne. Donc, si on a $\zeta = g(\theta)$, on estime la variance asymptotique de $\hat{\zeta}$ par :

$$\widehat{\text{Vas}}(\hat{\zeta}) = \frac{\partial g}{\partial \theta'}(\hat{\theta}) \widehat{\text{Vas}}(\hat{\theta}) \frac{\partial g}{\partial \theta'}(\hat{\theta})',$$

et dans le cas particulier $\zeta = \exp(\theta)$, on obtient :

$$\begin{aligned}\frac{\partial g}{\partial \theta}(\theta) &= \exp(\theta) \\ &= \zeta, \\ \widehat{\text{Vas}}(\hat{\zeta}) &= \exp(\hat{\theta})^2 \widehat{\text{Vas}}(\hat{\theta}) \\ &= \hat{\zeta}^2 \widehat{\text{Vas}}(\hat{\theta}).\end{aligned}$$

Une autre transformation est utile pour les paramètres qui doivent rester compris entre 0 et 1 :

$$\zeta = \frac{\exp(\theta)}{1 + \exp(\theta)},$$

dans ce cas :

$$\begin{aligned}\frac{\partial g}{\partial \theta}(\theta) &= \frac{\exp(\theta)}{(1 + \exp(\theta))^2} \\ &= \zeta(1 - \zeta),\end{aligned}$$

donc :

$$\begin{aligned}\widehat{\text{Vas}}(\hat{\zeta}) &= \frac{\exp(\hat{\theta})^2}{(1 + \exp(\hat{\theta}))^4} \widehat{\text{Vas}}(\hat{\theta}) \\ &= \hat{\zeta}^2 (1 - \hat{\zeta})^2 \widehat{\text{Vas}}(\hat{\theta})\end{aligned}$$

3.4 Statistiques utiles

Les paramètres des modèles de durée ne sont pas toujours interprétables directement. On doit donc calculer des paramètres complémentaires comme la moyenne, la médiane et toute quantité utile dans le contexte de l'étude réalisée. Pour cela il faut également appliquer le théorème de Slutsky si l'on veut construire un intervalle de confiance asymptotique. Dans tous les cas, il faut s'assurer que les moments existent car dans de nombreux cas $E(T^k)$ n'existe pas pour certaines valeurs des paramètres. C'est la raison pour laquelle on préfère souvent raisonner sur les quantiles qui ne posent pas ce problème, comme la médiane. Pour faire une prévision, on aura souvent des résultats plus stables avec la médiane qu'avec la moyenne. Pour fixer les idées, prenons le cas de la loi log-logistique :

$$E(T^k) = \left(\frac{1}{h}\right)^{k/\alpha} B(1 - k/\alpha, 1 + k/\alpha), \quad \alpha > k.$$

On ne pourra estimer la durée moyenne que si $\alpha > 1$, et la variance seulement si $\alpha > 2$. C'est le premier point à vérifier. Supposons que l'on veuille estimer la durée moyenne, après correction de la censure, car le maximum de vraisemblance en tient compte. On prendra simplement :

$$\widehat{E}(T) = \left(\frac{1}{\widehat{h}}\right)^{1/\widehat{\alpha}} B(1 - 1/\widehat{\alpha}, 1 + 1/\widehat{\alpha}),$$

où $\widehat{\zeta} = (\widehat{\alpha}, \widehat{h})'$ est l'estimateur de maximum de vraisemblance du modèle log-logistique. Notre estimateur peut donc s'écrire comme une transformation des paramètres initiaux :

$$E(T) \triangleq m = g(\zeta)$$

et sa variance asymptotique peut s'estimer par le théorème de Slutsky :

$$\widehat{\text{Vas}}(\widehat{m}) = \frac{\partial g}{\partial \zeta'}(\widehat{\zeta}) \widehat{\text{Vas}}(\widehat{\zeta}) \frac{\partial g}{\partial \zeta'}(\widehat{\zeta})',$$

on applique la même méthode pour les quantiles.

3.5 Estimation des fonctions

Les représentations graphiques nécessitent d'estimer les fonctions en un nombre suffisant de points. Ces points doivent être choisis par l'utilisateur et constituent une grille sur laquelle on effectuera les représentations graphiques.

Nous prendrons le cas de la loi exponentielle pour fixer les idées. Supposons que l'application porte sur des données en mois et que nous souhaitons faire un graphique sur une période de deux ans, nous prendrions la grille suivante :

$$t = 1, \dots, 24$$

plus généralement, on prendrait :

$$t = t^-, t^- + x, t^- + 2x, \dots, t^+,$$

où t^- est la borne inférieure de l'intervalle de représentation graphique, x le pas de la représentation et t^+ sa borne supérieure. Avec certaines distributions, il est nécessaire d'utiliser une grille plus fine entre les valeurs $t = 0$ et $t = 1$ pour obtenir un beau graphique. On s'adapte au cas par cas. On calcule les valeurs des fonctions par une boucle car elle prend très peu de temps de calcul. Par convention, on range les fonctions par colonne, chaque ligne indiquant une valeur différente de la durée. Il s'agit du format standard des sorties sous SAS.

Exemple 3.1. *Pour évaluer la probabilité qu'une durée de chômage soit supérieure à un an, avec des données mensuelles, on estimera simplement $S_T(12)$. Dans ce cas, il est inutile de calculer toutes les valeurs de la fonction de survie.*

Pour une loi exponentielle, on a simplement :

$$\begin{aligned}\widehat{S}_T(t) &= \exp(-\widehat{h}t), \\ \widehat{h}_T(t) &= \widehat{h}, \\ \widehat{F}_T(t) &= 1 - \exp(-\widehat{h}t), \\ \widehat{f}_T(t) &= \widehat{h} \exp(-\widehat{h}t),\end{aligned}$$

où t est une liste de valeurs certaines choisies par l'utilisateur. Ici, il faut faire attention, les logiciels usuels prennent souvent $t = y_i$ par défaut, de sorte que dans les sorties standard des logiciels les fonctions sont évaluées en des points qui dépendent de l'individu. On peut parfois corriger ce point en utilisant les options appropriées des procédures. On peut également calculer des intervalles de confiance en utilisant le théorème de Slutsky. Prenons le cas de la fonction de survie d'une loi exponentielle. On voit que :

$$\frac{\partial S}{\partial h}(t) = -t \exp(-ht)$$

$$\begin{aligned}\text{Vas}(\widehat{S}(t)) &= \left[\frac{\partial S(t)}{\partial h} \right]^2 \text{Vas}(\widehat{h}) \\ &= t^2 S(t)^2 \text{Vas}(\widehat{h})\end{aligned}$$

on utilisera donc :

$$\widehat{\text{Vas}}(\widehat{S}(t)) = t^2 \widehat{S}(t)^2 \widehat{\text{Vas}}(\widehat{h})$$

et l'intervalle de confiance asymptotique à 95% sera donné par :

$$\widehat{S}(t) \pm 1,96 \times \sqrt{\widehat{\text{Vas}}(\widehat{S}(t))} = \widehat{S}(t) \pm 1,96 \times t \widehat{S}(t) \sqrt{\widehat{\text{Vas}}(\widehat{h})},$$

et on peut procéder de la même manière pour les autres fonctions.

CHAPITRE 4

Pratique de l'estimation paramétrique

CE CHAPITRE PRÉSENTE les applications tirées des chapitres précédents. Nous avons introduit les différentes notions très progressivement de la manière indiquée dans le tableau 4.1. Nous commencerons par rappeler, pour chaque modèle, les notions théoriques nécessaires à l'estimation. L'algorithme d'optimisation que nous utiliserons par défaut est celui de Newton-Raphson. Pour chaque programme, nous verrons la programmation des modules de lecture, d'estimation et de prévision sous SAS-IML. Nous réaliserons l'ensemble des estimations sur l'enquête SIP (Santé et Itinéraires Professionnels), qui est facilement accessible.

4.1 Loi exponentielle

4.1.1 Log-vraisemblance

Les logarithmes du hasard et de la survie sont donnés par :

$$\begin{aligned}\ln h_T(y_i) &= \ln h, \forall i \\ \ln S_T(y_i) &= -hy_i,\end{aligned}$$

donc :

$$\begin{aligned}\ell(h) &= (\ln h) \sum_{i=1}^N d_i - h \sum_{i=1}^N y_i \\ &= N_d \ln h - h \sum_{i=1}^N y_i\end{aligned}$$

TABLEAU 4.1 – TECHNIQUES D'ESTIMATION APPLIQUÉES

| Distribution | Algorithme | Calcul | Page |
|----------------|--------------------------------------|---------------------|------|
| Exponentielle | Langage SAS-IML de base | | 47 |
| Gamma 1 | Routine d'optimisation <i>nlpnra</i> | Dérivées numériques | 66 |
| Lomax | Routine d'optimisation <i>nlpnra</i> | Gradient analytique | 80 |
| Etat absorbant | Balayage et routine <i>nlpnra</i> | Gradient analytique | 90 |
| Weibull | Routine d'optimisation <i>nlpnra</i> | Hessien analytique | 110 |
| Gamma 2 | Optimisation programmée | Dérivées numériques | 127 |
| Log-normal | Optimisation programmée | Gradient analytique | 139 |
| Log-logistique | Optimisation programmée | Hessien analytique | 151 |
| Pareto | Méthode BHHH | Gradient analytique | 165 |
| Gamma 3 | Balayage et routine <i>nlpnra</i> | Dérivées numériques | 177 |
| Burr 12 | Balayage et routine <i>nlpnra</i> | Gradient analytique | 201 |

avec $N_d = \sum_{i=1}^N d_i$, d'où le score :

$$\frac{\partial \ell}{\partial h} = \frac{N_d}{h} - \sum_{i=1}^N y_i$$

et le hessien :

$$\frac{\partial^2 \ell}{\partial h^2} = -\frac{N_d}{h^2} < 0.$$

4.1.2 Estimation

Le maximum est unique et donné par la condition du premier ordre :

$$\frac{N_d}{\hat{h}} - \sum_{i=1}^N y_i = 0 \Leftrightarrow \hat{h} = \frac{N_d}{\sum_{i=1}^N y_i}$$

et l'on peut estimer sa variance asymptotique par :

$$\widehat{\text{Vas}}(\hat{h}) = \frac{N_d}{\hat{h}^2} = \frac{(\sum_{i=1}^N y_i)^2}{N_d}.$$

Cette variance peut servir à calculer un intervalle de confiance asymptotique pour le paramètre h de la manière suivante :

$$\hat{h} \pm u_{1-\alpha/2} \sqrt{\widehat{\text{Vas}}(\hat{h})}$$

où $u_{1-\alpha/2}$ est le quantile $1 - \alpha/2$ de la loi normale centrée et réduite. Pour un intervalle de confiance asymptotique de niveau 95%, on prend donc $u_{0,975} = 1.96$. Le t de Student est donné par :

$$\hat{t}_h = \frac{\hat{h}}{\sqrt{\widehat{\text{Vas}}(\hat{h})}}.$$

4.1.3 Statistiques utiles

Dans les sorties, la moyenne et la médiane sont également utiles, ainsi que leur intervalle de confiance. La moyenne est estimée par :

$$\hat{m} = \frac{1}{\hat{h}}$$

on estime sa variance en appliquant le théorème de Slutsky :

$$\begin{aligned} m &= g(h) = \frac{1}{h}, \quad \frac{\partial g}{\partial h} = -\frac{1}{h^2}, \\ \widehat{\text{Vas}}(\widehat{m}) &= \frac{\partial g}{\partial h}(\widehat{h})^2 \text{Vas}(\widehat{h}) \\ &= \frac{1}{\widehat{h}^4} \text{Vas}(\widehat{h}). \end{aligned}$$

On estime la médiane et sa variance asymptotique par :

$$\begin{aligned} \text{Me} &= g(h) = \frac{\ln 2}{h}, \quad \frac{\partial g}{\partial h} = -\frac{\ln 2}{h^2} \\ \widehat{\text{Vas}}(\widehat{\text{Me}}) &= \frac{\partial g}{\partial h}(\widehat{h})^2 \text{Vas}(\widehat{h}) \\ &= \frac{(\ln 2)^2}{\widehat{h}^4} \text{Vas}(\widehat{h}). \end{aligned}$$

Arrivés à ce stade, on remarque que les t de Student asymptotiques des estimateurs de la moyenne et de la médiane sont égaux :

$$\widehat{t}_m = \frac{\widehat{m}}{\sqrt{\widehat{\text{Vas}}(\widehat{m})}} = \frac{\widehat{h}}{\sqrt{\widehat{\text{Vas}}(\widehat{h})}} = \widehat{t}_h$$

et

$$\widehat{t}_{\text{Me}} = \frac{\widehat{\text{Me}}}{\sqrt{\widehat{\text{Vas}}(\widehat{\text{Me}})}} = \frac{\widehat{h}}{\sqrt{\widehat{\text{Vas}}(\widehat{h})}} = \widehat{t}_h.$$

Enfin, on peut vouloir tracer les fonctions qui définissent la distribution. Pour cela il faut choisir une plage de valeurs de t , par exemple :

$$t = 1, 2, \dots, t^+,$$

on estime alors les fonctions pour tous les points définis par la grille choisie par l'utilisateur :

$$\begin{aligned} \widehat{f}_T(t) &= \widehat{h} \exp(-\widehat{h}t), \\ \widehat{h}_T(t) &= \widehat{h}, \\ \widehat{F}_T(t) &= 1 - \exp(-\widehat{h}t), \\ \widehat{S}_T(t) &= \exp(-\widehat{h}t). \end{aligned}$$

4.1.4 Programmation I : Une introduction à SAS-IML

Nous réalisons l'estimation sous SAS-IML. Comme il n'y a pas d'optimisation, cette section visera juste à introduire les notions de base de ce langage. Le programme suivant sera commenté ligne à ligne. Nous utiliserons ici une programmation par module car certains modules sont communs à plusieurs programmes d'estimation, ce qui permet de procéder par copier-coller, tout en clarifiant l'exposé. Les instructions IML sont résumées dans les tableaux 4.2 à 4.6.

1. `%let tab=duree;`
`%let y=duree_emp;`
`%let d=presence;`
`%let outp=_outp;`
`%let outi=_outi;`

Ce premier bloc contient des macro-variables créées par une instruction `%let`. On appelle ces macro variables en ajoutant un caractère `&` devant leur nom. Il s'agit des paramètres du programme : nom du tableau contenant les données (`&tab`), nom de la variable de durée (`&y`), nom de l'indicatrice de durée complète (`&d`), nom du tableau de sortie des paramètres (`&outp`, fixé à `_outp` par défaut), nom du tableau de sortie des prévisions (`&outi`, fixé à `_outi` par défaut). Les macro variables fonctionnent selon le principe du copier-coller. Chaque macro variables rencontrée est remplacée par le texte qu'elle contient, et le programme résultant est envoyé au compilateur SAS.

2. `proc iml;`

On lance la procédure IML ("Interactive Matrix Language") qui est un langage de programmation matriciel. Il comporte également de nombreuses fonctions et routines statistiques et économétriques.

3. `start lecture;`

Le module de lecture est déclaré par une instruction `start` et finira par une instruction `finish`. Ce module vise à lire les données et à effectuer un premier affichage résumant les données sur lesquelles on travaille, le nombre d'observations et le nombre de censures.

4. `nom_y={&y};`

Crée une variable IML de type caractère `nom_y` qui contient le nom de la variable de durée. Il faut mettre des accolades.

5. `nom_d={&d};`

Crée une variable IML de type caractère `nom_d` qui contient le nom de l'indicatrice de durée complète. Il faut mettre des accolades.

6. `use &tab;`
Ouvre le tableau SAS dont le nom figure dans la macro variable `&tab`.
7. `read all var(nom_y) into yi;`
Lit toutes (*all*) les observations de la variable de durée `nom_y` et les stocke dans le vecteur colonne IML nommé `yi`. Il faut mettre des parenthèses quand les noms sont dans un vecteur IML, ce qui est notre cas ; si on entrait les noms des variables directement, il faudrait mettre des accolades à la place des parenthèses.
8. `read all var(nom_d) into di;`
Lit toutes (*all*) les observations de l'indicateur de durée complète `nom_d` et les stocke dans le vecteur colonne IML nommé `di`.
9. `nd=sum(di); n=nrow(yi); nc=n-nd;`
Effectue la somme des indicateurs de durée complète, $\sum_{i=1}^n d_i$, ce qui donne le nombre de durées complètes N_d . L'instruction `nrow` donne le nombre de lignes d'une matrice ou d'un vecteur. Le nombre de lignes du vecteur `yi` est égal au nombre d'observations N . On en déduit le nombre de durées censurées $N_c = N - N_d$.
10. `print "Time variable =" nom_y,`
`"Censoring variable (=0 if censored) =" nom_d,`
`"Number of observations =" n,`
`"Complete =" nd,`
`"Censored =" nc;`
Les cinq lignes qui précèdent constituent une seule instruction *print*, qui sert à imprimer des matrices et des vecteurs. La ligne précédente imprime d'abord un texte "Time variable =", qui doit être mis entre guillemets, puis imprime le nom de la variable de durée étudiée, qui est stocké dans `nom_y`. La virgule qui suit indique qu'il faut effectuer un *saut de ligne*. Le résultat sera donc : Time variable = duree_emp. Ensuite on imprime le nom de la variable de censure, en précisant qu'une censure correspond à $d_i = 0$, puis on imprime le nombre total d'observations, le nombre d'observations complètes et le nombre d'observations censurées.
11. `finish lecture;`
Fin du module de lecture. Ce module ne sera pas exécuté avant d'avoir été lancé explicitement par l'utilisateur, à la fin du programme. Il est clairement possible d'utiliser ce module pour toutes les estimations de ce chapitre car il n'utilise pas l'hypothèse de loi exponentielle. Le lecteur est encouragé à se constituer une bibliothèque de modules, stockés au for-

mat texte, afin de programmer plus vite et de réduire les risques d'erreur de syntaxe.

12. `start estim;`

Début du module d'estimation. Ce module est spécifique à la loi exponentielle et ne peut pas être utilisé tel quel pour les autres distributions.

13. `print "Exponential distribution", ,
"S(t)=exp(-h.t)", , ;`

On commence par rappeler à l'utilisateur quelle est la distribution dont on estime les paramètres. Pour éliminer les risques d'erreur d'interprétation des paramètres, on rappelle l'expression de la fonction de survie qui est utilisée dans le programme. Pour des distributions plus complexes que la loi exponentielle, il existe souvent plusieurs paramétrages possibles, c'est la raison pour laquelle il faut rappeler celui qu'utilise le programme.

14. `h=nd/sum(yi); vh=h##2/nd; sh=sqrt(vh);
th=abs(h)/sh; ph=1-probchi(th##2,1);`

On calcule l'estimateur du maximum de vraisemblance \hat{h} que l'on range dans l'élément IML `h`. La fonction `sum` additionne tous les éléments d'une matrice ou d'un vecteur. On calcule ensuite l'estimation de sa variance asymptotique $\widehat{\text{Var}}(\hat{h})$ que l'on range dans `vh`, de son écart-type asymptotique $\sqrt{\widehat{\text{Var}}(\hat{h})}$ que l'on range dans `sh`, et de son t de Student asymptotique que l'on range dans `th`. La probabilité critique est calculée en utilisant la normalité asymptotique du t de Student; son carré suit donc asymptotiquement une distribution du χ^2 centrée à 1 degré de liberté sous $H_0 : h = 0$. On en déduit que $ph = 1 - F_{\chi_1^2}(th^2)$ où $F_{\chi_1^2}$ est la fonction de répartition du χ^2 centré à 1 degré de liberté. Sous SAS-IML, on a le choix entre deux syntaxes pour la fonction de répartition à d degrés de liberté prise au point x : soit `probchi(x,d)` soit `cdf("chisquare",x,d)`. Il est également possible de traiter les distributions décentrées mais nous n'en avons pas besoin.

15. `mt=1/h; vmt=mt##2#vh; smt=sqrt(vmt);
tmt=abs(mt)/smt; pmt=1-probchi(tmt##2,1);`

Calcul de l'estimation de l'espérance mathématique `mt` de la variable de durée après correction de la censure. On calcule également les statistiques associées en utilisant la méthode de Slutsky.

16. `md=log(2)/h; vmd=log(2)##2#vh/h##4;
smd=sqrt(vmd); tmd=abs(md)/smd;
pmd=1-probchi(tmd##2,1);`

Calcul de l'estimation de la médiane md de la variable de durée après correction de la censure. On y ajoute les statistiques associées.

17. `lv=sum(di#log(h)-h#yi) ;`

Calcul de la log-vraisemblance définie par $\sum_{i=1}^N d_i \ln h - h y_i$. On commence par constituer le vecteur d'éléments $d_i \ln h - h y_i$ avant de faire la somme de ses éléments par la fonction `sum`. L'opérateur `#` est très utile car il sert à réaliser quatre types d'opérations : (i) la multiplication de deux nombres réels ; (ii) la multiplication de tous les éléments d'un vecteur ou d'une matrice par un nombre réel ; (iii) le produit de Hadamard (noté \odot) ou la multiplication terme à terme de tous les éléments de deux matrices de format identique, $A = [a_{ij}]$, $B = [b_{ij}]$, $A \odot B = [a_{ij} b_{ij}]$; (iv) le produit de chaque ligne d'une matrice par une pondération rangée en colonne, $A = [a_i]$, $B = [b_{ij}]$, $A \# B = [a_i b_{ij}]$. Ici, on commence par multiplier chaque élément du vecteur d_i par $\ln h$, ce qui donne le vecteur colonne $[d_i \ln h]$, puis on multiplie chaque élément du vecteur y_i par h , ce qui donne le vecteur colonne $[h y_i]$, la différence de ces deux vecteurs donne le vecteur colonne $[d_i \ln h - h y_i]$. La log-vraisemblance est alors obtenue comme la somme des éléments de ce vecteur.

18. `resp=(lv|.|.|.|.|.)/(h|sh|th|ph)//
(md|smd|tmd|pmd)/(mt|smt|tmt|pmt) ;`

On crée une matrice `resp` qui contient tous les résultats d'estimation. Notre but est d'afficher les résultats sous forme synthétique et de les exporter plus tard dans une feuille de calcul. L'opérateur `| |` (touche `AltGr+6`) sert à concaténer horizontalement deux vecteurs ou matrices ayant le même nombre de lignes. Ainsi $A||B$ donne la matrice (A, B) . L'opérateur `//` sert à concaténer verticalement deux vecteurs ou matrices ayant le même nombre de colonnes. Donc :

$$A//B = \begin{pmatrix} A \\ B \end{pmatrix}$$

La première ligne de la matrice `resp` contient la log-vraisemblance. Comme on ne calcule pas de statistique additionnelle sur cette quantité, on remplit les autres colonnes par des valeurs manquantes notées par un point `(.)` sous SAS-IML. On empile cette ligne sur la ligne des résultats portant sur le paramètre h , puis sur la médiane, puis sur la moyenne.

19. `labc="Parameter"|"Std_Dev"|"Student"|"P_Value" ;`

Le vecteur `labc` contient les noms des colonnes de la matrice `resp`. Il s'agit de l'estimation du paramètre, de l'écart-type, du t de Student et de la probabilité critique. Nous avons fait en sorte que ces noms soient des

noms de variables admissibles sous SAS car ils serviront de noms de colonne lors de l'exportation des résultats. On remplace les espaces par des caractères de soulignement (touche 8).

20. `labr="Log-likelihood"// "h"// "Median"// "Mean" ;`
Le vecteur `labr` indique les noms des lignes de la matrice `resp`. La log-vraisemblance, le paramètre h , la médiane et la moyenne.
21. `print resp[colname=labc rowname=labr] ;`
On imprime les résultats rangés dans la matrice `resp`. Le logiciel utilisera le vecteur `labc` pour les noms de colonne (*colname*) et `labr` pour les noms de ligne (*rowname*), ce qui rendra l'affichage plus clair.
22. `finish estim ;`
Fin du module d'estimation. Il n'est pas exécuté à ce stade, il faudra l'appeler explicitement plus loin par une instruction `run`.
23. `start prev ;`
Début du module de prévision. Il servira à tracer les fonctions qui caractérisent la distribution des durées
24. `tmin=min(yi) ; tmax=max(yi) ; tstep=1 ;`
Par défaut, le programme effectue des prévisions pour toutes les valeurs comprises entre le minimum et le maximum de la base de données. Il est possible de modifier ces bornes selon les besoins, ainsi que l'incrément `tstep`.
25. `do time=tmin to tmax by tstep ;`
Boucle allant, par défaut, de $t = \min(y_i)$ jusqu'à $t = \max(y_i)$ par pas de 1. Donc, dans le cas standard avec `tmin=1`, cela donne 1, 2, 3 etc.
26. `st=exp(-h*time) ;`
Fonction de survie $\hat{S}(t) = \exp(-\hat{h} \times t)$.
27. `ht=h ;`
Fonction de hasard, constante par définition de la loi exponentielle, $\hat{h}(t) = \hat{h}$.
28. `ct=1-st ;`
Fonction de répartition : $\hat{F}(t) = 1 - \hat{S}(t)$
29. `ft=h*exp(-h*time) ;`
Densité : $\hat{f}(t) = \hat{h} \times \exp(-\hat{h} \times t)$.
30. `resi=resi/(time || st || ht || ct || ft) ;`
Chaque prévision donne une ligne de données, qui est stockée avec les lignes précédentes. Quand `time=tmin` on obtient en fait l'instruction `resi=(time || st || ht || ct || ft)` car la matrice `resi` n'existe pas

encore. Ensuite, on range la nouvelle ligne de données sous les lignes précédentes grâce aux opérateurs de concaténation. Cette syntaxe est pratique car il n'est pas nécessaire d'initialiser le vecteur quand $time=tmin$.

31. `end;`

Fin de la boucle.

32. `labi=nom_y || "SDF" || "Hasard" || "CDF" || "Density";`

Vecteur contenant les noms de colonnes de la matrice `resi` : le nom de la variable de durée stockée dans `nom_y`, SDF pour "Survival Distribution Function" ou fonction de survie, "Hasard" pour la fonction de hasard, CDF pour "Cumulative Distribution Function" ou fonction de répartition, et "Density" pour la densité. Il s'agit de noms admissibles comme noms SAS, ce qui est important pour créer des tableaux en sortie.

33. `finish prev;`

Fin du module de prévision.

34. `start sortie;`

Début du module de sortie, qui crée les tableaux SAS en sortie. Il s'agit d'un module à caractère général, comme le module de lecture, et il pourra donc être utilisé dans tous les autres programmes.

35. `create &outp from resp[colname=labc];`

`append from resp;`

Crée un tableau SAS de nom donné par la macro variable `&outp`, soit `_outp` dans le répertoire de travail ("Work") par défaut. Ce tableau contient les paramètres. Les noms de colonne sont pris dans le vecteur `labc`.

36. `create _np from labr[colname="Statistic"];`

`append from labr;`

Crée un tableau SAS contenant les noms des lignes de la matrice `resp`. Sous IML, on ne peut pas former de matrice combinant des caractères et des nombres. On doit donc sauvegarder cette première colonne à part, et la fusionner ensuite sous SAS après la sortie d'IML. La colonne contenant les noms des paramètres s'appellera "Statistic" une fois le tableau SAS créé. On peut entrer le nom de variable directement dans l'option `colname`.

37. `create &outi from resi[colname=labi];`

`append from resi;`

Crée un tableau SAS contenant les prévisions. Ce tableau portera le nom donné par la macro-variable `&outi`, soit `_outi` dans le répertoire de travail ("Work") par défaut. Les noms de colonne sont donnés par `labi`.

38. `finish sortie;`
Fin du module de sortie.
39. `run lecture;`
L'instruction `run` suivi d'un nom de module lance l'exécution de ce module. Ici, on lance le module `lecture`.
40. `run estim;`
Lance le module `estim`.
41. `run prev;`
Lance le module `prev`.
42. `run sortie;`
Lance le module `sortie`.
43. `quit;`
Quitte SAS-IML, on revient au langage SAS standard.
44. `data &outp; merge _np &outp;`
On fusionne le tableau contenant les estimations avec les noms de ligne.
45. `run;`
Lance toutes les instructions précédentes.

Arrivé à ce stade, on dispose de la sortie d'écran et des tableaux SAS qui seront stockés dans le répertoire de travail par défaut. Il est recommandé d'exporter ensuite les tableaux SAS `_outp` et `_outi` dans un format de feuille de calcul, soit en utilisant le menu "File → Export Data", soit en utilisant la procédure *export* de SAS. Voici un exemple pour une exportation au format Excel :

```
proc export data=_outp
outfile="c:\Estim.XLS"
replace
dbms=excel5;
run;
```

Cette procédure va chercher le fichier SAS `_outp` situé dans le répertoire de travail et l'exporter dans un fichier Excel de nom `Estim.XLS`, situé sous la racine du lecteur "c:". L'instruction `replace` autorise la réécriture sur un fichier déjà existant de même nom; l'instruction `dbms=excel5` est nécessaire pour que la procédure fonctionne bien sous certaines configurations. Elle donne un format Excel 95 que l'on pourra mettre à jour sous le tableur lui-même.

4.1.5 Application I

Nous travaillerons sur une extraction de la base SIP (Santé et Itinéraires Professionnels) qui permet de suivre la carrière des individus. Dans un premier temps, nous allons nous intéresser à la durée du premier emploi occupé, de nom DUREE_EMP, exprimée en années. Le lancement du programme fournit la sortie suivante.

Sortie SAS 4.1.

```

                                nom_y
Time variable = DUREE_EMP                                nom_d

Censoring variable (=0 if censored) = PRESENCE
                                                n

Number of observations =      12695
                        nd

Complete =      11277
          nc

Censored =      1418

Exponential distribution

S(t)=exp(-h.t)

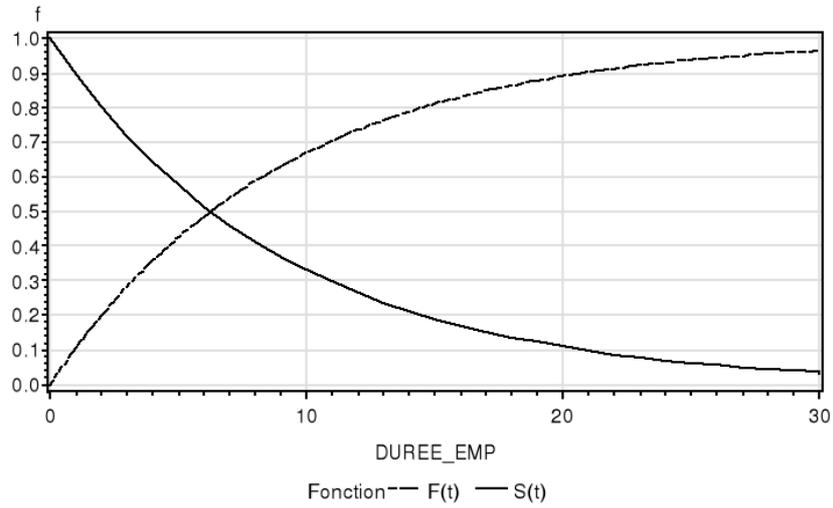
                                resp
Parameter   Std_Dev   Student   P_Value

Log-likelihood -36091.6      .           .
h              0.1107521 0.0010429 106.19322    0
Median         6.2585468 0.0589355 106.19322    0
Mean           9.0291744 0.0850259 106.19322    0

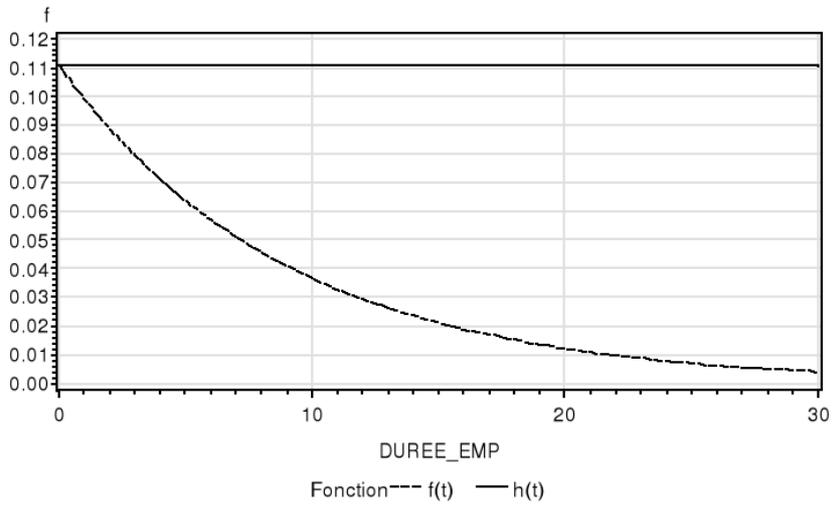
```

Sur les $N = 12645$ observations, $N_d = 1418$ sont censurées, soit un taux de 11,2%. Ce faible taux de censure s'explique par la définition de la variable retenue : le premier emploi est assez éloigné pour la plupart des individus de la base, de sorte qu'on observe entièrement cette période pour 88,8% des individus. Le paramètre de la loi exponentielle est estimé

$$\hat{h} = 0.11$$



GRAPHIQUE 4.1 – LOI EXPONENTIELLE, $\hat{S}_T(t)$ ET $\hat{F}_T(t)$



GRAPHIQUE 4.2 – LOI EXPONENTIELLE, $\hat{h}_T(t)$ ET $\hat{f}_T(t)$

et l'on peut construire son intervalle de confiance asymptotique de niveau 95% de la manière suivante :

$$[0.11 - 1.96 \times 0.001, 0.11 + 1.96 \times 0.001] = [0.10804, 0.11196]$$

ce qui est très précis. La précision vient de la taille de la base de données. A partir de ce paramètre, on peut estimer les durées moyenne et médiane du premier emploi occupé par les individus au début de leur carrière. La durée moyenne est estimée par :

$$\widehat{E}(T) = 9 \text{ ans}$$

avec un intervalle de confiance asymptotique de niveau 95% égal à :

$$[8.86, 9.16],$$

et la médiane est plus faible :

$$\widehat{\text{Me}}(T) = 6.3 \text{ ans},$$

avec un intervalle de confiance asymptotique de niveau 95% égal à :

$$[6.1, 6.4].$$

On peut également tracer les fonctions usuelles. Nous avons choisi de le faire pour les durées inférieures à 30 ans et en normalisant les fonctions en 0, soit :

$$S_T(0) = 1, F_T(0) = 0, h_T(0) = h, f_T(0) = h,$$

ce qui donne les graphiques 4.1 et 4.2. Sur le graphique 4.1 on reconnaît la fonction de survie parce qu'elle est décroissante et la fonction de répartition parce qu'elle est croissante. On reconnaît la fonction de hasard sur le graphique 4.2 parce qu'elle est toujours située au dessus de la densité.

TABLEAU 4.2 – INSTRUCTIONS ET FONCTIONS IML (1/5)

| Instructions IML | Signification |
|---|--|
| start Z; finish Z; run Z; | Début du module Z Fin du module Z (nom optionnel) Lancement du module Z |
| start f(x) global(a,b); return (z); finish f; | Début de la fonction f de paramètre entrant x. Les paramètres a et b sont disponibles en entrée et en sortie. Renvoie la quantité z vers l'appel de la fonction définie plus haut f(x). Fin de la fonction f (nom optionnel) |
| use TAB; read all var{Z1 Z2} into M; close TAB; | Lit le tableau SAS de nom TAB Lit toutes les observations des variables SAS Z1 et Z2 et les range dans la matrice IML de nom M Ferme le tableau TAB (optionnel) |
| read all var(NZ) into M; | Variante : lit les variables dont les noms figurent dans le vecteur de caractères IML de nom NZ et les range dans la matrice IML de nom M. |
| create SORTIE from M; append from M; | Crée le tableau SAS SORTIE à partir de la matrice IML M. Remplit le tableau SORTIE avec la matrice IML M (obligatoire). |
| from M[colname=("A" "B")]; | Variante : affecte les noms A et B aux variables du tableau SAS créé à partir de la matrice M |
| from M[colname=NZ]; | Variante : affecte les noms figurant dans le vecteur de caractères IML NZ aux variables du tableau SAS créé à partir de la matrice M. |

TABLEAU 4.3 – INSTRUCTIONS ET FONCTIONS IML (2/5)

| Instructions et fonctions | Signification |
|--|---|
| m1+m2 | Addition de deux matrices IML (m1 et m2) de mêmes dimensions, ou de deux nombres |
| m1-m2 | Soustraction de deux matrices IML (m1 et m2) de mêmes dimensions, ou de deux nombres |
| m1/m2 | Division terme à terme de deux matrices IML (m1 et m2) de mêmes dimensions, ou de deux nombres. Ne donne <i>pas</i> $(m1)^{-1}m2$. |
| m1#m2 | Produit de Hadamar. Multiplication terme à terme de deux matrices IML (m1 et m2) de mêmes dimensions, ou de deux nombres. Ne donne <i>pas</i> le produit matriciel. Voir les détails page 54 |
| a##b | a^b . Si a est une matrice, tous ses éléments sont élevés à la puissance b . Il ne s'agit <i>pas</i> de la puissance au sens matriciel du terme. |
| m1[i,j] m1[i,] m1[,j] | Element situé sur la i-ème ligne et la j-ème colonne de la matrice m1 i-ème ligne de la matrice m1 (vecteur ligne). j-ième colonne de la matrice m1 (vecteur colonne). |
| m1[+,] m1[,+] m1[+,+] ou sum(m1) t(m1) m1*m2 | Vecteur ligne des sommes de chaque colonne de la matrice m1. Vecteur colonne des sommes de chaque ligne de la matrice m1. Somme des éléments de la matrice m1. Transposée de la matrice m1 Produit matriciel de m1 et m2. Le nombre de colonnes de m1 doit être égal au nombre de lignes de m2. |

TABLEAU 4.4 – INSTRUCTIONS ET FONCTIONS IML (3/5)

| Instructions et fonctions | Signification |
|---------------------------------|--|
| abs(x) | Valeur absolue |
| beta(x,y) | Fonction Beta, $B(x, y) = \Gamma(x)\Gamma(y)/\Gamma(x + y)$ |
| char(x) | Transforme une donnée numérique en caractère |
| cdf("dist",x,p1,...,pk) | Fonction de répartition de la distribution dist, prise au point x, avec les paramètres p1,..., pk. Voir la documentation pour la liste des distributions et le paramétrage spécifique à chacune d'entre elles. |
| cinv(x,d) | Fonction inverse (quantile) de la fonction de répartition du χ^2 à d degrés de liberté |
| digamma(x) | Dérivée du logarithme de la fonction Gamma |
| exp(x) | Exponentielle |
| gamma(x) | Fonction Gamma |
| log(x) | Logarithme népérien |
| pdf("dist",x,p1,...,pk) | Densité. Voir cdf. |
| probchi(x,dl) | Fonction de répartition du χ^2 à dl degrés de liberté, prise au point x |
| quantile("dist",prob,p1,...,pk) | Inverse de la fonction de répartition. Voir cdf. |
| sqrt(x) | Racine carrée |
| trigamm(x) | Dérivée seconde du logarithme de la fonction Gamma |

TABLEAU 4.5 – INSTRUCTIONS ET FONCTIONS IML (4/5)

| Instructions et fonctions | Signification |
|---------------------------|---|
| a :b | Génère un vecteur ligne dont les valeurs vont de a à b par pas de 1 : a, a+1, a+2,...,b. |
| m1 m2 | Donne (m1,m2), on ajoute les colonnes de m2 à celles de m1, les matrices doivent avoir le même nombre de lignes. |
| m1//m2 | Donne $\begin{pmatrix} m1 \\ m2 \end{pmatrix}$, on ajoute les lignes de m2 à celle de m1, les matrices doivent avoir le même nombre de colonnes. |
| ginv(m1) | Inverse de Penrose-Moore de la matrice m1. Si la matrice est de plein rang colonne elle est égale à $(m1'm1)^{-1}m1'$. |
| inv(m1) | Inverse de la matrice m1. Elle doit être carrée et de plein rang |
| j(a,b,c) | Crée une matrice de a lignes, b colonnes, dont tous les éléments valent c. |
| min(m1) | Plus petit élément de la matrice m1 |
| max(m1) | Plus grand élément de la matrice m1 |
| ncol(m1) | Nombre de colonnes de la matrice m1 |
| nrow(m1) | Nombre de lignes de la matrice m1 |
| vecdiag(m1) | Sélectionne la diagonale de la matrice m1 et la range dans un vecteur colonne. |

TABLEAU 4.6 – INSTRUCTIONS ET FONCTIONS IML (5/5)

| Instructions et fonctions | Signification |
|---|---|
| do i=a to b by c; instructions; end; | Boucle s'exécutant pour i=a à b par pas de c. Le pas est en option, et vaut 1 par défaut |
| do while (condition); instructions; end; | Boucle s'exécutant tant que la condition est remplie |
| do until (condition); instructions; end; | Boucle s'exécutant jusqu'à ce que la condition soit remplie. |
| call nlpfdd(f0,g0,h0,"f",x0); call nlpfdd(f0,g0,h0,"f",x0) grd="sc"; | Routine de dérivation : "f" est la fonction à dériver et x0 le point auquel on évalue toutes les fonctions. f0 est la valeur de la fonction en x0, g0 le gradient et h0 le hessien. Le module qui calcule la fonction "f" et x0 sont donnés en entrée, et f0, g0 et h0 sont obtenus en sortie. Même routine, mais le gradient est calculé par la fonction "sc" qui est donnée en entrée. |
| opt=j(11,1,0); opt[1]=1; call nlpnra(rc,x,"f",x0,opt); call nlpnra(rc,x,"f",x0,opt) grd="sc"; call nlpnra(rc,x,"f",x0,opt) grd="sc" hes="h"; | Optimisation par la méthode de Newton-Raphson. opt est le vecteur d'options en entrée. Il contient 11 composantes, opt[1]=1 indique une maximisation, opt[2] indique le degré de détail à l'affichage, entre 0 et 5. Le module "f" indique la fonction à maximiser, x0 la valeur initiale de l'algorithme. En sortie, on obtient le code de résultat rc (qui oit être positif) et la valeur de la variable qui maximise la fonction Même routine, mais le gradient est calculé par la fonction "sc". Même routine, mais le hessien est calculé par la fonction "h". |
| call sort(m,a); call sort(m,a,a); | Trie la matrice m par ordre croissant de sa a-ième colonne. Trie la matrice m par ordre décroissant de sa a-ième colonne. Il existe des possibilités de tri plus avancées. |

4.2 Loi Gamma à un paramètre

4.2.1 Log-vraisemblance

Les logarithmes de la densité et de la survie sont donnés par :

$$\begin{aligned}\ln f_T(y_i, \beta) &= (\beta - 1) \ln y_i - y_i - \ln \Gamma(\beta), \\ \ln S_T(y_i, \beta) &= \ln(1 - \gamma(\beta, y_i)),\end{aligned}$$

donc :

$$\ell(\beta) = \sum_{i=1}^N d_i \ln f_T(y_i, \beta) + (1 - d_i) \ln S_T(y_i, \beta).$$

Il n'existe pas d'expression explicite pour l'estimateur de β , il faut donc procéder par optimisation. Ici, il faudra prendre une précaution. La quantité $\gamma(\beta, y_i)$ est une intégrale qui peut être légèrement tronquée par les routines de calcul. Il faut s'assurer qu'elle n'atteint jamais la valeur 1 (le cas limite $\gamma(\beta, +\infty)$), car dans ce cas la quantité $\ln S_T(y_i, \beta)$ ne serait plus définie. Nous la remplaçons donc par :

$$\min(\gamma(\beta, x), 1 - \varepsilon) = \begin{cases} \gamma(\beta, x) & \text{si } \gamma(\beta, x) \leq 1 - \varepsilon \\ 1 - \varepsilon & \text{sinon} \end{cases}$$

dans la pratique une valeur $\varepsilon = 10^{-6}$ est suffisante. La fonction Gamma tronquée normalisée doit par ailleurs être obtenue via la fonction de répartition du χ^2 :

$$\gamma(\beta, x) = F_{\chi^2, 2\beta}(2x),$$

on prendra donc :

$$\gamma(\beta, x) = \min(F_{\chi^2, 2\beta}(2x), 1 - \varepsilon),$$

avec $\varepsilon = 10^{-6}$. Nous définirons un module spécifique pour le calcul de cette fonction. Il servira également pour les lois Gamma à 2 et 3 paramètres.

4.2.2 Estimation

Pour l'estimation, on a besoin que $\beta > 0$ donc on effectue le changement de paramètre suivant :

$$\beta = \exp(\theta), \theta \in \mathbb{R}.$$

On réécrit la vraisemblance en fonction de ce nouveau paramètre :

$$\begin{aligned}\ell_\zeta(\theta) &= \ell(\zeta(\theta)) \\ &= \sum_{i=1}^N d_i \ln f_T(y_i, \exp(\theta)) + (1 - d_i) \ln S_T(y_i, \exp(\theta)),\end{aligned}$$

et on optimise en θ :

$$\hat{\theta} = \operatorname{argmax}_{\theta} \ell(\theta).$$

Pour revenir au paramètre β , on utilise la propriété d'invariance fonctionnelle du maximum de vraisemblance :

$$\hat{\beta} = \exp(\hat{\theta})$$

et la formule de Slutsky :

$$\operatorname{Vas}(\hat{\beta}) = \exp(\hat{\theta})^2 \operatorname{Vas}(\hat{\theta}) = \hat{\beta}^2 \operatorname{Vas}(\hat{\theta}).$$

4.2.3 Statistiques utiles

Dans les sorties, la moyenne et la médiane sont également utiles. La moyenne est estimée simplement par :

$$\hat{m} = \hat{\beta},$$

et la variance de l'estimateur $\hat{\beta}$ donne directement celle de l'estimateur de la moyenne. On estime la médiane par :

$$\operatorname{Me} = \frac{1}{2} \times F_{\chi^2, 2\hat{\beta}}^{-1}\left(\frac{1}{2}\right).$$

Pour calculer sa variance asymptotique, il faudrait recourir à des algorithmes d'intégration numérique, ce qui dépasse le cadre de cet ouvrage. Les fonctions qui définissent la distribution peuvent également être tracées. Pour cela il faut choisir une plage de valeurs de t , par exemple :

$$t = 1, 2, \dots, t^+,$$

on estime alors les fonctions en tous les points définis par la grille choisie par l'utilisateur, en utilisant les relations :

$$\begin{aligned} \hat{f}_T(t) &= \frac{t^{\hat{\beta}-1} \exp(-t)}{\Gamma(\hat{\beta})}, \\ \hat{F}_T(t) &= \gamma(\hat{\beta}, t) \\ \hat{S}_T(t) &= 1 - \hat{F}_T(t) \\ \hat{h}_T(t) &= \frac{\hat{f}_T(t)}{\hat{S}_T(t)}. \end{aligned}$$

4.2.4 Programmation II : Une introduction à l'optimisation numérique

Nous allons maximiser la vraisemblance à l'aide de l'algorithme de Newton-Raphson décrit en annexe (E.3). Nous verrons tous les détails de la programmation de cet algorithme plus loin dans cet ouvrage. Pour ce premier contact nous adopterons une optique d'économètre utilisateur, et évoluerons progressivement vers une optique d'économètre programmeur. Sous SAS-IML cette méthode d'optimisation est obtenue par la routine IML *nlpnra*. Le préfixe *nlp* est l'acronyme de "non linear programing" qui signifie optimisation non linéaire en anglais. On retrouve ce préfixe pour toutes les routines IML qui concernent ce type d'optimisation, ainsi que pour la routine de calcul numérique des dérivées. L'acronyme *nra* signifie "Newton-Raphson algorithm". Une deuxième routine IML nous sera utile pour calculer le hessien et donc pour estimer la variance asymptotique de l'estimateur. Il s'agit de la routine *nlpfdd*, où *fdd* signifie "Finite difference derivatives", où calcul des dérivées par la méthode des différences finies. La routine *nlpnra* possède 11 paramètres, nous n'en utiliserons que deux. Le premier permet d'indiquer si on maximise la fonction où si on la minimise. Par défaut, *nlpnra* fait une minimisation, il faudra donc impérativement fixer la bonne valeur pour ce paramètre (1 pour une maximisation). Le deuxième paramètre que nous utiliserons fixe la quantité de détails à imprimer sur l'exécution de la routine, entre 0 (pas d'impression) et 5 (avec dérivées à chaque étape). Nous commencerons par imprimer le plus grand niveau de détails. Voici le programme d'estimation.

```
1. proc iml;
  start lecture;
  nom_y={&y};
  nom_d={&d};
  use &tab; read all var(nom_y) into yi;
  read all var(nom_d) into di;
  nd=sum(di); n=nrow(yi); nc=n-nd;
  print "Time variable =" nom_y,
  "Censoring variable (=0 if censored) =" nom_d,
  "Number of observations =" n,
  "Complete =" nd,
  "Censored =" nc;
  finish lecture;
```

Nous utilisons le même module de lecture que pour la loi exponentielle.

```
2. start igf(beta,x);
```

L'instruction `start` sert également à définir des fonctions, et pas seule-

ment des modules. Cette fonction calcule la fonction Gamma tronquée normalisée (décrite en annexe). La seule différence de syntaxe porte sur l'existence de paramètres (`beta`, `x`) qui sont les arguments de la fonction `igf` (pour "incomplete Gamma function").

3. `f=probchi(2#x,2#beta);`
Ces instructions calculent $F_{\chi^2, 2\beta}(2x)$. Si x est un vecteur, on obtient les applications de la fonction à chacun des éléments du vecteur pris séparément. On range ces valeurs dans un vecteur `f` qui est temporaire car créé à l'intérieur de la fonction.
4. `do i=1 to nrow(f);`
On effectue une boucle sur les éléments du vecteur `f`. Le nombre d'éléments du vecteur est égal à son nombre de lignes car c'est un vecteur colonne.
5. `f[i]=min(f[i],1-1e-6);`
On s'assure que les éléments de `f` n'atteignent pas la valeur 1. Si c'est le cas, on fixe leur valeur à $1 - 10^{-6}$. Sous SAS IML 10^{-6} s'écrit `1e-6`.
6. `end;`
Fin de la boucle de vérification.
7. `return (f);`
On renvoie le vecteur `f` (le résultat de la fonction `igf`) vers l'appel de la fonction. Cet appel se situe généralement à l'intérieur d'une formule.
8. `finish;`
Fin de la définition de la fonction `igf`.
9. `start lv(theta) global(di,yi,lyi);`
On définit la fonction `lv` qui calcule la log-vraisemblance de l'échantillon. Son argument est `theta`. Des arguments additionnels interviennent également dans cette fonction, ils sont mis dans l'option `global`. On n'optimise pas par rapport à ces arguments donc on les met à part. L'option `global` permet à la fois d'accéder à des éléments autres que l'argument de la fonction mais aussi d'envoyer des éléments à l'extérieur de la fonction autrement que par l'instruction `return`. Le vecteur `lyi` contient $\ln y_i$ (voir plus loin).
10. `beta=exp(theta);`
Comme la vraisemblance est définie par rapport à θ , on pose $\beta = \exp(\theta)$ au début de la fonction et on écrit la vraisemblance par rapport à ses paramètres originaux.
11. `lfi=(beta-1)#lyi-yi-lgamma(beta);`

Calcul du logarithme de la densité de la loi Gamma pour l'observation i :

$$(\beta - 1) \ln y_i - y_i - \ln \Gamma(\beta),$$

le point important à remarquer est que :

on utilise la fonction `lgamma(beta)` et non `log(gamma(beta))`

En effet, $\Gamma(\beta)$ est une fonction qui tend très rapidement vers l'infini, de sorte que pour des valeurs élevées de β , le logiciel peut rencontrer un dépassement sur $\Gamma(\beta)$ alors qu'il n'y en a pas sur $\ln \Gamma(\beta)$, car l'application du logarithme réduit considérablement la valeur d'une fonction. Il faut donc toujours utiliser la fonction `lgamma` quand c'est possible pour éviter des problèmes de dépassement de capacité ("overflow", en anglais).

12. `lsi=log(1-igf(beta,yi));`

Calcul du logarithme de la fonction de survie pour l'observation i :

$$\ln(1 - \gamma(\beta, y_i))$$

la définition de la fonction `igf` permet de s'assurer que ce logarithme existe toujours.

13. `lv=sum(di#lfi+(1-di)#lsi);`

Calcul de la log-vraisemblance :

$$\sum_{i=1}^N d_i \ln f_T(y_i, \exp(\theta)) + (1 - d_i) \ln S_T(y_i, \exp(\theta))$$

14. `return (lv);`

Renvoi de la log-vraisemblance vers l'appel de la fonction. Cet appel sera fait automatiquement par la routine d'optimisation.

15. `finish;`

Fin de la définition de la log-vraisemblance.

16. `start estim;`

Début du module d'estimation.

17. `print "One parameter Gamma distribution", ,
"S(t)=1-g(beta,t)", ,
"where g(,) is the regularized incomplete Gamma
function", ;`

On rappelle la forme de la fonction de survie utilisée.

18. `lyi=log(yi)` ;

On calcule le vecteur $[\ln y_i]$ une seule fois au début du programme, pour économiser du temps de calcul. Le calculer à chaque fois que l'on calcule la vraisemblance allongerait considérablement la durée d'estimation, car la fonction `lv` sert aussi à calculer les dérivées premières et secondes.

19. `beta0=sum(yi)/nd` ;

Valeur initiale du paramètre β . On n'a pas besoin d'un estimateur convergent. Une bonne valeur initiale est une valeur qui mène au maximum sans rencontrer de problème en route. Ici, on utilise le fait que le paramètre β est égal à l'espérance mathématique de la variable de durée. Le terme en $\sum_{i=1}^n y_i / N_d$ est l'estimateur que l'on pourrait déduire de la loi exponentielle. Son ordre de grandeur devrait donc être raisonnable.

20. `theta0=log(beta0)` ;

Il s'agit de la valeur initiale de l'algorithme, car on n'optimise pas sur β mais sur $\theta = \ln \beta$. On prend donc :

$$\theta_0 = \ln \left(\frac{\sum_{i=1}^N y_i}{N_d} \right).$$

21. `opt=j(11,1,0)` ;

Vecteur d'options de la routine IML `nlpnra`. L'instruction `j(a,b,c)` crée une matrice de dimensions $a \times b$ donc tous les éléments sont égaux à c . On crée donc un vecteur colonne de zéros. Il s'agit de la valeur initiale des 11 paramètres de la routine `nlpnra`.

22. `opt[1]=1` ;

L'instruction `m[i]` donne le i -ème élément de la matrice `m`. L'instruction précédente fixe le premier paramètre à 1. Ce paramètre est très important : il indique que l'on veut réaliser une *maximisation*.

23. `opt[2]=5` ;

Fixe le paramètre d'impression à 5, nous aurons le maximum de détails sur l'optimisation qui sera réalisée. Les 9 autres paramètres sont laissés à leur valeur par défaut.

24. `call nlpnra(rc,theta,"lv",theta0,opt)` ;

L'instruction `call` lance l'exécution de la routine IML `nlpnra`. Les deux premiers paramètres de cette routine sont des paramètres en sortie, tous les autres sont des paramètres en entrée. Le premier paramètre `rc` est un code de résultat ("result code"). Il est positif quand l'optimisation s'est (relativement) bien passée. Il est nul sinon. Le deuxième paramètre est

l'estimateur du maximum de vraisemblance $\hat{\theta}$. Les paramètres en entrée sont les suivants : le nom de la fonction à optimiser (définie par une instruction `start` et dont l'argument est `theta`), ici il s'agit de la log-vraisemblance définie par la fonction "lv". On doit mettre le nom entre guillemets. Ensuite, on met la valeur initiale de l'algorithme, donnée par `theta0`, puis le vecteur d'options `opt`. Cette ligne réalise toute l'optimisation. Nous aurons donc juste à gérer les résultats.

25. `print "Result code (must be > 0) =" rc;`
Impression du code de résultat. Il doit être positif.

26. `call nlpfdd(lv1,gr,he,"lv",theta);`
On a besoin de la dérivée seconde à l'optimum pour estimer la covariance asymptotique. Elle doit être négative pour que l'on soit bien à un maximum et que l'estimateur de la variance du paramètre soit bien positif. `nlpfdd` est une routine de calcul de dérivées qui, dans sa version de base, comporte 5 paramètres (ce chiffre peut monter à 7 comme nous le verrons dans les autres applications). Les trois premiers paramètres sont des quantités en sortie : la valeur de la fonction (`lv1`), de sa dérivée première (`gr`) et de sa dérivée seconde (`he`). Les deux autres paramètres sont des quantités en entrée : le nom de la fonction (correctement définie sous SAS IML), ici il s'agit de la log-vraisemblance, et le point auquel on souhaite évaluer les dérivées, ici $\hat{\theta}$. En sortie on obtient donc :

$$\begin{aligned} \text{lv1} &= \ell_{\zeta}(\hat{\theta}), \\ \text{gr} &= \frac{\partial \ell_{\zeta}}{\partial \theta}(\hat{\theta}), \\ \text{he} &= \frac{\partial^2 \ell_{\zeta}}{\partial \theta^2}(\hat{\theta}). \end{aligned}$$

27. `elas=abs(theta#gr/lv1);`

Le gradient doit être nul à l'optimum, mais cette quantité est sensible à l'échelle dans laquelle sont exprimées les variables. On préfère donc souvent calculer l'élasticité de la log-vraisemblance au paramètre. Cette quantité est définie par :

$$e(\hat{\theta}) = \frac{\partial \ell_{\zeta}}{\partial \theta}(\hat{\theta}) \frac{\hat{\theta}}{\ell_{\zeta}(\hat{\theta})},$$

quand $\hat{\theta}$ augmente de 1%, la log-vraisemblance varie de $e(\hat{\theta})\%$. Cette quantité est insensible aux effets d'échelle et doit être très proche de 0 à l'optimum. On prend sa valeur absolue en utilisant la fonction IML `abs`.

```
28. print "Log-likelihood at maximum =" lv1,
      "Gradient" gr,
      "Elasticity =" elas,
      "Second order derivative =" he ;
```

Impression des résultats de la maximisation. La quantité `lv1` sert à faire (au moins) les tests du rapport des maxima de vraisemblance, `gr` et `elas` servent à vérifier que la dérivée est bien nulle en $\hat{\theta}$. La quantité `he` sert à vérifier que l'on est bien parvenu à un maximum ($\partial^2 \ell / \partial \theta^2 (\hat{\theta}) < 0$).

```
29. vtheta=-1/he ;
```

Estimation de la variance asymptotique de l'estimateur :

$$\widehat{\text{Vas}}(\hat{\theta}) = - \left[\frac{\partial^2 \ell_{\zeta}}{\partial \theta^2}(\hat{\theta}) \right]^{-1}.$$

```
30. beta=exp(theta) ;
```

On revient au paramètre d'intérêt :

$$\hat{\beta} = \exp(\hat{\theta})$$

```
31. vb=beta##2#vtheta ;
```

On applique le théorème de Slutsky pour obtenir une estimation de la variance asymptotique de $\hat{\beta}$:

$$\widehat{\text{Vas}}(\hat{\beta}) = \hat{\beta}^2 \widehat{\text{Vas}}(\hat{\theta}).$$

```
32. sb=sqrt(vb) ;
```

Estimation de l'écart-type asymptotique de $\hat{\beta}$.

```
33. tb=beta/sb ;
```

Calcul du t de Student asymptotique.

```
34. pb=1-probchi(tb##2,1) ;
```

Calcul de la probabilité critique correspondante.

```
35. resp=beta || sb || tb || pb ;
```

On met les résultats dans un vecteur ligne `resp`.

```
36. labc="Estimation" || "Std_Dev" || "Student" || "Pvalue" ;
```

Noms des colonnes du futur tableau SAS de résultats.

```
37. me=cinv(0.5,2#beta)/2 ;
```

Estimation de la médiane de la distribution de durée (la moyenne est donnée directement par $\hat{\beta}$). La fonction `cinv(x, p)` donne l'inverse de la fonction de répartition du χ_p^2 . On peut également utiliser la syntaxe `quantile("chisquare",0.5,2#beta)/2`.

38. `resp=resp/(me||.||.||.);`
 On ajoute la médiane au tableau de résultat, en remplissant avec des valeurs manquantes quand cela est nécessaire.
39. `labr="Beta (Mean)"//"Median";`
 Noms des lignes du futur tableau SAS de résultats.
40. `print resp[colname=labc rowname=labr];`
 Impression des résultats.
41. `finish estim;`
 Fin du module d'estimation.
42. `start prev;`
 Début du module de prévision. Sa structure est identique à celui utilisé pour la loi exponentielle, on change juste la valeur des fonctions.
`tmin=min(yi); tmax=max(yi); tstep=1;`
`do time=tmin to tmax by tstep;`
`fi=time##(beta-1)#exp(-time)/gamma(beta);`
`si=1-igf(beta,time);`
`hi=fi/si;`
`ci=1-si;`
`resi=resi/(time||fi||si||hi||ci);`
`end;`
`labi="&y"||"Density"||"SDF"||"Hasard"||"CDF";`
`finish prev;`
 Fin du module de prévision
43. `start sortie; create &outp from resp[colname=labc];`
`append from resp;`
`create _np from labr[colname="Statistic"];`
`append from labr;`
`create &outi from resi[colname=labi];`
`append from resi;`
`finish sortie;`
`run lecture;`
`run estim;`
`run prev;`
`run sortie;`
`quit;`
`data &outp; merge _np &outp;`
`run;`
 Le module de sortie est identique à celui du cas exponentiel.

4.2.5 Application II

On reprend la base de données précédente. La loi Gamma, contrairement à la loi exponentielle, autorise le hasard à varier avec la durée, ce que nous pourrions examiner graphiquement. La sortie nous montre d'abord le fonctionnement détaillé de la routine *nlpnra* qui réalise l'optimisation selon la méthode de Newton-Raphson. Ce degré de détail est obtenu en mettant l'option d'impression à 5 lors de l'appel de la routine. La première partie de la sortie donne la valeur initiale, appelée X1 par la routine. Plus généralement le p -ième paramètre du modèle est appelé X p par la routine. Elle indique également le gradient au point initial (-41408) et la log-vraisemblance au point initial (-55335.77708).

La routine présente ensuite les paramètres d'optimisation qui seront utilisés. Nous avons pris les valeurs par défaut qui sont affichées dans la sortie. Dans les cas usuel, on n'a pas besoin de les changer. La routine affiche ensuite les résultats de chaque itération. Les plus importants sont la valeur de la log-vraisemblance ("objective function", $\ell(\theta^{(k)})$), sa variation par rapport au point précédent ("obj fun change", $\ell(\theta^{(k)}) - \ell(\theta^{(k-1)})$) et la plus grande valeur absolue des composantes du gradient ("Max Abs Gradient Element", $\max |s(\theta^{(k)})|$). Les deux dernières quantités tendent vers 0 quand on s'approche de l'optimum. Ensuite, la routine affiche le paramètre $\theta^{(k)}$ et le gradient $s(\theta^{(k)})$. L'affichage continue jusqu'à la convergence de l'algorithme ou jusqu'à ce que le nombre maximum d'itérations autorisées soit atteint. Ici, l'algorithme converge en 5 itérations.

Sortie SAS 4.2.

```

One parameter Gamma distribution
      S(t)=1-g(beta,t)
where g(.) is the regularized incomplete Gamma function

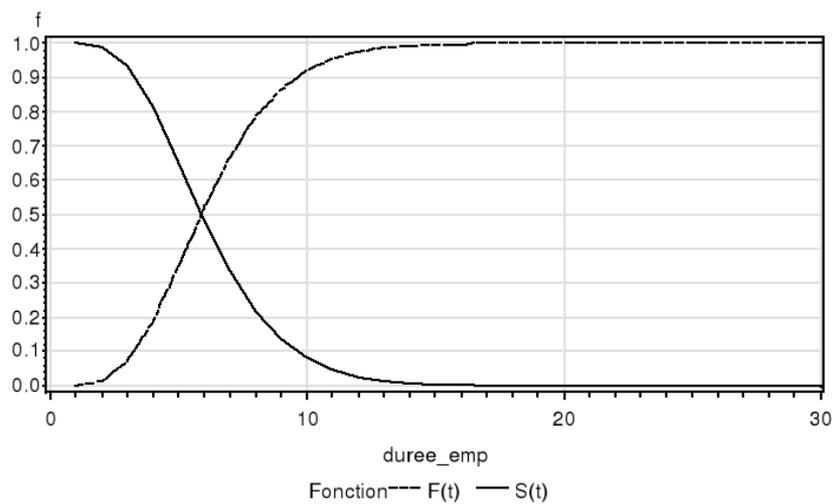
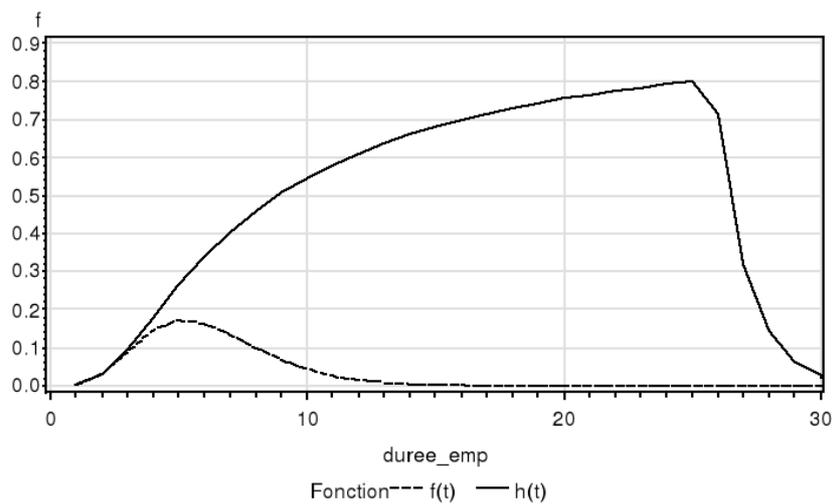
      Optimization Start
      Parameter Estimates      1

      N Parameter      Estimate      Gradient
      Objective
      Function
1 X1      2.200461      -41408

Value of Objective Function = -55335.77708

Newton-Raphson Optimization with Line Search

```

GRAPHIQUE 4.3 – LOI GAMMA 1, $\hat{S}_T(t)$ ET $\hat{F}_T(t)$ GRAPHIQUE 4.4 – LOI GAMMA 1, $\hat{h}_T(t)$ ET $\hat{f}_T(t)$

```

Minimum Iterations           0
Maximum Iterations           200
Maximum Function Calls       500
ABSGCONV Gradient Criterion   0.00001
GCONV Gradient Criterion     1E-8
GCONV2 Gradient Criterion    0
ABSFCONV Function Criterion  0
FCONV Function Criterion     2.220446E-16
FCONV2 Function Criterion    0
FSIZE Parameter              0
ABSXCONV Parameter Change Criterion 0
XCONV Parameter Change Criterion 0
XSIZE Parameter              0
ABSCONV Function Criterion   1.340781E154
Line Search Method            2
Starting Alpha for Line Search 1
Line Search Precision LSPRECISION 0.9
DAMPSTEP Parameter for Line Search .
MAXSTEP Parameter for Line Search 0
FD Derivatives: Accurate Digits in Obj.F 15.653559775
Singularity Tolerance (SINGULAR) 1E-8
    
```

Newton-Raphson Optimization with Line Search
 Without Parameter Scaling
 Gradient Computed by Finite Differences
 CRP Jacobian Computed by Finite Differences

Parameter Estimates
 Optimization Start

```

Active Constraints           0
Objective Function -55335.77708
Max Abs Gradient   41408.429288
Element
    
```

| Iter | Rest arts | Func Calls | Act Con | Objective Function | Max Abs | | Step Size |
|------|-----------|------------|---------|--------------------|----------------|------------------|-----------|
| | | | | | Obj Fun Change | Gradient Element | |
| 1 | 0 | 3 | 0 | -48913 | 6422.4 | 9098.1 | 0.997 |

Slope of
 Search
 Direction
 -11067

X1
 Parmes 1.9340777727
 Grad 9098.1145629

| Rest | Func | Act | Objective | Obj Fun | Max Abs Gradient | Step |
|------|------|-----|-----------|---------|------------------|------|
|------|------|-----|-----------|---------|------------------|------|

| Iter | arts | Calls | Con | Function | Change | Element | Size |
|------|------|-------|-----|----------|--------|---------|-------|
| 2 | 0 | 4 | 0 | -48458 | 454.9 | 857.7 | 1.000 |

Slope of
Search
Direction
-843.3

X1
Parms 1.8413901082
Grad 857.69476268

| Iter | Rest arts | Func Calls | Act Con | Objective Function | Obj Fun Change | Max Abs Gradient Element | Step Size |
|------|--------------|---------------|------------|-----------------------|-------------------|--------------------------------|--------------|
| 3 | 0 | 5 | 0 | -48452 | 6.3580 | 192.7 | 1.000 |

Slope of
Search
Direction
-8.879

X1
Parms 1.8310377679
Grad 192.66465564

| Iter | Rest arts | Func Calls | Act Con | Objective Function | Obj Fun Change | Max Abs Gradient Element | Step Size |
|------|--------------|---------------|------------|-----------------------|-------------------|--------------------------------|--------------|
| 4 | 0 | 6 | 0 | -48452 | 0.2285 | 0.2966 | 1.000 |

Slope of
Search
Direction
-0.457

X1
Parms 1.8286680813
Grad 0.2965590736

| Iter | Rest arts | Func Calls | Act Con | Objective Function | Obj Fun Change | Max Abs Gradient Element | Step Size |
|------|--------------|---------------|------------|-----------------------|-------------------|--------------------------------|--------------|
| 5 | 0 | 7 | 0 | -48452 | 8.413E-7 | 0.0727 | 1.000 |

Slope of
Search
Direction
-109E-8

```

                                X1
      Parms      1.8286644179
      Grad       0.0726726029

      Optimization Results
Iterations      5  Function Calls      8
Hessian Calls  6  Active Constraints   0
Objective Function -48451.84303  Max Abs Gradient  0.0726726029
                                Element
Slope of Search -1.086412E-6  Ridge      0
Direction
GCONV convergence criterion satisfied.
NOTE: At least one element of the (projected) gradient is
      greater than 1e-3.

```

```

      Optimization Results
      Parameter Estimates

      N Parameter      Estimate      Gradient
                                Objective
                                Function
      1 X1              1.828664      -0.072673

```

Value of Objective Function = -48451.84303

```

                                rc
      Result code (must be >0) =      6
                                lvl

```

Log-likelihood at maximum = -48451.84

```

                                gr
      Gradient -0.072673
                                elas
      Elasticity = 2.7428E-6
                                he
      Second order derivative = -80932.46

```

```

                                resp
      Estimation  Std_Dev  Student  Pvalue
Log-likelihood  -48451.84  .        .
Beta            6.2255663  0.0218835  284.48632  0
Median          5.8955938  .        .

```

On trouve que la log-vraisemblance est maximisée en un point où le gradient est de -0.072673 . Cette valeur peut paraître élevée à première vue, mais un examen plus approfondi montre que l'élasticité de la log-vraisemblance en ce point est de 2.7428×10^{-6} , ce qui est très faible. En faisant varier le paramètre β de 1%, on ne pourrait faire varier la log-vraisemblance que de 3 millièmes de pourcentage. On a donc atteint le maximum. Notons dès maintenant, que le maximum atteint est plutôt faible. En effet, avec la loi exponentielle, la log-

vraisemblance maximale était de -36091 , alors qu'avec la loi Gamma à un paramètre elle est de -48452 , ce qui est beaucoup plus faible. Ici, on peut penser que le modèle Gamma est beaucoup moins bon que le modèle exponentiel, ce qui explique sans doute les résultats que nous obtenons.

Le paramètre β est estimé à $\hat{\beta} = 6.22$. Il s'agit aussi de la durée moyenne estimée. Elle s'éloigne assez de la durée trouvée pour la loi exponentielle (9 ans). Par contre la médiane, estimée à 5,9 ans est proche de l'estimation obtenue pour la loi exponentielle (6.3 ans).

L'examen graphique fournit également des informations intéressantes. La fonction de survie (graphique 4.3) décroît très rapidement et la fonction de hasard présente un profil en forme de vague, typique de la distribution Gamma (graphique 4.4). Notons ici que ce type de variation de la fonction de hasard est rarement souhaitable car, au voisinage du hasard maximum, il implique une chute du hasard entre deux durées assez proches, ce qui n'est pas toujours crédible. Globalement, l'estimation associée à la loi Gamma à un paramètre ne semble pas très adaptée aux données que nous avons. Ce point sera confirmé par les estimations ultérieures.

4.3 Loi de Lomax à un paramètre

4.3.1 Log-vraisemblance

Les logarithmes du hasard et de la survie sont donnés par :

$$\begin{aligned}\ln h_T(y_i) &= \ln h - \ln(1 + hy_i), \\ \ln S_T(y_i) &= -\ln(1 + hy_i),\end{aligned}$$

donc :

$$\begin{aligned}\ell(h) &= \sum_{i=1}^N d_i (\ln h - \ln(1 + hy_i)) - \ln(1 + hy_i), \\ &= N_d \ln h - \sum_{i=1}^N (1 + d_i) \ln(1 + hy_i)\end{aligned}$$

avec $N_d = \sum_{i=1}^N d_i$. Le paramètre d'intérêt doit être positif.

4.3.2 Estimation

Pour imposer la condition de positivité sur h , on effectue le changement de paramètre :

$$h = \exp(\theta),$$

et l'on réécrit la vraisemblance sous la forme :

$$\begin{aligned}\ell_{\zeta}(\theta) &= \ell(\zeta(\theta)) \\ &= N_d \theta - \sum_{i=1}^N (1 + d_i) \ln(1 + \exp(\theta) y_i),\end{aligned}$$

le vecteur du score est donné par :

$$\begin{aligned}\frac{\partial \ell_{\zeta}}{\partial \theta} &= N_d - \sum_{i=1}^N (1 + d_i) \frac{\exp(\theta) y_i}{1 + \exp(\theta) y_i} \\ &= N_d - \sum_{i=1}^N (1 + d_i) \frac{h y_i}{1 + h y_i}\end{aligned}$$

et le hessien par :

$$\frac{\partial^2 \ell_{\zeta}}{\partial \theta^2} = - \sum_{i=1}^N (1 + d_i) \frac{\exp(\theta) y_i}{(1 + \exp(\theta) y_i)^2} \ll 0.$$

La variance asymptotique de $\hat{\theta}$ est donné par :

$$\widehat{\text{Vas}}(\hat{\theta}) = \left[- \frac{\partial^2 \ell_{\zeta}}{\partial \theta^2}(\hat{\theta}) \right]^{-1}.$$

On revient au paramètre d'intérêt en utilisant :

$$\hat{h} = \exp(\hat{\theta}),$$

et on obtient la variance par le théorème de Slutsky :

$$\begin{aligned}\text{Vas}(\hat{h}) &= \exp(\hat{\theta})^2 \text{Vas}(\hat{\theta}) \\ &= \hat{h}^2 \text{Vas}(\hat{\theta}).\end{aligned}$$

4.3.3 Statistiques utiles

Cette distribution n'admet pas de moyenne mais on peut estimer sa médiane :

$$\begin{aligned}\text{Me} &= g(h) = \frac{1}{h}, \quad \frac{\partial g}{\partial h} = -\frac{1}{h^2}, \\ \text{Vas}(\widehat{\text{Me}}) &= \frac{\partial g}{\partial h}(\hat{h})^2 \text{Vas}(\hat{h}) \\ &= \frac{1}{\hat{h}^4} \text{Vas}(\hat{h}).\end{aligned}$$

Les fonctions s'estiment par :

$$\begin{aligned}\hat{h}_T(t) &= \frac{\hat{h}}{1 + \hat{h}t}, \\ \hat{S}_T(t) &= \frac{1}{1 + \hat{h}t}, \\ \hat{f}_T(t) &= \hat{h}_T(t) \hat{S}_T(t), \\ \hat{F}_T(t) &= 1 - \hat{S}_T(t).\end{aligned}$$

4.3.4 Programmation III : Optimisation avec gradient analytique

Nous allons maintenant utiliser l'expression théorique des dérivées premières. On recourt à cette méthode quand on veut accélérer les routines d'optimisation et de dérivation. Nous intégrerons une nouvelle fonction qui calcule la dérivée première puis nous l'utiliserons dans les routines *nlpnra* et *nlpfdd*. Nous avons également vérifié, dans un premier temps, que notre dérivée théorique donne le même résultat que les dérivées numériques afin d'éviter des erreurs de recopie. Nous commentons maintenant le programme d'estimation.

1.

```
proc iml;
  start lecture;
  nom_y={&y};
  nom_d={&d};
  use &tab; read all var(nom_y) into yi;
  read all var(nom_d) into di;
  nd=sum(di); n=nrow(yi); nc=n-nd;
  print "Time variable =" nom_y,
  "Censoring variable (=0 if censored) =" nom_d,
  "Number of observations =" n,
  "Complete =" nd,
  "Censored =" nc;
  finish lecture;
  Ce module de lecture est identique aux précédents.
```
2.

```
start lv(theta) global(di,yi);
```


Fonction définissant la vraisemblance.
3.

```
h=exp(theta);
```


On optimise par rapport au paramètre θ , pour écrire la vraisemblance on utilise le paramètre d'origine $h = \exp(\theta)$.
4.

```
lsi=-log(1+h#yi);
```


Logarithme de la fonction de survie.

5. `lv=sum(di#(theta+lsi)+lsi);`

Calcul de la log-vraisemblance :

$$\ell_{\zeta}(\theta) = \sum_{i=1}^N d_i (\ln h - \ln(1 + hy_i)) - \ln(1 + hy_i).$$

6. `return (lv);`

Renvoi de la log-vraisemblance vers l'appel de la fonction (i.e., la routine d'optimisation).

7. `finish;`

Fin de la fonction qui définit la vraisemblance.

8. `start sc(theta) global(di,yi,nd);`

Fonction définissant la dérivée de la log-vraisemblance. Cette dérivée doit être calculée par rapport au même paramètre que celui utilisé dans la définition de la vraisemblance, c'est-à-dire θ et non h . La quantité `nd` a été ajoutée à l'option `global` car le nombre de durées complètes entre dans l'expression de la dérivée.

9. `h=exp(theta);`

On exprimera cette dérivée par rapport à h car cette formulation est plus pratique.

10. `der=nd-sum((1+di)#h#yi/(1+h#yi));`

Calcul du vecteur du score (la dérivée première de la log-vraisemblance) :

$$\frac{\partial \ell_{\zeta}}{\partial \theta}(\theta) = N_d - \sum_{i=1}^N \frac{(1 + d_i) h y_i}{1 + h y_i}.$$

11. `return (der);`

Renvoi de la dérivée vers l'appel de la fonction (i.e., la routine d'optimisation).

12. `finish;`

Fin de la définition de la fonction définissant la dérivée.

13. `start estim;`

Début du module d'estimation.

14. `print "One parameter Lomax distribution", ,
"S(t)=1/(1+h.t)", ;`

On rappelle la formulation utilisée pour la fonction de survie.

15. `h0=sum(yi)/nd;`

Valeur initiale du paramètre h .

16. `theta0=log(h0);`
Valeur initiale du paramètre d'optimisation θ .
17. `opt=j(11,1,0); opt[1]=1; opt[2]=1;`
On fixe les options. Avec la deuxième option à 1, on demande un affichage minimal sur le déroulement de la routine d'optimisation.
18. `call nlpnra(rc,theta,"lv",theta0,opt) grd="sc";`
On lance la routine d'optimisation. Par rapport au cas précédent (loi Gamma à un paramètre), on ajoute l'option `grd="sc"` qui donne le nom de la fonction qui calcule la dérivée de la log-vraisemblance (`grd` est l'abréviation de gradient). Cette fonction remplacera l'évaluation numérique des dérivées et sera également utilisée pour faire une évaluation numérique des dérivées secondes. Elle accélère l'estimation.
19. `print "Result code (must be > 0) =" rc;`
Impression du code de résultat, il doit être positif.
20. `call nlpfdd(lv1,gr,he,"lv",theta) grd="sc";`
On calcule la valeur de la log-vraisemblance au maximum ainsi que ses dérivées. Par rapport au cas précédent (loi Gamma à un paramètre) on ajoute l'option `grd="sc"` qui indique à la routine d'utiliser directement l'expression du gradient donnée par la fonction `sc`, et d'utiliser cette fonction pour calculer les dérivées secondes numériquement. Ceci permet d'accélérer le calcul.
21. `elas=abs(theta#gr/lv1);`
Calcul de l'élasticité de la log-vraisemblance par rapport au paramètre d'optimisation θ .
22. `print "Log-likelihood at maximum =" lv1,`
`"Gradient" gr,`
`"Elasticity =" elas,`
`"Second order derivative =" he;`
Impression de la valeur de la log-vraisemblance à son maximum et des dérivées. La dérivée première doit être nulle et la dérivées seconde doit être négative.
23. `vtheta=-1/he;`
Estimation de la variance asymptotique de $\hat{\theta}$.
24. `h=exp(theta);`
On revient au paramètre d'intérêt $\hat{h} = \exp(\hat{\theta})$.
25. `vb=h##2#vtheta;`
Estimation de la variance asymptotique de \hat{h} .

26. `sb=sqrt(vb);`
Estimation de l'écart-type asymptotique de \hat{h} .
27. `tb=h/sb;`
Estimation du t de Student asymptotique de \hat{h} .
28. `pb=1-probchi(tb##2,1);`
Estimation de la probabilité critique.
29. `resp=h||sb||tb||pb;`
Matrice des estimations (un vecteur ligne à ce stade).
30. `me=1/h; dg=-1/h##2; sme=sqrt(dg##2#vb);`
`tme=abs(me)/sme; pme=1-probchi(tme##2,1);`
Estimation de la médiane corrigée des censures. La quantité dg contient la dérivée de la médiane par rapport au paramètre h , on applique donc le théorème de Slutsky.
31. `resp=resp/(me||sme||tme||pme);`
On ajoute la médiane à la matrice des statistiques.
32. `labc="Parameter"||"Std_Dev"||"Student"||"Pvalue";`
Noms de colonne du futur tableau SAS contenant les résultats.
33. `labr="h"//"Median";`
Noms de ligne du futur tableau SAS contenant les résultats.
34. `print "This distribution has NO mean", ,resp`
`[colname=labc rowname=labr];`
On rappelle à l'utilisateur que cette distribution n'a pas de moyenne et on imprime les résultats pertinents.
35. `finish estim;`
Fin du module d'estimation.
36. `start prev;`
`tmin=min(yi); tmax=max(yi); tstep=1;`
`do time=tmin to tmax by tstep;`
`si=1/(1+h#time);`
`hi=h/(1+h#time);`
`fi=si#hi;`
`ci=1-si;`
`resi=resi/(time||fi||si||hi||ci);`
`end;`
Ce module de prévision a la même structure que les modules précédents. On change juste les valeurs des fonctions.
37. `labi="&y"||"Density"||"SDF"||"Hasard"||"CDF";`
Noms des colonnes du futur tableau SAS contenant les prévisions.

38. `finish prev;`
Fin du module de prévision.
39. `start sortie;`
`create &outp from resp[colname=labc];`
`append from resp;`
`create _np from labr[colname="Statistic"];`
`append from labr;`
`create &outi from resi[colname=labi];`
`append from resi;`
`finish sortie;`
Module de sortie, identique aux cas précédents (exponentiel et Gamma à un paramètre).
40. `run lecture;`
`run estim;`
`run prev;`
`run sortie;`
`quit;`
`data &outp; merge _np &outp;`
`run;`
Identique aux programmes précédents.

4.3.5 Application III

On reprend les données de l'application précédente. Pour les représentations graphiques, on ajoutera les valeurs correspondants à $t = 0$:

$$S_T(0) = 1, F_T(0) = 0, h_T(t) = h, f_T(t) = h.$$

Sortie SAS 4.3.

One parameter Lomax distribution

$$S(t)=1/(1+h.t)$$

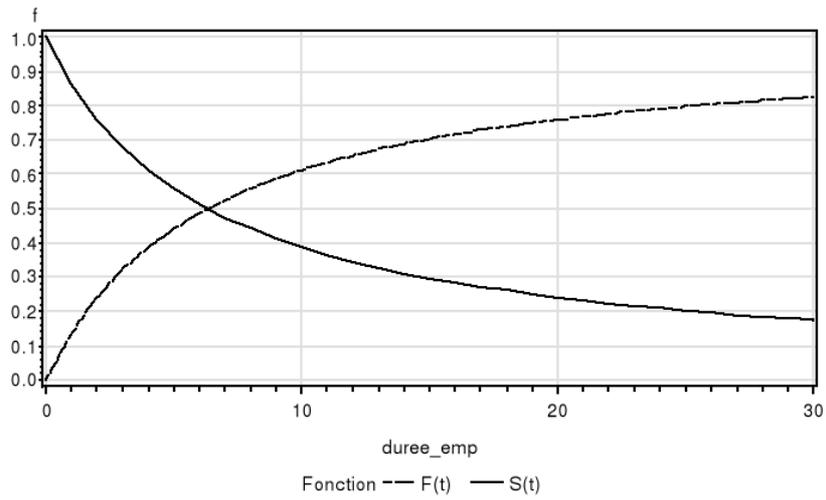
Newton-Raphson Optimization with Line Search

Without Parameter Scaling

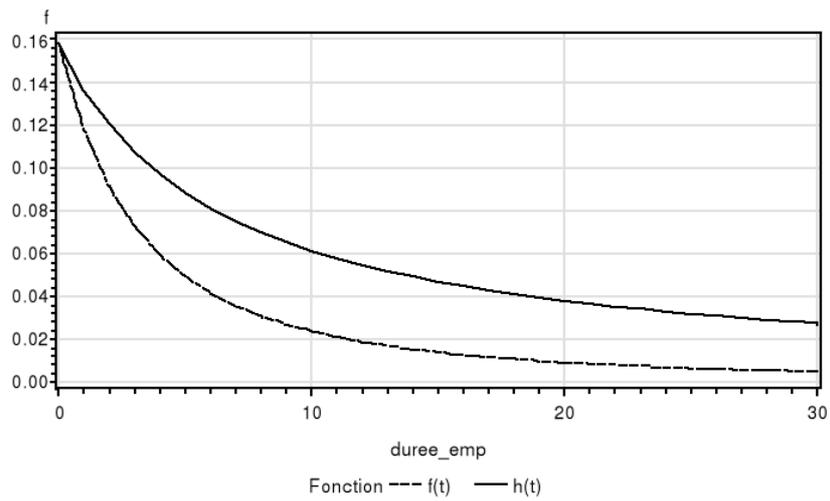
CRP Jacobian Computed by Finite Differences

Parameter Estimates

1



GRAPHIQUE 4.5 – LOI DE LOMAX, $\hat{S}_T(t)$ ET $\hat{F}_T(t)$



GRAPHIQUE 4.6 – LOI DE LOMAX, $\hat{h}_T(t)$ ET $\hat{f}_T(t)$

Optimization Start

Active Constraints 0 Objective Function -69801.39143
 Max Abs Gradient 12093.444823
 Element

| Iter | Rest arts | Func Calls | Act Con | Objective Function | Obj Fun Change | Max Abs Gradient Element | Step Size |
|------|--------------|---------------|------------|---------------------------------|-------------------|--------------------------------|--------------|
| 1 | 0 | 2 | 0 | -47040 | 22761.1 | 8812.7 | 0.100 |
| | | | | Slope of Search Direction | | | |
| | | | | -252968 | | | |

| Iter | Rest arts | Func Calls | Act Con | Objective Function | Obj Fun Change | Max Abs Gradient Element | Step Size |
|------|--------------|---------------|------------|---------------------------------|-------------------|--------------------------------|--------------|
| 2* | 0 | 3 | 0 | -39948 | 7092.6 | 4628.0 | 1.000 |
| | | | | Slope of Search Direction | | | |
| | | | | -25585 | | | |

| Iter | Rest arts | Func Calls | Act Con | Objective Function | Obj Fun Change | Max Abs Gradient Element | Step Size |
|------|--------------|---------------|------------|---------------------------------|-------------------|--------------------------------|--------------|
| 3* | 0 | 4 | 0 | -37720 | 2227.7 | 703.2 | 1.000 |
| | | | | Slope of Search Direction | | | |
| | | | | -5013.2 | | | |

| Iter | Rest arts | Func Calls | Act Con | Objective Function | Obj Fun Change | Max Abs Gradient Element | Step Size |
|------|--------------|---------------|------------|-----------------------|-------------------|--------------------------------|--------------|
| 4* | 0 | 5 | 0 | -37673 | 46.7838 | 1.2688 | 1.000 |
| | | | | Slope of Search | | | |

Direction

-93.471

| Iter | Rest arts | Func Calls | Act Con | Objective Function | Obj Fun Change | Max Abs Gradient Element | Step Size |
|------|--------------|---------------|------------|-----------------------|-------------------|--------------------------------|--------------|
| 5 | 0 | 6 | 0 | -37673 | 0.000153 | 0.000013 | 1.000 |

Slope of
Search
Direction

-0.0003

Optimization Results

| | | | |
|------------------------------|--------------|-----------------------------|------------|
| Iterations | 5 | Function Calls | 7 |
| Hessian Calls | 6 | Active Constraints | 0 |
| Objective Function | -37673.28087 | Max Abs Gradient Element | 0.00001291 |
| Slope of Search Direction | -0.000305788 | Ridge | 0 |

GCONV convergence criterion satisfied.

rc

Result code (must be >0) = 6

lvl

Log-likelihood at maximum = -37673.28
gr

Gradient -0.000013

elas

Elasticity = 6.321E-10
he

Second order derivative = -5264.906

This distribution has NO mean

resp

| | Parameter | Std_Dev | Student | Pvalue |
|----------------|-----------|-----------|-----------|--------|
| Log-likelihood | -37673.28 | . | . | . |
| h | 0.1581174 | 0.0021791 | 72.559669 | 0 |
| Median | 6.3244137 | 0.0871616 | 72.559669 | 0 |

La convergence a lieu après 5 itérations. L'estimation du paramètre de la loi de Lomax donne des résultats proches de la loi exponentielle, bien que la loi de Lomax n'admette pas de moyenne. Sa médiane est estimée à 6,3 ans, la même valeur que pour la loi exponentielle, mais sa log-vraisemblance maximale est plus faible (-37673 contre -36091). Globalement, la loi exponentielle domine les autres distributions à un paramètre dans notre application. Ceci ne signifie pas, toutefois, que la loi exponentielle soit une bonne hypothèse. Nous réexaminerons sa performance quand nous estimerons les distributions à plusieurs paramètres. La fonction de survie montre une décroissance plus faible qu'avec la loi Gamma (graphique 4.5) et la fonction de hasard est décroissante par hypothèse (graphique 4.6). Le modèle de Weibull nous permettra, plus loin, d'examiner si le hasard est constant ou décroissant.

4.4 Loi exponentielle avec état absorbant

4.4.1 Log-vraisemblance

Les logarithmes de la densité et de la survie sont donnés par :

$$\begin{aligned}\ln f_T(y_i) &= \ln(1 - \pi) + \ln h - hy_i \\ \ln S_T(y_i) &= \ln(\pi + (1 - \pi) \exp(-hy_i))\end{aligned}$$

donc la log-vraisemblance s'écrit :

$$\ell(\zeta) = \sum_{i=1}^N d_i (\ln(1 - \pi) + \ln h - hy_i) + (1 - d_i) \ln(\pi + (1 - \pi) \exp(-hy_i)),$$

avec

$$\zeta = \begin{pmatrix} \pi \\ h \end{pmatrix}.$$

Le gradient est égal à :

$$\begin{aligned}\frac{\partial \ell}{\partial \pi}(\zeta) &= \sum_{i=1}^N (1 - d_i) \frac{1 - S_T(y_i)}{S_\pi(y_i)} - \frac{d_i}{1 - \pi}, \\ \frac{\partial \ell}{\partial h}(\zeta) &= \sum_{i=1}^N d_i \left(\frac{1}{h} - y_i \right) - (1 - d_i) (1 - \pi) y_i \frac{S_T(y_i)}{S_\pi(y_i)},\end{aligned}$$

avec

$$S_T(y_i) = \exp(-hy_i) \text{ et } S_\pi(y_i) = \pi + (1 - \pi) S_T(y_i).$$

4.4.2 Estimation

Ici, on doit imposer $\pi \in [0, 1[$ et $h > 0$, on fait donc le changement de paramètres :

$$\pi = \frac{\exp(\theta_1)}{1 + \exp(\theta_1)}, \quad h = \exp(\theta_2), \quad \theta = \begin{pmatrix} \theta_1 \\ \theta_2 \end{pmatrix},$$

ce qui donne la vraisemblance :

$$\begin{aligned} \ell_\zeta(\theta) &= \ell(\zeta(\theta)) \\ &= \sum_{i=1}^N d_i (\theta_2 - \exp(\theta_2) y_i) + (1 - d_i) \ln(\exp(\theta_1) + \exp(-\exp(\theta_2) y_i)) \\ &\quad - \ln(1 + \exp(\theta_1)), \end{aligned}$$

les dérivées permettent d'accélérer l'optimisation. On pourrait dériver directement mais la pratique la plus courante consiste à dériver en chaîne. Ici nos paramètres sont entièrement séparés (π ne dépend que de θ_1 et h ne dépend que de θ_2) donc :

$$\frac{\partial \ell_\zeta}{\partial \theta_1} = \sum_{i=1}^N \frac{\partial \ell_i}{\partial \pi} \frac{\partial \pi}{\partial \theta_1}, \quad \frac{\partial \ell_\zeta}{\partial \theta_2} = \sum_{i=1}^N \frac{\partial \ell_i}{\partial h} \frac{\partial h}{\partial \theta_2},$$

avec :

$$\frac{\partial \pi}{\partial \theta_1} = \pi(1 - \pi), \quad \frac{\partial h}{\partial \theta_2} = h,$$

ce qui donne :

$$\begin{aligned} \frac{\partial \ell_\zeta}{\partial \theta_1} &= \sum_{i=1}^N (1 - d_i) \pi(1 - \pi) \frac{1 - S_T(y_i)}{S_\pi(y_i)} - \pi d_i, \\ \frac{\partial \ell_\zeta}{\partial \theta_2} &= \sum_{i=1}^N d_i (1 - hy_i) - (1 - d_i) (1 - \pi) hy_i \frac{S_T(y_i)}{S_\pi(y_i)}, \end{aligned}$$

Pour retrouver les bons paramètres après l'optimisation, on utilise les transformations :

$$\hat{\pi} = \frac{\exp(\hat{\theta}_1)}{1 + \exp(\hat{\theta}_1)}, \quad \hat{h} = \exp(\hat{\theta}_2),$$

et on utilise les dérivées suivantes pour obtenir la matrice de covariance asymptotique à partir du théorème de Slutsky :

$$\frac{\partial g}{\partial \theta'} = \begin{pmatrix} \frac{\exp(\theta_1)}{(1 + \exp(\theta_1))^2} & 0 \\ 0 & \exp(\theta_2) \end{pmatrix} = \begin{pmatrix} \pi(1 - \pi) & 0 \\ 0 & h \end{pmatrix}.$$

4.4.3 Statistiques utiles

Cette distribution comporte des individus pour lesquels l'espérance de durée est infinie donc on ne calcule pas l'espérance mathématique sur toute la population mais seulement pour les individus qui sortent du processus étudié (en proportion $1 - \pi$). Les formules usuelles de la loi exponentielle s'appliquent pour cette sous population. Pour la médiane sur l'ensemble de la population tout dépend de la valeur de π . Si elle est supérieure à $1/2$ la médiane n'existe pas non plus car plus de la moitié de la population a une durée infinie.

Ici nous adopterons la convention qui consiste à calculer tous les moments sur la sous-population qui sort du processus, et qui admet donc une moyenne et une médiane, comme dans la section sur la loi exponentielle. Il faudra bien le préciser lors de l'impression. Pour les graphiques, on raisonne sur l'ensemble de la population.

On calcule les fonctions usuelles, définies sur l'ensemble de la population, par :

$$\begin{aligned} \hat{f}_\pi(t) &= (1 - \hat{\pi}) \hat{h} \exp(-\hat{h}t), \\ \hat{S}_\pi(t) &= \hat{\pi} + (1 - \hat{\pi}) \exp(-\hat{h}t), \\ \hat{h}_\pi(t) &= \frac{\hat{f}_\pi(t)}{\hat{S}_\pi(t)}, \\ \hat{F}_\pi(t) &= 1 - \hat{S}_\pi(t). \end{aligned}$$

4.4.4 Application IV : Estimation par balayage et avec gradient analytique

Il existe des cas où l'algorithme de Newton-Raphson n'est pas très performant. Dans cet ouvrage, le problème se posera pour les proportions des modèles à état absorbant et pour le paramètre de forme des distributions à trois paramètres. Comme le problème ne vient que d'un seul paramètre, nous effectuerons une estimation par balayage pour le paramètre problématique avant d'utiliser l'algorithme de Newton-Raphson au voisinage de la bonne valeur.

Cette méthode est généralement plus efficace que les méthodes qui modifient la méthode de Newton-Raphson. Plus précisément, soit :

$$\theta = \begin{pmatrix} \theta_1 \\ \theta_2 \end{pmatrix},$$

le paramètre à estimer. Par hypothèse, le paramètre θ_1 ne pose pas de problème d'optimisation, contrairement au paramètre θ_2 . Ceci signifie que si l'on optimise par rapport à θ , l'algorithme de Newton-Raphson ne parvient pas à trouver le maximum de la fonction. Par contre, si on fixe la valeur de θ_2 , l'algorithme fonctionne toujours. La méthode est la suivante :

1. On fait une boucle sur les valeurs du paramètre θ_2 .
 - (a) Il est recommandé de transformer ce paramètre pour qu'il soit défini sur un intervalle de bornes finies. Dans notre exemple $\pi \in [0, 1]$ donc on pourra utiliser une grille de valeur de 0.01 à 0.99 par pas de 0.01. Si le paramètre était défini sur $\theta_2 \in [0, +\infty[$, la grille serait plus difficile à établir. On aurait donc intérêt à transformer le paramètre par la fonction $g(\theta_2) = \theta_2 / (1 + \theta_2) \in [0, 1]$ et à utiliser la même grille que précédemment. Si le paramètre est défini sur $\theta_2 \in]-\infty, +\infty[$ on peut utiliser la transformation $g(\theta_2) = \exp(\theta_2) / (1 + \exp(\theta_2))$.
 - (b) Plus généralement, on a intérêt à utiliser l'intervalle $[0, 1]$ car cela permet d'utiliser le même module de recherche du maximum pour toutes les estimations.
2. On réalise l'optimisation partielle suivante par la méthode de Newton-Raphson :

$$\theta_1^*(\theta_2) = \operatorname{argmax}_{\theta_1} \ell(\theta_1, \theta_2)$$

elle donne l'estimation de θ_1 qui maximise la vraisemblance pour une valeur donnée de θ_2 . Il ne s'agit pas du maximum de vraisemblance (car θ_2 n'est pas dans l'optimisation) mais d'une suite de paramètres dont une des valeurs est proche de l'estimateur du maximum de vraisemblance.

- (a) On aura intérêt à utiliser les dérivées premières pour accélérer l'algorithme d'optimisation car on doit calculer un grand nombre de valeurs de $\theta_1^*(\theta_2)$, de l'ordre d'une centaine.
- (b) On trace un graphique des log-vraisemblances obtenues en fonction de θ_2 :

$$\ell_{\zeta}^*(\theta_2) = \ell_{\zeta}(\theta_1^*(\theta_2), \theta_2),$$

et on choisit θ_2^* tel que :

$$\theta_2^* = \operatorname{argmax}_{\theta_2} \ell_{\zeta}^*(\theta_2),$$

si la grille est suffisamment fine θ_2^* est l'estimateur du maximum de vraisemblance de θ_2 et $\theta_1^*(\theta_2^*)$ est l'estimateur du maximum de vraisemblance de θ_1 :

$$\begin{aligned} \theta^* &= (\theta_1^*(\theta_2^*), \theta_2^*), \\ \hat{\theta} &= \theta^* \text{ si la grille est suffisamment fine} \end{aligned}$$

3. Dans la pratique, on obtiendra l'estimateur du maximum de vraisemblance de la manière suivante : on optimise la log-vraisemblance en θ en prenant θ^* comme valeur initiale. En effet, en ce point, le gradient est quasiment nul, et l'algorithme de Newton-Raphson ne fera qu'une ou deux itérations pour parvenir au maximum global, selon la précision de l'ordinateur.

- (a) On fera donc :

$$\hat{\theta} = \operatorname{argmax}_{\theta} \ell_{\zeta}(\theta) \text{ avec } \theta_0 = \theta^*$$

- (b) Cette méthode permettra de calculer la matrice de covariance simplement, en inversant l'opposé de la matrice hessienne.

Nous aurons donc un programme d'estimation en deux parties : la première partie calcule $\ell_{\zeta}^*(\theta_2)$ sur la grille de valeurs choisies pour θ_2 . La seconde partie fait une optimisation standard en prenant θ^* comme valeur initiale, calcule les estimations et les prévisions. Le programme qui réalise le balayage est le suivant :

- ```
1. %let pi_min=0.01;
 %let pi_max=0.99;
 %let pi_by=0.01;
```

On ajoute trois macro-variables pour définir l'intervalle de balayage et son pas.

- ```
2. %let outb=_outb;
   %let outp=_inp;
```

On ajoute deux macro-variables pour les sorties. La variable `&outp` servira à sauvegarder $(\pi, h^*(\pi), \ell_{\zeta}^*(\pi))$ pour toutes les valeurs de π , alors que `&outb` servira à sauvegarder $(\pi^*, h^*(\pi^*), \ell_{\zeta}^*(\pi^*))$. Cette dernière sortie constituera une ligne de données en entrée pour le programme qui réalise l'estimation finale, c'est la raison pour laquelle le tableau est appelé `_inp` par défaut.

```

3. proc iml;
  start lecture;
  nom_y={&y};
  nom_d={&d};
  use &tab; read all var(nom_y) into yi;
  read all var(nom_d) into di;
  nd=sum(di); n=nrow(yi); nc=n-nd; lyi=log(yi);
  print "Time variable =" nom_y,
  "Censoring variable (=0 if censored) =" nom_d,
  "Number of observations =" n,
  "Complete =" nd,
  "Censored =" nc;
  finish lecture;
  Module de lecture identique aux programmes précédents.

```

```

4. start lv(theta) global(di,yi,pi);
  h=exp(theta);
  lfi=log(1-pi)+theta-h#yi;
  lsi=log(pi+(1-pi)#exp(-h#yi));
  lv=sum(di#(lfi-lsi)+lsi);
  return (lv);
  finish;

```

Fonction définissant la log-vraisemblance par rapport à un seul paramètre $\theta = \ln h$ pour une valeur fixée de π . La valeur de π est lue via l'option *global*. La quantité *lfi* contient $\ln f_{\pi}(y_i)$ et *lsi* contient $\ln S_{\pi}(y_i)$.

```

5. start sc(theta) global(di,yi,pi);
  h=exp(theta);
  st=exp(-h#yi);
  spi=pi+(1-pi)#st;
  der=sum(di#(1-h#yi)-(1-di)#h#yi#(1-pi)#st/spi);
  return (der);
  finish;

```

Fonction calculant la dérivée de la log-vraisemblance par rapport à θ pour une valeur fixée de π . La valeur de π est lue via l'option *global*. La variable *st* contient $S_T(y_i)$ et la variable *spi* contient $S_{\pi}(y_i)$.

```

6. start estim;

```

Début du module d'estimation.

```

7. print "Exponential distribution with absorbing
state", , "S(t)=pi+(1-pi).exp(-h.t)", , ,
"FIRST STEP : select pi and save it for the second

```

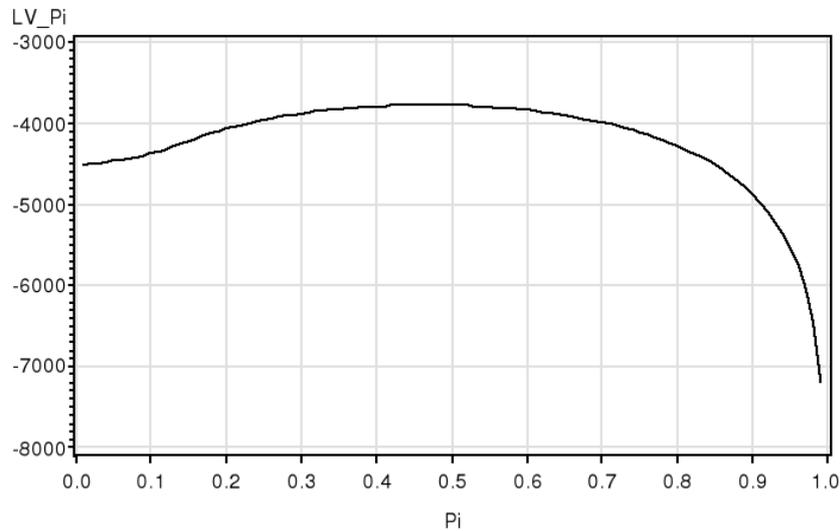
```
step", , i
```

On rappelle la forme de la fonction de survie utilisée, et qu'il s'agit d'un programme qui ne réalise que la première étape de l'estimation.

8. `h0=sum(yi)/nd; theta0=log(h0);`
Valeur initiale, on prend $h_0 = \sum y_i / N_d$ donc $\theta_0 = \ln h_0$.
9. `opt=j(11,1,0); opt[1]=1; opt[2]=0;`
Options de la routine IML `nlpnra`, on réalise une maximisation (première option à 1) et on annule l'affichage (deuxième option à 0).
10. `pimin=&pi_min; pimax=&pi_max; piby=&pi_by;`
`do pi=pimin to pimax by piby;`
Boucle de balayage sur les valeur de π , définie par les macro-variables.
11. `call nlpnra(rc,theta,"lv",theta0,opt) grd="sc";`
On estime la valeur de h qui maximise la log-vraisemblance pour une valeur donnée de π , ce qui donne $h^*(\pi)$. On utilise un gradient analytique pour accélérer l'estimation.
12. `outb=outb/(pi||exp(theta)||lv(theta));`
On empile les résultats de chaque itération sous la forme $(\pi, h^*(\pi), \ell_\zeta^*(\pi))$.
13. `theta0=theta;`
On prend le résultat de l'itération précédente comme valeur initiale de l'itération courante pour accélérer l'algorithme. Donc pour l'itération k on prend $\theta_0^{(k)} = \theta^{*(k-1)}$.
14. `end;`
Fin de la boucle de balayage.
15. `finish estim;`
Fin du module d'estimation.
16. `start sortie;`
Début du module de sortie.
17. `nb="pi"||"h"||"lv_pi";`
Noms de colonne des futures matrices de sortie.
18. `create &outb from outb[colname=nb];`
`append from outb;`
On sauvegarde $(\pi, h^*(\pi), \ell_\zeta^*(\pi))$ dans une matrice dont le nom est définie par `&outb` (`_outb` dans le répertoire de travail, par défaut).
19. `call sort(outb,3,3);`
On trie la matrice par ordre décroissant des log-vraisemblances calculées, pour que le maximum se situe sur la première ligne. Toutes les colonnes sont affectées par ce tri. Pour trier une matrice m par rapport à

sa k -ième colonne par ordre croissant, on utilise la routine IML `call sort(m,k)` avec seulement deux arguments. Pour trier cette même matrice par rapport à sa k -ième colonne par ordre décroissant on indique le numéro de la colonne dans le troisième argument de l'appel : `call sort(m,k,k)`.

20. `outp=outb[1,];`
On sélectionne la première ligne de la matrice triée `outb` et on la range dans le vecteur ligne `outp`; elle contient les paramètres relatifs à la plus grande vraisemblance trouvée lors du balayage $(\pi^*, h^*(\pi^*), \ell_{\zeta}^*(\pi^*))$. Pour sélectionner la ligne i d'une matrice m , on utilise l'instruction `m[i,]` pour sélectionner la colonne j de cette matrice, on utilise l'instruction `m[,j]`.
21. `pim=outp[1];`
La variable `pim` contient π^* , la valeur de π qui maximise la vraisemblance sur l'intervalle de balayage étudié.
22. `if pim=pimin then print "WARNING: Corner solution on the left";`
`if pim=pimax then print "WARNING: Corner solution on the right";`
`print "Maximum likelihood at " outp[colname=nb];`
Si π^* est situé sur les bornes de l'intervalle, on envoie un avertissement à l'utilisateur car il s'agit d'une solution en coin. Dans ce cas il faut refaire un balayage situé avant la borne inférieure de l'intervalle de balayage si le maximum est à gauche, ou un balayage situé après la borne supérieure de l'intervalle de balayage si le maximum est à droite. Si on obtient une solution intérieure, et que l'intervalle de balayage est suffisamment grand, on devrait se trouver près du maximum de vraisemblance global en $\hat{\theta}$.
23. `create &outp from outp[colname=nb];`
`append from outp;`
Sauvegarde du paramètre trouvé par balayage dans le tableau défini par la macro-variable `&outp` (`_outp` dans le répertoire de travail, par défaut).
24. `finish sortie;`
Fin du module de sortie.
25. `run lecture;`
`run estim;`
`run sortie;`
Lancement des modules du programme.

GRAPHIQUE 4.7 – LOI EXPONENTIELLE À ÉTAT ABSORBANT, $\ell^*(\pi)$

26. quit;
run;

27. On quitte IML et on lance le programme.

Pour l'application, on utilise également les données de l'enquête SIP. On étudie ici la durée d'un handicap. Celui-ci peut être permanent ou temporaire. A peu près la moitié de l'échantillon est censurée. En effet, si un handicap est permanent, sa durée sera nécessairement censurée, ce qui explique le taux de censure élevé dans cet échantillon. On trouve que la valeur de π qui maximise la vraisemblance est $\pi^* = 0.47$, ce qu'illustre le graphique 4.7. La sortie du programme est la suivante.

Sortie SAS 4.4.

```

                                nom_y
Time variable = DUREE_HAND
                                nom_d
Censoring variable (=0 if censored) = D_HAND
n

```

```

Number of observations =      2140
                      nd
Complete =              1036
                      nc
Censored =              1104

```

Exponential distribution with absorbing state

$$S(t) = \pi + (1 - \pi) \cdot \exp(-h \cdot t)$$

FIRST STEP : select pi and save it for the second step

```

                      outp
Maximum likelihood at  pi      h      lv_pi
                      0.47 0.2406297 -3767.157

```

| pi | h | lv_pi |
|------|---------|----------|
| 0.01 | 0.03600 | -4511.15 |
| 0.02 | 0.03758 | -4500.38 |
| 0.03 | 0.03937 | -4488.92 |
| 0.04 | 0.04144 | -4476.67 |
| 0.05 | 0.04385 | -4463.49 |
| 0.06 | 0.04674 | -4449.23 |
| 0.07 | 0.05033 | -4433.64 |
| 0.08 | 0.05498 | -4416.38 |
| 0.09 | 0.06142 | -4396.95 |
| 0.10 | 0.07123 | -4374.51 |
| 0.11 | 0.08708 | -4347.84 |
| 0.12 | 0.10707 | -4316.51 |
| 0.13 | 0.12416 | -4282.57 |
| 0.14 | 0.13739 | -4248.25 |
| 0.15 | 0.14795 | -4214.68 |
| 0.16 | 0.15668 | -4182.36 |
| 0.17 | 0.16412 | -4151.49 |
| 0.18 | 0.17059 | -4122.16 |
| 0.19 | 0.17629 | -4094.36 |
| 0.20 | 0.18140 | -4068.07 |
| 0.21 | 0.18600 | -4043.24 |
| 0.22 | 0.19019 | -4019.82 |
| 0.23 | 0.19403 | -3997.74 |
| 0.24 | 0.19757 | -3976.95 |
| 0.25 | 0.20084 | -3957.39 |
| 0.26 | 0.20388 | -3939.02 |
| 0.27 | 0.20672 | -3921.78 |
| 0.28 | 0.20938 | -3905.63 |
| 0.29 | 0.21187 | -3890.53 |
| 0.30 | 0.21422 | -3876.44 |
| 0.31 | 0.21644 | -3863.32 |
| 0.32 | 0.21854 | -3851.14 |

| | | |
|------|---------|----------|
| 0.33 | 0.22053 | -3839.88 |
| 0.34 | 0.22241 | -3829.50 |
| 0.35 | 0.22421 | -3819.98 |
| 0.36 | 0.22592 | -3811.31 |
| 0.37 | 0.22756 | -3803.45 |
| 0.38 | 0.22912 | -3796.40 |
| 0.39 | 0.23061 | -3790.13 |
| 0.40 | 0.23204 | -3784.64 |
| 0.41 | 0.23342 | -3779.92 |
| 0.42 | 0.23474 | -3775.95 |
| 0.43 | 0.23601 | -3772.72 |
| 0.44 | 0.23723 | -3770.23 |
| 0.45 | 0.23840 | -3768.48 |
| 0.46 | 0.23954 | -3767.45 |
| 0.47 | 0.24063 | -3767.16 |
| 0.48 | 0.24169 | -3767.59 |
| 0.49 | 0.24271 | -3768.75 |
| 0.50 | 0.24370 | -3770.65 |
| 0.51 | 0.24465 | -3773.28 |
| 0.52 | 0.24558 | -3776.66 |
| 0.53 | 0.24647 | -3780.79 |
| 0.54 | 0.24734 | -3785.68 |
| 0.55 | 0.24819 | -3791.33 |
| 0.56 | 0.24901 | -3797.78 |
| 0.57 | 0.24981 | -3805.02 |
| 0.58 | 0.25058 | -3813.08 |
| 0.59 | 0.25133 | -3821.97 |
| 0.60 | 0.25206 | -3831.72 |
| 0.61 | 0.25278 | -3842.35 |
| 0.62 | 0.25347 | -3853.88 |
| 0.63 | 0.25415 | -3866.36 |
| 0.64 | 0.25481 | -3879.80 |
| 0.65 | 0.25545 | -3894.26 |
| 0.66 | 0.25608 | -3909.75 |
| 0.67 | 0.25669 | -3926.35 |
| 0.68 | 0.25729 | -3944.08 |
| 0.69 | 0.25787 | -3963.01 |
| 0.70 | 0.25844 | -3983.20 |
| 0.71 | 0.25900 | -4004.72 |
| 0.72 | 0.25954 | -4027.65 |
| 0.73 | 0.26007 | -4052.06 |
| 0.74 | 0.26059 | -4078.06 |
| 0.75 | 0.26110 | -4105.75 |
| 0.76 | 0.26160 | -4135.26 |
| 0.77 | 0.26209 | -4166.71 |
| 0.78 | 0.26257 | -4200.28 |
| 0.79 | 0.26303 | -4236.13 |
| 0.80 | 0.26349 | -4274.47 |
| 0.81 | 0.26394 | -4315.55 |

| | | |
|------|---------|----------|
| 0.82 | 0.26438 | -4359.63 |
| 0.83 | 0.26481 | -4407.05 |
| 0.84 | 0.26524 | -4458.19 |
| 0.85 | 0.26565 | -4513.51 |
| 0.86 | 0.26606 | -4573.57 |
| 0.87 | 0.26646 | -4639.04 |
| 0.88 | 0.26685 | -4710.79 |
| 0.89 | 0.26724 | -4789.87 |
| 0.90 | 0.26762 | -4877.66 |
| 0.91 | 0.26799 | -4975.97 |
| 0.92 | 0.26835 | -5087.26 |
| 0.93 | 0.26871 | -5214.97 |
| 0.94 | 0.26907 | -5364.15 |
| 0.95 | 0.26941 | -5542.61 |
| 0.96 | 0.26975 | -5763.47 |
| 0.97 | 0.27009 | -6051.29 |
| 0.98 | 0.27042 | -6461.23 |
| 0.99 | 0.27074 | -7169.30 |

Une fois une solution intérieure trouvée, nous allons utiliser le tableau SAS &outp comme point de départ d'un algorithme global de maximisation de la vraisemblance. Ici, il faudra juste faire attention à un point, nous avons maintenant un *vecteur* de paramètres et non un paramètre réel, ce qui va nous amener à modifier le module d'estimation sur plusieurs points importants.

```
1. %let inp=_inp;
   %let outp=_outp;
   %let outi=_outi;
```

La première macro-variable indique le nom du tableau qui contient le résultat de l'estimation par balayage. Il s'agit de $(\pi^*, h^*(\pi^*))$.

```
2. proc iml;
   start lecture;
   nom_y={&y}; nom_d={&d};
   use &tab; read all var(nom_y) into yi;
   read all var(nom_d) into di;
   nd=sum(di); n=nrow(yi); nc=n-nd; lyi=log(yi);
   print "Time variable =" nom_y,
        "Censoring variable (=0 if censored) =" nom_d,
        "Number of observations =" n,
        "Complete =" nd,
        "Censored =" nc;
   finish lecture;
```

Le module de lecture standard.

```

3. start cor(x) global(epsilon);
   do i=1 to nrow(x);
   x[i]=min(x[i],1-epsilon);
   end;
   return (x);
   finish;

```

Nous allons réaliser une estimation globale sur le vecteur (π, h) donc ne nous pourrions pas contrôler la valeur de π . La fonction `cor` s'assure que cette valeur n'est jamais égale à 1 car la log-vraisemblance n'est pas définie en ce point. La fonction réalise l'opération suivante :

$$\min(x, 1 - \varepsilon) = \begin{cases} x & \text{si } x \leq 1 - \varepsilon \\ 1 - \varepsilon & \text{sinon} \end{cases}$$

```

4. start lv(theta) global(di, yi, lyi, epsilon);
   Calcul de la log-vraisemblance avec deux paramètres d'optimisation rangés dans  $\theta$ .
5. t1=theta[1]; t2=theta[2];
   On récupère  $\theta_1^{(k)}$  et  $\theta_2^{(k)}$ , où  $k$  est l'itération de l'algorithme.
6. pi=cor(exp(t1)/(1+exp(t1))); h=exp(t2);
   On recalcule les paramètres d'intérêt :

```

$$\pi = \min\left(\frac{\exp(\theta_1)}{1 + \exp(\theta_1)}, 1 - \varepsilon\right) \text{ et } h = \exp(\theta_2).$$

```

7. lfi=log(1-pi)+t2-h#yi;
   Logarithme de la densité.
8. lsi=log(pi+(1-pi)#exp(-h#yi));
   Logarithme de la survie.
9. lv=sum(di#(lfi-lsi)+lsi);
   Log-vraisemblance.
10. return (lv);
   Renvoi du résultat vers l'appel de la fonction.
11. finish;
   Fin de la fonction.
12. start sc(theta) global(di, yi, pi);
   Fonction calculant le gradient, également appelé le score. La routine d'optimisation nlpnra exige que le vecteur des paramètres et celui des dérivées soient des vecteurs-lignes.

```

13. `t1=theta[1]; t2=theta[2];`
On récupère les paramètres.
14. `pi=cor(exp(t1)/(1+exp(t1))); h=exp(t2);`
On calcule les paramètres d'intérêt.
15. `st=exp(-h#yi);`
Survie en l'absence d'état absorbant.
16. `spi=pi+(1-pi)#st;`
Survie en présence d'état absorbant.
17. `der1=sum((1-di)#pi#(1-pi)#(1-st)/spi-di#pi);`
Dérivée par rapport au premier paramètre $\partial \ell_{\zeta} / \partial \theta_1$.
18. `der2=sum(di#(1-h#yi)-(1-di)#(1-pi)#h#yi#st/spi);`
Dérivée par rapport au second paramètre $\partial \ell_{\zeta} / \partial \theta_2$.
19. `der=(der1 || der2);`
On forme un vecteur ligne avec les deux dérivées, en respectant bien l'ordre des paramètres.
20. `return (der);`
On renvoie le score vers l'appel de la fonction.
21. `finish;`
Fin de la fonction de calcul du score.
22. `start estim;`
Début du module d'estimation.
23. `use &inp; read all var {pi h} into dzeta0;`
Lecture du résultat du balayage. Dans l'instruction `var` on entre directement les noms des paramètres, il faut donc utiliser des accolades (et non des parenthèses). On obtient un *vecteur-ligne* qui servira de valeur initiale ζ_0 .
24. `print "Exponential distribution with absorbing state", , "S(t)=pi+(1-pi).exp(-h.t)", ;`
Rappel de la fonction de survie utilisée. Les deux virgules à la suite signifient que l'on saute deux lignes.
25. `epsilon=1e-6;`
On fixe le paramètre ε à 10^{-6} .
26. `theta0=log(dzeta0[1]/(1-dzeta0[1])) || log(dzeta0[2]);`
On calcule la valeur initiale du paramètre qui servira à l'optimisation :

$$\theta_0 = \left(\ln \frac{\pi^*}{1-\pi^*}, \ln h^*(\pi^*) \right).$$

27. `opt=j(11,1,0); opt[1]=1; opt[2]=1;`
Paramètres de la routine *nlpnra* : maximisation et impression minimale.
28. `call nlpnra(rc,theta,"lv",theta0,opt) grd="sc";`
Appel de la routine d'optimisation avec gradient analytique donné par la fonction *sc*.
29. `print "Result code (must be > 0) =" rc;`
Code de résultat, il doit être positif.
30. `call nlpfdd(lv1,gr,he,"lv",theta) grd="sc";`
Appel de la routine de calcul des dérivées, avec gradient analytique donné par la fonction *sc*. En sortie, on obtient la log-vraisemblance maximale (*lv1*), le gradient à l'optimum (*gr*) et le hessien à l'optimum (*he*).
31. `theta=t(theta); gr=t(gr);`
On transpose le paramètre et le gradient, car on travaille habituellement sur des vecteurs-colonnes en économétrie. La transposée de la matrice *m* s'obtient par *t(m)*.
32. `elas=abs(theta#gr/lv1);`
Elasticités de la log-vraisemblance par rapport aux paramètres. On obtient un vecteur colonne donnant les deux élasticités, par rapport à π et par rapport à *h*.
33. `vp=eigval(he);`
La fonction *eigval* donne les valeurs propres d'une matrice qu'elle range dans un vecteur colonne. Elles sont triées par ordre décroissant. Pour que nous soyons au maximum, il faut que toutes les valeurs propres soient négatives (donc que le hessien soit défini négatif). "Eigenvalue" signifie "valeur propre" en anglais, d'où le nom de fonction *eigval*.
34. `print "Log-likelihood at maximum =" lv1,`
`"Gradient" gr, "Elasticities =" elas, "Hessian`
`eigenvalues =" vp;`
`vtheta=inv(-he);`
Estimation de la matrice de covariance asymptotique de $\widehat{\theta}$, $\widehat{\text{Var}}(\widehat{\theta})$.
35. `t1=theta[1]; t2=theta[2];`
On récupère $\widehat{\theta}_1$ et $\widehat{\theta}_2$.
36. `pi=cor(exp(t1)/(1+exp(t1))); h=exp(t2);`
`b=pi/h;`
On calcule $\widehat{\zeta}$ que l'on range en colonne dans le vecteur IML *b* :

$$\widehat{\zeta} = \begin{pmatrix} \widehat{\pi} \\ \widehat{h} \end{pmatrix}$$

37. $dg = (\text{pi} \# (1 - \text{pi}) \mid \mid 0) // (0 \mid \mid h)$;

On applique le théorème de Slutsky à la fonction :

$$\zeta = g(\theta) = \begin{pmatrix} \exp(\theta_1) / (1 + \exp(\theta_1)) \\ \exp(\theta_2) \end{pmatrix}.$$

38. $vb = dg * v\theta * t(dg)$;

Calcul de l'estimation de la variance

$$\widehat{\text{Vas}}(\hat{\zeta}) = \frac{\partial g(\theta)}{\partial \theta'} \widehat{\text{Vas}}(\hat{\theta}) \left(\frac{\partial g(\theta)}{\partial \theta'} \right)'$$

39. $sb = \text{sqrt}(\text{vecdiag}(vb))$;

Calcul des estimations des écarts-types des estimateurs. L'instruction *vecdiag* sélectionne la diagonale d'une matrice et la range dans un vecteur colonne. La diagonale de $\widehat{\text{Vas}}(\hat{\zeta})$ contient les estimations des variances des estimateurs. Ensuite on applique la fonction *sqrt* qui prend la racine carrée ("square root", en anglais) des éléments du vecteur (N.B. : cette fonction ne calcule pas la racine carrée d'une matrice). On obtient donc le vecteur des estimations des écarts-types des estimateurs.

40. $tb = b / sb$;

Calcul des *t* de Student asymptotiques, il suivent asymptotiquement une loi normale $N(0, 1)$ sous l'hypothèse de nullité des coefficients.

41. $pb = 1 - \text{probchi}(tb \# \# 2, 1)$;

Calcul des probabilités critiques correspondantes.

42. $resp = b \mid \mid sb \mid \mid tb \mid \mid pb$;

Stockage des résultats dans la matrice *resp*.

43. $vh = vb[2, 2]$;

Pour calculer les statistiques qui suivront, on a besoin de l'estimation de la variance asymptotique de \hat{h} , elle est située en position (2,2) dans la matrice de covariance asymptotique $\widehat{\text{Vas}}(\hat{\zeta})$.

44. $mt = 1/h$; $vmt = vh/h \# \# 4$; $smt = \text{sqrt}(vmt)$;

$tmt = \text{abs}(mt) / smt$; $pmt = 1 - \text{probchi}(tmt \# \# 2, 1)$;

Calcul de l'estimation de la durée moyenne pour le sous-échantillon des individus dont la durée est finie, et des statistiques associées.

45. $md = \log(2)/h$; $vmd = \log(2) \# \# 2 \# vh/h \# \# 4$; $smd = \text{sqrt}(vmd)$;

$pmd = 1 - \text{probchi}(tmd \# \# 2, 1)$; $tmd = \text{abs}(md) / smd$;

Calcul de l'estimation de la durée médiane pour le sous-échantillon des individus dont la durée est finie, et des statistiques associées.

46. `resp=resp// (mt || smt || tmt || pmt) //
(md || smd || tmd || pmd);`
On ajoute les statistiques à la matrice des résultats, pour une sauvegarde ultérieure.
47. `labc="Parameter" || "Std_Dev" || "Student" || "Pvalue";`
Futur noms de colonne du tableau SAS en sortie.
48. `labr="Pi (T infinite)"//"H (T finite)"//
"Mean (T finite)"//"Median (T finite)";`
Futurs noms de ligne du tableau SAS en sortie.
49. `print "Estimates", ,resp[colname=labc rowname=labr];`
Impression des résultats.
50. `finish estim;`
Fin du module d'estimation.
51. `start prev;`
Début du module de prévision. Sa structure est la même que celle des modules précédents.
52. `tmin=min(yi); tmax=max(yi); tby=1;
do time=tmin to tmax by tby;
st=exp(-h#time); ft=h#exp(-h#time);`
Calcul de $\hat{S}_T(t)$ et de $\hat{f}_T(t)$.
53. `si=pi+(1-pi)#st;`
Calcul de $\hat{S}_\pi(t) = \hat{\pi} + (1 - \hat{\pi}) \hat{S}_T(t)$.
54. `fi=(1-pi)#ft;`
Calcul de $\hat{f}_\pi(t) = (1 - \hat{\pi}) \hat{f}_T(t)$.
55. `hi=fi/si;`
Calcul de $\hat{h}_\pi(t) = \hat{f}_\pi(t) / \hat{S}_\pi(t)$.
56. `ci=1-si;`
Calcul de $\hat{F}_\pi(t) = 1 - \hat{S}_\pi(t)$.
57. `resi=resi// (time || fi || si || hi || ci);`
Stockage des résultats.
58. `end;`
Fin de la boucle.
59. `labi="&y" || "Density" || "SDF" || "Hasard" || "CDF";`
Futurs noms de colonne du tableau SAS contenant les prévisions.
60. `finish prev;`
Fin du module de prévision.

```

61. start sortie;
    create &outp from resp[colname=labc];
    append from resp;
    create _np from labr[colname="Statistic"];
    append from labr;
    create &outi from resi[colname=labi];
    append from resi;
    finish sortie;
    Module de sortie identique aux précédents.

62. run lecture;
    run estim;
    run prev;
    run sortie;
    Lancement des modules.

63. quit;
    On sort de SAS-IML.

64. data &outp; merge _np &outp;
    run;
    Identique aux programmes précédents.

```

Sortie SAS 4.5.

```

                                nom_y
Time variable = DUREE_HAND
                                nom_d
Censoring variable (=0 if censored) = D_HAND
                                n
Number of observations =          2140
                                nd
Complete =          1036
                                nc
Censored =          1104

```

Exponential distribution with absorbing state

$$S(t) = \pi + (1 - \pi) \cdot \exp(-h \cdot t)$$

Newton-Raphson Optimization with Line Search

Without Parameter Scaling
 CRP Jacobian Computed by Finite Differences

```

Parameter Estimates                2
Optimization Start
Active Constraints                0 Objective Function -3767.157388
Max Abs Gradient 1.7304887381
Element

```

| Iter | Rest arts | Func Calls | Act Con | Objective Function | Obj Fun Change | Max Abs Gradient Element | Step Size |
|------|-----------|------------|---------|--------------------|----------------|--------------------------|-----------|
| 1 | 0 | 3 | 0 | -3767 | 0.00331 | 0.000630 | 1.000 |

Slope of
Search
Direction
-0.0066

| Iter | Rest arts | Func Calls | Act Con | Objective Function | Obj Fun Change | Max Abs Gradient Element | Step Size |
|------|-----------|------------|---------|--------------------|----------------|--------------------------|-----------|
| 2 | 0 | 4 | 0 | -3767 | 6.86E-10 | 5.36E-10 | 1.000 |

Slope of
Search
Direction
-14E-10

Optimization Results

```

Iterations                2 Function Calls                5
Hessian Calls            3 Active Constraints            0
Objective Function -3767.154075 Max Abs Gradient      5.36114E-10
Element
Slope of Search -1.408086E-9 Ridge                0
Direction

```

GCONV convergence criterion satisfied.

```

rc
Result code (must be >0) = 6

```

lv1

Log-likelihood at maximum = -3767.154

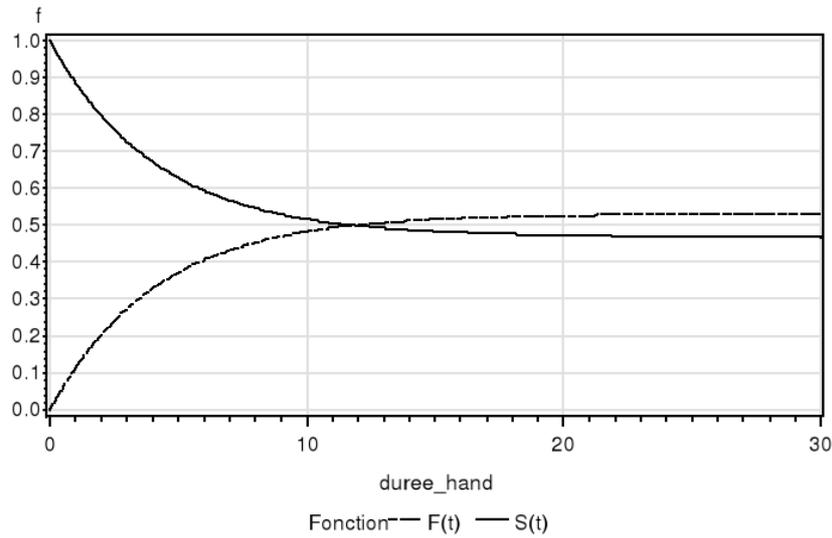
gr

Gradient -2.19E-10

-5.36E-10

elas

Elasticities = 7.211E-15

GRAPHIQUE 4.8 – LOI EXPONENTIELLE À ÉTAT ABSORBANT, $\hat{S}_\pi(t)$ ET $\hat{F}_\pi(t)$

```

2.028E-13
                                vp
Hessian eigenvalues = -440.2535
                                -862.0766

```

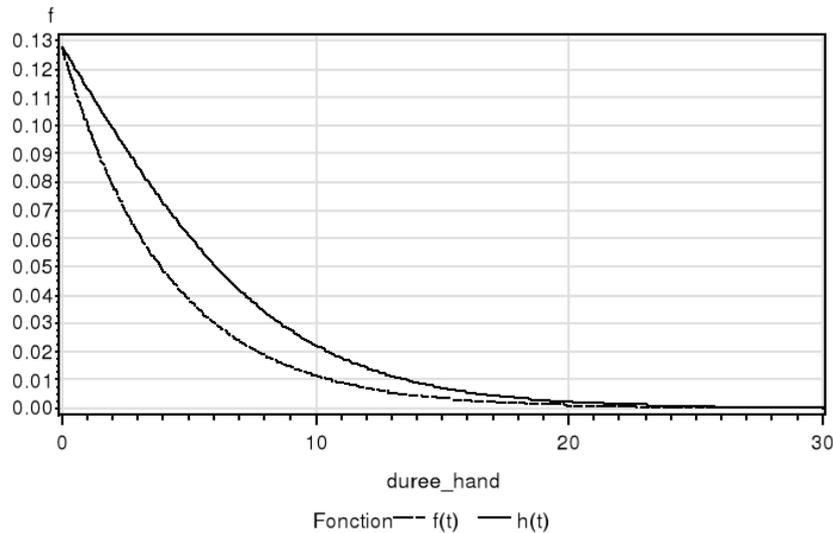
Estimates

| | Parameter | Std_Dev | Student | Pvalue |
|-------------------|-----------|-----------|-----------|--------|
| Pi (T infinite) | 0.4690463 | 0.0117167 | 40.03235 | 0 |
| H (T finite) | 0.240527 | 0.0083943 | 28.653593 | 0 |
| Mean (T finite) | 4.1575373 | 0.1450965 | 28.653593 | 0 |
| Median (T finite) | 2.8817853 | 0.1005733 | 28.653593 | 0 |

L'algorithme de Newton-Raphson converge en deux itérations car la première étape de l'itération donne une valeur très proche du maximum global. On trouve une estimation égale à :

$$\hat{\zeta} = (\hat{\pi}, \hat{h}) = (0.469, 0.241),$$

donc à peu près la moitié des handicaps sont permanents. Pour les handicaps non permanents, la durée médiane du handicap est de 2.88 ans (2 ans et 11 mois), ce qui signifie que la moitié des handicaps non permanents donnent



GRAPHIQUE 4.9 – LOI EXPONENTIELLE À ÉTAT ABSORBANT, $\hat{h}_\pi(t)$ ET $\hat{f}_\pi(t)$

lieu à une guérison dans ces délais. La durée moyenne du handicap est plus élevée, à 4.2 ans.

Le fait que π soit significativement positif implique que la fonction de survie présente un plateau en π , ce que confirme clairement le graphique 4.8. De même, on constate que le hasard, la probabilité conditionnelle de guérison, décroît rapidement avec la durée du handicap, comme le montre le graphique 4.9.

4.5 Loi de Weibull

4.5.1 Log-vraisemblance

Les logarithmes du hasard et de la survie sont donnés par :

$$\begin{aligned}\ln h_T(y_i) &= \ln h + \ln \alpha + (\alpha - 1) \ln y_i \\ \ln S_T(y_i) &= -h y_i^\alpha\end{aligned}$$

les paramètres sont rangés dans un vecteur $\zeta = (\alpha, h)'$ d'où la log-vraisemblance, avec $N_d = \sum_i d_i$:

$$\ell(\zeta) = (\ln h + \ln \alpha) N_d + (\alpha - 1) \sum_{i=1}^N d_i \ln y_i - h \sum_{i=1}^N y_i^\alpha,$$

le score est donné par :¹

$$\begin{aligned} \frac{\partial \ell}{\partial \alpha} &= \frac{N_d}{\alpha} + \sum_{i=1}^N d_i \ln y_i - h \sum_{i=1}^N (\ln y_i) y_i^\alpha \\ \frac{\partial \ell}{\partial h} &= \frac{N_d}{h} - \sum_{i=1}^N y_i^\alpha \end{aligned}$$

et l'opposée de la matrice hessienne, avec $\zeta = (\alpha, h)'$:

$$-\frac{\partial^2 \ell}{\partial \zeta \partial \zeta'}(\zeta) = \begin{bmatrix} N_d/\alpha^2 + h \sum_{i=1}^N (\ln y_i)^2 y_i^\alpha & \sum_{i=1}^N (\ln y_i) y_i^\alpha \\ \sum_{i=1}^N (\ln y_i) y_i^\alpha & N_d/h^2 \end{bmatrix}.$$

L'estimateur du maximum de vraisemblance $\hat{\zeta}$ de ζ est donc défini par les conditions du premier ordre. On obtient donc une première relation :

$$\hat{h} = \frac{N_d}{\sum_{i=1}^N y_i^{\hat{\alpha}}}$$

qui généralise la formule du cas exponentiel (obtenue quand $\hat{\alpha} = 1$). En reportant cette définition dans la seconde équation, on obtient :

$$\frac{N_d}{\hat{\alpha}} + \sum_{i=1}^N d_i \ln y_i - \frac{N_d}{\sum_{i=1}^N y_i^{\hat{\alpha}}} \sum_{i=1}^N (\ln y_i) y_i^{\hat{\alpha}} = 0,$$

et l'on peut résoudre cette équation numériquement en $\hat{\alpha}$. On obtient ensuite \hat{h} . Dans notre application, nous appliquerons directement un algorithme de Newton-Raphson sur le paramètre ζ .

Les conditions du premier ordre sont également intéressantes pour effectuer un test du score d'indépendance temporelle. Le test est défini par :

$$\begin{cases} H_0 : \alpha = 1 \\ H_1 : \alpha \neq 1 \end{cases}$$

1. Rappel :

$$\frac{\partial (y_i^\alpha)}{\partial \alpha} = \frac{\partial}{\partial \alpha} (e^{\alpha \ln y_i}) = (\ln y_i) e^{\alpha \ln y_i} = (\ln y_i) y_i^\alpha$$

Pour cela, il faut considérer la quantité $\partial \ell / \partial \alpha$ sous l'hypothèse $\alpha = 1$. Sous l'hypothèse nulle, cette dérivée doit être égale à 0. On obtient :

$$\begin{aligned}\widehat{s}_0 &= \frac{\partial \ell}{\partial \alpha}(\widehat{h}_0, 1) \\ &= \sum_{i=1}^N d_i + \sum_{i=1}^N d_i \ln y_i - \frac{\sum_{i=1}^N d_i}{\sum_{i=1}^N y_i} \sum_{i=1}^N (\ln y_i) y_i,\end{aligned}$$

où \widehat{h}_0 désigne l'estimateur de h sous H_0 ($N_d / \sum_i y_i$). Si cette dérivée est proche de 0, on se trouve dans le cas d'un modèle exponentiel. La matrice de covariance asymptotique de $\widehat{\zeta}$ sous l'hypothèse nulle est donnée par :

$$\left\{ E \left(- \frac{\partial^2 \ell}{\partial \zeta \partial \zeta'} \right) \right\}^{-1} = \begin{pmatrix} J_{\alpha\alpha} & J_{h\alpha} \\ J_{\alpha h} & J_{hh} \end{pmatrix}^{-1}$$

de sorte que la variance asymptotique de s_0 peut s'estimer par :

$$\widehat{V}(\widehat{s}_0) = \left(\widehat{J}_{\alpha\alpha} - \widehat{J}_{\alpha h} \widehat{J}_{hh}^{-1} \widehat{J}_{h\alpha} \right)^{-1},$$

d'où la statistique de test :

$$S_0 = \frac{\widehat{s}_0^2}{\widehat{V}(\widehat{s}_0)} \xrightarrow[H_0]{L} \chi_1^2;$$

on rejette l'hypothèse nulle au seuil de 5% si cette statistique dépasse 3,84. Mais on peut également faire un test de Student asymptotique, qui possède l'avantage de révéler le sens du rejet :

$$U_0 = \frac{s_0}{\sqrt{\widehat{V}(s_0)}} \xrightarrow[H_0]{L} N(0, 1),$$

si $U_0 > 0$ cela veut dire que $s_0 > 0$ et donc que l'on peut augmenter la vraisemblance en augmentant la valeur de α . Si l'on rejette l'hypothèse nulle, on est donc dans le cas où $\alpha > 1$. Inversement si $U_0 < 0$, cela signifie, en cas de rejet de l'hypothèse nulle, que $\alpha < 1$.

Enfin, on remarque que dans le cas des données complètes, $d_i = 1$ et $y_i = t_i, \forall i$, de sorte que :

$$\begin{aligned}\frac{s_0}{N} &\stackrel{pp}{=} 1 + \frac{1}{N} \sum_{i=1}^N \ln t_i - \frac{1/N \sum_{i=1}^N (\ln t_i) t_i}{1/N \sum_{i=1}^N t_i} \\ &\stackrel{pp}{=} 1 + E(\ln T) - \frac{E(T \ln T)}{E(T)},\end{aligned}$$

or, quand la loi est exponentielle (annexe D.1) :

$$\text{Cov}(T, \ln T) = E(T)$$

donc :

$$\begin{aligned} \frac{s_0}{N} &\stackrel{pp}{=} 1 + E(\ln T) - \frac{\text{Cov}(T, \ln T) + E(\ln T)E(T)}{E(T)} \\ &\stackrel{pp}{=} 1 - \frac{\text{Cov}(T, \ln T)}{E(T)} \\ &\stackrel{pp}{=} 0 \end{aligned}$$

4.5.2 Estimation

Pour l'estimation proprement dite, on effectue des changements de paramètres :

$$\alpha = \exp(\theta_1), \quad h = \exp(\theta_2), \quad \theta = \begin{pmatrix} \theta_1 \\ \theta_2 \end{pmatrix},$$

on remarque que :

$$\frac{\partial \alpha}{\partial \theta_1} = \alpha, \quad \frac{\partial^2 \alpha}{\partial \theta_1^2} = \alpha, \quad \frac{\partial h}{\partial \theta_2} = h, \quad \frac{\partial^2 h}{\partial \theta_2^2} = h.$$

et le vecteur du score devient :

$$\begin{aligned} \frac{\partial \ell_\zeta}{\partial \theta_1} &= \sum_{i=1}^N \frac{\partial \ell}{\partial \alpha} \frac{\partial \alpha}{\partial \theta_1} \\ &= N_d + \alpha \sum_{i=1}^N (d_i - h y_i^\alpha) \ln y_i \\ \frac{\partial \ell_\zeta}{\partial \theta_2} &= \sum_{i=1}^N \frac{\partial \ell}{\partial \alpha} \frac{\partial \alpha}{\partial \theta_2} \\ &= N_d - h \sum_{i=1}^N y_i^\alpha \end{aligned}$$

On estime la matrice de covariance de $\hat{\theta}$ par la relation :

$$\text{Vas}(\hat{\theta}) = -\frac{\partial^2 \ell_\zeta}{\partial \theta \partial \theta'}(\hat{\theta}),$$

puis :

$$\hat{\alpha} = \exp(\hat{\theta}_1), \quad \hat{h} = \exp(\hat{\theta}_2),$$

et le théorème de Slutsky :

$$g(\theta) = (\exp(\theta_1), \exp(\theta_2)), \quad \frac{\partial g}{\partial \theta}(\theta) = \begin{pmatrix} \exp(\theta_1) & 0 \\ 0 & \exp(\theta_2) \end{pmatrix}$$

$$\text{Vas}(\hat{\zeta}) = \frac{\partial g}{\partial \theta'}(\theta) \text{Vas}(\hat{\theta}) \frac{\partial g}{\partial \theta'}(\theta)'$$

Pour la matrice des dérivées secondes avec changement de paramètre, on utilise :

$$\begin{aligned} \frac{\partial^2 \ell_{\zeta}}{\partial \theta_1^2} &= \frac{\partial}{\partial \theta_1} \sum_{i=1}^N \frac{\partial \ell}{\partial \alpha} \frac{\partial \alpha}{\partial \theta_1} \\ &= \sum_{i=1}^N \frac{\partial^2 \ell}{\partial \alpha^2} \left(\frac{\partial \alpha^2}{\partial \theta_1} \right)^2 + \frac{\partial \ell}{\partial \alpha} \frac{\partial^2 \alpha}{\partial \theta_1^2} \\ &= \alpha \sum_{i=1}^N (d_i - h y_i^\alpha (\alpha \ln y_i + 1)) \ln y_i \end{aligned}$$

$$\begin{aligned} \frac{\partial^2 \ell_{\zeta}}{\partial \theta_1 \partial \theta_2} &= \frac{\partial}{\partial \theta_2} \sum_{i=1}^N \frac{\partial \ell}{\partial \alpha} \frac{\partial \alpha}{\partial \theta_1} \\ &= \sum_{i=1}^N \frac{\partial^2 \ell}{\partial \alpha \partial h} \frac{\partial h}{\partial \theta_2} \frac{\partial \alpha}{\partial \theta_1} + \frac{\partial \ell}{\partial \alpha} \underbrace{\frac{\partial^2 \alpha}{\partial \theta_1 \partial \theta_2}}_0 \\ &= h \alpha \sum_{i=1}^N \frac{\partial^2 \ell}{\partial \alpha \partial h} \\ &= -h \alpha \sum_{i=1}^N (\ln y_i) y_i^\alpha \end{aligned}$$

$$\begin{aligned} \frac{\partial^2 \ell_{\zeta}}{\partial \theta_2^2} &= \sum_{i=1}^N \frac{\partial^2 \ell}{\partial h^2} \left(\frac{\partial h}{\partial \theta_2} \right)^2 + \frac{\partial \ell}{\partial h} \frac{\partial^2 h}{\partial \theta_2^2} \\ &= -h \sum_{i=1}^N y_i^\alpha \end{aligned}$$

4.5.3 Statistiques utiles

Dans les sorties, on ajoute les estimations de la moyenne et de la médiane. Pour la moyenne, on a :

$$\hat{m} = \frac{\Gamma(1 + \hat{\alpha})}{\hat{h}^{1/\hat{\alpha}}},$$

on applique le théorème de Slutsky avec :

$$g(\theta) = \frac{\Gamma(1 + \alpha)}{\widehat{h}^{1/\alpha}} = \exp\left(-\frac{1}{\alpha} \ln h\right) \Gamma(1 + \alpha)$$

donc :

$$\frac{\partial g}{\partial \alpha} = h^{-1/\alpha} \left(\frac{\ln h}{\alpha^2} \Gamma(1 + \alpha) + \Gamma'(1 + \alpha) \right),$$

dans la pratique on utilise la fonction digamma (voir annexe, p. A.4.3) :

$$\psi^{(0)}(x) \triangleq \frac{\partial \ln \Gamma(x)}{\partial x} = \frac{\Gamma'(x)}{\Gamma(x)} \Leftrightarrow \Gamma'(x) = \psi^{(0)}(x) \Gamma(x),$$

ce qui implique :

$$\begin{aligned} \frac{\partial g}{\partial \alpha} &= h^{-1/\alpha} \left(\frac{\ln h}{\alpha^2} \Gamma(1 + \alpha) + \psi^{(0)}(1 + \alpha) \Gamma(1 + \alpha) \right) \\ &= h^{-1/\alpha} \Gamma(1 + \alpha) \left(\frac{\ln h}{\alpha^2} + \psi^{(0)}(1 + \alpha) \right) \end{aligned}$$

et d'autre part :

$$\frac{\partial g}{\partial h} = -\frac{1}{\alpha} h^{-1/\alpha - 1} \Gamma(1 + \alpha),$$

en posant :

$$\frac{\partial g}{\partial \theta'} = \left(\frac{\partial g}{\partial \alpha}, \frac{\partial g}{\partial h} \right),$$

on peut estimer la variance de l'estimateur de la moyenne par :

$$\text{Vas}(\widehat{m}) = \frac{\partial g}{\partial \theta'}(\widehat{\theta}) \text{Vas}(\widehat{\theta}) \frac{\partial g}{\partial \theta}(\widehat{\theta})'.$$

La médiane s'estime plus simplement par la formule suivante (il y a bien deux logarithmes appliqués successivement dans la formule qui suit) :

$$\widehat{\text{Me}} = g(\theta) = \left(\frac{\ln 2}{h} \right)^{1/\alpha} = \exp\left(\frac{1}{\alpha} (\ln(\ln 2) - \ln h)\right)$$

donc :

$$\begin{aligned} \frac{\partial g}{\partial \alpha}(\theta) &= -\frac{1}{\alpha^2} \times \ln\left(\frac{\ln 2}{h}\right) \times \left(\frac{\ln 2}{h}\right)^{1/\alpha} \\ \frac{\partial g}{\partial h}(\theta) &= -\frac{1}{\alpha h} \left(\frac{\ln 2}{h}\right)^{1/\alpha}, \end{aligned}$$

et on applique la formule de Slutsky de la même manière que précédemment. Pour estimer toutes les fonctions qui définissent la distribution, il faut choisir une plage de valeurs de t , par exemple :

$$t = 1, 2, \dots, t^+,$$

on estime alors les fonctions en tous les points définis par la grille choisie par l'utilisateur :

$$\begin{aligned}\hat{h}_T(t) &= \hat{h}\hat{\alpha}t^{\hat{\alpha}-1} \\ \hat{S}_T(t) &= \exp(-\hat{h}t^{\hat{\alpha}}), \\ \hat{f}_T(t) &= \hat{h}_T(t)\hat{S}_T(t) \\ \hat{F}_T(t) &= 1 - \hat{S}_T(t).\end{aligned}$$

4.5.4 Programmation V : Optimisation avec hessien analytique

Nous réalisons cette estimation avec des dérivées premières et secondes analytiques. Ceci implique de modifier l'appel de la routine *nlpnra* et de supprimer l'utilisation de la routine *nlpfdd* désormais inutile.

1. `%let tab=duree; %let y=dureejour; %let d=d1;`
`%let outp=_outp; %let outi=_outi;`
 Macro-variables identiques aux programmes précédents.
2. `proc iml;`
`start lecture;`
`nom_y={&y}; nom_d={&d};`
`use &tab; read all var(nom_y) into yi;`
`read all var(nom_d) into di;`
`nd=sum(di); n=nrow(yi); nc=n-nd;`
`print "Time variable =" nom_y,`
`"Censoring variable (=0 if censored) =" nom_d,`
`"Number of observations =" n,`
`"Complete =" nd,`
`"Censored =" nc;`
`finish lecture;`
 Module de lecture identique aux modules précédents.
3. `start lv(theta) global(di,yi,lyi,nd);`
`t1=theta[1]; t2=theta[2]; alpha=exp(t1);`
`h=exp(t2);`
 Lecture des paramètres d'optimisation $\theta_1^{(k)}$ et $\theta_2^{(k)}$ pour la k -ième itération et calcul des paramètres d'intérêt.

4. `lv=nd*(t1+t2)+sum((alpha-1)*di*lyi-h*yi**alpha);`
Calcul de la log-vraisemblance.
5. `return (lv);`
Renvoi de la log-vraisemblance vers l'appel de la fonction.
6. `finish;`
Fin de la fonction qui calcule la log-vraisemblance.
7. `start sc(theta) global(di,yi,lyi,nd);`
Fonction calculant le score.
8. `t1=theta[1]; t2=theta[2]; alpha=exp(t1);`
`h=exp(t2);`
`yia=yi**alpha;`
On met $[y_i^\alpha]$ dans un vecteur `yia` car cette quantité est appelée plusieurs fois par la suite; ceci permet de gagner du temps de calcul.
9. `der1=nd+alpha*sum(lyi*(di-h*yia));`
`der2=nd-h*sum(yia);`
`der=(der1 || der2);`
On range les dérivées $\partial \ell_\zeta / \partial \theta_1$ et $\partial \ell_\zeta / \partial \theta_2$ dans un vecteur ligne car c'est sous ce format qu'il faut envoyer les dérivées à la routine `nlpnra`.
10. `return (der);`
Renvoi du vecteur des dérivées vers l'appel de la fonction.
11. `finish;`
Fin de la fonction (vectorielle) qui calcule le score.
12. `start hes(theta) global(di,yi,lyi);`
Module de calcul de la matrice hessienne.
13. `t1=theta[1]; t2=theta[2]; alpha=exp(t1);`
`h=exp(t2);`
`yia=yi**alpha;`
`he11=alpha*sum(lyi*(di-(alpha*lyi+1)*h*yia));`
`he12=-alpha*h*sum(lyi*yia);`
`he22=-h*sum(yia);`
`he=(he11 || he12)/(he12 || he22);`
On range les dérivées dans une matrice; il s'agit de la forme standard.
14. `return (he);`
Renvoi du hessien vers l'appel de la fonction.
15. `finish;`
Fin de la fonction (matricielle) qui calcule le hessien.

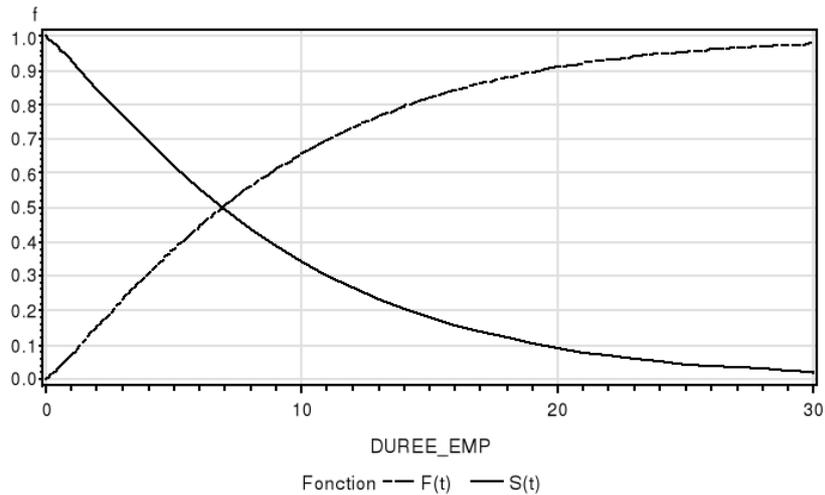
16. `start estim;`
Début du module d'estimation.
17. `print "Weibull distribution", ,`
`"S(t)=exp(-h.t^alpha)", , ;`
`lyi=log(yi);`
On range $[\ln y_i]$ dans le vecteur `lyi` pour ne plus avoir à calculer ces logarithmes par la suite ; ceci permet d'accélérer le programme.
18. `h0=nd/sum(yi); alpha0=1;`
Valeur initiale : on prend les valeurs du modèle exponentiel, puisqu'il est situé sur un point intérieur de l'ensemble des paramètres.
19. `theta0=log(alpha0 | h0);`
Paramètre d'optimisation $\theta = (\ln \alpha, \ln h)$, on doit utiliser un vecteur ligne pour la routine `nlpnra`.
20. `opt=j(11,1,0); opt[1]=1; opt[2]=1;`
Paramètres de la procédure d'optimisation : on maximise (`opt[1]=1`) et on demande un affichage minimal (`opt[2]=1`).
21. `call nlpnra(rc,theta,"lv",theta0,opt) grd="sc"`
`hes="hes";`
On lance la routine d'optimisation par la méthode de Newton-Raphson avec les options `grd=` et `hes=` qui indiquent respectivement les noms des fonctions qui calculent le gradient et le hessien de la log-vraisemblance.
22. `print "Result code (must be > 0) =" rc;`
Code de résultat de l'optimisation ; il doit être positif.
23. `lv1=lv(theta); gr=t(sc(theta)); he=hes(theta);`
Calcul de la log-vraisemblance maximale, du gradient à l'optimum et du hessien à l'optimum. Cette méthode directe est plus rapide que l'utilisation de la routine `nlpfdd`. On exprime le gradient en colonne, car c'est plus pratique pour l'affichage.
24. `theta=t(theta); elas=abs(theta#gr/lv1);`
`vp=eigval(he);`
On remet le vecteur des paramètres en colonne, puis on calcule les élasticités de la log-vraisemblance par rapport aux paramètres d'optimisation et les valeurs propres du hessien.
25. `print "Log-likelihood at maximum =" lv1,`
`"Gradient" gr,`
`"Elasticities =" elas,`
`"Hessian eigenvalues =" vp;`

Affichage du résultat de l'optimisation. Le gradient (donc le vecteur des élasticités) doit être nul et les valeurs propres du hessien doivent toutes être négatives pour que l'on soit à un maximum.

26. $v_{\theta} = \text{inv}(-h_e)$;
Estimation de la matrice de covariance asymptotique de l'estimateur du paramètre d'optimisation $\widehat{\text{Var}}(\hat{\theta})$.
27. $b = \exp(\theta)$;
On calcule l'estimation du paramètre d'intérêt. La fonction exponentielle est appliquée ligne à ligne (N.B. : la fonction `exp` ne calcule pas l'exponentielle d'une matrice). Le vecteur b correspond au paramètre ζ .
28. $\alpha = b[1]$; $h = b[2]$;
On extrait les paramètres d'intérêt pour pouvoir appliquer les formules plus facilement.
29. $dg = (\alpha \mid 0) / (0 \mid h)$;
Matrice des dérivées de $\zeta = g(\theta)$ pour appliquer le théorème de Slutsky. La matrice dg contient $\partial g(\theta) / \partial \theta'$.
30. $vb = dg * v_{\theta} * t(dg)$;
Estimation de la matrice de covariance asymptotique de l'estimateur du paramètre d'intérêt $\widehat{\text{Var}}(\hat{\zeta})$.
31. $sb = \text{sqrt}(\text{vecdiag}(vb))$;
Estimation des écarts-types asymptotiques des estimateurs du paramètre d'intérêt.
32. $tb = b / sb$;
Estimation des t de Student asymptotiques.
33. $pb = 1 - \text{probchi}(tb \# 2, 1)$;
Estimation des probabilités critiques correspondantes.
34. $resp = b \mid sb \mid tb \mid pb$;
On stocke les premiers résultats dans la matrice `resp`.
35. $mt = \text{gamma}(1 + \alpha) \# h \# (-1 / \alpha)$;
Estimation de durée moyenne.
36. $dmt = mt \# (\text{digamma}(1 + \alpha) + \log(h) / \alpha \# 2)$
 $\mid mt \# (-1 / \alpha / h)$;
Dérivée de la durée moyenne par rapport aux paramètres d'intérêt.
37. $vmt = dmt * vb * t(dmt)$; $smt = \text{sqrt}(vmt)$; $tmt = mt / smt$;
 $pmt = 1 - \text{probchi}(tmt \# 2, 1)$;
Application du théorème de Slutsky et calcul des statistiques associées à l'estimateur de la moyenne.

38. `md=(log(2)/h)##(1/alpha);`
Estimation de la durée médiane.
39. `dmd=-md#(log(log(2)/h)/alpha##2) ||`
`(-md/h/alpha);`
Dérivée de la durée médiane par rapport aux paramètres d'intérêt.
40. `vmd=dmd*vb*t(dmd); smd=sqrt(vmd); tmd=md/smd;`
`pmd=1-probchi(tmd##2,1);`
Application du théorème de Slutsky et calcul des statistiques associées à l'estimateur de la médiane.
41. `resp=(lv1||.||. ||.||.)//resp//(md||smd||tmd||pmd)//`
`(mt||smt||tmt||pmt);`
Stockage des résultats. On met la log-vraisemblance en première ligne, avec des valeurs manquantes pour remplir la matrice.
42. `labc="Parameter"||"Std_Dev"||"Student"||"P_Value";`
Futurs noms des colonnes du tableau SAS de sortie.
43. `labr="Log-likelihood"//"alpha"//"h"//"Median"//`
`"Mean";`
Futurs noms des lignes du tableau SAS en sortie.
44. `print resp[colname=labc rowname=labr];`
Affichage des résultats.
45. `test=abs(alpha-1)/sb[1];`
`ptest=1-probchi(test##2,1);`
Statistique du test de Student examinant si les estimations sont compatibles avec la loi exponentielle (indépendance temporelle). On calcule également la probabilité critique.
46. `print "Test H0 : alpha=1 (exponential`
`distribution)", "Student =" test,`
`"p value =" ptest;`
Affichage des résultats du test.
47. `sy=sum(yi); syly=sum(yi#lyi);`
`s0=nd+sum(di#lyi)-nd#syly/sy;`
`vs0=nd#(1+sum(yi#lyi##2)/sy-(syly/sy)##2);`
`score=s0##2/vs0;`
`pscore=1-probchi(score,1);`
`print "Score test of the exponential model",`
`"Stat =" score,"p-value =" pscore;`
Réalisation du test du score d'indépendance temporelle.

48. `finish estim;`
Fin du module d'estimation.
49. `start prev;`
Module de prévision. La structure est identique à celle des modules précédents, on adapte juste la définition des fonctions.
50. `tmin=min(yi); tmax=max(yi); tby=1;`
`do time=tmin to tmax by tby;`
`st=exp(-h#time##alpha);`
`ht=h#alpha#time##(alpha-1);`
`ct=1-st;`
`ft=ht#st;`
`resi=resi/(time||st||ht||ct||ft);`
`end;`
`labi=nom_y||"SDF"||"Hasard"||"CDF"||"Density";`
`finish prev;`
Fin du module de prévision.
51. `start sortie;`
`create &outp from resp[colname=labc];`
`append from resp;`
`create _np from labr[colname="Estimate"];`
`append from labr;`
`create &outi from resi[colname=labi];`
`append from resi;`
`finish sortie;`
Module de sortie, identique aux modèles précédents.
52. `run lecture;`
`run estim;`
`run prev;`
`run sortie;`
`quit;`
Sortie de SAS-IML
53. `data &outp; merge _np &outp;`
`run;`
On ajoute les noms de lignes dans la matrice des résultats et on lance le programme.

GRAPHIQUE 4.10 – LOI DE WEIBULL, $\hat{S}_T(t)$ ET $\hat{F}_T(t)$

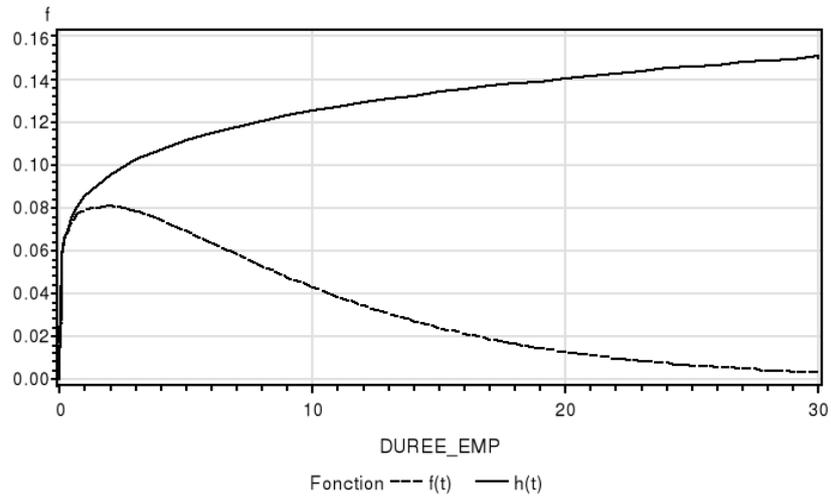
4.5.5 Application V

On reprend l'application sur la durée du premier emploi. Pour les graphiques nous avons ajouté les valeurs des fonctions pour des points (non observables) compris entre 0 et 1, afin d'éviter une représentation coudée en début de graphe. La densité présente un maximum et le hasard est strictement croissant. Ce résultat semble crédible à première vue car nous étudions la durée du premier emploi, c'est-à-dire du premier contrat de travail de toute une carrière. Ce contrat peut être interrompu par de nombreux éléments de la vie et la probabilité de rester dans la même entreprise devrait donc décroître avec le temps. Toutefois, une autre représentation est, en théorie possible, avec un maximum. Dans ce dernier cas, il existerait une durée au delà de laquelle la probabilité de quitter l'entreprise décroît. Le modèle de Weibull ne permet pas de tester cette hypothèse, il faut donc rester prudent à ce stade de l'analyse.

Sortie SAS 4.6.

```
Weibull distribution
```

```
S(t)=exp(-h.t^alpha)
```



GRAPHIQUE 4.11 – LOI DE WEIBULL, $\hat{h}_T(t)$ ET $\hat{f}_T(t)$

Newton-Raphson Optimization with Line Search

Without Parameter Scaling

Parameter Estimates 2

Optimization Start

Active Constraints 0 Objective Function -36091.59799
 Max Abs Gradient 2941.4035624
 Element

| Iter | Rest arts | Func Calls | Act Con | Objective Function | Obj Fun Change | Max Abs Gradient Element | Step Size |
|------|-----------|------------|---------|--------------------|----------------|--------------------------|-----------|
| 1 | 0 | 2 | 0 | -36041 | 50.1460 | 2627.7 | 0.100 |

Slope of Search Direction

-528.2

| Rest | Func | Act | Objective | Obj Fun | Max Abs Gradient | Step |
|------|------|-----|-----------|---------|------------------|------|
|------|------|-----|-----------|---------|------------------|------|

| Iter | arts | Calls | Con | Function | Change | Element | Size |
|------|------|-------|-----|----------|--------|---------|-------|
| 2* | 0 | 3 | 0 | -35936 | 105.6 | 1678.0 | 0.305 |

Slope of
Search
Direction

-411.3

| Iter | Rest arts | Func Calls | Act Con | Objective Function | Obj Fun Change | Max Abs Gradient Element | Step Size |
|------|--------------|---------------|------------|-----------------------|-------------------|--------------------------------|--------------|
| 3* | 0 | 4 | 0 | -35855 | 80.4258 | 729.6 | 1.000 |

Slope of
Search
Direction

-173.4

| Iter | Rest arts | Func Calls | Act Con | Objective Function | Obj Fun Change | Max Abs Gradient Element | Step Size |
|------|--------------|---------------|------------|-----------------------|-------------------|--------------------------------|--------------|
| 4* | 0 | 5 | 0 | -35853 | 2.1405 | 13.8117 | 1.000 |

Slope of
Search
Direction

-4.225

| Iter | Rest arts | Func Calls | Act Con | Objective Function | Obj Fun Change | Max Abs Gradient Element | Step Size |
|------|--------------|---------------|------------|-----------------------|-------------------|--------------------------------|--------------|
| 5 | 0 | 6 | 0 | -35853 | 0.00114 | 0.0103 | 1.000 |

Slope of
Search
Direction

-0.0023

| Iter | Rest arts | Func Calls | Act Con | Objective Function | Obj Fun Change | Max Abs Gradient Element | Step Size |
|------|--------------|---------------|------------|-----------------------|-------------------|--------------------------------|--------------|
|------|--------------|---------------|------------|-----------------------|-------------------|--------------------------------|--------------|

6 0 8 0 -35853 6.04E-10 0.000164 1.016

Slope of
Search
Direction

-12E-10

Optimization Results

| | | | |
|------------------------------|--------------|--------------------|--------------|
| Iterations | 6 | Function Calls | 9 |
| Hessian Calls | 7 | Active Constraints | 0 |
| Objective Function | -35853.2851 | Max Abs Gradient | 0.0001639734 |
| | | Element | |
| Slope of Search Direction | -1.215475E-9 | Ridge | 0 |

GCONV convergence criterion satisfied.

rc

Result code (must be >0) = 6

lvl

Log-likelihood at maximum = -35853.29
gr

Gradient 0.000164
0.0000317
elas

Elasticities = 7.069E-10
2.314E-9

vp
Hessian eigenvalues = -1877.316
-131616.2

| | Parameter | Std_Dev | Student | P_Value |
|----------------|-----------|-----------|-----------|---------|
| Log-likelihood | -35853.29 | . | . | . |
| alpha | 1.1671422 | 0.0078849 | 148.02209 | 0 |
| h | 0.0729647 | 0.0016228 | 44.963321 | 0 |
| Median | 6.8817354 | 0.0626982 | 109.75976 | 0 |
| Mean | 10.198811 | 0.1041027 | 97.96872 | 0 |

Test H0 : alpha=1 (exponential distribution)
test

```

Student = 21.197705
          ptest
p value =          0

Score test of the exponential model
          score

Stat = 447.78161
      pscore

p-value =          0

```

L'algorithme converge en 6 itérations et fournit les estimations :

$$\hat{h} = 0.073, \hat{\alpha} = 1.17,$$

on effectue alors un test $\alpha = 1$ pour savoir si l'on peut admettre la loi exponentielle. La réponse est claire : non. Le t de Student asymptotique est égal à 21.2, ce qui excède largement la valeur critique pour un test au seuil de 5% (1.96). On rejette donc la loi exponentielle au seuil de 5%. Ce résultat est étroitement relié à la forme de la fonction de hasard (graphique 4.11) : elle est croissante avec la durée passée dans le premier emploi. Les paramètres étant différents de ceux d'une loi exponentielle, on doit réévaluer la durée moyenne (10.2 ans) et la durée médiane (6.9 ans), qui sont plus élevées que sous l'hypothèse de loi exponentielle.

Nous aurions également pu effectuer un test du rapport des maxima de vraisemblance pour tester la validité de la loi exponentielle. La statistique est donnée par :

$$LR = -2(\ell(\hat{\theta}_0) - \ell(\hat{\theta})),$$

où $\ell(\hat{\theta}_0)$ est la log-vraisemblance de la loi exponentielle et $\ell(\hat{\theta})$ celle du modèle de Weibull. Sous l'hypothèse nulle $\alpha = 1$, elle suit asymptotiquement un χ_1^2 car la contrainte ne porte que sur un paramètre. Ici, on trouve :

$$LR = -2 \times (-36091.6 + 35853.3) = 476.6,$$

ce qui excède largement la valeur critique du test de niveau 95% (ici, 3.84). On préférera donc la loi de Weibull à la loi exponentielle.

Il reste à effectuer le test du score. En toute rigueur ce test aurait dû être associé à la loi exponentielle, car il ne nécessite que des quantités disponibles sous l'hypothèse nulle $\alpha = 1$. On trouve une statistique de 447.8 qui dépasse largement la valeur critique du χ_1^2 au seuil de 5%.

4.6 Loi Gamma à deux paramètres

Dans cette section, nous allons voir comment optimiser sans routine d'optimisation. Ceci peut s'avérer utile quand on veut modifier la routine standard. Ici, nous traiterons le cas de base. Nous allons donc entrer dans les détails de la routine de Newton-Raphson. Toutefois, toutes les dérivées seront calculées numériquement grâce à la routine *nlpfdd*.

4.6.1 Log-vraisemblance

Les logarithmes de la densité et de la survie sont donnés par :

$$\begin{aligned}\ln f_T(y_i) &= \beta \ln h + (\beta - 1) \ln y_i - h y_i - \ln \Gamma(\beta) \\ \ln S_T(y_i) &= \ln(1 - \gamma(\beta, h y_i)),\end{aligned}$$

le paramètre à estimer est :

$$\zeta = \begin{pmatrix} \beta \\ h \end{pmatrix},$$

d'où la log-vraisemblance :

$$\ell(\zeta) = \sum_{i=1}^N d_i (\beta \ln h + (\beta - 1) \ln y_i - h y_i - \ln \Gamma(\beta)) + (1 - d_i) \ln(1 - \gamma(\beta, h y_i))$$

4.6.2 Estimation

Dans la pratique, on effectue le changement de paramètre :

$$\beta = \exp(\theta_1), \quad h = \exp(\theta_2)$$

et l'on optimise la quantité suivante par rapport au paramètre θ :

$$\ell_{\zeta}(\theta) = \ell(\zeta(\theta)) = \ell(\exp(\theta_1), \exp(\theta_2)).$$

Nous allons procéder à cette estimation par un algorithme de Newton-Raphson avec dérivées numériques. Les étapes de l'algorithme sont les suivants :

1. On fixe une valeur initiale θ_0 . Nous avons pris $h_0 = N_d / \sum_{i=1}^N y_i$ car quand $\beta = 1$, h_0 est l'EMV de h . Pour choisir la valeur initiale de β , nous avons considéré le cas des données non censurées ; la condition d'optimisation sur h donne $\beta = h \times \sum y_i / N$, en remplaçant h par h_0 , on obtient $\beta_0 = N_d / N$. Ces valeurs ne sont pas convergentes mais devraient donner un bon ordre de grandeur.

2. Calcul du gradient et du hessien numériques, respectivement $s(\theta^{(k)})$ et $H(\theta^{(k)})$.
3. Calcul de la nouvelle valeur du paramètre
 - (a) On calcule d'abord $\theta^{(k+1)} = \theta^{(k)} - \lambda_k H^{-1}(\theta^{(k)}) s(\theta^{(k)})$, $k \geq 0$ avec $\lambda_k = 1$.
 - (b) On vérifie ensuite la condition suivante : $\ell(\theta^{(k+1)}) \geq \ell(\theta^{(k)})$, l'algorithme doit être croissant.
 - (c) Si cette condition est remplie, on passe à l'itération suivante (étape 2), sinon on pose $\lambda_k = 1/2, (1/2)^2, \dots, (1/2)^c$ jusqu'à ce qu'elle soit remplie. Dans le programme, on pose $c = 10$, et on affiche la valeur de λ_k à chaque itération pour voir comment elle s'est passée.
4. Pour arrêter l'algorithme, on utilise les élasticités de la fonction objectif par rapport au paramètre d'optimisation. La condition, pour un vecteur de p paramètres est :

$$e(\theta^{(k+1)}) = \max_j \left| \frac{\theta_j^{(k+1)} s(\theta_j^{(k+1)})}{\ell_\zeta(\theta_j^{(k+1)})} \right| < \bar{e},$$

avec $\bar{e} = 10^{-6}$. On peut modifier cette valeur dans les macros variables.

5. On affiche les résultats.

4.6.3 Statistiques utiles

La moyenne peut être estimée par :

$$\hat{m} = \frac{\hat{\beta}}{\hat{h}},$$

et l'on utilise :

$$m = g(\theta) = \frac{\beta}{h}, \quad \frac{\partial g}{\partial \theta'} = \left(\frac{1}{h}, -\frac{\beta}{h^2} \right)$$

$$\text{Vas}(\hat{m}) = \frac{\partial g}{\partial \theta'}(\hat{\theta}) \text{Vas}(\hat{\theta}) \left(\frac{\partial g}{\partial \theta'}(\hat{\theta}) \right)'$$

pour la médiane, on utilise :

$$\widehat{\text{Me}} = \frac{1}{2\hat{h}} \times F_{\chi^2, 2\hat{\beta}}^{-1} \left(\frac{1}{2} \right).$$

Pour estimer toutes les fonctions qui définissent la distribution, il faut choisir une plage de valeurs de t , par exemple :

$$t = 1, 2, \dots, t^+,$$

on estime alors les fonctions en tous les points définis par la grille choisie par l'utilisateur, en utilisant :

$$\begin{aligned}\hat{f}_T(t) &= \frac{\hat{h}^{\hat{\beta}} t^{\hat{\beta}-1} \exp(-\hat{h}t)}{\Gamma(\hat{\beta})}, \\ \hat{S}_T(t) &= 1 - \gamma(\hat{\beta}, \hat{h}t), \\ \hat{F}_T(t) &= 1 - \hat{S}_T(t), \\ \hat{h}_T(t) &= \frac{\hat{f}_T(t)}{\hat{S}_T(t)}.\end{aligned}$$

4.6.4 Programmation VI : Algorithme de Newton-Raphson sans routine d'optimisation et avec dérivées numériques

Voici le programme d'estimation et son commentaire. Nous avons vérifié qu'il donne les mêmes résultats que la routine *nlpnra*.

1. `%let tab=duree; %let y=dureejour; %let d=d1;`
`%let outp=_outp; %let outi=_outi;`
 Les paramètres habituels du programme.
2. `%let i=100; %let e=1e-6; %let c=10;`
 Les paramètres spécifiques au programme : la macro-variable `&i` donne le nombre maximum d'itérations que l'on fixe par sécurité. Les routines d'optimisation peuvent aller jusqu'à 250 par défaut, mais l'algorithme de Newton-Raphson requiert relativement peu d'itérations, nous fixons un maximum à 100, chiffre déjà très élevé pour cette routine. La macro-variable `&e` fixe $\bar{e} = 10^{-6}$, et la macro-variable `&c` fixe $c = 10$ soit un pas minimal $\lambda_k = 2^{-10}$.
3. `proc iml;`
`start lecture;`
`nom_y={&y};`
`nom_d={&d};`
`use &tab; read all var(nom_y) into yi;`
`read all var(nom_d) into di;`
`nd=sum(di); n=nrow(yi); nc=n-nd;`
`print "Time variable =" nom_y,`

```
"Censoring variable (=0 if censored) =" nom_d,
"Number of observations =" n,
"Complete =" nd,
"Censored =" nc;
finish lecture;
Module de lecture habituel.
```

4.

```
start igf(beta,x) global(epsilon);
f=probchi(2#x,2#beta);
do i=1 to nrow(x);
f[i]=min(f[i],1-epsilon);
end;
return (f);
finish;
```

Calcul de la fonction Gamma tronquée normalisée (annexe A.4.4).

5.

```
start lv(theta) global(di,yi,lyi);
t1=theta[1]; t2=theta[2];
beta=exp(t1); h=exp(t2);
lfi=beta#t2+(beta-1)#lyi-h#yi-lgamma(beta);
lsi=log(1-igf(beta,h#yi));
lv=sum(di#(lfi-lsi)+lsi);
return (lv);
finish;
```

Calcul de la log-vraisemblance de l'échantillon.

6.

```
start estim;
print "Two-parameter Gamma distribution", ,
"S(t)=1-g(beta,h#t)",
"where g(,) is the regularized incomplete Gamma
function", ;
```

On rappelle la forme de la fonction de survie utilisée.

7.

```
epsilon=1e-6;
```

Paramètre ε pour le calcul de la fonction Gamma normalisée.
8.

```
lyi=log(yi);
```

On calcule $[\ln y_i]$ une seule fois au début du programme pour limiter le temps de calcul.
9.

```
my=sum(yi)/nd; theta0=log(nd/n) || log(1/my);
```

Calcul de la valeur initiale du paramètre d'optimisation.
10.

```
lv0=lv(theta0);
```

Log-vraisemblance au point initial.

11. `print "Starting parameter =" theta0,`
`"Starting log-likelihood =" lv0, , ;`
 Affichage de la valeur initiale et de la log-vraisemblance initiale.
12. `maxc=&c; maxiter=&i; maxe=&e;`
 Passage des macro-variables à des variables IML. Il s'agit des paramètres de l'algorithme.
13. `labopt="Iteration" || "Log-likelihood" || "Max Abs`
`Elasticity" || "Sub-Iteration" ;`
 Noms des colonnes de l'affichage des itérations.
14. `do iter=1 to maxiter;`
 Début de l'algorithme de Newton-Raphson.
15. `call nlpfdd(lv1,sc,he,"lv",theta0);`
 Calcul de la log-vraisemblance (lv1), du score (sc) et du hessien (he) de l'itération courante.
16. `lambda=1;`
 On réinitialise le paramètre de pas à 1 pour chaque itération.
17. `step=sc*inv(he);`
 Pas de l'itération courante, écrit pour un vecteur ligne. Habituellement, avec des vecteurs colonne et un gradient écrit en colonne, on a :

$$\theta^{(k+1)} = \theta^{(k)} - \lambda_k \times H(\theta^{(k)})^{-1} s(\theta^{(k)}),$$

en transposant cette formule, on obtient :

$$\theta^{(k+1)'} = \theta^{(k)'} - \lambda_k \times s(\theta^{(k)})' H(\theta^{(k)})^{-1}$$

car $H(\theta^{(k)})$ est symétrique. Donc avec un gradient écrit en ligne, on obtient la commande : `pas = s(\theta^{(k)})' H(\theta^{(k)})^{-1}`.

18. `theta=theta0-lambda#step;`
 Itération. Le vecteur ligne `theta` correspond à $\theta^{(k+1)'}$ et `theta0` à $\theta^{(k)'}$.
19. `vp=max(eigval(he)); print "vp =" vp;`
 Calcul de la plus grande valeur propre du hessien, pour vérifier qu'il n'y a pas de problème. Elle doit être négative; si ce n'est pas le cas, il faut changer de valeur initiale...ou d'algorithme.
20. `lv1=lv(theta);`
 Calcul de la nouvelle log vraisemblance $\ell_{\zeta}(\theta^{(k+1)})$.

21. `if lv1 < lv0 then do;`
Si la nouvelle log-vraisemblance n'est pas supérieure à l'ancienne, c'est que le pas était trop important, il faut donc le réduire.
22. `do c=1 to maxc;`
Boucle de réduction du pas de l'itération.
23. `lambda=lambda/2;`
On divise par 2 à chaque tour de boucle, ce qui donne $(1/2)^c$.
24. `theta=theta0-lambda#step;`
Paramètre avec un pas plus faible.
25. `lv1=lv(theta);`
Log-vraisemblance associée au nouveau paramètre.
26. `if lv1>=lv0 then do; print lambda; c=maxc; end;`
Si le nouveau paramètre fournit une log-vraisemblance supérieure, on arrête la boucle, sinon on continue jusqu'à $c = 10$.
27. `end;`
Fin de la boucle.
28. `end;`
Fin de l'instruction *then do*.
29. `elas=max(abs(theta#sc/lv1));`
Plus grande élasticité absolue de la log-vraisemblance par rapport au paramètre d'optimisation.
30. `riter=(iter||lv1||elas||lambda);`
Résultat de l'itération $(k+1, \ell_{\zeta}(\theta^{(k+1)}), e(\theta^{(k+1)}), \lambda_k)$
31. `print riter[colname=labopt];`
Affichage des résultats de l'itération.
32. `theta0=theta; lv0=lv1;`
On prend $\theta^{(k+1)}$ comme point de départ de la nouvelle itération.
33. `if elas < &e then iter=maxiter;`
Si la plus grande élasticité absolue passe sous le seuil \bar{e} , on arrête l'algorithme.
34. `end;`
Fin de la boucle de l'algorithme de Newton-Raphson.
35. `call nlpfdd(lv1,gr,he,"lv",theta);`
Calcul de la log-vraisemblance, du gradient et du hessien numériques au maximum.

36. `gr=t(gr); theta=t(theta); elas=abs(theta#gr/lv1);`
`vp=eigval(he);`
 On remet le gradient et le paramètre en colonnes, on calcule les élasticités absolues à l'optimum ainsi que les valeurs propres du hessien.
37. `print "Log-likelihood at maximum =" lv1,`
`"Gradient" gr, "Elasticities =" elas,`
`"Hessian eigenvalues =" vp;`
 Affichage des résultats à l'optimum. Les élasticités doivent être nulles et le hessien défini négatif (toutes les valeurs propres doivent donc être négatives).
38. `vtheta=inv(-he);`
 Calcul de l'estimation de la matrice de covariance asymptotique.
39. `b=exp(theta); beta=b[1]; h=b[2];`
 On range le vecteur des paramètres d'intérêt ζ dans le vecteur IML `b`. On extrait les composantes de ce vecteur pour simplifier l'écriture des formules qui vont suivre.
40. `db=(beta||0)/(0||h); vb=db*vtheta*t(db);`
 Estimation de la matrice de covariance asymptotique $\widehat{\text{Var}}(\hat{\zeta})$.
41. `sb=sqrt(vecdiag(vb));`
 Estimation des écarts-types asymptotiques.
42. `tb=b/sb;`
 Calcul des t de Student asymptotiques.
43. `pb=1-probchi(tb##2,1);`
 Calcul des probabilités critiques associées.
44. `resp=b||sb||tb||pb;`
 Stockage des statistiques dans une matrice `resp`.
45. `labc="Params"||"Std_Dev"||"Student"||"Pvalue";`
 Futurs noms de colonne du tableau SAS en sortie.
46. `labr="Log-likelihood"//"Beta"//"H"//"Mean"//`
`"Median";`
 Futurs noms de ligne du tableau SAS en sortie.
47. `md=cinv(0.5,2#beta)/(2#h);`
 Estimation de la durée médiane.
48. `mt=1/h; smt=sqrt((1/h)##2#sb[2]); tmt=mt/smt;`
`pmt=1-probchi(tmt##2,1);`
 Estimation de la durée moyenne et des statistiques associées.

- ```

49. resp=(lv1|||.|||.||.)//resp//(mt||smt||tmt||pmt)
 //(md|||.|||.||.);
 Regroupement de l'ensemble des statistiques dans la matrice resp.
50. print resp[colname=labc rowname=labr];
 Affichage des résultats.
51. stat_exp=abs(beta-1)/sb[1];
 pval_exp=1-probchi(stat_exp##2,1);
 stat_gam=abs(h-1)/sb[2];
 pval_gam=1-probchi(stat_gam##2,1);
 print "Test H0 : beta=1 (loi exponentielle)"
 stat_exp, "p-value =" pval_exp, , ,
 "Test H0 : h=1 (loi Gamma à 1 paramètre)"
 stat_gam, "p-value" pval_gam;
 Tests de pertinence de la loi exponentielle (statistique stat_exp) et de
 la loi Gamma à un paramètre (statistique stat_gam) et affichage des
 résultats.
52. finish estim;
 Fin du module d'estimation.
53. start prev;
 Début du module de prévision. Sa structure est identique à celle des mo-
 dèles précédents, seules les fonctions changent.
54. tmin=min(yi); tmax=max(yi); tby=1;
 do time=tmin to tmax by tby;
 fi=h##beta#time##(beta-1)#exp(-h#time)/
 gamma(beta);
 si=1-igf(beta,h#time);
 hi=fi/si;
 ci=1-si;
 resi=resi//(time||fi||si||hi||ci);
 end;
 labi="&y"||"Density"||"SDF"||"Hasard"||"CDF";
 create &outi from resi[colname=labi];
 append from resi;

55. finish prev;
 Fin du module de prévision.
56. start sortie;
 Module de sortie identique aux précédents.

```

```

create &outp from resp[colname=labc];
append from resp;
create _np from labr[colname="Estimate"];
append from labr;
create &outi from resi[colname=labi];
append from resi;
finish sortie;
57. run lecture;
run estim;
run prev;
quit;
Lancement des modules et sortie de SAS-IML.
58. data &outp; merge _np &outp;
run;
Ajout des noms de lignes à la matrice de résultats sur les paramètres et
lancement du programme.

```

#### 4.6.5 Application VI

On reprend l'étude de la durée du premier emploi. Les fonctions de survie (graphique 4.12) et de hasard (graphique 4.13) estimées pour la loi Gamma à deux paramètres ont des formes similaires à celles du modèle de Weibull. Toutefois, la loi Gamma possède une log-vraisemblance supérieure ( $-35513.7$  contre  $-35853.3$  pour le modèle de Weibull). Ce modèle n'est donc pas à négliger dans cette application.

#### Sortie SAS 4.7.

Two-parameter Gamma distribution

$$S(t)=1-g(\text{beta},h\#t)$$

where  $g(,)$  is the regularized incomplete Gamma function

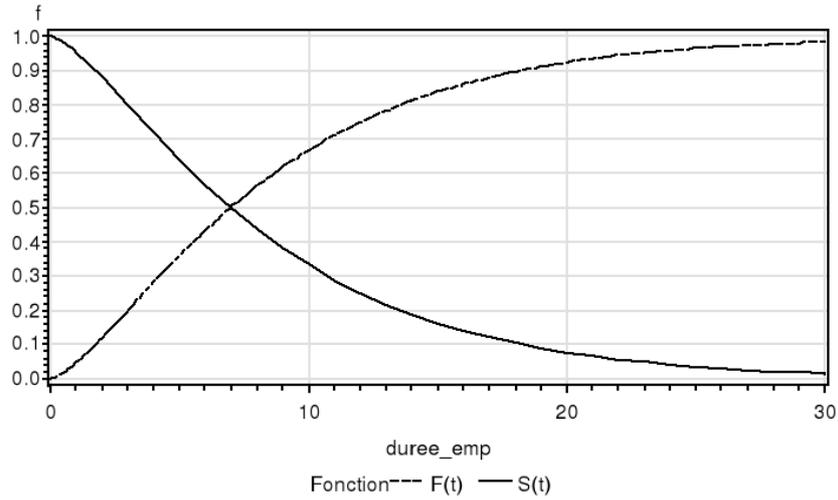
theta0

Starting parameter = -0.118443 -2.200461  
lv0

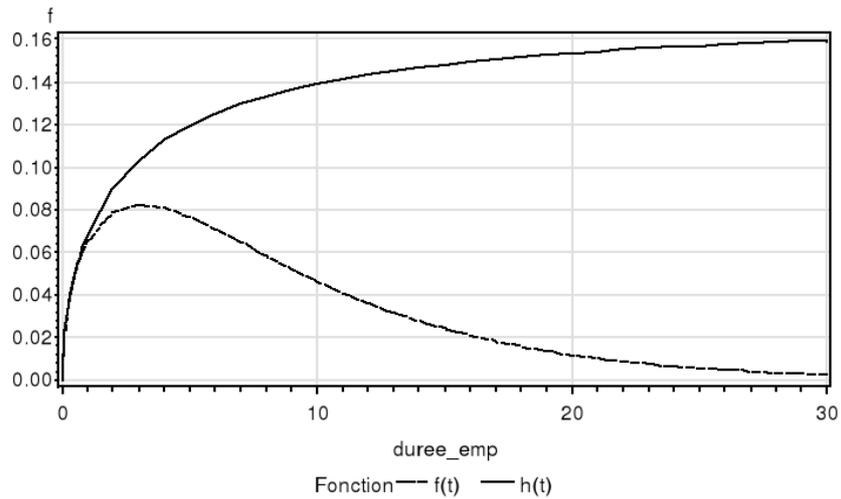
Starting log-likelihood = -36511.91

vp

vp = -2163.463

GRAPHIQUE 4.12 – LOI GAMMA 2,  $\hat{S}_T(t)$  ET  $\hat{F}_T(t)$ 

riter

GRAPHIQUE 4.13 – LOI GAMMA 2,  $\hat{h}_T(t)$  ET  $\hat{f}_T(t)$

|      | Iteration | Log-likelihood | Max Abs | Elasticity | Sub-Iteration |
|------|-----------|----------------|---------|------------|---------------|
| ROW1 | 1         | -35635.87      |         | 0.0726365  | 1             |

vp

vp = -3131.416

riter

|      | Iteration | Log-likelihood | Max Abs | Elasticity | Sub-Iteration |
|------|-----------|----------------|---------|------------|---------------|
| ROW1 | 2         | -35514.71      |         | 0.0641672  | 1             |

vp

vp = -2984.023

riter

|      | Iteration | Log-likelihood | Max Abs | Elasticity | Sub-Iteration |
|------|-----------|----------------|---------|------------|---------------|
| ROW1 | 3         | -35513.7       |         | 0.0060375  | 1             |

vp

vp = -2977.117

riter

|      | Iteration | Log-likelihood | Max Abs | Elasticity | Sub-Iteration |
|------|-----------|----------------|---------|------------|---------------|
| ROW1 | 4         | -35513.7       |         | 0.0000447  | 1             |

vp

vp = -3034.538

riter

|      | Iteration | Log-likelihood | Max Abs | Elasticity | Sub-Iteration |
|------|-----------|----------------|---------|------------|---------------|
| ROW1 | 5         | -35513.7       |         | 4.2385E-6  | 0.0009766     |

vp

vp = -2968.819

lambda

0.25

riter

|  | Iteration | Log-likelihood | Max Abs | Elasticity | Sub-Iteration |
|--|-----------|----------------|---------|------------|---------------|
|--|-----------|----------------|---------|------------|---------------|

ROW1            6            -35513.7            4.3084E-6            0.25

vp

vp = -2992.287

lambda

0.25

riter

Iteration Log-likelihood Max Abs Elasticity Sub-Iteration

ROW1            7            -35513.7            3.0412E-6            0.25

vp

vp = -2989.785

riter

Iteration Log-likelihood Max Abs Elasticity Sub-Iteration

ROW1            8            -35513.7            7.6904E-7            1

lvl

Log-likelihood at maximum = -35513.7

gr

Gradient 0.0710229

0.032382

elas

Elasticities = 8.5994E-7

1.5905E-6

vp

Hessian eigenvalues = -2961.672

-41095.03

resp

Params    Std\_Dev    Student    Pvalue

|                |           |           |           |   |
|----------------|-----------|-----------|-----------|---|
| Log-likelihood | -35513.7  | .         | .         | . |
| Beta           | 1.537253  | 0.0185049 | 83.072609 | 0 |
| H              | 0.1747604 | 0.0025748 | 67.872595 | 0 |
| Mean           | 5.7221214 | 0.2903563 | 19.707238 | 0 |
| Median         | 6.9797055 | .         | .         | . |

stat\_exp

```
Test H0 : beta=1 (loi exponentielle) 29.032961
 pval_exp
```

```
p-value = 0
```

```
stat_gam
```

```
Test H0 : h=1 (loi Gamma à 1 paramètre) 320.50263
 pval_gam
```

```
p-value 0
```

Les estimations donnent  $\hat{\beta} = 1.54$  et  $\hat{h} = 0.17$ . On se retrouve donc loin des cas particuliers de la loi Gamma à deux paramètres. Le test de Student asymptotique de l'hypothèse  $\beta = 1$  (modèle exponentiel) donne une statistique de 29 qui aboutit au rejet du modèle exponentiel (valeur critique de 1.96 au seuil de 5%), et le test de Student asymptotique de l'hypothèse  $h = 1$  (modèle Gamma à un paramètre) donne une statistique de 320, qui amène à rejeter l'hypothèse nulle à tous les seuils conventionnels.

Les moments estimés par la loi Gamma à deux paramètres sont assez différents des précédents car la moyenne estimée (5.7 ans) est inférieure à la médiane estimée (7 ans). Le modèle Gamma généralisé peut être utilisé pour trancher entre les modèles de Weibull et le modèle Gamma à deux paramètres.

## 4.7 Loi log-normale

### 4.7.1 Log-vraisemblance

Les logarithmes de la densité et de la survie sont donnés par :

$$\begin{aligned}\ln f_T(y_i) &= -\ln \sigma - \ln y_i + \ln \varphi\left(\frac{\ln y_i - m}{\sigma}\right) \\ \ln S_T(y_i) &= \ln\left(1 - \Phi\left(\frac{\ln y_i - m}{\sigma}\right)\right)\end{aligned}$$

on opère une première simplification en utilisant le changement de paramètres suivant :

$$\alpha = \frac{1}{\sigma}, \mu = \frac{m}{\sigma},$$

ce qui donne :

$$\begin{aligned}\ln f_T(y_i) &= \ln \alpha - \ln y_i + \ln \varphi(\alpha \ln y_i - \mu) \\ \ln S_T(y_i) &= \ln(1 - \Phi(\alpha(\ln y_i - \mu)))\end{aligned}$$

Pour simplifier les notations, on pose :

$$z_i = \alpha \ln y_i - \mu,$$

d'où la vraisemblance :

$$\ell(\zeta) = \sum_{i=1}^N d_i (\ln \alpha - \ln y_i + \ln \varphi(z_i)) + (1 - d_i) \ln(1 - \Phi(z_i)),$$

avec

$$\zeta = \begin{pmatrix} \alpha \\ \mu \end{pmatrix}.$$

Le score est donné par (on utilise la relation  $\varphi'(z) = -z\varphi(z)$ ) :

$$\begin{aligned} \frac{\partial \ell}{\partial \alpha} &= \sum_{i=1}^N d_i \left( \frac{1}{\alpha} - z_i \ln y_i \right) - (1 - d_i) \ln y_i \times \frac{\varphi(z_i)}{1 - \Phi(z_i)} \\ \frac{\partial \ell}{\partial \mu} &= \sum_{i=1}^N d_i z_i + (1 - d_i) \frac{\varphi(z_i)}{1 - \Phi(z_i)} \end{aligned}$$

#### 4.7.2 Estimation

On impose la positivité du premier paramètre seulement (car il s'agit d'un écart type) :

$$\begin{aligned} \alpha &= \exp(\theta_1), \quad \mu = \theta_2, \quad \theta = \begin{pmatrix} \theta_1 \\ \theta_2 \end{pmatrix}, \\ \frac{\partial \alpha}{\partial \theta_1} &= \alpha. \end{aligned}$$

ce qui donne la log-vraisemblance :

$$\ell_\zeta(\theta) = \ell(\zeta(\theta)),$$

les dérivées premières permettent d'accélérer l'algorithme d'optimisation :

$$\begin{aligned} \frac{\partial \ell_\zeta}{\partial \theta_1} &= \sum_{i=1}^N \frac{\partial \ell_i}{\partial \alpha} \frac{\partial \alpha}{\partial \theta_1} \\ &= N_d - \alpha \sum_{i=1}^N \ln y_i \left\{ d_i z_i + (1 - d_i) \frac{\varphi(z_i)}{1 - \Phi(z_i)} \right\} \\ \frac{\partial \ell_\zeta}{\partial \theta_2} &= \frac{\partial \ell}{\partial \mu}. \end{aligned}$$

Pour la log-vraisemblance comme pour les dérivées, il convient de s'assurer que l'on a toujours  $1 - \Phi(z_i) > 0$ , on utilisera donc :

$$\max(1 - \Phi(z_i), \varepsilon)$$

avec  $\varepsilon = 10^{-6}$  (modifiable par l'utilisateur). Pour retrouver les paramètres  $(\sigma, m)$ , on doit poser :

$$\zeta = \begin{pmatrix} \sigma \\ m \end{pmatrix} = \begin{pmatrix} 1/\alpha \\ \mu/\alpha \end{pmatrix} = \begin{pmatrix} \exp(-\theta_1) \\ \theta_2 \exp(-\theta_1) \end{pmatrix},$$

donc on utilise le théorème de Slutsky avec :

$$\frac{\partial \mathbf{g}}{\partial \theta'} = \begin{pmatrix} -\exp(-\theta_1) & 0 \\ -\theta_2 \exp(-\theta_1) & \exp(-\theta_1) \end{pmatrix}.$$

### 4.7.3 Statistiques utiles

Si  $\ln T$  suit une loi normale  $N(m, \sigma^2)$ , l'espérance mathématique est donnée par :

$$E(T) = g(\zeta) = \exp(m + \sigma^2/2)$$

pour calculer la variance asymptotique, on utilisera donc :

$$\frac{\partial g}{\partial m} = \exp(m + \sigma^2/2), \quad \frac{\partial g}{\partial \sigma} = \sigma \exp(m + \sigma^2/2).$$

La médiane est simplement égale à :

$$\text{Me} = g(\zeta) = \exp(m)$$

donc on utilisera

$$\frac{\partial g}{\partial m} = \exp(m).$$

Les fonctions peuvent être estimées par :

$$\begin{aligned} \hat{f}_T(t) &= \frac{1}{\hat{\sigma} t} \varphi\left(\frac{\ln t - \hat{m}}{\hat{\sigma}}\right), \\ \hat{F}_T(t) &= \Phi\left(\frac{\ln t - \hat{m}}{\hat{\sigma}}\right), \\ \hat{S}_T(t) &= 1 - \hat{F}_T(t), \\ \hat{h}_T(t) &= \frac{\hat{f}_T(t)}{\hat{S}_T(t)}. \end{aligned}$$

#### 4.7.4 Programmation VII : Algorithme de Newton-Raphson sans routine d'optimisation et avec gradient analytique

Voici le programme d'estimation commenté.

```
1. proc iml;
 start lecture;
 nom_y={&y};
 nom_d={&d};
 use &tab; read all var(nom_y) into yi;
 read all var(nom_d) into di;
 nd=sum(di); n=nrow(yi); nc=n-nd;
 print "Time variable =" nom_y,
 "Censoring variable (=0 if censored) =" nom_d,
 "Number of observations =" n,
 "Complete =" nd, "Censored =" nc;
 finish lecture;
 Module de lecture identique aux précédents.

2. start cphi(x) global(epsilon);
 f=1-probnorm(x);
 do i=1 to nrow(x);
 f[i]=max(f[i],epsilon);
 end;
 return (f);
 finish;
 Fonction de calcul de la survie :
```

$$cphi(x) = \max(1 - \Phi(x), \epsilon),$$

avec :

$$x = \alpha \ln y_i - \mu, \epsilon = 10^{-6} \text{ (par défaut, modifiable par l'utilisateur).}$$

```
3. start lv(theta) global(di,lyi,epsilon);
 t1=theta[1]; alpha=exp(t1); mu=theta[2];
 zi=alpha#lyi-mu;
 lfi=t1-lyi+log(pdf("normal",zi,0,1));
 lsi=log(cphi(zi));
 lv=sum(di#lfi+(1-di)#lsi);
 return (lv);
 finish;
 Fonction calculant la log-vraisemblance de l'échantillon.
```

- ```

4. start sc(theta) global(di,lyi,epsilon,nd);
   alpha=exp(theta[1]); mu=theta[2];
   zi=alpha#lyi-mu;
   wi=di#zi+(1-di)#pdf("normal",zi,0,1)/cphi(zi);
   der1=nd-alpha#sum(lyi#wi);
   der2=sum(wi);
   der=(der1||der2);
   return (der);
   finish;
   Fonction calculant le score de l'échantillon.
5. start estim;
   Début du module d'estimation.
6. print "Log-normal distribution", ,
   "S(t)=1-Phi((ln(t)-m)/sigma)", ,
   "where Phi() is the CDF of the standard normal
   distribution", ;
   On rappelle la forme de la fonction de survie.
7. epsilon=1e-6;
   On fixe  $\varepsilon = 10^{-6}$ .
8. lyi=log(yi);
   On stocke  $[\ln(y_i)]$  dans un vecteur pour économiser du temps de calcul.
9. mly=sum(lyi)/nd; sly=sqrt(ssq(lyi)/nd-mly##2);
   Calcul des valeurs initiales  $(\sigma_0, m_0)$ . On prend les moments empiriques
   sur les données tronquées, donc ce n'est pas un estimateur convergent.
   Il devrait toutefois donner des ordres de grandeurs pertinents.
10. theta0=(log(1/sly)||mly/sly);
   Calcul de la valeur initiale du paramètre d'optimisation :

```
- $$\theta_0 = (\alpha_0, \mu_0) = (\ln(1/\sigma_0), m_0/\sigma_0).$$
- ```

11. lv0=lv(theta0);
 Log-vraisemblance au point initial.
12. print "Starting parameter =" theta0,
 "Starting log-likelihood =" lv0, , ;
 Affichage des données du point de départ.
13. maxc=&c; maxiter=&i; maxe=&e;
 Macro-variables de la procédure de Newton-Raphson, identiques à celles
 du programme précédent.

```

14. labopt="Iteration" || "Log-likelihood" || "Max Abs Elasticity" || "Sub-Iteration";  
Noms des colonnes de l'affichage des itérations.
15. do iter=1 to maxiter;  
Boucle des itérations de la procédure de Newton-Raphson.
16. call nlpfdd(lv1,sc,he,"lv",theta0) grd="sc";  
Calcul du hessien numérique à partir des dérivées premières analytiques données par la fonction sc.
17. lambda=1;  
On réinitialise le paramètre de pas à 1 à chaque itération.
18. step=sc\*inv(he);  
Pas de l'itération.
19. theta=theta0-lambda#step;  
Paramètre candidat de l'itération.
20. vp=max(eigval(he));
21. print "vp =" vp;  
Calcul de la plus grande valeur propre du hessien et affichage.
22. lv1=lv(theta);  
Log-vraisemblance à la valeur du paramètre candidat de l'itération.
23. if lv1 < lv0 then do;  
Si la log-vraisemblance est inférieure au paramètre candidat, on réduit le pas de l'itération jusqu'à ce qu'elle soit supérieure.
24. do c=1 to maxc;  
lambda=lambda/2;  
theta=theta0-lambda#step;  
lv1=lv(theta);  
if lv1 >= lv0 then c=maxc;  
end;  
Fin de la boucle de réduction du pas.  
end;  
Fin du *then do*.
25. elas=max(abs(theta#sc/lv1));  
Critère d'arrêt, la plus grande des élasticités de la log-vraisemblance par rapport aux paramètres.
26. riter=(iter||lv1||elas||lambda);  
print riter[colname=labopt];  
Affichage du résultat de l'itération courante.

27. `theta0=theta; lv0=lv1;`  
On accepte le paramètre candidat, après une modification éventuelle de réduction de pas.
28. `if elas < &e then iter=maxiter;`  
Si la plus grande des élasticités de la log-vraisemblance par rapport aux paramètres est suffisamment faible, on arrête l'algorithme. Sinon, on refait une itération.
29. `end;`  
Fin de l'algorithme de Newton-Raphson.
30. `call nlpfdd(lv1,gr,he,"lv",theta) grd="sc";`  
Calcul du hessien à l'optimum.
31. `gr=t(gr); theta=t(theta); elas=abs(theta#gr/lv1);`  
`vp=eigval(he);`  
`print "Log-likelihood at maximum =" lv1,`
32. `"Gradient =" gr,"Elasticities =" elas,`  
`"Hessian eigenvalues =" vp;`  
Calcul des résultats de l'optimisation et affichage.
33. `vtheta=inv(-he);`  
Calcul de l'estimation de la variance asymptotique de l'estimateur du paramètre d'optimisation,  $\widehat{Vas}(\hat{\theta})$ .
34. `t1=theta[1]; t2=theta[2];`  
`b=exp(-t1)//t2#exp(-t1);`  
`sigma=b[1]; m=b[2];`  
Estimation du paramètre d'intérêt  $\hat{\zeta}$ .
35. `db=(-exp(-t1)||0)//(-t2#exp(-t1)||exp(-t1));`  
Calcul de  $\partial\zeta/\partial\theta'(\hat{\theta})$ .
36. `vb=db*vtheta*t(db);`  
Calcul de  $\widehat{Vas}(\hat{\zeta})$ .
37. `sb=sqrt(vecdiag(vb));`  
Calcul des estimations des écarts-types asymptotiques de l'estimateur du paramètre d'intérêt.
38. `tb=b/sb;`  
Estimation des  $t$  de Student asymptotiques correspondants.
39. `pb=1-probchi(tb##2,1);`  
Calcul des probabilités critiques correspondantes.
40. `resp=b||sb||tb||pb;`  
`labc="Params"||"Std_Dev"||"Student"||"Pvalue";`

```
labr="Log-likelihood"// "Sigma"// "m"// "Mean"//
"Median";
```

Constitution de la matrice des résultats, des noms de colonnes et de lignes du futur tableau de sortie SAS.

```
41. md=exp(m); smd=sqrt(md##2#vb[2,2]); tmd=md/smd;
pmd=1-probchi(tmd##2,1);
```

Estimation de la durée médiane et des statistiques associées.

```
42. mt=exp(m+sigma##2/2); dmt=(sigma#mt||mt);
smt=sqrt(dmt*vb*t(dmt)); tmt=mt/smt;
pmt=1-probchi(tmt##2,1);
```

Estimation de la durée moyenne et des statistiques associées.

```
43. resp=(lv1||.||.||||.)//resp//(mt||smt||tmt||pmt)//
(md||smd||tmd||pmd);
```

Constitution de la matrice des résultats incluant la log-vraisemblance, et les estimations de la moyenne et de la médiane.

```
44. start has(x) global(alpha,mu);
z=alpha#x-mu;
f=pdf("normal",z,0,1)#alpha/exp(x);
s=cphi(z);
has=log(f)-log(s);
return (has);
finish;
```

Fonction qui calcule le hasard en fonction de la durée  $h_T(t)$ . Elle sert à trouver le point auquel le hasard est maximum, puisqu'il existe toujours avec une loi log-normale. On maximisera le logarithme du hasard. La durée elle-même sera modélisée sous forme exponentielle  $t = \exp(x)$  afin de garantir que les itérations ne donnent que des valeurs positives.

```
45. opt=j(11,1,0); opt[1]=1; opt[2]=0; alpha=exp(t1);
mu=t2; lt0=log(md);
```

Paramètres de la procédure nlpnra, maximisation sans affichage. La durée médiane est prise comme valeur initiale  $x_0 = \ln(\text{Me})$ .

```
46. call nlpnra(rc,lt1,"has",lt0,opt);
Recherche du maximum de la fonction de hasard.
```

```
47. tmax=exp(lt1); hmax=exp(has(lt1));
```

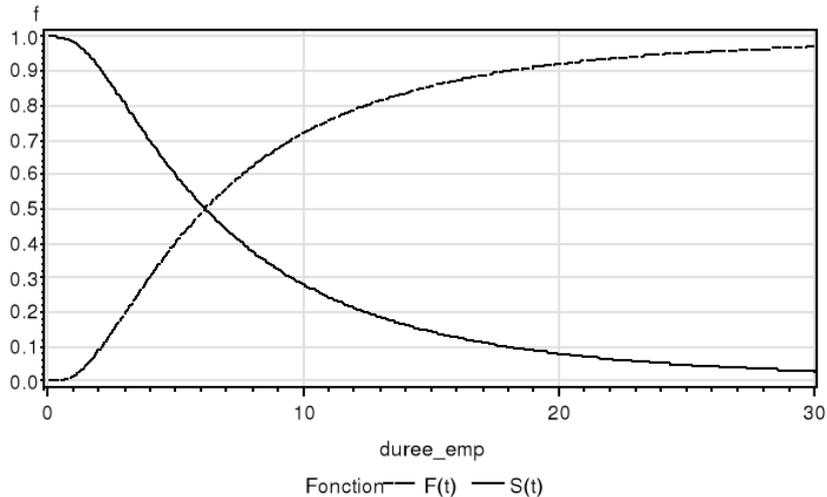
Calcul du point qui maximise le hasard  $t^* = \exp(x^*)$  et du hasard correspondant  $h^* = h_T(t^*)$ .

```
48. print "The hasard function reaches its maximum at
T =" tmax, "It reaches the value h =" hmax;
```

- ```

print resp[colname=labc rowname=labr];
Affichage des résultats.
49. finish estim;
Fin du module d'estimation.
50. start prev;
Début du module de prévision. Sa structure est identique aux modules
des programmes précédents, seules les fonctions changent.
51. tmin=min(yi); tmax=max(yi); tby=1;
alpha=exp(t1); mu=t2;
do time=tmin to tmax by tby;
zi=alpha#log(time)-mu;
fi=pdf("normal",zi,0,1)#alpha/time;
si=cphi(zi);
hi=fi/si;
ci=1-si;
resi=resi/(time||fi||si||hi||ci);
end;
labi("&y"||"Density"||"SDF"||"Hasard"||"CDF");
finish prev;
Fin du module de prévision.
52. start sortie;
Module de sortie identique aux précédents.
create &outp from resp[colname=labc];
append from resp;
create _np from labr[colname="Estimate"];
append from labr;
create &outi from resi[colname=labi];
append from resi;
finish sortie;
53. run lecture;
run estim;
run prev;
run sortie;
quit;
Lancement des modules et sortie de SAS-IML.
54. data &outp; merge _np &outp;
run;
Ajout des noms de lignes de la matrice des résultats et lancement du pro-
gramme.

```

GRAPHIQUE 4.14 – LOI LOG-NORMALE, $\hat{S}_T(t)$ ET $\hat{F}_T(t)$

4.7.5 Application VII

Nous reprenons l'estimation des durées du premier emploi avec une loi log-normale. Une caractéristique importante de cette distribution est qu'elle impose que le hasard admet un maximum (graphique 4.14). On trouve par le calcul que ce maximum est situé à $T = 5.67$ ans. La log-vraisemblance maximale présente également une valeur intéressante, puisqu'il s'agit de la plus élevée que nous ayons trouvée jusqu'à maintenant (-34272.35). Il est possible d'examiner l'adéquation de cette loi à partir de la loi Gamma généralisée que nous verrons plus loin.

Sortie SAS 4.8.

```
Log-normal distribution
```

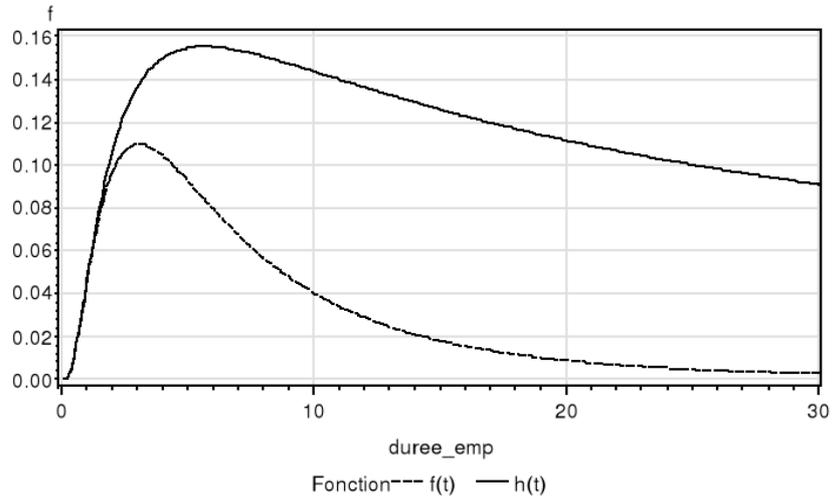
```
S(t)=1-Phi((ln(t)-m)/sigma)
```

```
where Phi() is the CDF of the standard normal distribution
```

```
theta0
```

```
Starting parameter = 0.2271609 2.1846851
                    lv0
```

```
Starting log-likelihood = -34351.08
```



GRAPHIQUE 4.15 – LOI LOG-NORMALE, $\hat{h}_T(t)$ ET $\hat{f}_T(t)$

vp

vp = -3457.245

riter

Iteration Log-likelihood Max Abs Elasticity Sub-Iteration

| | | | | |
|------|---|-----------|-----------|---|
| ROW1 | 1 | -34272.82 | 0.0732448 | 1 |
|------|---|-----------|-----------|---|

vp

vp = -3186.519

riter

Iteration Log-likelihood Max Abs Elasticity Sub-Iteration

| | | | | |
|------|---|-----------|----------|---|
| ROW1 | 2 | -34272.35 | 0.001439 | 1 |
|------|---|-----------|----------|---|

vp

vp = -3177.523

riter

Iteration Log-likelihood Max Abs Elasticity Sub-Iteration

| | | | | |
|------|---|-----------|-----------|---|
| ROW1 | 3 | -34272.35 | 0.0000324 | 1 |
|------|---|-----------|-----------|---|

```

                                vp
                                vp = -3177.38

                                riter
Iteration Log-likelihood Max Abs Elasticity Sub-Iteration
ROW1      4      -34272.35      2.3321E-9      1

                                lvl
Log-likelihood at maximum = -34272.35
                                gr
                                Gradient = -6.49E-10
                                -1.87E-11
                                elas

                                Elasticities = 3.42E-15
                                1.19E-15
                                vp

                                Hessian eigenvalues = -3177.372
                                -84832.53

                                tmax

The hazard function reaches a maximum at T = 5.6759982
                                hmax

Its maximum value is h = 0.1553491
                                resp
                                Params      Std_Dev      Student      Pvalue
Log-likelihood -34272.35      .      .      .
Sigma          0.8348341  0.0056114  148.77549      0
m              1.8186529  0.007578  239.99218      0
Mean           8.7331567  0.079803  109.43396      0
Median         6.1635501  0.0467072  131.96151      0

```

Nous trouvons que le logarithme de la durée du premier emploi suivrait une loi log-normale de paramètres estimés :

$$(\hat{m}, \hat{\sigma}) = (1.82, 0.83),$$

ce qui correspond à une durée moyenne de 8.7 ans et à une durée médiane de 6.2 ans. La moitié des contrats de travail d'un premier emploi parviendraient donc à leur terme après 6.2 ans. Le taux de sortie du premier emploi serait éga-

lement d'abord croissant pendant les 5.7 premières années, une valeur proche de la médiane, avant de décroître lentement.

4.8 Loi log-logistique

4.8.1 Log-vraisemblance

Les logarithmes du hasard et de la survie sont donnés par :

$$\begin{aligned}\ln h_T(y_i) &= \ln \alpha + \ln h + (\alpha - 1) \ln y_i - \ln(1 + hy_i^\alpha) \\ \ln S_T(y_i) &= -\ln(1 + hy_i^\alpha),\end{aligned}$$

et la vraisemblance peut s'écrire :

$$\ell(\zeta) = N_d(\ln \alpha + \ln h) + (\alpha - 1) \sum_{i=1}^N d_i \ln y_i - \sum_{i=1}^N (1 + d_i) \ln(1 + hy_i^\alpha).$$

avec :

$$\zeta = \begin{pmatrix} \alpha \\ h \end{pmatrix}.$$

4.8.2 Estimation

Pour exprimer les dérivées, on posera :

$$z_i = \frac{hy_i^\alpha}{1 + hy_i^\alpha},$$

de sorte que le score s'écrit :

$$\begin{aligned}\frac{\partial \ell}{\partial \alpha} &= \frac{N_d}{\alpha} + \sum_{i=1}^N (d_i - (1 + d_i) z_i) \ln y_i \\ \frac{\partial \ell}{\partial h} &= \frac{N_d}{h} - \frac{1}{h} \sum_{i=1}^N (1 + d_i) z_i\end{aligned}$$

et les dérivées secondes sont égales à :

$$\begin{aligned}\frac{\partial^2 \ell}{\partial \alpha^2} &= -\frac{N_d}{\alpha^2} - \sum_{i=1}^N (1 + d_i) z_i (1 - z_i) (\ln y_i)^2 \\ \frac{\partial^2 \ell}{\partial \alpha \partial h} &= -\frac{1}{h} \sum_{i=1}^N (1 + d_i) z_i (1 - z_i) \ln y_i \\ \frac{\partial^2 \ell}{\partial h^2} &= -\frac{N_d}{h^2} + \frac{1}{h^2} \sum_{i=1}^N (1 + d_i) z_i^2\end{aligned}$$

Pour l'optimisation, on posera :

$$\alpha = \exp(\theta_1), \quad h = \exp(\theta_2), \quad \theta = \begin{pmatrix} \theta_1 \\ \theta_2 \end{pmatrix}$$

on remarque alors que :

$$\frac{\partial \alpha}{\partial \theta_1} = \alpha, \quad \frac{\partial h}{\partial \theta_2} = h,$$

de sorte que les dérivées premières par rapport aux paramètres d'estimation sont égales à :

$$\begin{aligned} \frac{\partial \ell_\zeta}{\partial \theta_1} &= \alpha \sum_{i=1}^N \frac{\partial \ell_i}{\partial \alpha} \\ &= N_d + \alpha \sum_{i=1}^N (d_i - (1 + d_i) z_i) \ln y_i \\ \frac{\partial \ell_\zeta}{\partial \theta_2} &= h \sum_{i=1}^N \frac{\partial \ell_i}{\partial h} \\ &= N_d - \sum_{i=1}^N (1 + d_i) z_i \end{aligned}$$

et les dérivées secondes à :

$$\begin{aligned} \frac{\partial^2 \ell_\zeta}{\partial \theta_1^2} &= \alpha \sum_{i=1}^N \frac{\partial \ell_i}{\partial \alpha} + \alpha^2 \sum_{i=1}^N \frac{\partial^2 \ell_i}{\partial \alpha^2} \\ &= \alpha \sum_{i=1}^N \ln y_i (d_i - (1 + d_i) z_i - \alpha \ln y_i \times (1 + d_i) z_i (1 - z_i)) \\ \frac{\partial^2 \ell_\zeta}{\partial \theta_1 \partial \theta_2} &= \alpha h \sum_{i=1}^N \frac{\partial^2 \ell_i}{\partial \alpha \partial h} \\ &= -\alpha \sum_{i=1}^N (1 + d_i) z_i (1 - z_i) \ln y_i \\ \frac{\partial^2 \ell_\zeta}{\partial \theta_2^2} &= h \sum_{i=1}^N \frac{\partial \ell_i}{\partial h} + h^2 \sum_{i=1}^N \frac{\partial^2 \ell_i}{\partial h^2} \\ &= -\sum_{i=1}^N (1 + d_i) z_i (1 - z_i) \end{aligned}$$

Après l'optimisation, on retourne aux paramètres initiaux par la transformation :

$$\zeta = \begin{pmatrix} \alpha \\ h \end{pmatrix} = g(\theta) = \begin{pmatrix} \exp(\theta_1) \\ \exp(\theta_2) \end{pmatrix}$$

en utilisant :

$$\frac{\partial g}{\partial \theta'} = \begin{pmatrix} \exp(\theta_1) & 0 \\ 0 & \exp(\theta_2) \end{pmatrix}$$

et

$$\text{Vas}(\hat{\zeta}) = \frac{\partial g}{\partial \theta'}(\hat{\theta}) \text{Vas}(\hat{\theta}) \frac{\partial g}{\partial \theta'}(\hat{\theta})'.$$

4.8.3 Statistiques utiles

Plusieurs paramétrages sont utilisés avec la loi log-logistique. Le premier consiste à recouvrir les paramètres de la distribution logistique de base :

$$\zeta_1 = \begin{pmatrix} \delta \\ m \end{pmatrix}, \delta = \frac{1}{\alpha}, m = -\frac{1}{\alpha} \ln h$$

on utilise alors le théorème de Slutsky avec :

$$\frac{\partial \zeta_1}{\partial (\alpha, h)} = \begin{pmatrix} -1/\alpha^2 & 0 \\ -(\ln h)/\alpha^2 & -1/(\alpha h) \end{pmatrix},$$

le second paramétrage est celui de la loi log-normale pour faciliter les comparaisons. La moyenne se calcule comme dans le cas du paramétrage de la loi normale, mais il faut faire une modification pour obtenir d'écart-type. En notant σ l'écart-type de la distribution logistique de départ, on a :²

$$\sigma = \frac{\pi \delta}{\sqrt{3}} = \frac{\pi}{\alpha \sqrt{3}}$$

de sorte que l'on utilise :

$$\frac{\partial \sigma}{\partial \alpha} = -\frac{\pi}{\alpha^2 \sqrt{3}}.$$

L'espérance de la loi log-logistique existe si $\alpha > 1$, elle est donnée par :

$$E(T) = g(\zeta) = \left(\frac{1}{h}\right)^{1/\alpha} B(1 - 1/\alpha, 1 + 1/\alpha)$$

2. Les deux premiers moments d'une variable aléatoire logistique X de paramètres (m, δ) sont donnés par :

$$E(X) = m \text{ et } V(X) = \frac{\pi^2 \delta^2}{3}.$$

on a donc :

$$\begin{aligned}\frac{\partial g}{\partial \alpha} &= \left(\frac{1}{h}\right)^{1/\alpha} B(1-1/\alpha, 1+1/\alpha) \left(\frac{\ln h}{\alpha^2} + \psi^{(0)}(1-1/\alpha) + \psi^{(0)}(1+1/\alpha)\right) \\ &= E(T) \left(\frac{\ln h}{\alpha^2} + \psi^{(0)}(1-1/\alpha) + \psi^{(0)}(1+1/\alpha)\right),\end{aligned}$$

où $\psi^{(0)}$ est la fonction digamma et :

$$\begin{aligned}\frac{\partial g}{\partial h} &= -\frac{1}{\alpha} h^{-(1/\alpha+1)} B(1-1/\alpha, 1+1/\alpha) \\ &= -\frac{E(T)}{\alpha h}\end{aligned}$$

et on applique le théorème de Slutsky pour obtenir la variance asymptotique de l'estimateur de la moyenne. La médiane est donnée par :

$$\text{Me}(T) = g(\zeta) = \left(\frac{1}{h}\right)^{1/\alpha},$$

on a donc :

$$\begin{aligned}\frac{\partial g}{\partial \alpha} &= \frac{\ln h}{\alpha^2} \left(\frac{1}{h}\right)^{1/\alpha} = \frac{\ln h}{\alpha^2} \times \text{Me}(T) \\ \frac{\partial g}{\partial h} &= -\frac{1}{\alpha} h^{-(1/\alpha+1)} = -\frac{1}{\alpha h} \times \text{Me}(T)\end{aligned}$$

et on applique le théorème de Slutsky. Pour estimer toutes les fonctions qui définissent la distribution, il faut choisir une plage de valeurs de t , par exemple :

$$t = 1, 2, \dots, t^+,$$

on estime alors les fonctions en tous les points définis par la grille choisie par l'utilisateur :

$$\begin{aligned}\hat{h}_T(t) &= \frac{\hat{\alpha} \hat{h} t^{\hat{\alpha}-1}}{1 + \hat{h} t^{\hat{\alpha}}}, \\ \hat{S}_T(t) &= \frac{1}{1 + \hat{h} t^{\hat{\alpha}}}, \\ \hat{f}_T(t) &= \hat{h}_T(t) \hat{S}_T(t), \\ \hat{F}_T(t) &= 1 - \hat{S}_T(t).\end{aligned}$$

4.8.4 Programmation VIII : Algorithme de Newton-Raphson sans routine d'optimisation et avec hessien analytique

Le programme d'estimation est le suivant.

1.

```
proc iml;
  start lecture;
  nom_y={&y}; nom_d={&d};
  use &tab; read all var(nom_y) into yi;
  read all var(nom_d) into di;
  nd=sum(di); n=nrow(yi); nc=n-nd;
  print "Time variable =" nom_y,
    "Censoring variable (=0 if censored) =" nom_d,
    "Number of observations =" n,
    "Complete =" nd,
    "Censored =" nc;
  finish lecture;
  Même module de lecture que précédemment.
```
2.

```
start lv(theta) global(di,lyi,epsilon,yi,nd);
  t1=theta[1]; t2=theta[2];
  alpha=exp(t1); h=exp(t2);
  lv=nd*(t1+t2)+sum((alpha-1)#di#lyi
    -(1+di)#log(1+h#yi##alpha));
  return (lv);
  finish;
  Calcul de la log-vraisemblance.
```
3.

```
start sc(theta) global(di,lyi,epsilon,yi,nd);
  t1=theta[1]; t2=theta[2];
  alpha=exp(t1); h=exp(t2);
  z1=h#yi##alpha;
  zi=z1/(1+z1);
  der1=nd+alpha#sum(lyi#(di-(1+di)#zi));
  der2=nd-sum((1+di)#zi);
  der=(der1||der2);
  return (der);
  finish;
  Calcul du score.
```
4.

```
start hes(theta) global(di,lyi,epsilon,yi,nd);
  t1=theta[1]; t2=theta[2];
  alpha=exp(t1); h=exp(t2);
```

```

zi1=h#yi##alpha;
zi=zi1/(1+zi1);
zi2=zi*(1-zi);
h11=alpha#sum(lyi*(di-(1+di)#zi
-alpha*(1+di)#zi2#lyi));
h12=-alpha#sum((1+di)#zi2#lyi);
h22=-sum((1+di)#zi2);
he=(h11|h12)/(h12|h22);
return (he);
finish;
Calcul du hessien.

```

5. start estim;
Début du module d'estimation.
6. print "Log-logistic", , "S(t)=1/(1+h.t^alpha)", ,
"we also give S(t)=1-F((ln(t)-m)/delta)",
"where F is the CDF of the (0,1) Logistic
distribution",
"and sigma=delta.pi/sqrt(3) for comparison with
the log-normal distribution",;
On rappelle la forme de la fonction de survie utilisée et on précise la définition des autres paramètres qui seront estimés.
7. epsilon=1e-6; lyi=log(yi);
pi=arcs(-1);
Une manière de calculer le nombre $\pi = \cos^{-1}(-1)$.
8. mly=sum(lyi)/n; sly=sqrt(ssq(lyi)/n-mly##2);
Valeur initiale : on prend les moments empiriques correspondants sur les logarithmes des durées tronquées. Cette valeur initiale n'est pas convergente mais devrait fournir un ordre de grandeur acceptable par l'algorithme.
9. alpha0=pi/sly/sqrt(3); h0=exp(-alpha0#mly);
theta0=(log(alpha0)|log(h0));
lv0=lv(theta0);
print "Starting parameter =" theta0,
"Starting log-likelihood =" lv0, ,;
Calcul et affichage des paramètres initiaux θ_0 et de la log-vraisemblance associée.
10. maxc=&c; maxiter=&i; maxe=&e;
Paramètres du module d'optimisation.

11. labopt="Iteration" || "Log-likelihood" || "Max Abs Elasticity" || "Sub-Iteration";
Futures colonnes du tableau SAS en sortie.
12. theta=theta0;
Initialisation de l'algorithme.
13. do iter=1 to maxiter;
Début de l'algorithme de Newton-Raphson.
14. lambda=1;
sc=sc(theta); he=hes(theta);
step=sc*inv(he);
Calcul du pas de l'itération courante.
15. theta=theta0-lambda#step;
Point candidat de l'itération.
16. vp=max(eigval(he));
print "vp =" vp;
Calcul et affichage de la plus grande valeur propre du hessien ; elle doit être négative pour une itération sans problème.
17. lv1=lv(theta);
Valeur de la log-vraisemblance au point candidat.
18. if lv1 < lv0 then do;
Si le point candidat n'améliore pas la log-vraisemblance, on réduit le pas de l'itération.
19. do c=1 to maxc;
Boucle de réduction du pas de l'itération.
20. lambda=lambda/2;
theta=theta0-lambda#step;
lv1=lv(theta);
if lv1>=lv0 then c=maxc;
end;
Fin de la boucle de réduction du pas.
21. end;
Fin du *then do* qui lance la boucle de réduction du pas.
22. elas=max(abs(theta#sc/lv1));
Calcul du critère d'arrêt : la plus grande élasticité de la log-vraisemblance par rapport aux paramètres.
23. riter=(iter||lv1||elas||lambda);
print riter[colname=labopt];
Affichage des résultats de l'itération.

24. `theta0=theta; lv0=lv1;`
On lancera la prochaine boucle à partir du point candidat qui vient d'être accepté.
25. `if elas<&e then iter=maxiter;`
Si le critère d'arrêt est rempli, on arrête l'algorithme.
26. `end;`
Fin de la boucle d'itération par la méthode de Newton-Raphson.
27. `lv1=lv(theta); gr=t(sc(theta)); he=hes(theta);`
`theta=t(theta); elas=abs(theta#gr/lv1);`
`vp=eigval(he);`
`print "Log-likelihood at maximum =" lv1,`
`"Gradient =" gr,`
`"Elasticities =" elas,`
`"Hessian eigenvalues =" vp;`
Affichage des résultats globaux de la procédure d'optimisation. Le gradient doit être nul et les valeurs propres du hessien toutes négatives.
28. `vtheta=inv(-he);`
Estimation de la matrice de covariance asymptotique du paramètre d'optimisation $\widehat{\text{Vas}}(\hat{\theta})$.
29. `t1=theta[1]; t2=theta[2];`
`b=exp(theta);`
`alpha=b[1]; h=b[2];`
Calcul des paramètres d'intérêt $\alpha = \exp(\theta_1)$ et $h = \exp(\theta_2)$.
30. `db=(alpha||0)/(0||h);`
Matrice des dérivées utilisée pour appliquer le théorème de Slutsky.
31. `vb=db*vtheta*t(db);`
Calcul de la matrice de covariance du paramètre d'intérêt $\widehat{\text{Vas}}(\hat{\zeta})$.
32. `sb=sqrt(vecdiag(vb));`
`tb=b/sb;`
`pb=1-probchi(tb##2,1);`
Calcul des statistiques usuelles relatives au paramètre d'intérêt.
33. `b1=(1/alpha)/(-log(h)/alpha);`
Second paramétrage en (δ, m) habituellement associée à la loi log logistique.
34. `db1=((-1/alpha##2)||0)/`
`((-log(h)/alpha##2)||(-1/alpha/h));`
Matrice des dérivées utilisée pour appliquer le théorème de Slutsky.

35. `vb1=db1*vb*t(db1);`
`sb1=sqrt(vecdiag(vb1));`
`tb1=b1/sb1;`
`pb1=1-probchi(tb1##2,1);`
 Calcul des statistiques usuelles relatives à $\hat{\zeta}_1 = (\hat{\delta}, \hat{m})$.
36. `sigma=pi/alpha/sqrt(3);`
 Troisième paramétrage en (σ, m) habituellement associé à la loi log normale. Seule l'estimation de σ est nécessaire car celle de m est donnée par le paramétrage log logistique.
37. `dsig=-pi/alpha##2/sqrt(3);`
 Dérivées utilisée pour appliquer le théorème de Slutsky.
38. `vsig=dsig##2#vb[1,1];`
`vb[1,1]` contient $\widehat{Vas}(\hat{\alpha})$.
39. `ssig=sqrt(vsig);`
`tsig=sigma/ssig;`
`psig=1-probchi(tsig##2,1);`
 Calcul des statistiques usuelles relatives à $\hat{\sigma}$.
40. `resp=(lv1||.||.||.)/(b||sb||tb||pb)/(b1||sb1||tb1||pb1)/(sigma||ssig||tsig||psig);`
 Stockage de l'ensemble des résultats dans une matrice.
41. `if alpha>1 then do;`
 Si $\alpha > 1$ la moyenne existe et le hasard admet un maximum. On calcule donc des statistiques supplémentaires.
42. `mt=(1/h)##(1/alpha)#beta(1-1/alpha,1+1/alpha);`
`dmt=(mt#(log(h)/alpha##2+digamma(1-1/alpha)`
`+digamma(1+1/alpha))|(-mt/alpha/h);`
`vmt=dmt*vb*t(dmt); smt=sqrt(vmt); tmt=mt/smt;`
`pmt=1-probchi(tmt##2,1);`
 Estimation de la durée moyenne corrigée de la censure et des statistiques associées.
43. `resp=resp/(mt||smt||tmt||pmt);`
 On ajoute les statistiques relatives à la durée moyenne aux résultats.
44. `tmax=((alpha-1)/h)##(1/alpha);`
`hmax=alpha#h#tmax##(alpha-1)/(1+h#tmax##alpha);`
 Calcul du point auquel le hasard est maximum et de la valeur maximale du hasard.

```

45. print , "The hasard function reaches a maximum at
    T =" tmax, "Its maximum value is h =" hmax,;
    Affichage des résultats sur le hasard.

46. end;
    Fin de la condition  $\alpha > 1$ .

47. else do;
    Début de la condition  $\alpha \leq 1$ . La distribution n'admet pas de moyenne et
    le hasard n'admet pas de maximum intérieur.

48. print "The MEAN does not exist since alpha<=1.";
    resp=resp//(.||.||.);
    On affiche un message d'alerte et la matrice des résultats est complétée
    par des valeurs manquantes.

49. end;
    Fin des instructions conditionnelles sur la valeur de  $\alpha$ .

50. md=(1/h)##(1/alpha);
    dmd=(md#log(h)/alpha##2)||(-md/alpha/h);
    vmd=dmd*vb*t(dmd); smd=sqrt(vmd); tmd=md/smd;
    pmd=1-probchi(tmd##2,1);
    Estimation de la médiane et des statistiques associées.

51. resp=resp//(md||smd||tmd||pmd);
    Ajout des résultats sur la médiane à la matrice des résultats.

52. labc="Params"||"Std_Dev"||"Student"||"Pvalue";
    Futurs noms de colonne du tableau SAS en sortie.

53. labr="Log-likelihood"//"alpha"//"h"//"delta"//
    "m"//"sigma"//"Mean"//"Median";
    Futurs noms de ligne du tableau SAS en sortie.

54. print resp[colname=labc rowname=labr];
    Affichage des résultats d'estimation.

55. test_lom=abs(alpha-1)/sb[1];
    p_lom=1-probchi(test_lom##2,1);
    print "Test H0 : alpha=1 (one parameter Lomax
    distribution) " test_lom, "p-value =" p_lom;
    Test de l'hypothèse nulle  $\alpha = 1$  qui correspond à la loi de Lomax à un
    paramètre et affichage des statistiques associées à ce test.

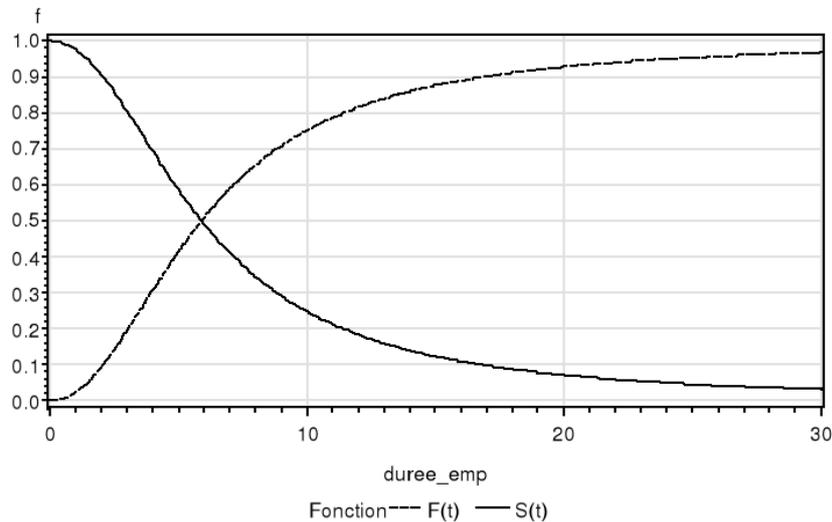
56. finish estim;
    Fin du module d'estimation.

```

- ```
57. start prev;
 Module de prévision. Sa structure est identique aux modules précédents,
 seules les valeurs des fonctions changent. tmin=min(yi); tmax=max(yi);
 tby=1;
 do time=tmin to tmax by tby;
 zi=1+h#time##alpha;
 hi=alpha#h#time##(alpha-1)/zi;
 si=1/zi;
 fi=hi#si;
 ci=1-si;
 resi=resi/(time||fi||si||hi||ci);
 end;
 labi="&y"||"Density"||"SDF"||"Hasard"||"CDF";
 finish prev;
 Fin du module de prévision.
```
- ```
58. start sortie;
    create &outp from resp[colname=labc];
    append from resp;
    create _np from labr[colname="Estimate"];
    append from labr;
    create &outi from resi[colname=labi];
    append from resi;
    finish sortie;
    Module de sortie identique aux précédents.
```
- ```
59. run lecture;
 run estim;
 run prev;
 run sortie;
 quit;
 Lancement des modules et sortie d'IML.
```
- ```
60. data &outp; merge _np &outp;
    run;
    Ajout des noms de lignes à la matrice des résultats et lancement de l'en-
    semble des instructions.
```

4.8.5 Application VIII

Nous estimons maintenant le modèle log-logistique sur la durée du premier emploi. Par rapport au modèle log-normal, le modèle log-logistique n'im-

GRAPHIQUE 4.16 – LOI LOG-LOGISTIQUE, $\hat{S}_T(t)$ ET $\hat{F}_T(t)$

pose pas la présence d'un maximum. Nous en trouvons cependant un, en $T = 6.2$ ans, une valeur un peu plus élevée que pour la loi log-normale (5.7 ans). La log-vraisemblance maximale du modèle log-logistique est toutefois un peu plus faible que celle du modèle log-normal (-34286.22).

Sortie SAS 4.9.

```

Log-logistic

S(t)=1/(1+h.t^alpha)

we also give S(t)=1-F((ln(t)-m)/delta)
where F is the CDF of the (0,1) Logistic distribution
and sigma=delta.pi/sqrt(3) for comparison with the log-normal
distribution

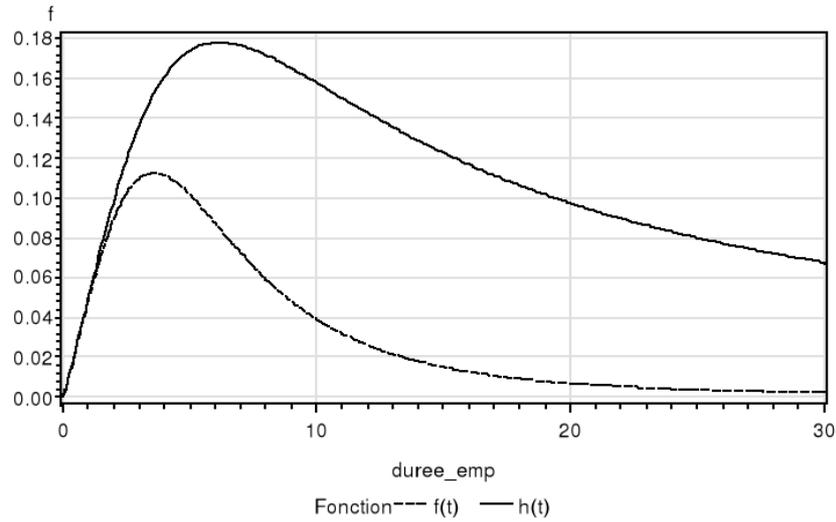
theta0

Starting parameter = 0.8225847 -3.96258
                    lv0

Starting log-likelihood = -34349.4

vp

```



GRAPHIQUE 4.17 – LOI LOG-LOGISTIQUE, $\hat{h}_T(t)$ ET $\hat{f}_T(t)$

vp = -931.5842

| riter | | | | |
|-------|-----------|----------------|--------------------|---------------|
| | Iteration | Log-likelihood | Max Abs Elasticity | Sub-Iteration |
| ROW1 | 1 | -34286.78 | 0.0518848 | 1 |

vp

vp = -932.9691

| riter | | | | |
|-------|-----------|----------------|--------------------|---------------|
| | Iteration | Log-likelihood | Max Abs Elasticity | Sub-Iteration |
| ROW1 | 2 | -34286.22 | 0.0050506 | 1 |

vp

vp = -931.0493

| riter | | | | |
|-------|-----------|----------------|--------------------|---------------|
| | Iteration | Log-likelihood | Max Abs Elasticity | Sub-Iteration |
| ROW1 | 3 | -34286.22 | 0.0000489 | 1 |

vp

vp = -931.0382

riter

Iteration Log-likelihood Max Abs Elasticity Sub-Iteration

| | | | | |
|------|---|-----------|-----------|-----------|
| ROW1 | 4 | -34286.22 | 5.3105E-9 | 0.0009766 |
|------|---|-----------|-----------|-----------|

lvl

Log-likelihood at maximum = -34286.22

gr

Gradient = -0.000245

-0.000033

elas

Elasticities = 5.3053E-9

3.5551E-9

vp

Hessian eigenvalues = -931.0382

-71813.9

tmax

The hazard function reaches a maximum at T = 6.1747984

hmax

Its maximum value is h = 0.1782061

resp

| | Params | Std_Dev | Student | Pvalue |
|----------------|-----------|-----------|-----------|--------|
| Log-likelihood | -34286.22 | . | . | . |
| alpha | 2.1003866 | 0.016399 | 128.08022 | 0 |
| h | 0.0240399 | 0.0007704 | 31.203777 | 0 |
| delta | 0.4761028 | 0.0037172 | 128.08022 | 0 |
| m | 1.7749315 | 0.0281741 | 62.998794 | 0 |
| sigma | 0.863555 | 0.0067423 | 128.08022 | 0 |
| Mean | 8.8494982 | 0.2792337 | 31.692086 | 0 |
| Median | 5.8998769 | 0.0441166 | 133.73373 | 0 |

test_lom

Test H0 : alpha=1 (one parameter Lomax distribution) 67.100863

p_lom

p-value = 0

On obtient les estimations suivantes des paramètres :

$$(\hat{a}, \hat{h}) = (2.1, 0.024),$$

qui permettent de tester le cas particulier du modèle de Lomax ($\alpha = 1$). La version asymptotique du test de Student donne $\hat{t} = 67.1$ de sorte que l'on rejette l'hypothèse nulle aux seuils usuels (valeur critique pour un test de niveau 95% : 1.96). Le paramétrage originel de loi logistique mène aux estimations :

$$(\hat{m}, \hat{\delta}) = (1.77, 0.48),$$

mais il n'est pas très pratique parce que δ n'est pas l'écart-type de la loi logistique. Pour comparer cette loi à la loi log-normale, il faut utiliser le troisième paramétrage :

$$(\hat{m}, \hat{\sigma}) = (1.77, 0.86),$$

qui donne des estimations très proches de celles obtenues avec une loi log-normale (1.82, 0.83). La durée moyenne du premier emploi est estimée à 8.7 ans et la durée médiane à 6.2 ans (contre respectivement 8.8 ans et 5.9 ans avec la loi log-normale).

4.9 Loi de Pareto

4.9.1 Log-vraisemblance

Les logarithmes du hasard et de la survie sont donnée par :

$$\begin{aligned} \ln h_T(y_i) &= \ln a + \ln h - \ln(a + hy_i), \\ \ln S_T(y_i) &= a \ln a - a \ln(a + hy_i) \end{aligned}$$

d'où la log-vraisemblance :

$$\ell(\zeta) = \sum_{i=1}^N d_i \ln h + (d_i + a) (\ln a - \ln(a + hy_i))$$

avec :

$$\zeta = \begin{pmatrix} a \\ h \end{pmatrix}.$$

Les dérivées premières individuelles, que nous utiliserons pour l'algorithme de Berndt-Hall-Hausman (1974), sont données par :

$$\begin{aligned} \frac{\partial \ell_i}{\partial a} &= \ln \frac{a}{a + hy_i} + \frac{h(d_i + a)y_i}{a(a + hy_i)} \\ \frac{\partial \ell_i}{\partial h} &= \frac{d_i}{h} - \frac{(d_i + a)y_i}{a + hy_i} \end{aligned}$$

4.9.2 Estimation

On effectue le changement de paramètres :

$$a = \exp(\theta_1), h = \exp(\theta_2), \theta = \begin{pmatrix} \theta_1 \\ \theta_2 \end{pmatrix}$$

avec :

$$\frac{\partial a}{\partial \theta_1} = a, \frac{\partial h}{\partial \theta_2} = h.$$

d'où la log-vraisemblance :

$$\ell_{\zeta}(\theta) = \ell(\zeta(\theta))$$

et les dérivées premières dont on a besoin pour l'algorithme :

$$\begin{aligned} \frac{\partial \ell_{\zeta i}}{\partial \theta_1} &= a \frac{\partial \ell_i}{\partial a} = a \ln \left(\frac{a}{a + h y_i} \right) + \frac{(d_i + a) h y_i}{a + h y_i} \\ \frac{\partial \ell_{\zeta i}}{\partial \theta_2} &= h \frac{\partial \ell_i}{\partial h} = d_i - \frac{(d_i + a) h y_i}{a + h y_i} \end{aligned}$$

Pour l'algorithme de BHHH, on utilisera donc :

$$s_{\zeta}(\theta) = \frac{\partial \ell_{\zeta}}{\partial \theta} = \sum_{i=1}^N \frac{\partial \ell_{\zeta i}}{\partial \theta}$$

et

$$H_{\zeta}(\theta) = - \sum_{i=1}^N \frac{\partial \ell_{\zeta i}}{\partial \theta} \frac{\partial \ell_{\zeta i}}{\partial \theta'}$$

qui approxime le hessien (auquel cette matrice est égale en espérance). On retrouve les estimations des paramètres d'intérêt par les formules :

$$\hat{a} = \exp(\hat{\theta}_1), \hat{h} = \exp(\hat{\theta}_2),$$

et leur matrice de covariance par le théorème de Slutsky.

4.9.3 Statistiques utiles

L'espérance mathématique existe quand $a > 1$, elle est alors égale à :

$$E(T) = g(\zeta) = \frac{a}{(a-1)h}$$

donc :

$$\begin{aligned}\frac{\partial g}{\partial a} &= -\frac{1}{(a-1)^2 h}, \\ \frac{\partial g}{\partial h} &= -\frac{a}{(a-1)h^2} = -\frac{E(T)}{h}\end{aligned}$$

et on applique le théorème de Slutsky. Toutefois, il faut se méfier des valeurs de a proches de 1 car les expressions précédentes tendent toutes vers l'infini dans ce cas. La médiane ne pose pas ce problème :

$$\text{Me}(T) = g(\zeta) = \frac{a}{h}(2^{1/a} - 1),$$

avec :

$$\begin{aligned}\frac{\partial g}{\partial a} &= \frac{1}{h} \left(2^{1/a} \left(1 - \frac{\ln 2}{a} \right) - 1 \right), \\ \frac{\partial g}{\partial h} &= -\frac{a}{h^2} (2^{1/a} - 1) = -\frac{\text{Me}(T)}{h}.\end{aligned}$$

Les fonctions qui caractérisent la distribution sont données par :

$$\begin{aligned}\hat{S}_T(t) &= \left(\frac{\hat{a}}{\hat{a} + \hat{h}t} \right)^{\hat{a}}, \\ \hat{h}_T(t) &= \frac{\hat{a}\hat{h}}{\hat{a} + \hat{h}t}, \\ \hat{f}_T(t) &= \hat{h}_T(t)\hat{S}_T(t), \\ \hat{F}_T(t) &= 1 - \hat{S}_T(t)\end{aligned}$$

4.9.4 Programmation IX : Estimation par la méthode de Berndt Hall Hausman

Voici le programme d'estimation.

```
1. proc iml;
  start lecture;
  nom_y={&y}; nom_d={&d};
  use &tab; read all var(nom_y) into yi;
  read all var(nom_d) into di;
  nd=sum(di); n=nrow(yi); nc=n-nd;
  print "Time variable =" nom_y,
  "Censoring variable (=0 if censored) =" nom_d,
```

```
"Number of observations =" n,
"Complete =" nd, "Censored =" nc;
finish lecture;
```

Même module de lecture que les précédents.

```
2. start lv(theta) global(di,epsilon,yi,nd);
t1=theta[1]; t2=theta[2];
a=exp(t1); h=exp(t2);
lv=sum(di#t2+(di+a)#(t1-log(a+h#yi)));
return (lv);
finish;
```

Calcul de la log-vraisemblance.

```
3. start sc(theta) global(di,epsilon,yi,b3h);
t1=theta[1]; t2=theta[2];
a=exp(t1); h=exp(t2);
hyi=h#yi;
zi=(di+a)#hyi/(a+hyi);
der1=a#log(a/(a+hyi))+zi;
der2=di-zi;
deri=(der1 | der2);
der=deri[+,];
b3h=-t(deri)*deri;
return (der);
finish;
```

Calcul du gradient et de l'approximation du hessien par la méthode de Berndt-Hall-Hall-Hausman. Pour faire ce dernier calcul, on utilise une expression matricielle qui permet de gagner du temps par rapport à une boucle. On pose :

$$\underset{(N,2)}{\text{deri}} = \begin{pmatrix} \frac{\partial \ell_1}{\partial \theta'}(\hat{\theta}) \\ \vdots \\ \frac{\partial \ell_i}{\partial \theta'}(\hat{\theta}) \\ \vdots \\ \frac{\partial \ell_N}{\partial \theta'}(\hat{\theta}) \end{pmatrix}$$

de sorte que :

$$-\underset{(2,N)}{\text{deri}}' \underset{(N,2)}{\text{deri}} = -\sum_{i=1}^N \frac{\partial \ell_i}{\partial \theta'}(\hat{\theta})' \frac{\partial \ell_i}{\partial \theta'}(\hat{\theta}) = H_{\zeta}(\hat{\theta}),$$

d'autre part l'expression $\text{deri}[\text{+}, \text{,}]$ additionne la matrice der sur les lignes, ce qui donne le score exprimé en vecteur ligne :

$$\text{deri}[\text{+}, \text{,}]_{(1,2)} = \sum_{i=1}^N \frac{\partial \ell_i}{\partial \theta'}(\hat{\theta}),$$

le vecteur du score est renvoyé vers l'appel de la fonction et la matrice $H_{\zeta}(\hat{\theta})$ est rendue disponible dans tout le programme après l'appel de la fonction via l'option *global*.

4. `start estim;`
Début du module d'estimation.
5. `print "Estimation of the parameters of Log-
logistic distribution", , ,
"S(t)=(a/(a+h.t))^a", , , ;`
Rappel du paramétrage de la fonction de survie utilisée.
6. `epsilon=1e-6;`
Critère d'arrêt de l'algorithme.
7. `h0=sum(yi)/nd; a0=1;
theta0=log(a0 | |h0);
lv0=lv(theta0);
print "Starting parameter =" theta0,
"Starting log-likelihood =" lv0, , ;`
Calcul des valeurs initiales et affichage de la log-vraisemblance au point initial.
8. `maxc=&c; maxiter=&i; maxe=&e;`
Paramètre du module d'optimisation.
9. `labopt="Iteration" | | "Log-likelihood" | | "Max Abs
Elasticity" | | "Sub-Iteration";`
Noms de colonne de l'affichage des résultats intermédiaires de l'algorithme.
10. `theta=theta0;`
Initialisation de l'algorithme.
11. `do iter=1 to maxiter;`
Début de l'algorithme.
12. `lambda=1;
sc=sc(theta);
step=sc*inv(b3h);`
Calcul du pas de l'algorithme BHHH. La matrice `b3h` mise à jour a été

rendue disponible dès que la fonction `sc` a été appelée à la ligne précédente.

13. `theta=theta0-lambda#step;`
Point candidat de l'itération.
14. `lv1=lv(theta);`
`if lv1 < lv0 then do;`
On vérifie que le point candidat augmente la vraisemblance.
15. `do c=1 to maxc;`
Si le point candidat n'améliore pas la vraisemblance, on réduit la taille du pas.
16. `lambda=lambda/2;`
`theta=theta0-lambda#step;`
`lv1=lv(theta);`
`if lv1 >= lv0 then c=maxc;`
17. Si le point candidat modifié permet d'augmenter la vraisemblance, on arrête la boucle.
18. `end;`
Fin de la boucle de réduction du pas.
19. `end;`
Fin du *then do* qui lance la boucle de réduction du pas.
20. `elas=max(abs(theta#sc/lv1));`
`riter=(iter||lv1||elas||lambda);`
`print riter[colname=labopt];`
Affichage des résultats de l'itération en cours.
21. `theta0=theta; lv0=lv1;`
Préparation de la prochaine boucle.
22. `if elas<&e then iter=maxiter;`
Si le gradient est suffisamment proche du vecteur nul, on arrête l'algorithme.
23. `end;`
Fin de la boucle de l'algorithme de BHHH.
24. `call nlpfdd(lv1,gr,he,"lv",theta) grd="sc";`
Calcul de la matrice hessienne au point de convergence de l'algorithme, en utilisant le gradient analytique pour gagner du temps.

25. `theta=t(theta); gr=t(gr); elas=abs(theta#gr/lv1);`
On remet les vecteurs en colonne et on calcule les élasticités de la log-vraisemblance par rapport aux paramètres.
26. `vp=eigval(he);`
Calcul des valeurs propres du hessien.
27. `print "Log-likelihood at maximum =" lv1,`
`"Gradient =" gr,`
`"Elasticities =" elas,`
`"Hessian eigenvalues =" vp;`
Affichage des résultats de l'algorithme.
28. `vtheta=inv(-he);`
Calcul de la matrice de covariance asymptotique de l'estimateur du paramètre d'optimisation, à partir des dérivées secondes numériques :

$$\widehat{\text{Var}}(\hat{\theta}) = \left(- \sum_{i=1}^N \frac{\partial^2 \ell}{\partial \theta \partial \theta'}(\hat{\theta}) \right)^{-1}.$$

29. `t1=theta[1]; t2=theta[2]; b=exp(theta);`
Calcul du paramètre d'intérêt $a = \exp(\theta_1)$ et $h = \exp(\theta_2)$.
30. `a=b[1]; h=b[2];`
`db=(a|0)/(0|h);`
`vb=db*vtheta*t(db);`
`sb=sqrt(vecdiag(vb));`
`tb=b/sb;`
`pb=1-probchi(tb##2,1);`
`resp=(b|sb|tb|pb);`
Calcul des statistiques associées au paramètre d'intérêt $\zeta = (a, h)'$ et stockage dans une matrice.
31. `if a>1 then do;`
Si $a > 1$ la moyenne existe.
32. `mt=a/(a-1)/h;`
`dmt=(-1/(a-1)##2/h)|(-mt/h);`
`vmt=dmt*vb*t(dmt);`
`smt=sqrt(vmt);`
`tmt=mt/smt;`
`pmt=1-probchi(tmt##2,1);`
`resp=resp/(mt|smt|tmt|pmt);`
Calcul des statistiques associées à la moyenne et ajout à la matrice des résultats.

33. end;
Fin de la condition $a > 1$.
34. else do;
Début de la condition $a \leq 1$, la moyenne n'existe pas.
35. print "The distribution has NO mean since a<=1";
resp=resp/(. || . || . || . || .);
On affiche un message d'avertissement et on remplit la matrice des résultats avec une ligne de valeurs manquantes.
36. end;
Fin de la conditions $a \leq 1$.
37. md=a#(2##(1/a)-1)/h;
dmd=((2##(1/a)#(1-log(2)/a)-1)/h) || (-md/h);
vmd=dmd*vb*t(dmd);
smd=sqrt(vmd);
tmd=md/smd;
pmd=1-probchi(tmd##2,1);
Estimation de la médiane et calcul des statistiques associées.
38. resp=(lv1 || . || . || . || .)//resp/(md || smd || tmd || pmd);
On ajoute la log-vraisemblance et les résultats sur la médiane à la matrice globale des résultats.
39. labc="Params" || "Std_Dev" || "Student" || "Pvalue";
Futurs noms de colonne du tableau SAS en sortie.
40. labr="Log-likelihood"//"a"//"h"//"Mean"//"Median";
Futurs noms de ligne du tableau SAS en sortie.
41. print resp[colname=labc rowname=labr];
Affichage de l'ensemble des résultats.
42. test_lom=abs(a-1)/sb[1];
p_lom=1-probchi(test_lom##2,1);
print "Test H0 : a=1 (one parameter Lomax distribution)" test_lom,
"p-value =" p_lom;
Test de l'hypothèse nulle $\alpha = 1$, qui correspond à la loi de Lomax à un paramètre, et affichage des statistiques associées.
43. finish estim;
Fin du module d'estimation.
44. start prev;
Module de prévision. Sa structure est identique aux modules précédents, à l'exception des valeurs des fonctions.

- ```

45. tmin=min(yi); tmax=max(yi); tby=1;
 do time=tmin to tmax by tby;
 zi=a/(a+h#time);
 si=zi##a;
 hi=h#zi;
 fi=hi#si;
 ci=1-si;
 resi=resi//((time||fi||si||hi||ci));
 end;
 labi="&y"||"Density"||"SDF"||"Hasard"||"CDF";
 finish prev;
 Fin du module de prévision.

46. start sortie;
 create &outp from resp[colname=labc];
 append from resp;
 create _np from labr[colname="Estimate"];
 append from labr;
 create &outi from resi[colname=labi];
 append from resi;
 finish sortie;
 Module de sortie identique aux précédents.

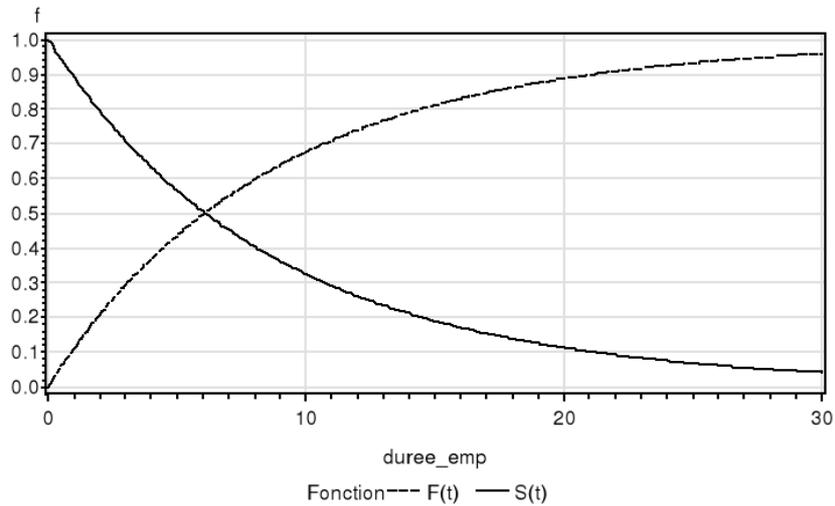
47. run lecture;
 run estim;
 run prev;
 run sortie;
 quit;
 Lancement des modules et sortie de SAS-IML.

48. data &outp; merge _np &outp;
 run;
 Ajout des noms des lignes à la matrice des résultats et lancement du programme.

```

#### 4.9.5 Application IX

Nous reprenons les données sur la durée du premier emploi afin d'estimer les paramètres de la loi de Pareto. On trouve un hasard décroissant, imposé par la loi, de sorte que la probabilité de sortie du premier emploi diminuerait avec sa durée. Au vu des estimations qui précèdent, il faut être prudent sur ce point, d'autant plus que la log-vraisemblance maximale obtenue avec cette loi est la

GRAPHIQUE 4.18 – LOI DE PARETO,  $\hat{S}_T(t)$  ET  $\hat{F}_T(t)$ 

plus faible de celle obtenue pour les modèles à deux paramètres ( $-36074.62$ ).

#### Sortie SAS 4.10.

Estimation of the parameters of the Pareto distribution

$$S(t) = (a / (a + h \cdot t))^a$$

theta0

Starting parameter = 0 2.2004609

lv0

Starting log-likelihood = -69801.39

riter

|      | Iteration | Log-likelihood | Max Abs | Elasticity | Sub-Iteration |
|------|-----------|----------------|---------|------------|---------------|
| ROW1 | 1         | -57167.31      |         | 0.2249737  | 1             |

riter

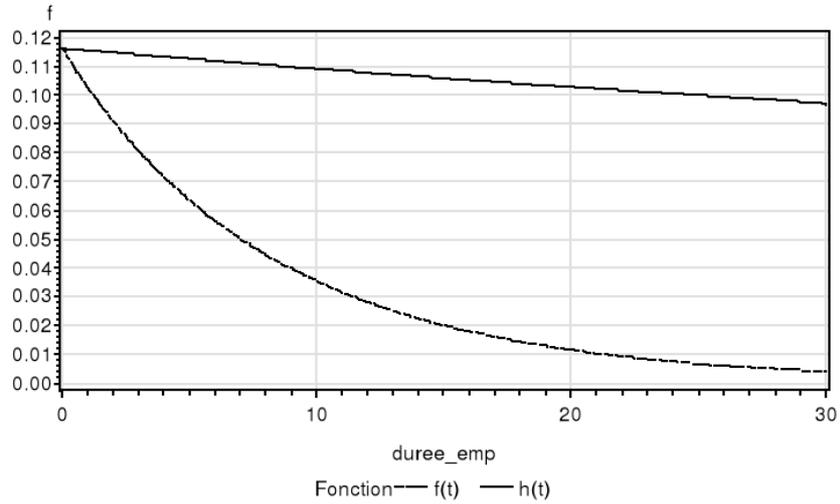
|      | Iteration | Log-likelihood | Max Abs | Elasticity | Sub-Iteration |
|------|-----------|----------------|---------|------------|---------------|
| ROW1 | 2         | -45011.42      |         | 0.0553659  | 1             |

riter

|      | Iteration | Log-likelihood | Max Abs | Elasticity | Sub-Iteration |
|------|-----------|----------------|---------|------------|---------------|
| ROW1 | 3         | -36729.83      |         | 0.4336092  | 1             |

riter

|      | Iteration | Log-likelihood | Max Abs | Elasticity | Sub-Iteration |
|------|-----------|----------------|---------|------------|---------------|
| ROW1 | 4         | -36351.9       |         | 0.0654319  | 0.5           |



GRAPHIQUE 4.19 – LOI DE PARETO,  $\hat{h}_T(t)$  ET  $\hat{f}_T(t)$

| riter |           |                |     |     |            |               |
|-------|-----------|----------------|-----|-----|------------|---------------|
|       | Iteration | Log-likelihood | Max | Abs | Elasticity | Sub-Iteration |
| ROW1  | 5         | -36079.08      |     |     | 0.1135681  | 1             |
| riter |           |                |     |     |            |               |
|       | Iteration | Log-likelihood | Max | Abs | Elasticity | Sub-Iteration |
| ROW1  | 6         | -36075.41      |     |     | 0.0180251  | 1             |
| riter |           |                |     |     |            |               |
|       | Iteration | Log-likelihood | Max | Abs | Elasticity | Sub-Iteration |
| ROW1  | 7         | -36074.64      |     |     | 0.0013062  | 0.5           |
| riter |           |                |     |     |            |               |
|       | Iteration | Log-likelihood | Max | Abs | Elasticity | Sub-Iteration |
| ROW1  | 8         | -36074.62      |     |     | 0.0008531  | 0.5           |
| riter |           |                |     |     |            |               |
|       | Iteration | Log-likelihood | Max | Abs | Elasticity | Sub-Iteration |
| ROW1  | 9         | -36074.62      |     |     | 0.0001687  | 1             |
| riter |           |                |     |     |            |               |
|       | Iteration | Log-likelihood | Max | Abs | Elasticity | Sub-Iteration |
| ROW1  | 10        | -36074.62      |     |     | 0.0000773  | 0.5           |
| riter |           |                |     |     |            |               |
|       | Iteration | Log-likelihood | Max | Abs | Elasticity | Sub-Iteration |
| ROW1  | 11        | -36074.62      |     |     | 0.0000131  | 1             |
| riter |           |                |     |     |            |               |
|       | Iteration | Log-likelihood | Max | Abs | Elasticity | Sub-Iteration |
|       |           |                |     |     |            |               |

```
ROW1 12 -36074.62 7.3697E-6 0.5
```

```

 riter
Iteration Log-likelihood Max Abs Elasticity Sub-Iteration
ROW1 13 -36074.62 1.0371E-6 1
```

```

 riter
Iteration Log-likelihood Max Abs Elasticity Sub-Iteration
ROW1 14 -36074.62 7.1175E-7 0.5
```

```

 lvl
Log-likelihood at maximum = -36074.62
```

```

 gr
Gradient = -0.000194
 -0.001347
 elas
Elasticities = 1.5468E-8
 8.0354E-8
```

```

 vp
Hessian eigenvalues = -31.00263
 -10340.41
```

```

 resp
 Params Std_Dev Student Pvalue
Log-likelihood -36074.62 . . .
a 17.752827 3.1852933 5.5733728 2.4985E-8
h 0.1162039 0.0014646 79.339294 0
Mean 9.1192403 0.0934219 97.613468 0
Median 6.0828997 0.0654545 92.933264 0
```

```
test_lom
```

```
Test H0 : a=1 (one parameter Lomax distribution) 5.25943
```

```

 p_lom
p-value = 1.445E-7
```

On trouve les estimations suivantes pour les paramètres :

$$(\hat{a}, \hat{h}) = (17.8, 0.12),$$

et la valeur élevée pour  $\hat{a}$  fait penser à la loi exponentielle. En effet quand  $a \rightarrow +\infty$ , la loi de Pareto correspond exactement à la loi exponentielle. Ce sentiment est renforcé par une comparaison des log-vraisemblances maximales :  $-36091.6$  pour la loi exponentielle contre  $-36074.6$  pour la loi de Pareto. Cette généralisation ne semble qu'améliorer faiblement les prévisions de la loi exponentielle. Nous pourrions tester l'adéquation de cette loi grâce à la loi de Burr que nous verrons plus loin. Toutefois, à ce stade, la balance penche plutôt en faveur de la loi log-logistique. La durée moyenne du premier emploi est estimée à 9.1 ans et la durée médiane à 6.1 ans.

On peut également tester la validité de la loi de Lomax à un paramètre ( $a = 1$ ). Le  $t$  de Student asymptotique de 5.2 amène à rejeter cette hypothèse nulle aux seuils usuels (valeur critique au seuil de 5% : 1.96).

## 4.10 Loi Gamma généralisée

Cette loi comporte trois paramètres. Nous utiliserons la réécriture de Prentice (1974), qui sera utile pour  $q > 0$ . Si on obtenait une valeur de  $q$  très proche de 0, il faudrait utiliser la distribution log-normale. Les logarithmes de la densité et de la survie sont donnés par :

$$\begin{aligned}\ln f_T(y_i) &= \ln q + q^{-2}(\psi_0 + qu_i) - \exp(\psi_0 + qu_i) - \ln \sigma - \ln y_i - \ln \Gamma(q^{-2}) \\ \ln S_T(y_i) &= \ln(1 - \gamma(q^{-2}, \exp(\psi_0 + qu_i)))\end{aligned}$$

avec :

$$u_i = \frac{\ln y_i - m}{\sigma}, \quad \psi_0 = \psi^{(0)}(q^{-2}),$$

où  $\psi^{(0)}(\cdot)$  est la fonction digamma. D'où la vraisemblance :

$$\ell(\zeta) = \sum_{i=1}^N d_i \ln f_T(y_i) + (1 - d_i) \ln(1 - \gamma(q^{-2}, \exp(\psi_0 + qu_i))).$$

Pour effectuer cette optimisation, on procède en deux étapes. Dans une première étape, on estime  $q$  par balayage puis, dans une seconde étape, on estime simultanément  $\zeta = (q, \sigma, m)'$  par un algorithme de Newton-Raphson.

### 4.10.1 Estimation

Pour l'estimation de ce modèle nous utiliserons un algorithme en deux parties. Une première partie réalisera une estimation par balayage du paramètre  $q$  tout en appliquant un algorithme de Newton-Raphson sur les paramètres  $(m, \sigma)$ . Une seconde partie réalisera une estimation de l'ensemble du paramètre par la méthode de Newton-Raphson en prenant le résultat de la première partie comme valeur initiale. Les étapes de l'estimation sont les suivantes :

1. Pour fixer la grille de balayage, on se ramène à l'intervalle  $]0, 1[$  en prenant la transformation :

$$B = \frac{q}{1 + q},$$

qui vérifie :

$$\lim_{q \rightarrow 0} B = 0, \quad \lim_{q \rightarrow +\infty} B = 1.$$

On réalise les estimations pour  $B^- = 0.01$  à  $B^+ = 0.99$  en utilisant :

$$q = \frac{B}{1-B},$$

ceci permet de couvrir l'intervalle  $[0.01, 99]$  pour  $\beta$ . Au besoin, on peut élargir les bornes si la log-vraisemblance semble présenter une solution en coin. Par exemple, on peut fixer  $B^- = 0.001$  et  $B^+ = 0.999$  en gardant un pas suffisamment élevé pour que le calcul ne soit pas trop long. On peut aussi prendre une grille très fine une fois la zone du maximum global trouvée.

2. On fixe une grille de valeurs de  $q$  et l'on calcule :

$$(\sigma(q), m(q)) = \arg \max_{(\sigma, m)} \ell(q, \sigma, m),$$

3. On choisit une valeur  $\beta^*$  telle que :

$$q^* = \arg \max_{\beta} \ell^*(q, \sigma(q), m(q))$$

4. On prend le point  $\zeta_0 = (q^*, \sigma(q^*), h(q^*))$  comme valeur initiale d'un algorithme de Newton-Raphson à trois paramètres  $\zeta = (q, \sigma, m)$ . Dans la mesure où le point  $\zeta_0$  correspond à un gradient proche du vecteur nul, il y a très peu d'itérations à effectuer. On peut alors utiliser les routines classiques pour déterminer les statistiques usuelles.

#### 4.10.2 Statistiques utiles

On peut commencer par réécrire les paramètres du modèle :

$$\beta = q^{-2}, \quad \alpha = \frac{q}{\sigma} \quad \text{et} \quad h = \exp\left(\psi^{(0)}(q^{-2}) - \frac{q}{\sigma} m\right)$$

et l'on utilisera les dérivées suivantes pour calculer la nouvelle matrice de covariance asymptotique :

$$\frac{\partial \beta}{\partial q} = -2q^{-3}, \quad \frac{\partial \beta}{\partial \sigma} = 0, \quad \frac{\partial \beta}{\partial m} = 0$$

$$\frac{\partial \alpha}{\partial q} = \frac{1}{\sigma}, \quad \frac{\partial \alpha}{\partial \sigma} = -\frac{q}{\sigma^2}, \quad \frac{\partial \alpha}{\partial m} = 0$$

$$\frac{\partial h}{\partial q} = -\left(2q^{-3}\psi^{(1)}(q^{-2}) + \frac{m}{\sigma}\right) \times h$$

où  $\psi^{(1)}(\cdot)$  est la fonction trigamma (annexe A.4.3),

$$\frac{\partial h}{\partial \sigma} = \frac{qm}{\sigma^2} \times h,$$

et

$$\frac{\partial h}{\partial m} = -\frac{q}{\sigma} \times h.$$

On forme ensuite la matrice :

$$\frac{\partial g(\zeta)}{\partial \zeta'} = \begin{pmatrix} -2q^{-3} & 0 & 0 \\ \sigma^{-1} & -q\sigma^{-2} & 0 \\ -(2q^{-3}\psi^{(1)}(q^{-2}) + m\sigma^{-1}) \times h & qm\sigma^{-2} \times h & -q\sigma^{-1} \times h \end{pmatrix}$$

et on utilisera :

$$\hat{\zeta}_1 = \begin{pmatrix} \hat{\beta} \\ \hat{\alpha} \\ \hat{h} \end{pmatrix} = g(\hat{\zeta})$$

$$\widehat{\text{Vas}}(\hat{\zeta}_1) = \frac{\partial g(\hat{\zeta})}{\partial \zeta'} \widehat{\text{Vas}}(\hat{\zeta}) \left( \frac{\partial g(\hat{\zeta})}{\partial \zeta'} \right)'.$$

La durée moyenne est donnée par :

$$E(T) = \frac{\Gamma(1/\alpha + \beta)}{h^{1/\alpha} \Gamma(\beta)} = \frac{\Gamma(1/\alpha)}{h^{1/\alpha} B(1/\alpha, \beta)},$$

et on l'estime en remplaçant les paramètres par leurs estimations. Pour la variance asymptotique on utilise le théorème de Slutsky avec :

$$\frac{\partial g}{\partial \beta} = E(T) (\psi^{(0)}(1/\alpha + \beta) - \psi^{(0)}(\beta))$$

$$\frac{\partial g}{\partial \alpha} = \frac{E(T)}{\alpha^2} (\ln h - \psi^{(0)}(1/\alpha + \beta))$$

$$\frac{\partial g}{\partial h} = -\frac{E(T)}{\alpha h}$$

et la médiane est égale à :

$$\text{Me}(T) = \left[ \frac{1}{2h} \times F_{\chi^2, 2\beta}^{-1} \left( \frac{1}{2} \right) \right]^{1/\alpha}.$$

On peut également réaliser les tests de Student asymptotiques pour les distributions de Weibull ( $q = 1, \hat{t}_W$ ) et Gamma à deux paramètres ( $q = \sigma, \hat{t}_{G2}$ ), qui suivent asymptotiquement une loi normale  $N(0, 1)$  :

$$\hat{t}_W = \frac{|\hat{q} - 1|}{\sqrt{\widehat{\text{Vas}}(\hat{q})}},$$

$$\hat{t}_{G2} = \frac{|\hat{q} - \hat{\sigma}|}{\sqrt{\widehat{\text{Vas}}(\hat{q}) + \widehat{\text{Vas}}(\hat{\sigma}) - 2\widehat{\text{Covas}}(\hat{q}, \hat{\sigma})}}.$$

Pour écrire le programme, il faudra également faire attention au cas  $\beta \rightarrow +\infty$  ( $q \rightarrow 0$ ). Pour cela, il faut fixer une valeur seuil de  $q$  en deçà de laquelle on utilisera les fonctions de densité et de survie de la loi log-normale. Pour fixer cette valeur, on utilise la propriété suivante. Si  $X$  suit une distribution du  $\chi_k^2$  alors :<sup>3</sup>

$$U = \frac{X - k}{\sqrt{2k}} \xrightarrow[k \rightarrow +\infty]{L} N(0, 1).$$

Or la fonction gamma tronquée régularisée est donnée par :

$$\gamma(\beta, x) = F_{\chi^2, 2\beta}(2x)$$

donc cette fonction sera proche de la fonction de répartition de la loi normale quand  $k = 2\beta \rightarrow +\infty$ . On peut donc considérer que :

$$q = \frac{1}{\sqrt{\beta}} = \frac{1}{\sqrt{k/2}}.$$

Un examen graphique des fonctions de répartition montre qu'elles sont déjà très proches quand  $k = 200$  (graphique 4.20). En prenant cette valeur comme référence, on obtiendrait un seuil  $\bar{q}$  égal à :

$$\bar{q} = \frac{1}{\sqrt{200/2}} = 0,1.$$

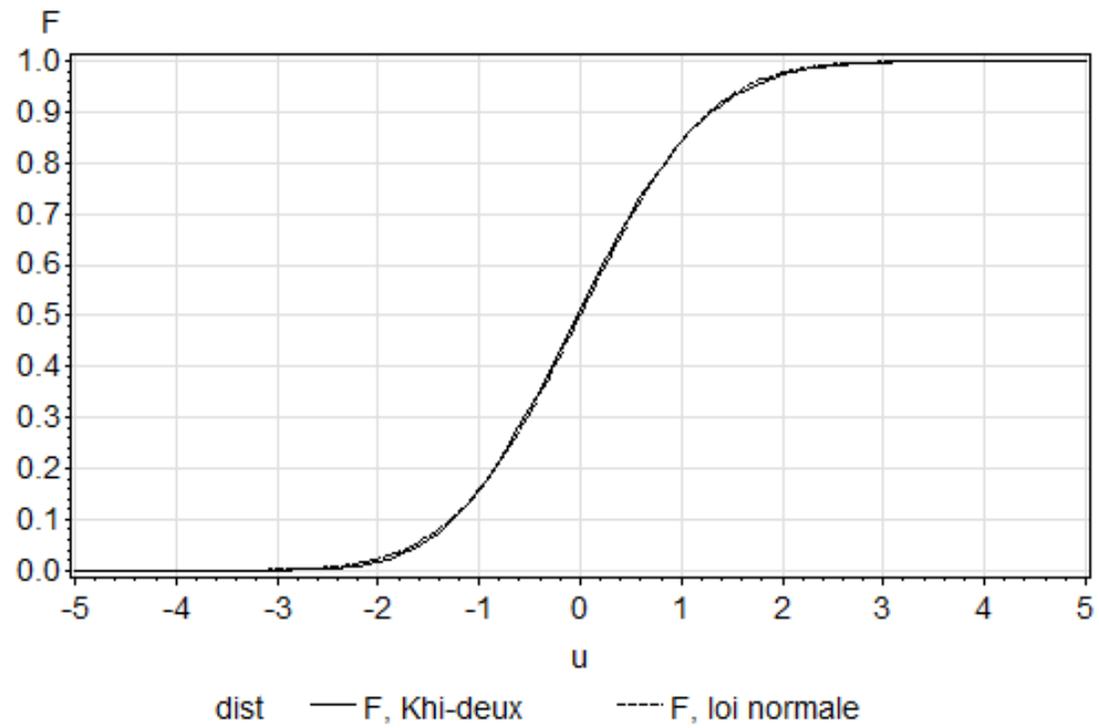
#### 4.10.3 Programmation X : Estimation par balayage avec routine d'optimisation

1. `proc iml;`

Lancement de SAS-IML.

---

3. Cette propriété vient du fait que la distribution du  $\chi_k^2$  est la somme des carrés de  $k$  variables normales indépendantes.

GRAPHIQUE 4.20 – COMPARAISON DES FONCTIONS DE RÉPARTITION POUR  $k = 200$

2. 

```
start lecture;
nom_y={&y}; nom_d={&d};
use &tab; read all var(nom_y) into yi;
read all var(nom_d) into di;
nd=sum(di); n=nrow(yi); nc=n-nd; lyi=log(yi);
print "Time variable =" nom_y,
"Censoring variable (=0 if censored) =" nom_d,
"Number of observations =" n,
"Complete =" nd, "Censored =" nc;
finish lecture;
```

Module de lecture habituel.
3. 

```
start igf(beta,x) global(epsilon);
bet=max(beta,epsilon);
f=probchi(2#x,2#bet);
do i=1 to nrow(x);
f[i]=min(f[i],1-epsilon);
end;
return (f);
finish;
```

Fonction Gamma tronquée normalisée.
4. 

```
start lv(theta) global(di,yi,lyi,q);
t1=theta[1]; t2=theta[2]; sigma=exp(t1);
m=t2; ui=(lyi-m)/sigma;
beta=q##(-2); psi=digamma(beta);
wi=psi+q#ui; lfi=log(q)-t1-lyi
-lgamma(beta)+beta#wi-exp(wi);
lsi=log(1-igf(beta,exp(wi)));
lv=sum(di#lfi+(1-di)#lsi);
return (lv);
finish;
```

Calcul de la log-vraisemblance, pour une valeur fixée de  $q$ .
5. 

```
start estim;
```

Début du module d'estimation.
6. 

```
print "Three-parameter Gamma distribution ", ,
"S(t)=1-g(beta,exp(digamma(beta)+q.u)",
"u=(ln(t)-m)/sigma",
"beta=q^(-2)",
"where g is the incomplete regularized Gamma
function", , ,
```

- "FIRST STEP : select  $q$  and save it for the second step", , ;  
Affichage de la fonction de survie et du paramétrage utilisé (Prentice, 1974).
7.  $lyi = \log(y_i)$  ; Calcul du vecteur contenant  $[\ln(y_i)]$ .
  8.  $\epsilon = 1e-6$  ; On fixe  $\epsilon$  à  $10^{-6}$ .
  9.  $m0 = \text{sum}(lyi) / n$  ;  
 $\text{sigma0} = \text{sqrt}(\text{sum}((lyi - j(n, 1, m0)) \# \# 2) / n)$  ;  
Calcul des valeurs initiales, on prend la moyenne empirique et l'écart type empirique des logarithmes des durées, sans tenir compte de la censure.
  10.  $\text{theta0} = (\log(\text{sigma0}) \mid \mid m0)$  ;  
Passage au paramètre d'optimisation initial  $\theta_0 = (\ln \sigma_0, m_0)$ .
  11.  $\text{opt} = j(11, 1, 0)$  ;  $\text{opt}[1] = 1$  ;  $\text{opt}[2] = 0$  ;  
Options de la routine d'optimisation, on maximise ( $\text{opt}[1] = 1$ ) et on ne demande pas d'affichage ( $\text{opt}[2] = 0$ ).
  12.  $\text{bmin} = \&b\_min$  ;  $\text{bmax} = \&b\_max$  ;  $\text{bby} = \&b\_by$  ;  
Paramètres de la boucle de balayage. Le paramètre  $B$  varie entre les bornes  $B^-$  ( $\text{bmin}$ ) et  $B^+$  ( $\text{bmax}$ ) avec un pas égal à  $\text{bby}$ . On remarque que l'on doit avoir  $B^- > 0$  et  $B^+ < 1$ .
  13.  $\text{nb} = "b" \mid \mid "q" \mid \mid "sigma" \mid \mid "m" \mid \mid "lv\_q"$  ;  
Noms de colonne de l'affichage des résultats des optimisations intermédiaires.
  14.  $\text{do } b = \text{bmin} \text{ to } \text{bmax} \text{ by } \text{bby}$  ;  
Début de la boucle de balayage.
  15.  $q = b / (1 - b)$  ;  
Calcul de la valeur de balayage du paramètre  $q = B / (1 - B) \in \mathbb{R}_+^*$ .
  16.  $\text{call } \text{nlpnra}(rc, \text{theta}, "lv", \text{theta0}, \text{opt})$  ;  
Optimisation de la log-vraisemblance.
  17.  $\text{call } \text{nlpfdd}(f, gr, he, "lv", \text{theta})$  ;  
 $\text{elas} = \max(\text{abs}(\text{theta} \# gr / f))$  ;  $\text{vp} = \max(\text{eigval}(he))$  ;  
Calcul de la plus grande élasticité de la log-vraisemblance et de la plus grand valeur propre du hessien. La première quantité doit être nulle et la seconde négative à chaque itération.
  18.  $\text{res} = (b \mid \mid q \mid \mid \exp(\text{theta}[1]) \mid \mid \text{theta}[2] \mid \mid lv(\text{theta}))$  ;  $\text{print res}[\text{colname} = \text{nb}]$ , "Gradient =" gr, "Max Elasticity =" elas,

- ```

"Max eigenvalue =" vp ;
Affichage des résultats intermédiaires.
19. outb=outb/(b||q||exp(theta[1])||theta[2]||
lv(theta));
Stockage des résultats intermédiaires.
20. theta0=theta;
On prend le résultat de l'optimisation précédente pour initialiser l'opti-
misation courante.
21. end;
Fin de la boucle de balayage.
22. finish estim;
Fin du module d'estimation.
23. start sortie;
Module de sortie.
24. create &outb from outb[colname=nb];
append from outb;
Création de la matrice de sortie, elle contient les résultats de toutes les
optimisations.
25. call sort(outb,5,5);
On trie la matrice de sortie par ordre décroissant par rapport à la log-
vraisemblance située dans la colonne 5.
26. outp=outb[1,];
La première ligne de la matrice contient les caractéristiques du maxi-
mum de la log-vraisemblance.
27. bm=outp[1];
Valeur de B qui correspond au maximum de vraisemblance.
28. if bm=bmin then print "WARNING: Corner solution
on the left";
if bm=bmax then print "WARNING: Corner solution
on the right";
On affiche un message d'avertissement en cas de solution en coin.
29. print "Maximum likelihood at " outp[colname=nb];
Affichage du résultat de l'optimisation.
30. create &outp from outp[colname=nb];
append from outp;
Stockage du paramètre qui maximise la vraisemblance.

```

- ```

31. finish sortie;
 run lecture;
 run estim;
 run sortie;
 quit;
 Lancement des modules et sortie d'IML.
32. symbol1 c=black line=1 i=join;
 proc gplot data=_outb; plot lv_q*q;
 Graphique de la log-vraisemblance en fonction de q .
33. run;
 Lancement du programme.

```

Ce premier programme permet de faire un balayage sur le paramètre  $q$ . Une fois ce programme lancé, on peut envoyer le programme d'estimation proprement dit.

#### 4.10.4 Application X

Nous reprenons les données sur la durée du premier emploi. Nous estimons d'abord le paramètre  $q$  par balayage (graphiques 4.21 et 4.22). Il apparaît clairement que nous sommes dans le cas log-normal puisque la valeur la plus petite que nous avons pu essayer  $q = 0.005$  maximise la vraisemblance, ce qui correspond à  $\beta = q^{-2} = 40000$ . En ce point la vraisemblance maximale est de  $-34277.1$ , inférieure à celle obtenue avec la loi log-normale ( $-34272.35$ ). Il faut donc se référer aux estimations de la loi log-normale, puisqu'on atteint les possibilités numériques du logiciel.

Voici la sortie du premier balayage, pour  $B \in [0.05, 0.95]$ . Nous avons enlevé quelques lignes pour des raisons de place. Le second balayage permet de descendre jusqu'à  $q = 0.005$ , ce qui correspond à  $\beta = q^{-2} = 40000$ . On se trouve clairement avec le cas d'une log-normale, qui est ici une solution en coin.

#### Sortie SAS 4.11.

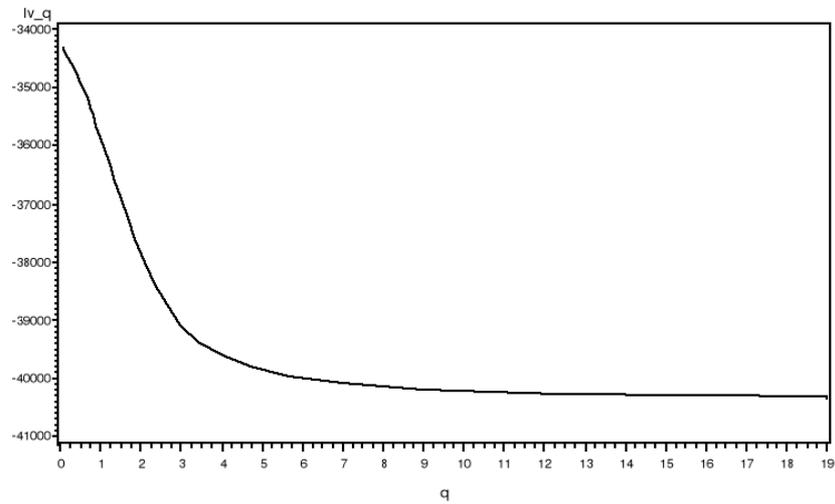
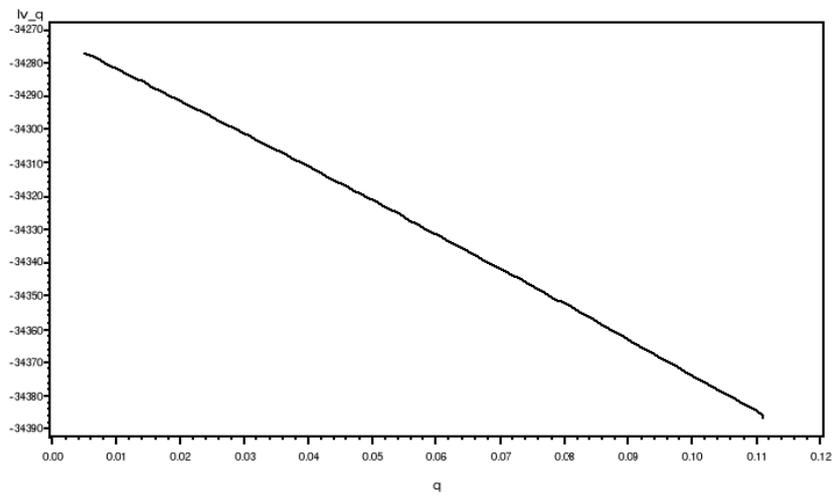
```

Three-parameter Gamma distribution
S(t)=1-g(beta,exp(digamma(beta)+q.u)
 u=(ln(t)-m)/sigma
 beta=q ^ (-2)
where g is the incomplete regularized Gamma function

FIRST STEP : select q and save it for the second step

 res
b q sigma m lv_q

```

GRAPHIQUE 4.21 – LOI GAMMA GÉNÉRALISÉE,  $\ell^*(q)$ GRAPHIQUE 4.22 – LOI GAMMA GÉNÉRALISÉE,  $\ell^*(q)$ ,  $0.005 \leq q \leq 0.110$

```

0.05 0.0526316 0.8380779 1.8181099 -34323.89
 gr
Gradient = 0.0278035 0.0315343
 elas
Max Elasticity = 1.6703E-6
 vp
Max eigenvalue = -17240.91

```

```

 res
b q sigma m lv_q
0.1 0.1111111 0.8414997 1.8173274 -34385.84
 gr
Gradient = -0.179477 0.0263437
 elas
Max Elasticity = 1.3923E-6
 vp
Max eigenvalue = -17096.69

```

```

 res
b q sigma m lv_q
0.15 0.1764706 0.8450778 1.816169 -34460.79
 gr
Gradient = -0.097796 -0.042999
 elas
Max Elasticity = 2.2662E-6
 vp
Max eigenvalue = -16637.31

```

```

 res
b q sigma m lv_q
0.2 0.25 0.8487418 1.8143723 -34552.2
 gr
Gradient = -0.233234 0.018217
 elas
Max Elasticity = 1.107E-6
 vp
Max eigenvalue = -16004.68

```

...pour des raisons de place, on passe directement à  $B=0.8$  ...

```

 res
b q sigma m lv_q
0.8 4 0.4674234 1.7438127 -39597.82
 gr
Gradient = -0.079877 0.0411081
 elas
Max Elasticity = 1.8103E-6
 vp
Max eigenvalue = -2663.589

```

```

 res
 b q sigma m lv_q
0.85 5.6666667 0.3411915 1.7947244 -39950.04
 gr
 Gradient = 0.1357571 0.0469984
 elas
 Max Elasticity = 3.6541E-6
 vp
 Max eigenvalue = -2606.434

 res
 b q sigma m lv_q
0.9 9 0.2202337 1.8304496 -40186.09
 gr
 Gradient = -0.06509 -0.055203
 elas
 Max Elasticity = 2.5145E-6
 vp
 Max eigenvalue = -2766.421

 res
 b q sigma m lv_q
0.95 19 0.1056604 1.8522932 -40313.24
 gr
 Gradient = 0.3813 0.2566123
 elas
 Max Elasticity = 0.0000213
 vp
 Max eigenvalue = -8782.455

WARNING: Corner solution on the left

 Maximum likelihood at
 outp
 b q sigma m lv_q
0.05 0.0526316 0.8380779 1.8181099 -34323.89

```

```

1. %let inp=_inp;
 %let qs=0.1;

```

On ajoute deux macro-variables. La première contient le résultat du balayage par le premier programme. Elle contient  $(q^*, m(q^*), \sigma(q^*))$ . Le second paramètre indique la limite du paramètre  $q$  en dessous de laquelle on considère que l'on est dans le cas log-normal. Ici, nous avons pris  $q = 0.1$ , ce qui correspond à  $\beta = q^{-2} = 100$ . Cette valeur est modi-

fiable par l'utilisateur.

2. `proc iml;`  
Début de la procédure SAS-IML.
3. `start lecture;`  
`nom_y={&y};`  
`nom_d={&d};`  
`use &tab; read all var(nom_y) into yi;`  
`read all var(nom_d) into di;`  
`nd=sum(di); n=nrow(yi); nc=n-nd; lyi=log(yi);`  
`print "Time variable =" nom_y,`  
`"Censoring variable (=0 if censored) =" nom_d,`  
`"Number of observations =" n,`  
`"Complete =" nd,`  
`"Censored =" nc;`  
`finish lecture;`  
 Module de lecture habituel.
4. `start cphi(x) global(epsilon);`  
`f=1-probnorm(x);`  
`do i=1 to nrow(x);`  
`f[i]=max(f[i],epsilon);`  
`end;`  
`return (f);`  
`finish;`

Calcul de la fonction utilisée pour la survie de la loi log-normale :

$$f = \max(1 - \Phi(x), \varepsilon)$$

avec  $\varepsilon = 10^{-6}$  par défaut, modifiable par l'utilisateur. Ceci garantit que l'on peut prendre le logarithme de cette fonction.

5. `start lphi(x) global(epsilon,pi);`  
`f=- (log(2#pi)+x##2)/2;`  
`return (f);`  
`finish;`  
 Fonction calculant le logarithme de la densité de la loi normale centrée-réduite :

$$\ln \varphi(x) = -\frac{1}{2} (\ln(2\pi) + x^2).$$

6. `start igf(beta,x) global(n,epsilon);`  
 Calcul de la fonction Gamma tronquée normalisée. Si  $\beta < 10^{10}$ , on prend

la formule qui utilise la fonction de répartition de la loi du khi-deux. On utilise :

$$\gamma(\beta, x) = \min(F_{\chi^2, 2\beta}(2x), \varepsilon)$$

avec  $\varepsilon = 10^{-6}$  pour que la fonction n'atteigne pas la valeur 1. Si  $\beta \geq 10^{10}$ , on remplace la fonction par sa limite quand  $\beta \rightarrow +\infty$ , pour éviter un dépassement lors du calcul ("overflow", en anglais). Cette limite est égale à 0.

- ```

7. if beta<1e10 then do;
  f=probchi(2#x,2#beta);
  do i=1 to nrow(x);
  f[i]=min(f[i],1-epsilon);
  end;
  Cas habituel.
  end;
  Fin du then do.
else f=j(n,1,0);
  Remplacement par la limite de la fonction quand  $\beta \geq 10^{10}$ .
  return (f);
  finish;
8. start lv(theta) global(di,yi,lyi,qs,flag);
  Calcul de la log-vraisemblance du modèle Gamma généralisé.
9. t1=theta[1]; t2=theta[2]; m=theta[3];
  Paramètres d'optimisation.
10. q=exp(t1); sigma=exp(t2);
  Paramètres d'intérêt. Le paramètre  $m$  est à la fois un paramètre d'optimisation et un paramètre d'intérêt.
11. ui=(lyi-m)/sigma;
  if q>qs then do;
  Cas usuel, correspond à  $q > 0$ .
12. beta=q##(-2); psi=digamma(beta); wi=psi+q#ui;
  lfi=log(q)-t2-lyi-lgamma(beta)+beta#wi-exp(wi);
  lsi=log(1-igf(beta,exp(wi)));
  flag=0;
  Variable qui indique si la log-vraisemblance Gamma généralisée a été utilisée (flag=0) ou si la log-vraisemblance log-normale a été utilisée (flag=1).
13. end;
  Fin du then do.

```

14. else do;
Début du cas particulier log-normal.
15. lfi=lphi(ui)-t2-lyi;
lsi=log(cphi(ui));
flag=1;
end;
Fin du else do.
16. lv=sum(di#lfi+(1-di)#lsi);
Calcul de la log-vraisemblance, quelque soit la définition utilisée.
17. return (lv);
finish;
Fin du module de calcul de la log-vraisemblance.
18. start estim;
Début du module d'estimation.
19. print "Three-parameter Gamma distribution ", ,
"S(t)=1-g(beta,exp(digamma(beta)+q.u) ",
"u=(ln(t)-m)/sigma",
"beta=q^(-2)",
"where g is the incomplete regularized Gamma
function", , ,
"SECOND STEP : Global optimization", , ;
Rappel du paramétrage.
20. flag=0; pi=acos(-1); epsilon=1e-6; qs=&qs;
Paramètres initiaux. On calcule le nombre π par la formule :

$$\pi = \cos^{-1}(-1).$$

21. lyi=log(yi);
Vecteur des logarithmes des durées, stocké pour usage ultérieur.
22. use &inp; read all var {q sigma m} into dzeta;
Lecture du résultat du balayage. Le vecteur dzeta contient :

$$(q^*, \sigma(q^*), m(q^*)).$$

23. theta0=(log(dzeta[1]) || log(dzeta[2]) || dzeta[3]);
Valeur initiale du paramètre d'optimisation :

$$(\theta_1^{(0)}, \theta_2^{(0)}, \theta_3^{(0)}) = (\ln q^*, \ln \sigma(q^*), m(q^*)).$$

24. `opt=j(11,1,0); opt[1]=1; opt[2]=1;`
Initialisation de la routine *nlpnra*. On demande une maximisation et un affichage minimal.
25. `call nlpnra(rc,theta,"lv",theta0,opt);`
Appel de la routine d'optimisation par la méthode de Newton-Raphson.
26. `print "Result code (must be >0) =" rc;`
Code de résultat provenant de la routine; il doit être positif.
27. `call nlpfdd(lv1,g,h,"lv",theta);`
Calcul de la log-vraisemblance, du gradient et du hessien à l'optimum.
28. `g=t(g); theta=t(theta); elas=abs(theta#g/lv1);`
`vp=eigval(h);`
On remet le gradient et le paramètre d'optimisation en colonnes, puis on calcule les élasticité de la log-vraisemblance aux paramètres et les valeurs propres du hessien.
29. `print "Log-likelihood at maximum =" lv1,`
`"Gradient" g, "Elasticities =" elas,`
`"Hessian eigenvalues =" vp;`
Affichage des résultats de l'optimisation.
30. `labc="Statistic" || "Std_Dev" || "Student" || "P_Value";`
Noms des colonnes du tableau des résultats.
31. `resp=(lv1 || . || . || .);`
Stockage de la log-vraisemblance maximale dans la matrice des résultats.
32. `labr="log-likelihood";`
Nom de ligne de la matrice des résultats.
33. `if flag=0 then do;`
Si l'on est dans le cas $q > 0$, on effectue les calculs suivants.
34. `vtheta=inv(-h);`
Estimation de la matrice de covariance asymptotique.
35. `q=exp(theta[1]); sigma=exp(theta[2]); m=theta[3];`
Paramétrage de Prentice.
36. `bp=q//sigma//m;`
On range les paramètres à la Prentice dans le vecteur *bp*.
37. `dbp=(q || 0 || 0) // (0 || sigma || 0) // (0 || 0 || 1);`
Dérivées du paramètre de Prentice par rapport au paramètre d'optimisation.

38. $vbp=dbp*vtheta*t(dbp);$
Estimation de la matrice de covariance asymptotique correspondant au paramétrage de Prentice.
39. $sbp=sqrt(vecdiag(vbp));$
Estimation des écarts-types asymptotiques.
40. $tbp=abs(bp)/sbp;$
Estimation des t de Student asymptotiques.
41. $pbp=1-probchi(tbp##2,1);$
Estimation des probabilités critiques asymptotiques.
42. $resp=resp/(bp||sbp||tbp||pbp);$
On met à jour la matrice des résultats en ajoutant une ligne.
43. $labr=labr/"q"/"sigma"/"m";$
On met à jour le vecteur des noms de ligne en ajoutant une ligne.
44. $beta=1/q##2; alpha=q/sigma;$
 $h=exp(digamma(beta)-alpha#m);$
 $bs=beta/alpha/h;$
Paramétrage de Stacy, défini à partir du paramétrage de Prentice. Stockage dans un vecteur bs .
45. $dbs=(-2#q##(-3)||0||0)//$
 $(1/sigma||-q#sigma##(-2)||0)//$
 $(-(2#q##(-3)#trigamma(q##(-2))+m/sigma)#h||$
 $q#m#sigma##(-2)#h||$
 $-q#h/sigma);$
Dérivées des paramètres de Stacy par rapport à ceux de Prentice, en vue de l'application du théorème de Slutsky.
46. $vbs=dbs*vbp*t(dbs);$
Estimation de la matrice de covariance asymptotique avec paramétrage à la Stacy.
47. $sbs=sqrt(vecdiag(vbs)); tbs=abs(bs)/sbs;$
 $pbs=1-probchi(tbs##2,1);$
Estimation des écarts-types, des t de Student et des probabilités asymptotiques associés au paramétrage à la Stacy.
48. $resp=resp/(bs||sbs||tbs||pbs);$
Mise à jour de la matrice des résultats.
49. $labr=labr/"beta"/"alpha"/"h";$
Mise à jour des noms de lignes.

- ```

50. mt=gamma(1/alpha)/(h##(1/alpha)
 #beta(1/alpha,beta));
 dmt=(mt#(digamma(1/alpha+beta)-digamma(beta)))||
 mt#(log(h)-digamma(1/alpha+beta))/alpha##2||
 -mt/alpha/h;
 vmt=dmt*vbs*t(dmt); smt=sqrt(vmt); tmt=mt/smt;
 pmt=1-probchi(tmt##2,1);
 Estimation de la durée moyenne et des statistiques associées.
51. md=(cinv(0.5,2#beta)/(2#h))##(1/alpha);
 Estimation de la durée médiane.
52. resp=resp/(mt||smt||tmt||pmt)/(md||.||.||||.);
 Mise à jour de la matrice des résultats.
53. labr=labr/"Mean"/"Median";
 Mise à jour des noms de ligne.
54. print resp[colname=labc rowname=labr];
 Affichage des résultats.
55. print "Tests with the Prentice
 parametrization", , ;
 tw=abs(q-1)/sqrt(vbp[1,1]);
 pw=1-probchi(tw##2,1);
 print "Test H0: q=1 (Weibull) ",
 "Statistic =" tw, "p-value =" pw;
 Test d'adéquation de la loi de Weibull ($H_0 : q = 1$).
56. tg2=abs(q-sigma)/sqrt(vbp[1,1]+vbp[2,2]
 -2#vbp[1,2]); pg2=1-probchi(tg2##2,1);
 print "Test H0: q=sigma (2-parameter Gamma)",
 "Statistic =" tg2,
 "p-value =" pg2;
 Test d'adéquation de la loi Gamma à deux paramètres ($H_0 : q = \sigma$).
57. end;
 Fin du then do correspondant à la loi Gamma généralisée.
58. else do;
 Début du cas particulier log-normal.
59. q=exp(theta[1]);
 if q > 0 then beta=q##(-2); else beta=.;
 On affiche la valeur de β correspondant à la valeur de q . Par convention,
 une valeur manquante correspond à $+\infty$.

```

```

60. print "The estimated value of beta is equal
 to =" beta, "therefore it is likely that the
 distribution is LOG-NORMAL, , , ,"
 "RESULTS FOR THE LOG-NORMAL DISTRIBUTION";
 Comme la valeur de q est inférieure au seuil défini par l'utilisateur, on
 considère que la loi est log-normale.
61. rg=2:3;
 Indices correspondants aux paramètres $(\theta_2, \theta_3) = (\ln \sigma, m)$.
62. theta=t(theta[rg]);
 On redéfinit θ comme (θ_2, θ_3) .
63. vtheta=inv(-h[rg,rg]);
 On calcule $\widehat{\text{Vas}}(\widehat{\theta}_2, \widehat{\theta}_3)$.
64. bn=exp(theta[1])/theta[2]; sigma=bn[1]; m=bn[2];
 Paramétrage log-normal, stocké dans le vecteur $\text{bn} = (\sigma, m)$.
65. dbn=(sigma|0)/(0|1);
 vbn=dbn*vtheta*t(dbn);
 sbn=sqrt(vecdiag(vbn));
 tbn=abs(bn)/sbn;
 pbn=1-probchi(tbn##2,1);
 Calcul des statistiques associées au paramètre (σ, m) .
66. resp=resp/(bn||sbn||tbn||pbn);
 On met à jour la matrice des résultats.
67. labr=labr/"sigma"/"m";
 On met à jour le vecteur des noms de ligne.
68. md=exp(m);
 Estimation de la durée médiane.
69. smd=sqrt(md##2#vbn[2,2]); tmd=md/smd;
 pmd=1-probchi(tmd##2,1);
 Calcul des statistiques associées à la durée médiane.
70. mt=exp(m+sigma##2/2);
 Estimation de la durée moyenne
71. dmt=(sigma#mt||mt);
 smt=sqrt(dmt*vbn*t(dmt)); tmt=mt/smt;
 pmt=1-probchi(tmt##2,1);
 Calcul des statistiques associées à la durée moyenne.
72. resp=resp/(mt||smt||tmt||pmt)//
 (md||smd||tmd||pmd);
 Mise à jour de la matrice des résultats.

```

73. `labr=labr// "Mean" // "Median" ;`  
Mise à jour du vecteur des noms de lignes.
74. `start has(x) global(alpha,mu);`  
`z=alpha#x-mu;`  
`f=pdf("normal",z,0,1)#alpha/exp(x);`  
`s=cphi(z);`  
`has=log(f)-log(s);`  
`return (has);`  
`finish;`  
Fonction définissant le logarithme de la fonction de hasard de la loi log-normale.
75. `opt=j(11,1,0); opt[1]=1; opt[2]=0;`  
Paramètres de l'optimisation qui recherche le maximum du hasard par rapport à  $t$ .
76. `alpha=1/sigma; mu=m/sigma; lt0=log(md);`  
Valeur initiale de l'algorithme, on prend la durée médiane.
77. `call nlpnra(rc,lt1,"has",lt0,opt);`  
Maximisation du hasard par la méthode de Newton-Raphson.
78. `tmax=exp(lt1); hmax=exp(has(lt1));`  
Durée maximisant le hasard ( $t_{max}$ ) et valeur du hasard au maximum ( $h_{max}$ ).
79. `print "The hasard function reaches a maximum at`  
`T =" tmax, "Its maximum value is h =" hmax;`  
`print resp[colname=labc rowname=labr];`  
Affichage des résultats.
80. `end;`  
Fin du cas log-normal.
81. `finish estim;`  
Fin du module d'estimation.
82. `start prev;`  
`tmin=min(yi); tmax=max(yi); tby=1;`  
`do time=tmin to tmax by tby;`  
`if flag=0 then do;`  
`ui=(lyi-m)/sigma;`  
`beta=q##(-2); psi=digamma(beta); wi=psi+q#ui;`  
`lfi=log(q)-t2-lyi-lgamma(beta)+beta#wi-exp(wi);`  
`lsi=log(1-igf(beta,exp(wi)));`

```

fi=exp(lfi);
si=exp(lsi);
ci=1-si;
hi=fi/si;
end;
else do;
zi=(log(time)-m)/sigma;
fi=pdf("normal",zi,0,1)/(sigma*time);
si=cphi(zi);
hi=fi/si;
ci=1-si;
end;
resi=resi/(time*|fi|*|si|*|hi|*|ci|);
end;
labi="&y"||"Density"||"SDF"||"Hasard"||"CDF";
finish prev;

```

Module de prévision habituel. On remplace juste les fonctions par leurs valeurs dans les cas Gamma généralisé et log-normal.

```

83. start sortie;
create &outp from resp[colname=labc];
append from resp;
create _np from labr[colname="Estimate"];
append from labr;
create &outi from resi[colname=labi];
append from resi;
finish sortie;

```

Module de sortie habituel.

```

84. run lecture;
run estim;
run prev;
run sortie;

```

Lancement des modules.

```

85. quit;
data &outp; merge _np &outp;
run;

```

On quitte SAS-IML et on ajoute les noms de ligne à la matrice des résultats.

Ce programme génère la sortie suivante.

**Sortie SAS 4.12.**

```

 nom_y
Time variable = DUREE_EMP
 nom_d
Censoring variable (=0 if censored) = PRESENCE
 n
Number of observations = 12695
 nd
Complete = 11277
 nc
Censored = 1418

Three-parameter Gamma distribution

S(t)=1-g(beta,exp(digamma(beta)+q.u)
u=(ln(t)-m)/sigma
beta=q^(-2)
where g is the incomplete regularized Gamma function

SECOND STEP : Global optimization

Newton-Raphson Optimization with Line Search

Without Parameter Scaling
Gradient Computed by Finite Differences
CRP Jacobian Computed by Finite Differences

Parameter Estimates 3

Optimization Start

Active Constraints 0 Objective Function -34385.83916
Max Abs Gradient 122.30624302
Element

 Max Abs
Iter Rest Func Act Objective Obj Fun Gradient Step
 arts Calls Con Function Change Element Size
 1 0 3 0 -34272 113.4 38.9343 1.000

 Slope of
 Search
 Direction

 -108.8

 Max Abs
Rest Func Act Objective Obj Fun Gradient Step

```

| Iter | arts | Calls | Con | Function | Change | Element | Size  |
|------|------|-------|-----|----------|--------|---------|-------|
| 2*   | 0    | 4     | 0   | -34272   | 0.0428 | 0.1365  | 1.000 |

Slope of  
Search  
Direction

-0.0849

| Iter | Rest<br>arts | Func<br>Calls | Act<br>Con | Objective<br>Function | Obj Fun<br>Change | Max Abs<br>Gradient<br>Element | Step<br>Size |
|------|--------------|---------------|------------|-----------------------|-------------------|--------------------------------|--------------|
| 3*   | 0            | 7             | 0          | -34272                | 2.958E-8          | 0.0539                         | 0.207        |

Slope of  
Search  
Direction

-107E-8

#### Optimization Results

|                              |              |                             |              |
|------------------------------|--------------|-----------------------------|--------------|
| Iterations                   | 3            | Function Calls              | 8            |
| Hessian Calls                | 4            | Active Constraints          | 0            |
| Objective Function           | -34272.35302 | Max Abs Gradient<br>Element | 0.0538751884 |
| Slope of Search<br>Direction | -1.07096E-6  | Ridge                       | 0.00001      |

GCONV convergence criterion satisfied.

NOTE: At least one element of the (projected) gradient is  
greater than 1e-3.

rc

Result code (must be >0) = 6

lvl

Log-likelihood at maximum = -34272.35  
g

|          |           |
|----------|-----------|
| Gradient | 0         |
|          | 0.0103404 |
|          | -0.053875 |
|          | elas      |

```
Elasticities = 0
 5.4465E-8
 2.8589E-6
 vp
```

```
Hessian eigenvalues = 0
 -17207.45
 -22423.84
```

beta

The estimated value of beta is equal to = 479.89428  
therefore it is likely that the distribution is LOG-NORMAL

#### RESULTS FOR THE LOG-NORMAL DISTRIBUTION

```

 tmax
The hasard function reaches a maximum at T = 5.6759629
 hmax
Its maximum value is h = 0.1553486
```

|                | Statistic | Std_Dev   | Student   | P_Value |
|----------------|-----------|-----------|-----------|---------|
| log-likelihood | -34272.35 | .         | .         | .       |
| sigma          | 0.8348369 | 0.0056166 | 148.63808 | 0       |
| m              | 1.8186531 | 0.0075794 | 239.94724 | 0       |
| Mean           | 8.7331793 | 0.0798095 | 109.42535 | 0       |
| Median         | 6.1635514 | 0.0467159 | 131.93678 | 0       |

En partant du point trouvé par balayage, le programme réalise trois itérations. La convergence a lieu sur un point où  $\hat{\beta} = 479.89$ , ce qui correspond au cas log-normal ( $\hat{q} = 0.046$ ). Le programme affiche donc les résultats obtenus avec le modèle log-normal. Notons ici que lorsque flag=1, le logiciel utilise une vraisemblance qui ne dépend que de  $(\sigma, m)$  donc le gradient numérique est nul par rapport à  $q$ . Ceci implique aussi qu'il y ait une valeur propre nulle pour le hessien (car la nullité de la dérivée première implique celle de la dérivée seconde). Ce cas est géré par le programme en extrayant le bloc qui correspond aux paramètres  $(\sigma, m)$  du hessien global avant d'inverser la matrices des dérivées secondes. Les graphiques sont donc identiques au cas log-normal et, pour cette raison, ne sont pas représentés ici.

## 4.11 Loi de Burr de type 12

### 4.11.1 Log-vraisemblance

Cette loi généralise les lois log-logistique ( $a = 1$ ), de Pareto ( $\alpha = 1$ ) et de Weibull ( $a \rightarrow +\infty$ ). Les logarithmes du hasard et de la survie sont donnés par :

$$\begin{aligned}\ln h_T(y_i) &= \ln a + \ln h + \ln \alpha + (\alpha - 1) \ln y_i - \ln(a + hy_i^\alpha) \\ \ln S_T(y_i) &= a \{ \ln a - \ln(a + hy_i^\alpha) \}\end{aligned}$$

d'où la vraisemblance :

$$\begin{aligned}\ell(\gamma) &= \sum_{i=1}^N d_i (\ln a + \ln h + \ln \alpha + (\alpha - 1) \ln y_i - \ln(a + hy_i^\alpha)) + a \ln a \\ &\quad - a \ln(a + hy_i^\alpha) \\ &= \sum_{i=1}^N d_i (\ln a + \ln h + \ln \alpha + (\alpha - 1) \ln y_i) + a \ln a - (a + d_i) \ln(a + hy_i^\alpha) \\ &= N \times a \ln a + N_d (\ln a + \ln h + \ln \alpha) + \sum_{i=1}^N (\alpha - 1) d_i \ln y_i \\ &\quad - (a + d_i) \ln(a + hy_i^\alpha)\end{aligned}$$

avec

$$N_d = \sum_{i=1}^N d_i, \quad \zeta = \begin{pmatrix} a \\ \alpha \\ h \end{pmatrix}.$$

On pose :

$$z_{1i} = a + hy_i^\alpha, \quad z_{2i} = \frac{hy_i^\alpha}{a + hy_i^\alpha}$$

et le vecteur du score s'écrit :

$$\begin{aligned}\frac{\partial \ell}{\partial a} &= N(1 + \ln a) + \frac{N_d}{a} - \sum_{i=1}^N \left( \ln z_{1i} + \frac{a + d_i}{z_{1i}} \right) \\ \frac{\partial \ell}{\partial \alpha} &= \frac{N_d}{\alpha} + \sum_{i=1}^N (d_i - (a + d_i) z_{2i}) \ln y_i \\ \frac{\partial \ell}{\partial h} &= \frac{N_d}{h} - \frac{1}{h} \sum_{i=1}^N (a + d_i) z_{2i}\end{aligned}$$

L'algorithme de Newton-Raphson ne fonctionne pas très bien quand on veut optimiser simultanément par rapport aux trois paramètres de cette dis-

tribution. Pour cette raison, nous ferons une estimation par balayage sur le paramètre  $a$ . Plus précisément, nous effectuerons les optimisations suivantes :

$$\ell^*(a) = \arg \max_{(a,h)} \ell(a, \alpha, h).$$

Nous avons choisi un balayage sur le paramètre  $a$  car, pour une valeur fixée de  $a$ , la log-vraisemblance est de forme similaire à celle d'un modèle de Weibull pour lequel la méthode de Newton-Raphson fonctionne très bien. Une fois une suite de valeurs de  $a$  obtenues, on recherche le maximum :

$$a^* = \arg \max_a \ell(a)$$

La grille de valeur de  $a$  que nous explorerons est ramenée sur l'intervalle  $]0, 1[$  pour faciliter la programmation. On recourt donc à la transformation :

$$B = \frac{a}{a+1},$$

on a :

$$\lim_{a \rightarrow 0} B = 0, \quad \lim_{a \rightarrow +\infty} B = 1,$$

on fera donc une grille variant de  $B = 0.05$  à  $B = 0.95$  par pas de 0.01, avec :

$$a = \frac{B}{1-B}.$$

Cette grille d'une centaine de points devrait permettre d'avoir un bon aperçu de la fonction. Une fois la valeur de  $a^*$  déterminée, on lance un algorithme de Newton-Raphson sur les trois paramètres pour finir l'optimisation. Dans la mesure où le point  $a^*$  correspond à un gradient proche du vecteur nul, il y a peu d'itérations à effectuer. On peut alors utiliser les routines classiques pour déterminer les statistiques usuelles.

#### 4.11.2 Estimation

On effectue le changement de paramètres :

$$a = \exp(\theta_1), \quad \alpha = \exp(\theta_2), \quad h = \exp(\theta_3), \quad \theta = \begin{pmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \end{pmatrix},$$

et on remarque que :

$$\frac{\partial a}{\partial \theta_1} = a, \quad \frac{\partial \alpha}{\partial \theta_2} = \alpha, \quad \frac{\partial h}{\partial \theta_3} = h.$$

La vraisemblance se réécrit sous la forme :

$$\ell_{\zeta}(\theta) = \ell(\zeta(\theta)).$$

Comme les changements de paramètres sont séparés, on a :

$$\begin{aligned} \frac{\partial \ell_{\zeta}}{\partial \theta_1} &= \frac{\partial \ell}{\partial a} \frac{\partial a}{\partial \theta_1} \\ &= Na(1 + \ln a) + N_d - a \sum_{i=1}^N \left( \ln z_{1i} + \frac{a + d_i}{z_{1i}} \right) \\ \frac{\partial \ell_{\zeta}}{\partial \theta_2} &= \frac{\partial \ell}{\partial \alpha} \frac{\partial \alpha}{\partial \theta_2} \\ &= N_d + \alpha \sum_{i=1}^N (d_i - (a + d_i) z_{2i}) \ln y_i \\ \frac{\partial \ell_{\zeta}}{\partial \theta_3} &= \frac{\partial \ell}{\partial h} \frac{\partial h}{\partial \theta_3} \\ &= N_d - \sum_{i=1}^N (a + d_i) z_{2i} \end{aligned}$$

Après l'estimation, on revient aux paramètres d'intérêt par la méthode de Slutsky, en utilisant :

$$\zeta = g(\theta) = \begin{pmatrix} \exp(\theta_1) \\ \exp(\theta_2) \\ \exp(\theta_3) \end{pmatrix}$$

et

$$\frac{\partial g}{\partial \theta'} = \begin{pmatrix} a & 0 & 0 \\ 0 & \alpha & 0 \\ 0 & 0 & h \end{pmatrix}.$$

### 4.11.3 Statistiques utiles

L'espérance mathématique existe si  $\alpha a > 1$ . Dans ce cas, elle est donnée par :

$$E(T) = g(\zeta) = a \left( \frac{a}{h} \right)^{1/\alpha} B(a - 1/\alpha, 1 + 1/\alpha),$$

pour calculer la variance asymptotique de l'estimateur, on utilise :

$$\begin{aligned} \frac{\partial g}{\partial a} &= g(\zeta) \left( \frac{1}{a} \left( \frac{1}{\alpha} + 1 \right) + \psi^{(0)}(a - 1/\alpha) - \psi^{(0)}(a + 1) \right), \\ \frac{\partial g}{\partial \alpha} &= \frac{g(\zeta)}{\alpha^2} \left( -\ln \left( \frac{a}{h} \right) + \psi^{(0)}(a - 1/\alpha) - \psi^{(0)}(1 + 1/\alpha) \right), \\ \frac{\partial g}{\partial h} &= -\frac{g(\zeta)}{\alpha h}. \end{aligned}$$

La médiane est donnée par :

$$\text{Me} = g(\zeta) = \left(\frac{a}{h}\right)^{1/\alpha} (2^{1/a} - 1)$$

et on utilise les dérivées suivantes pour calculer la variance asymptotique de l'estimateur de la médiane :

$$\begin{aligned} \frac{\partial g}{\partial a} &= \frac{g(\zeta)}{\alpha a} \left(1 - \left(\frac{\ln 2}{a}\right) \frac{2^{1/a}}{2^{1/a} - 1}\right), \\ \frac{\partial g}{\partial \alpha} &= -\frac{g(\zeta)}{\alpha^2} \left(\ln\left(\frac{a}{h}\right) + \ln(2^{1/a} - 1)\right), \\ \frac{\partial g}{\partial h} &= -\frac{g(\zeta)}{\alpha h}. \end{aligned}$$

On estime les fonctions par :

$$\begin{aligned} \hat{S}_T(t) &= \left(\frac{\hat{a}}{\hat{a} + \hat{h}t^{\hat{a}}}\right)^{\hat{a}}, \\ \hat{h}_T(t) &= \frac{\hat{a}\hat{h}\hat{a}t^{\hat{a}-1}}{\hat{a} + \hat{h}t^{\hat{a}}}, \\ \hat{f}_T(t) &= \hat{h}_T(t)\hat{S}_T(t), \\ \hat{F}_T(t) &= 1 - \hat{S}_T(t). \end{aligned}$$

#### 4.11.4 Programmation XI : estimation par balayage et avec gradient analytique

On reprend l'application sur la durée du premier emploi. Nous réaliserons une estimation par balayage similaire à celle de loi Gamma généralisée. Le paramètre de balayage sera  $a$  et nous utiliserons la transformation :

$$B = \frac{a}{1+a} \Leftrightarrow a = \frac{B}{1-B},$$

en faisant varier  $B$  de 0.05 à 0.95, nous couvrirons l'intervalle  $a \in [0.053, 19]$ , ce qui devrait être suffisant. Pour voir si on se rapproche du cas de Weibull, il suffira de comparer les log-vraisemblances à l'optimum. Si la log-vraisemblance de Weibull est plus faible que celles que nous pouvons atteindre sur l'intervalle précédent, c'est que nous ne sommes pas dans le cas  $a \rightarrow +\infty$ . Voici la première partie du programme d'estimation.

```

1. %let tab=duree;
 %let y=duree_emp;
 %let d=presence;
 %let b_min=0.05;
 %let b_max=0.95;
 %let b_by=0.05;
 %let outb=_outb;
 %let outp=_inp;

```

Macro-variables définissant les paramètres du programme. Les macro-variables `&b_min`, `&b_max` et `&b_by` indiquent l'intervalle sur laquelle on effectue le balayage [`&bmin`,`&bmax`] et le pas d'estimation `&b_by`. La macro-variable `&outb` indique le tableau qui contiendra le résultat du balayage  $(B, a, \alpha(a), h(a), \ell(a))$  pour chaque valeur de  $a$ . Ce tableau permet de tracer le graphique de la log-vraisemblance. Par défaut ce tableau se nomme `_outb` et se situe dans le répertoire de travail ("Work"). La macro-variable `&outp` contient la ligne relative au maximum de la log-vraisemblance sur l'intervalle de balayage. Elle contient :

$$(B^*, a^*, \alpha(a^*), h(a^*), \ell(a^*))$$

mais il peut s'agir d'un maximum local si l'intervalle n'est pas suffisamment large. Ce tableau sert à initialiser la seconde partie du programme, qui réalise une optimisation globale sur les paramètres  $(a, \alpha, h)$ . Ce second programme prend  $(a^*, \alpha(a^*), h(a^*))$  comme valeur initiale puis réalise une optimisation par la méthode de Newton-Raphson. Par défaut, ce tableau se nomme `_inp` et se situe dans le répertoire de travail ("Work").

```

2. proc iml;
 Début de la procédure SAS-IML.

3. start lecture;
 nom_y={&y}; nom_d={&d};
 use &tab; read all var(nom_y) into yi;
 read all var(nom_d) into di;
 nd=sum(di); n=nrow(yi); nc=n-nd; lyi=log(yi);
 print "Time variable =" nom_y,
 "Censoring variable (=0 if censored) =" nom_d,
 "Number of observations =" n,
 "Complete =" nd, "Censored =" nc;
 finish lecture;
 Module de lecture standard.

```

4. `start lv(theta) global(di, yi, lyi, a);`  
Fonction calculant la log-vraisemblance pour une valeur fixée de  $a$ . Cette valeur est lue par l'option *global*.
5. `t1=theta[1]; t2=theta[2]; alpha=exp(t1); h=exp(t2);`  
On repasse aux paramètres d'intérêt pour clarifier l'écriture de la log-vraisemblance.
6. `zi=log(1+(h/a)#yi##alpha);`  
`lv=sum(di#(t2+t1+(alpha-1)#lyi-zi)-a#zi);`  
`return (lv);`  
Renvoi de la log-vraisemblance vers l'appel de la fonction.
7. `finish;`  
Fin de la fonction calculant la log-vraisemblance.
8. `start sc(theta) global(di, yi, lyi, a, nd);`  
Fonction calculant le score pour une valeur fixée de  $a$ .
9. `t1=theta[1]; t2=theta[2]; alpha=exp(t1);`  
`h=exp(t2);`  
`hyia=h#yi##alpha;`  
`z2i=hyia/(a+hyia);`  
`der1=nd+alpha#sum((di-(a+di)#z2i)#lyi);`  
`der2=nd-sum((a+di)#z2i);`  
`der=(der1||der2);`  
Le gradient est écrit en ligne car ceci est requis par la routine *nlpnra*.
10. `return (der);`  
Renvoi du score vers l'appel de la fonction, réalisé par la procédure *nlpnra*.
11. `finish;`  
Fin de la fonction calculant le score.
12. `start estim;`  
Début du module d'estimation.
13. `print "Burr 12 distribution", ,`  
`"S(t)=(1+(h/a).t^alpha)^(-a)", ,`  
`"FIRST STEP : select a and save it for the`  
`second step", , ;`  
On rappelle le paramétrage utilisé.
14. `alpha0=1; h0=nd/sum(yi);`  
Valeur initiale, on prend le modèle exponentiel. Cette valeur n'est utilisée que pour la première valeur de  $B$ .

15. `theta0=log(alpha0 || h0);`  
Le paramètre d'optimisation est  $(\ln \alpha, \ln h)$  donc on adapte la valeur initiale.
16. `lyi=log(yi);`  
Calcul du vecteur  $[\ln y_i]$  car il revient par la suite. Ceci permet de réduire le temps de calcul.
17. `opt=j(11,1,0); opt[1]=1; opt[2]=0;`  
Paramètres de la routine d'optimisation `nlpnra`. On demande une maximisation (`opt[1]=1`) sans affichage (`opt[2]=0`).
18. `nb="b" || "a" || "alpha" || "h" || "lv_a";`  
Noms des colonnes des tableaux en sortie.
19. `bmin=&b_min; bmax=&b_max; bby=&b_by;`  
Paramètres de la boucle de balayage.
20. `do b=bmin to bmax by bby;`  
Début de la boucle de balayage.
21. `a=b/(1-b);`  
Calcul de  $a$ .
22. `call nlpnra(rc,theta,"lv",theta0,opt) grd="sc";`  
Calcul du maximum de la log-vraisemblance. On utilise un gradient analytique défini par la fonction `sc`.
23. `call nlpfdd(f,gr,he,"lv",theta);`  
Calcul du gradient et du hessien à l'optimum. On utilise un gradient analytique défini par la fonction `sc`.
24. `elas=max(abs(theta#gr/f)); vp=max(eigval(he));`  
Calcul de l'élasticité de la log-vraisemblance par rapport aux paramètres, et des valeurs propres du hessien.
25. `res=(b || a || exp(theta[1]) || exp(theta[2]) || lv(theta));`  
Vecteur contenant les résultats d'une itération de  $B$ .
26. `print res[colname=nb], "Gradient =" gr, "Max Elasticity =" elas, "Max eigenvalue =" vp;`  
Affichage des résultats de l'itération courante.
27. `outb=outb//res;`  
On ajoute les résultats de l'itération courante à la matrice des résultats.
28. `theta0=theta;`  
La valeur initiale de la prochaine optimisation est celle de l'optimisation

précédente car le paramètre  $a$  est nécessairement proche. Ceci permet de réduire le temps de calcul.

29. `end;`  
Fin de la boucle de balayage.
30. `finish estim;`  
Fin du module d'estimation.
31. `start sortie;`  
Début du module de sortie.
32. `create &outb from outb[colname=nb];`  
`append from outb;`  
Stockage de la matrice des résultats détaillés du balayage dans un tableau SAS dont le nom est donné par la macro-variable `&outb`.
33. `call sort(outb,5,5);`  
On trie la matrice des résultats par ordre décroissant de la log vraisemblance. Le maximum est donc située sur la première ligne de cette matrice.
34. `outp=outb[1,];`  
On extrait la ligne qui correspond au maximum. Elle servira de valeur initiale de l'algorithme d'optimisation globale de la second partie.
35. `bm=outp[1];`  
Valeur de  $B$  correspondant au maximum de la log-vraisemblance.
36. `if bm=bmin then print "WARNING: Corner solution on the left";`  
Message d'avertissement dans le cas d'une solution en coin à gauche.
37. `if bm=bmax then print "WARNING: Corner solution on the right";`  
Message d'avertissement dans le cas d'une solution en coin à droite.
38. `print "Maximum likelihood at " outp[colname=nb];`  
Affichage du résultat de l'optimisation.
39. `create &outp from outp[colname=nb];`  
`append from outp;`  
Stockage de la ligne de résultat correspondant au maximum de la log-vraisemblance.
40. `finish sortie;`  
Fin du module de sortie.
41. `run lecture;`  
`run estim;`

```

run sortie;
Lancement des modules

42. quit;
Sortie de SAS-IML

43. symbol1 c=black line=1 i=join;
proc gplot data=_outb; plot lv_a*a/grid;
run;
Affichage du graphique résumant l'ensemble du balayage.

```

#### 4.11.5 Application XI

On obtient la sortie suivante, dont nous éliminons les valeurs intermédiaires pour des raisons de place.

##### Sortie SAS 4.13.

```

 Burr 12 distribution
 S(t)=(a/(a+h.y ^alpha))^a

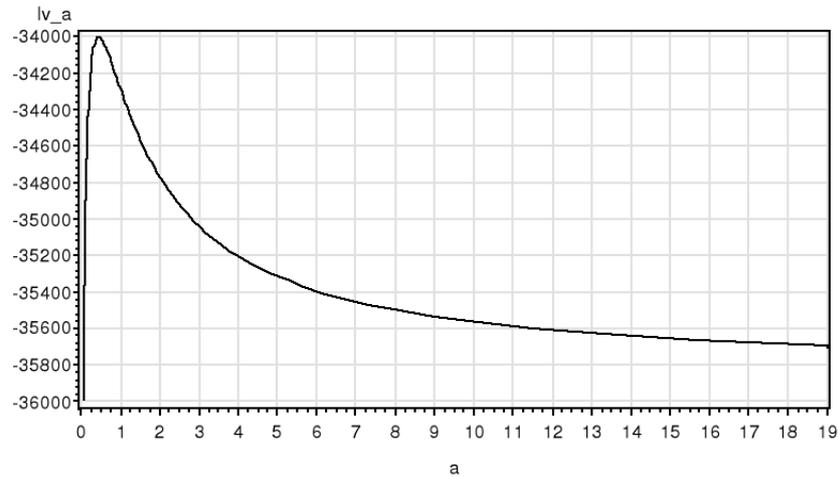
FIRST STEP : select a and save it for the second step
 res
 b a alpha h lv_a
0.05 0.0526316 12.500782 0.0000256 -35994.65
 gr
 Gradient = 2.1195E-7 2.4314E-8
 elas
 Max Elasticity = 1.487E-11
 vp
 Max eigenvalue = -96.54956
 res
 b a alpha h lv_a
0.06 0.0638298 10.740022 0.0000694 -35693.16
 gr
 Gradient = -0.001666 -0.000131
 elas
 Max Elasticity = 1.1078E-7
 vp
 Max eigenvalue = -124.6957
 res
 b a alpha h lv_a
0.07 0.0752688 9.4220624 0.0001449 -35461.48
 gr
 Gradient = -0.000322 -0.000021
 elas
 Max Elasticity = 2.0359E-8
 vp
 Max eigenvalue = -151.1888
 res
 b a alpha h lv_a
0.08 0.0869565 8.411473 0.0002526 -35270.84
 gr

```

```

Gradient = -0.000055 -1.013E-6
 elas
Max Elasticity = 3.2955E-9
 vp
Max eigenvalue = -174.8077
 res
b a alpha h lv_a
0.09 0.0989011 7.6210937 0.0003893 -35107.05
 gr
Gradient = -0.000012 -7.713E-7
 elas
Max Elasticity = 7.207E-10
 vp
Max eigenvalue = -195.558
 res
...
 res
b a alpha h lv_a
0.9 9 1.281234 0.0633903 -35534.95
 gr
Gradient = -2.256E-9 5.457E-10
 elas
Max Elasticity = 4.236E-14
 vp
Max eigenvalue = -1701.085
 res
b a alpha h lv_a
0.91 10.111111 1.2686993 0.0643786 -35567.47
 gr
Gradient = 1.5898E-9 -3.4E-10
 elas
Max Elasticity = 2.623E-14
 vp
Max eigenvalue = -1719.641
 res
b a alpha h lv_a
0.92 11.5 1.2564285 0.065361 -35599.86
 gr
Gradient = -3.06E-9 -4.733E-9
 elas
Max Elasticity = 3.627E-13
 vp
Max eigenvalue = -1738.006
 res
b a alpha h lv_a
0.93 13.285714 1.2444163 0.066337 -35632.11
 gr
Gradient = -0.006592 -0.001238
 elas
Max Elasticity = 9.4286E-8
 vp
Max eigenvalue = -1756.164
 res
b a alpha h lv_a
0.94 15.666667 1.2326572 0.0673065 -35664.21

```

GRAPHIQUE 4.23 – LOI DE BURR,  $\ell^*(a)$ , BALAYAGE DE  $a = 0$  À  $a = 19$ 

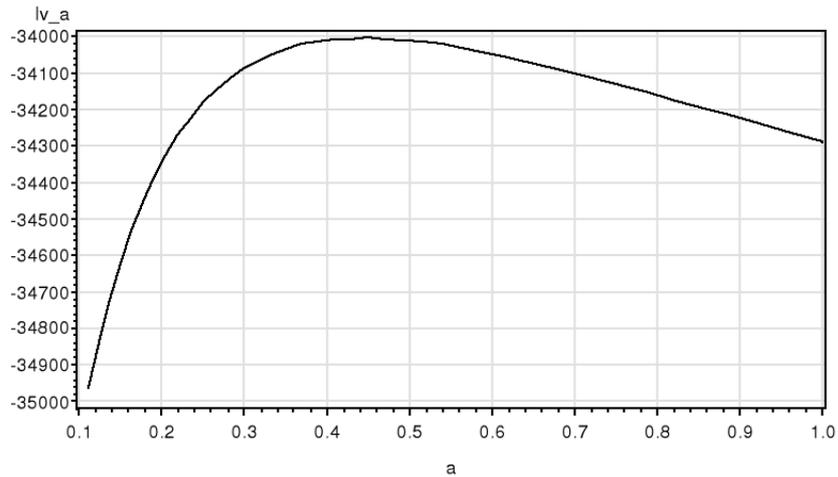
```

 gr
Gradient = -7.554E-9 -2.73E-10
 elas
Max Elasticity = 4.431E-14
 vp
Max eigenvalue = -1774.151
 res
b a alpha h lv_a
0.95 19 1.2211464 0.0682689 -35696.16
 gr
Gradient = -1.333E-8 -3.492E-9
 elas
Max Elasticity = 2.626E-13
 vp
Max eigenvalue = -1791.916
 outp
Maximum likelihood at b a alpha h lv_a
 0.31 0.4492754 3.0012237 0.0095276 -34003.5

```

Le graphique 4.23 résume le balayage. Il montre que le maximum se situe autour de  $a^* = 0.45$ . Pour mieux le voir, on peut lancer un deuxième balayage sur l'intervalle  $[0.1, 1]$ , ce qui donne le graphique 4.24. On voit également que le modèle de Weibull est disqualifié, puisque sa log-vraisemblance maximale est de  $-35853.29$  alors que celle que fournit le modèle de Burr avec  $a^* = 0.45$  est de  $-34003.5$ .

Nous pouvons maintenant passer à l'estimation proprement dite. On prend



GRAPHIQUE 4.24 – LOI DE BURR,  $\ell^*(a)$ , BALAYAGE DE  $a = 0.1$  À  $a = 1$

la valeur précédente comme valeur initiale, soit :

$$(a_0, \alpha_0, h_0) = (0.4492754, 3.0012237, 0.0095276).$$

On utilise le programme d'estimation suivant.

```
1. proc iml;
 start lecture;
 nom_y={&y}; nom_d={&d};
 use &tab; read all var(nom_y) into yi;
 read all var(nom_d) into di;
 nd=sum(di); n=nrow(yi); nc=n-nd; lyi=log(yi);
 print "Time variable =" nom_y,
 "Censoring variable (=0 if censored) =" nom_d,
 "Number of observations =" n,
 "Complete =" nd, "Censored =" nc;
 finish lecture;
 Module de lecture habituel.

2. start lv(theta) global(di,yi,lyi,n,nd);
 t1=theta[1]; t2=theta[2]; t3=theta[3];
 a=exp(t1); alpha=exp(t2); h=exp(t3);
 lv=n#a#t1+nd#(t1+t2+t3)+sum((alpha-1)#di#lyi
```

```

-(a+di)#log(a+h#yi##alpha));
return (lv);
finish;

```

Calcul de la log-vraisemblance définie sur les trois paramètres.

```

3. start sc(theta) global(di,yi,lyi,a,nd,n);
t1=theta[1]; t2=theta[2]; t3=theta[3];
a=exp(t1); alpha=exp(t2); h=exp(t3);
hya=h#yi##alpha;
zli=a+hya;
z2i=hya/(a+hya);
der1=n#a#(1+t1)+nd-a#sum(log(zli)+(a+di)/zli);
der2=nd+alpha#sum((di-(a+di)#z2i)#lyi);
der3=nd-sum((a+di)#z2i);
der=(der1||der2||der3);
return (der);
finish;

```

Calcul du gradient défini sur les trois paramètres.

4. start estim;  
Début du module d'estimation.
5. use &inp; read all var {a alpha h} into dzeta0;  
Lecture du résultat de l'estimation par balayage.
6. theta0=log(dzeta0);  
Calcul de la valeur initiale du paramètre d'optimisation.
7. opt=j(11,1,0); opt[1]=1; opt[2]=5;  
call nlpnra(rc,theta,"lv",theta0,opt) grd="sc";  
Optimisation par la méthode de Newton-Raphson avec gradient analytique.
8. call nlpfdd(f,g,he,"lv",theta) grd="sc";  
Calcul de la log-vraisemblance, du gradient et du hessien à l'optimum.
9. theta=t(theta); g=t(g); elas=g/f#theta;  
vp=eigval(he);  
On remet le paramètre et le gradient en colonne, puis on calcule les élasticités de la log-vraisemblance par rapport aux paramètres et les valeurs propres du hessien.
10. print "log-likelihood =" f,  
"Gradient =", g,  
"Elasticities =" elas,

```
"Eigenvalues of the hessian =" vp;
Affichage des résultats de l'optimisation.
```

11. `vtheta=inv(-he);`  
Estimation de la covariance asymptotique de l'estimateur du paramètre d'optimisation.
12. `b=exp(theta);`  
Paramètre d'intérêt.
13. `a=b[1]; alpha=b[2]; h=b[3];`  
`dg=(a | 0 | 0)/(0 | alpha | 0)/(0 | 0 | h);`  
Dérivée du paramètre d'intérêt par rapport au paramètre d'optimisation.
14. `vb=dg*vtheta*t(dg);`  
`sb=sqrt(vecdiag(vb));`  
`tb=abs(b)/sb;`  
`pb=1-probchi(tb##2,1);`  
`res_b=(b | sb | tb | pb);`  
Calcul des statistiques associées au paramètre d'intérêt et stockage des résultats dans une matrice `res_b`.
15. `labc="Estim"// "Ecart_Type"// "Student"// "PValue";`  
Noms de colonne de la matrice de résultat.
16. `labr="ALPHA"// "H"// "A"// "Mean"// "Median";`  
Noms de ligne de la matrice de résultat.
17. `test=alpha#a;`  
`if test > 1 then do;`  
`mt=a#(a/h)#(1/alpha)#beta(a-1/alpha,1+alpha);`  
`dmt=(mt#((1/alpha+1)/a+digamma(a-1/alpha)`  
`-digamma(a+1)) | |`  
`mt#(-log(a/h)+digamma(a-1/alpha)`  
`-digamma(1+1/alpha))/(alpha##2) | |`  
`-mt/alpha/h);`  
`vmt=dmt*vb*t(dmt);`  
`smt=sqrt(vecdiag(vmt));`  
`tmt=abs(mt)/smt;`  
`pmt=1-probchi(tmt##2,1);`  
`res_b=res_b/(mt | | smt | | tmt | | pmt);`  
`end;`  
Estimation de la durée moyenne quand elle existe, ainsi que des statistiques associées.

- ```

18. else do; print "La moyenne n'existe pas";
    res_b=res_b/(. || . || . || .);
    end;
    Quand la moyenne n'existe pas, on met des valeurs manquante dans la
    matrice des résultats.
19. md=(a/h)##(1/alpha)#(2##(1/a)-1)##(1/alpha);
    dmd=(
    md#(1-(log(2)/a)#2##(1/a)/(2##(1/a)-1))
    /(alpha#a) ||
    -md/alpha##2#(log(a/h)+log(2##(1/a)-1)) ||
    -md/alpha/h);
    vmd=dmd*vb*t(dmd);
    smd=sqrt(vecdiag(vmd));
    tmd=abs(md)/smd;
    pmd=1-probchi(tmd##2,1);
    res_b=res_b/(md || smd || tmd || pmd);
    Calcul de la médiane et des statistiques associées.
20. if alpha>1 then do;
    tmax=((alpha-1)#a/h)##(1/alpha);
    print "The hasard is maximum at T =" tmax;
    end;
    Calcul du maximum du hasard, quand il existe et affichage du résultat.
21. test_par=abs(alpha-1)/sb[2];
    pval_par=1-probchi(test_par##2,1);
    Test d'adéquation du modèle de Pareto ( $H_0: \alpha = 1$ ).
22. test_log=abs(a-1)/sb[1];
    pval_log=1-probchi(test_log##2,1);
    Test d'adéquation du modèle log-logistique ( $H_0: a = 1$ ).
23. print "Test H0 : Alpha=1 (Pareto) =" test_par,
    "P-value =" pval_par, , ,
    "Test H0 : A=1 (log-logistic) =" test_log,
    "p-value =" pval_log;
    Affichage des résultats des tests.
24. print "Paramètres de la loi de Burr type 12",
    res_b[colname=labc rowname=labr];
    Affichage des résultats de l'estimation.
25. finish estim;
    Fin du module d'estimation.

```

- ```

26. start prev;
 tmin=min(yi); tmax=max(yi); tby=1;
 do time=tmin to tmax by tby;
 si=(a/(a+h#time##alpha))##a;
 hi=a#h#alpha#time##(alpha-1)/(a+h#time##alpha);
 fi=hi#si;
 ci=1-si;
 res=res//((time||si||hi||fi||ci);
 end;
 nres="&y"||"SDF"||"Hasard"||"Density"||"CDF";
 finish prev;
 Module de prévision standard. On change juste la définition des fonc-
 tions.

27. start sortie;
 create &outp from res_b[colname=labc];
 append from res_b;
 create _np from labr[colname="Param"];
 append from labr;
 create &outi from res[colname=nres];
 append from res;
 finish sortie;
 Module de sortie standard.

28. run lecture;
 run estim;
 run prev;
 run sortie;
 Lancement des modules.

29. quit;
 On sort de SAS-IML.

30. data &outp; merge _np &outp;
 run;
 Ajout des noms de ligne au tableau des résultats.

 On obtient la sortie suivante.

```

**Sortie SAS 4.14.**

```

 nom_y
Time variable = DUREE_EMP
 nom_d
Censoring variable (=0 if censored) = PRESENCE

```

```

 n
Number of observations = 12695
 nd
Complete = 11277
 nc
Censored = 1418

```

```

dzeta0
0.4492754 3.0012237 0.0095276

```

```

Optimization Start
Parameter Estimates

```

| N Parameter | Estimate  | Gradient<br>Objective<br>Function |
|-------------|-----------|-----------------------------------|
| 1 X1        | -0.800119 | -7.985329                         |
| 2 X2        | 1.099020  | -3.088098E-8                      |
| 3 X3        | -4.653559 | -1.280932E-8                      |

```

Value of Objective Function = -34003.49793

```

```

Newton-Raphson Optimization with Line Search

```

```

Minimum Iterations 0
Maximum Iterations 200
Maximum Function Calls 500
ABSGCONV Gradient Criterion 0.00001
GCONV Gradient Criterion 1E-8
GCONV2 Gradient Criterion 0
ABSFCONV Function Criterion 0
FCONV Function Criterion 2.220446E-16
FCONV2 Function Criterion 0
FSIZE Parameter 0
ABSXCONV Parameter Change Criterion 0
XCONV Parameter Change Criterion 0
XSIZE Parameter 0
ABSCONV Function Criterion 1.340781E154
Line Search Method 2
Starting Alpha for Line Search 1
Line Search Precision LSPRECISION 0.9
DAMPSTEP Parameter for Line Search .
MAXSTEP Parameter for Line Search 0
FD Derivatives: Accurate Digits in Obj.F 15.653559775
Singularity Tolerance (SINGULAR) 1E-8

```

```

Newton-Raphson Optimization with Line Search

```

```

Without Parameter Scaling

```

```

CRP Jacobian Computed by Finite Differences

```

Parameter Estimates

3

Optimization Start

Active Constraints 0 Objective Function -34003.49793  
 Max Abs Gradient 7.9853286205  
 Element

|      | Rest | Func  | Act | Objective | Obj Fun | Max Abs  | Step  |
|------|------|-------|-----|-----------|---------|----------|-------|
| Iter | arts | Calls | Con | Function  | Change  | Gradient | Size  |
| 1    | 0    | 2     | 0   | -34003    | 0.00593 | 7.1864   | 0.100 |

Slope of  
Search  
Direction

-0.0624

|       | X1           | X2           | X3           |
|-------|--------------|--------------|--------------|
| Parms | -0.800901141 | 1.0994122731 | -4.654675016 |
| Grad  | 7.1863872515 | -0.000208823 | 0.000070202  |

|      | Rest | Func  | Act | Objective | Obj Fun | Max Abs  | Step  |
|------|------|-------|-----|-----------|---------|----------|-------|
| Iter | arts | Calls | Con | Function  | Change  | Gradient | Size  |
| 2*   | 0    | 3     | 0   | -34003    | 0.0127  | 5.0758   | 0.293 |

Slope of  
Search  
Direction

-0.0505

|       | X1           | X2           | X3          |
|-------|--------------|--------------|-------------|
| Parms | -0.80296437  | 1.1004474281 | -4.65762172 |
| Grad  | 5.0758295268 | -0.001383321 | 0.000787799 |

|      | Rest | Func  | Act | Objective | Obj Fun | Max Abs  | Step  |
|------|------|-------|-----|-----------|---------|----------|-------|
| Iter | arts | Calls | Con | Function  | Change  | Gradient | Size  |
| 3*   | 0    | 4     | 0   | -34003    | 0.0126  | 0.0144   | 1.000 |

Slope of  
Search  
Direction

-0.0252

|       | X1           | X2           | X3           |
|-------|--------------|--------------|--------------|
| Parms | -0.807930173 | 1.1029405796 | -4.664723164 |
| Grad  | -0.014422095 | -0.007120969 | 0.0041929156 |

|      | Rest | Func  | Act | Objective | Obj Fun  | Max Abs  | Step  |
|------|------|-------|-----|-----------|----------|----------|-------|
| Iter | arts | Calls | Con | Function  | Change   | Element  | Size  |
| 4*   | 0    | 5     | 0   | -34003    | 3.589E-8 | 1.663E-7 | 1.000 |

Slope of  
Search  
Direction  
-72E-9

|       | X1           | X2           | X3           |
|-------|--------------|--------------|--------------|
| Parms | -0.807925456 | 1.1029398649 | -4.664725305 |
| Grad  | -1.662975E-7 | -1.235658E-7 | 1.5397745E-8 |

#### Optimization Results

|                    |              |                    |              |
|--------------------|--------------|--------------------|--------------|
| Iterations         | 4            | Function Calls     | 6            |
| Hessian Calls      | 5            | Active Constraints | 0            |
| Objective Function | -34003.46674 | Max Abs Gradient   | 1.6629747E-7 |
|                    |              | Element            |              |
| Slope of Search    | -7.192195E-8 | Ridge              | 0.0001       |
| Direction          |              |                    |              |

GCONV convergence criterion satisfied.

#### Optimization Results

##### Parameter Estimates

| N | Parameter | Estimate  | Gradient<br>Objective<br>Function |
|---|-----------|-----------|-----------------------------------|
| 1 | X1        | -0.807925 | 0.000000166                       |
| 2 | X2        | 1.102940  | 0.000000124                       |
| 3 | X3        | -4.664725 | -1.539775E-8                      |

Value of Objective Function = -34003.46674

f  
log-likelihood = -34003.47  
Gradient =  
g  
1.663E-7  
1.2357E-7  
-1.54E-8

```

 elas
Elasticities = 3.951E-12
 -4.01E-12
 -2.11E-12

 vp
Eigenvalues of the hessian = -220.5592
 -2575.609
 -60621.12

 tmax
The hasard is maximum at T = 4.536936

 test_par
Test H0 : Alpha=1 (Pareto) = 37.754347

 pval_par
P-value = 0

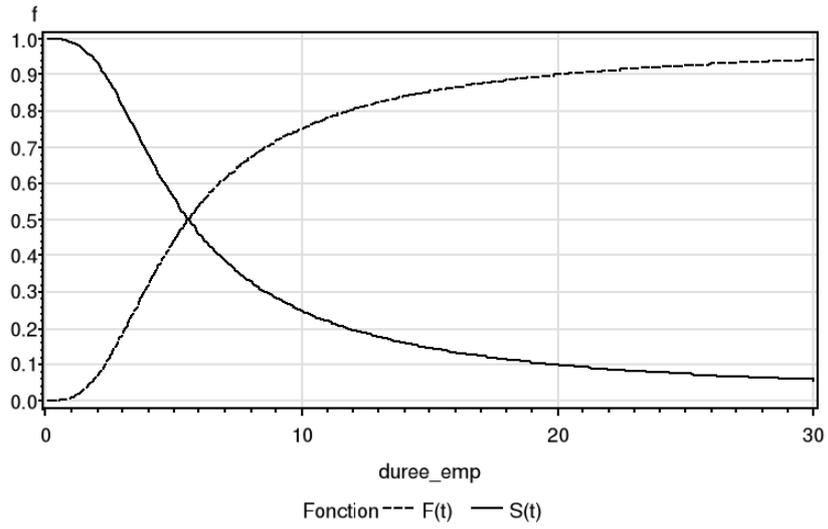
 test_log
Test H0 : A=1 (log-logistic) = 39.792983

 pval_log
p-value = 0

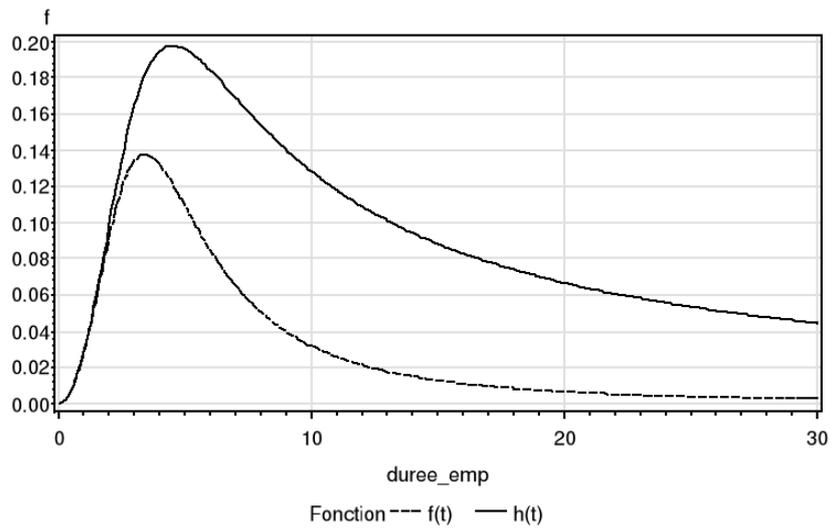
Paramètres de la loi de Burr type 12
 res_b
 Estim Ecart_Type Student PValue
ALPHA 0.4457819 0.0139275 32.00724 0
H 3.0130109 0.0533187 56.50951 0
A 0.0094218 0.0005692 16.553455 0
Mean 11.516164 0.479467 24.01868 0
Median 5.569846 0.043068 129.32666 0

```

Les résultats indiquent une log-vraisemblance de  $-34003.47$ , qui représente la valeur la plus élevée de tous les modèles estimé sur ces données. Il s'agit donc du meilleur ajustement paramétrique que nous avons obtenu. Enfin, les test de Student concluent au rejet des cas particuliers de Pareto et log-logistique.



GRAPHIQUE 4.25 – LOI DE BURR,  $\hat{S}_T(t)$  ET  $\hat{F}_T(t)$



GRAPHIQUE 4.26 – LOI DE BURR,  $\hat{h}_T(t)$  ET  $\hat{f}_T(t)$



## CHAPITRE 5

# Estimation du modèle à hasard constant par morceaux

CE MODÈLE a un statut particulier. On peut l'interpréter comme un cas intermédiaire entre l'estimation paramétrique et l'estimation non paramétrique. L'estimation non paramétrique, que nous verrons dans le chapitre suivant, se caractérise par une estimation point par point de la fonction de survie. L'estimateur à hasard constant par morceaux réalise le même type d'estimation mais par intervalle. On peut donc l'interpréter comme une forme hybride entre l'estimation paramétrique que nous venons de voir dans le chapitre précédent et l'estimation non paramétrique que nous verrons dans le chapitre suivant. Sa forme particulière justifie sa place dans un chapitre séparé. De plus, pour estimer ce modèle, nous adopterons une programmation par blocs, afin d'obtenir un temps de calcul raisonnable. Ceci nous amènera à réécrire la log-vraisemblance afin de l'adapter à une programmation plus efficace. Le type de programmation est donc assez différent de ceux des modèles paramétriques à trois paramètres que nous avons vu jusqu'à maintenant. Ici, c'est l'utilisateur qui choisit le nombre de paramètres qu'il souhaite estimer. Les paramètres sont simplement les valeurs que prend la fonction de hasard sur chacun des intervalles choisis par l'utilisateur. Dans notre application, les durées sont mesurées en années, de sorte que l'on peut effectuer une estimation point par point pour certaines valeurs de la durée, et une estimation par intervalles pour d'autres valeurs. Le choix des intervalles dépend du nombre d'observations disponibles dans chacun des intervalles. Idéalement, on souhaite toujours imposer le moins de contraintes possibles à la distribution de la durée. On prendra donc des intervalles très petits pour les durées où il y a beaucoup d'observations, et des intervalles plus grands quand il y a très peu de point. On peut donc

voir le modèle à hasard constant par morceaux comme la recherche du meilleur compromis entre ce que l'on souhaite estimer, un hasard sans contrainte, et ce que l'on peut estimer. En effet, nous verrons dans le chapitre suivant que les estimateurs non paramétriques estiment le hasard de manière très imprécise au voisinage des durées où il y a peu d'observations. En regroupant les durées par intervalles, on augmente le nombre d'observations disponibles localement pour l'estimation de la fonction de hasard. Le prix à payer est un biais si le hasard n'est pas constant sur l'intervalle.

## 5.1 Le modèle

Le modèle à hasard constant par morceaux est également appelé modèle exponentiel par morceaux; il est utilisé pour réaliser une approximation de l'approche non paramétrique de la fonction de hasard. Théoriquement, sa définition est très proche de celle de l'estimateur actuariel que nous rencontrerons dans le chapitre sur l'estimation non paramétrique. Ce modèle est très intéressant car on met très peu de contraintes sur la fonction de hasard. Pour cela, on commence par choisir les intervalles de durées sur lesquels le hasard est constant. Par exemple, on choisit les  $M$  intervalles suivants :

$$]c_0, c_1], ]c_1, c_2], \dots, ]c_{M-1}, c_M]$$

et l'on définit la fonction de hasard par :

$$h(t) = \begin{cases} h_1 & t \in ]c_0, c_1] \\ h_2 & t \in ]c_1, c_2] \\ \vdots & \\ h_M & t \in ]c_{M-1}, c_M] \end{cases}$$

on peut également utiliser une variable indicatrice de l'intervalle  $m \in \{1, \dots, M\}$ ,  $I_m$ , définie par :

$$\delta_m(t) = \begin{cases} 1 & \text{si } c_{m-1} < t \leq c_m \\ 0 & \text{sinon} \end{cases}$$

de sorte que :

$$h(t) = \sum_{m=1}^M h_m \delta_m(t)$$

où les valeurs  $h_1, \dots, h_M$  sont estimées librement. De ce fait, la fonction de hasard peut prendre toutes les formes possibles si le nombre d'intervalles est suf-

fisamment grand. Ceci suppose d'avoir suffisamment d'observations dans chaque intervalle. Le hasard cumulé est donné par :

$$\begin{aligned} H(t) &= \int_0^t h(x) dx \\ &= \int_0^t \sum_{m=1}^M h_m \delta_m(x) dx \\ &= \sum_{m=1}^M h_m \int_0^t \delta_m(x) dx \end{aligned}$$

avec

$$\int_0^t \delta_m(x) dx = \begin{cases} t - c_{m-1} & \text{si } c_{m-1} < t \leq c_m \\ 0 & \text{sinon} \end{cases}$$

donc le hasard cumulé sur le  $m$ -ième intervalle,  $t \in ]c_{m-1}, c_m]$ , est égal à

$$H_m(t) = \begin{cases} h_1(t - c_0) & \text{si } m = 1 \\ \sum_{j=1}^{m-1} h_j(c_j - c_{j-1}) + h_m(t - c_{m-1}) & \text{si } m \geq 2 \end{cases}$$

qui correspond à la somme des surfaces des rectangles situés sous le graphe de la fonction de hasard. En explicitant, on trouve :

$$H(t) = \begin{cases} h_1(t - c_0) & t \in ]c_0, c_1] \\ h_1(c_1 - c_0) + h_2(t - c_1) & t \in ]c_1, c_2] \\ \vdots & \vdots \\ \sum_{j=1}^{m-1} h_j(c_j - c_{j-1}) + h_m(t - c_{m-1}) & t \in ]c_{m-1}, c_M] \end{cases}$$

on remarque que cette fonction est continue. La survie sur l'intervalle  $m$  s'écrit :

$$S_m(t) = \exp(-H_m(t)),$$

qui correspond au  $m$ -ième intervalle de  $S(t) = \exp(-H(t))$ , et l'on retrouve la densité correspondante par la relation :

$$f_m(t) = h_m S_m(t).$$

Pour calculer la durée moyenne, on utilise la propriété (annexe A.1) :

$$E(T) = \int_0^{+\infty} S(x) dx,$$

et le fait que la distribution est exponentielle par morceaux. On obtient, avec  $c_0 = 0$  et  $c_M = \{+\infty\}$  :

$$\begin{aligned}
E(T) &= \int_0^{c_1} \exp(-h_1 x) dx + \dots + \int_{c_{m-1}}^{c_m} \exp(-h_m x) dx + \dots \\
&\quad + \int_{c_{M-1}}^{+\infty} \exp(-h_M x) dx \\
&= \frac{1}{h_1} (1 - e^{-h_1 c_1}) + \dots + \frac{1}{h_m} (e^{-h_m c_{m-1}} - e^{-h_m c_m}) + \dots + \frac{1}{h_M} e^{-h_M c_{M-1}} \\
&= \sum_{m=1}^{M-1} \frac{1}{h_m} (\exp(-h_m c_{m-1}) - \exp(-h_m c_m)) + \frac{1}{h_M} \exp(-h_M c_{M-1}),
\end{aligned}$$

Pour calculer la médiane, on remarque que la fonction  $H(t)$  est continue, donc  $S(t)$  l'est également et on peut calculer le quantile d'ordre  $\omega$ ,  $Q_\omega$ , par la relation :

$$S(Q_\omega) = 1 - \omega.$$

La médiane s'obtient par la relation :

$$S(\text{Me}) = \frac{1}{2},$$

donc si la fonction de survie atteint  $\frac{1}{2}$  sur l'intervalle  $m$ , on a :

$$\exp(-H_m(\text{Me})) = \frac{1}{2} \Leftrightarrow H_m(\text{Me}) = \ln 2.$$

Si  $m = 1$ , ceci implique :

$$h_1(\text{Me} - c_0) = \ln 2$$

donc :

$$\text{Me} = c_0 + \frac{\ln 2}{h_1},$$

soit  $\text{Me} = (\ln 2) / h_1$  dans le cas usuel  $c_0 = 0$ . Si  $m \geq 2$ , on a :

$$\begin{aligned}
\sum_{j=1}^{m-1} h_j (c_j - c_{j-1}) + h_m (\text{Me} - c_{m-1}) &= \ln 2 \\
\Leftrightarrow \text{Me} &= c_{m-1} + \frac{1}{h_m} \left( \ln 2 - \sum_{j=1}^{m-1} h_j (c_j - c_{j-1}) \right)
\end{aligned}$$

## 5.2 Log-vraisemblance

La log-vraisemblance de l'échantillon est donnée par :

$$\ell(\zeta) = \sum_{i=1}^N d_i \ln h_T(y_i) - H_T(y_i),$$

avec :

$$\zeta = \begin{pmatrix} h_1 \\ \vdots \\ h_M \end{pmatrix},$$

où  $H_T(\cdot)$  est le hasard intégré. Si l'on fait apparaître les intervalles, on obtient :

$$\ell(\zeta) = \sum_{i=1}^N \sum_{m=1}^M \delta_{im} (d_i \ln h_m - H_m(y_i)),$$

avec :

$$\delta_{im} = \begin{cases} 1 & \text{si } y_i \in ]c_{m-1}, c_m] \\ 0 & \text{sinon} \end{cases}, \quad m = 1, \dots, M$$

le problème de cette formulation provient de la double somme. Elle consiste, pour chaque individu, à additionner  $M - 1$  zéros au chiffre qui nous intéresse. Autant cela ne pose aucun problème théorique, autant cela pose un problème de temps de calcul car, pour chaque individu, on effectue  $M - 1$  opérations inutiles parmi  $M$ , soit  $N \times (M - 1)$  au total. Nous allons donc trier les individus selon leur durée  $y_i$  et raisonner par bloc afin d'éviter ces calculs inutiles. Une fois le tri effectué, on obtient des blocs d'individus qui figurent dans le même intervalle et dont les indices sont définis par :

$$\begin{aligned} & i_1^-, i_1^+ \text{ pour } h_1 \\ & i_2^-, i_2^+ \text{ pour } h_2 \\ & \vdots \\ & i_M^-, i_M^+ \text{ pour } h_M \end{aligned}$$

avec, par définition :

$$i_1^- = 1, \quad i_{m-1}^+ = i_m^- - 1 \text{ et } i_M^+ = N.$$

Considérons maintenant la première partie de la log-vraisemblance :

$$\begin{aligned}\ell_1(\zeta) &= \sum_{i=1}^N \sum_{m=1}^M \delta_{im} d_i \ln h_m \\ &= \sum_{i=i_1^-}^{i_1^+} d_i \ln h_1 + \dots + \sum_{i=i_M^-}^{i_M^+} d_i \ln h_M \\ &= \sum_{m=1}^M D_m \ln h_m,\end{aligned}$$

où  $D_m$  est le nombre d'individus *non censurés* dans l'intervalle  $m$  :

$$D_m = \sum_{i=i_m^-}^{i_m^+} d_i.$$

Cette réécriture de la log-vraisemblance donne lieu à un temps de calcul beaucoup plus court que sa version originale, car elle n'implique que la somme de  $M$  quantités au lieu de  $N \times M$ . Considérons maintenant la seconde partie de la log-vraisemblance :

$$\ell_2(\zeta) = - \sum_{i=1}^N \sum_{m=1}^M \delta_{im} H_m(y_i),$$

avec

$$H_m(t) = \begin{cases} h_1(t - c_0) & \text{si } m = 1 \\ \sum_{j=1}^{m-1} h_j(c_j - c_{j-1}) + h_m(t - c_{m-1}) & \text{si } m \geq 2 \end{cases}$$

Considérons le calcul de

$$\sum_{i=1}^N \sum_{m=1}^M \delta_{im} H_m(y_i),$$

pour le premier intervalle, on a :

$$[h_1(y_i - c_0)], \quad i_1^- \leq i \leq i_1^+$$

et l'on additionne ses composantes, ce qui donne :

$$h_1 \left( \sum_{i=i_1^-}^{i_1^+} y_i - N_1 c_0 \right),$$

pour les  $M - 1$  autres blocs, on décompose la somme en deux parties :

$$\sum_{i=i_m^-}^{i_m^+} \sum_{j=1}^{m-1} h_j(c_j - c_{j-1}) \quad \text{et} \quad \sum_{i=i_m^-}^{i_m^+} h_m(y_i - c_{m-1})$$

la première somme est constante par rapport à  $y_i$ , cette quantité est donc la même pour tous les individus de l'intervalle  $m$ . Elle est égale à :

$$N_m \sum_{j=1}^{m-1} h_j (c_j - c_{j-1}),$$

où  $N_m$  est le nombre total d'individus de l'intervalle  $m$  (censurés ou non). Ici on pose :

$$\underline{h}_k = (h_1, \dots, h_k)' \text{ et } \underline{g}_k = (c_1 - c_0, \dots, c_k - c_{k-1})',$$

et l'on obtient :

$$N_m \sum_{j=1}^{m-1} h_j (c_j - c_{j-1}) = N_m \underline{h}'_{m-1} \underline{g}_{m-1}$$

enfin, la seconde composante de la somme est égale à :

$$\sum_{i=i_m^-}^{i_m^+} h_m (y_i - c_{m-1}) = h_m \left( \sum_{i=i_m^-}^{i_m^+} y_i - N_m c_{m-1} \right).$$

A ce stade, on peut introduire la notation :

$$S_m = \sum_{i=i_m^-}^{i_m^+} y_i,$$

ce qui fait que, globalement, on obtient une nouvelle décomposition :

$$\begin{aligned} \ell(\zeta) &= \sum_{i=1}^N \sum_{m=1}^M \delta_{im} (d_i \ln h_m - H_m(y_i)) \\ &= \underbrace{\sum_{m=1}^M (D_m \ln h_m - h_m (S_m - N_m c_{m-1}))}_{\ell_a(\zeta)} - \underbrace{\sum_{m=2}^M N_m \underline{h}'_{m-1} \underline{g}_{m-1}}_{\ell_b(\zeta)}, \end{aligned}$$

l'intérêt principal de cette formulation est qu'elle n'implique pas de faire des doubles sommes. De plus, on peut calculer les statistiques  $S_m$  et  $N_m$  avant de lancer l'optimisation, de sorte qu'on ne les refera pas à chaque itération de l'algorithme d'optimisation. On se contente ici de faire une somme sur le nombre d'intervalles, ce qui est très rapide.

### 5.3 Estimation

Pour l'estimation, on utilisera un paramétrage qui garantit la positivité des hasards. Nous prendrons donc :

$$h_j = \exp(\theta_j), \quad j = 1, \dots, M$$

et maximiserons :

$$\ell_{\zeta}(\theta) = \ell(\zeta(\theta)),$$

par rapport à  $\theta$ . Nous utiliserons un algorithme de Newton-Raphson avec dérivées numériques. Pour la valeur initiale, nous prendrons simplement le cas particulier de la loi exponentielle où tous les hasards sont égaux :

$$h_j^{(0)} = \frac{N_d}{\sum_{i=1}^N y_i} \text{ et } \theta_j^{(0)} = \ln h_j^{(0)}, \forall j = 1, \dots, M,$$

où  $N_d$  est le nombre de durées complètes dans l'échantillon.

## 5.4 Statistiques utiles

On obtient des estimateurs convergents des durées médiane et moyenne en remplaçant  $h$  par son estimateur du maximum de vraisemblance. En gardant la convention  $c_0 = 0$ , on obtient donc :

$$\hat{E}(T) = \sum_{m=1}^{M-1} \frac{1}{\hat{h}_m} (\exp(-\hat{h}_m c_{m-1}) - \exp(-\hat{h}_m c_m)) + \frac{1}{\hat{h}_M} \exp(-\hat{h}_M c_{M-1}).$$

et

$$\hat{Me} = \begin{cases} c_0 + \frac{\ln 2}{\hat{h}_1} & \text{si } m = 1 \\ c_{m-1} + \frac{1}{\hat{h}_m} \left( \ln 2 - \sum_{j=1}^{m-1} \hat{h}_j (c_j - c_{j-1}) \right) & \text{si } m \geq 2 \end{cases}$$

## 5.5 Programmation XII : Programmation par bloc

Le programme d'estimation est le suivant. Nous le commentons ligne à ligne.

```
1. %let tab=duree;
 %let y=duree_emp;
 %let d=presence;
 %let outp=_outp;
 %let outi=_outi;
```

Macro-variables standard : nom du tableau contenant les données (&tab), nom de la variable de durée (&y), nom de l'indicatrice de durée complète (&d), nom du tableau de sortie des paramètres (&outp) et nom du tableau de sortie des prévisions (&outi).

2. `%let threshold=1 2 3 4 5 6 7 8 9 10 12 14 16 18 20  
24 28 32 36 40 44 ;`  
 Liste des seuils  $c_1, \dots, c_{M-1}$ . Par convention on fixe  $c_0 = 0$  et  $c_M = \{+\infty\}$ .  
 On réalise donc une estimation point par point de  $t = 1$  à  $t = 10$ , ensuite on utilise des intervalles de deux ans, de  $]10, 12]$  à  $]18, 20]$ , enfin on utilise des intervalles de 4 ans, de  $]20, 24]$  à  $]40, 44]$ , le dernier intervalle étant implicitement défini par  $]44, +\infty[$ . Il y aura donc 22 intervalles et les 22 paramètres du modèle seront les hasards constants sur chacun de ces intervalles.
3. `proc iml;`  
 Début de la procédure IML.
4. `start lecture;`  
`nom_y={&y};`  
`nom_d={&d};`  
`use &tab; read all var(nom_y) into yi;`  
`read all var(nom_d) into di;`  
`nd=sum(di); n=nrow(yi); nc=n-nd;`  
`print "Time variable =" nom_y,`  
`"Censoring variable (=0 if censored) =" nom_d,`  
`"Number of observations =" n,`  
`"Complete =" nd,`  
`"Censored =" nc;`  
`finish lecture;`  
 Module de lecture identique aux précédents.
5. `start lv(theta) global(di,yi,range,dc,c,m);`  
 Fonction calculant la log-vraisemblance.
6. `h=exp(theta);` On passe du paramètre d'optimisation au paramètre d'intérêt  $h_j = \exp(\theta_j)$ .
7. `lva=sum(range[,8]#log(t(h))-t(h)#(range[,10]  
-range[,6]#(0/c)));`  
 Première partie de la log-vraisemblance  $\ell_a$ :

$$\sum_{m=1}^M (D_m \ln h_m - h_m (S_m - N_m c_{m-1})),$$

les quantités utiles sont stockées dans une matrice nommée `range`. Pour l'instant, on retiendra que :

$$\begin{aligned} \text{range}[m, 8] &= D_m, \\ \text{range}[m, 10] &= S_m, \\ \text{range}[m, 6] &= N_m. \end{aligned}$$

d'autre part, le vecteur  $c$  contient les seuils suivants :

$$\begin{pmatrix} c_1 \\ \vdots \\ c_{M-1} \end{pmatrix},$$

on a donc :

$$(0//c) = \begin{pmatrix} 0 \\ c_1 \\ \vdots \\ c_{M-1} \end{pmatrix}$$

de sorte que le  $m$ -ième élément du vecteur  $c$  contient en fait le  $(m-1)$ -ième seuil. Le terme à l'intérieur de la somme est donc un vecteur colonne contenant :

$$\begin{pmatrix} D_1 \ln h_1 - h_1 (S_1 - N_1 c_0) \\ \vdots \\ D_M \ln h_M - h_M (S_M - N_M c_{M-1}) \end{pmatrix}$$

et l'on additionne ses composantes pour obtenir la première partie de la log-vraisemblance.

8.  $lvb=0$  ;

Initialisation de la seconde partie de la log-vraisemblance qui devra être égale à :

$$\sum_{m=2}^M N_m \underline{h}'_{m-1} \underline{g}_{m-1}$$

9. do  $j=2$  to  $m$  ;

$nj=\text{range}[j, 6]$  ;

Nombre d'observations total dans l'intervalle  $m$ ,  $N_m$ .

10.  $rgc j=1 : (j-1)$  ;

Liste servant à sélectionner les éléments de 1 à  $m-1$  (Ici la somme va de  $j=1$  à  $m$ , où  $j$  correspond à la notation théorique  $m$  et  $m$  correspond à la notation théorique  $M$ ).

11.  $lvb=lvb+nj\#(t(h[rgcj])*dc[rgcj]);$   
Ajout de l'élément  $N_m h'_{m-1} \underline{g}_{m-1}$  à la somme. On obtient  $\ell_b$ .
12. `end;`  
Fin de la boucle.
13.  $lv=lva-lvb;$   
Calcul de la log-vraisemblance.
14. `return (lv);`  
Renvoi du calcul vers l'appel de la fonction.
15. `finish;`  
Fin de la fonction calculant la log-vraisemblance.
16. `start estim;`  
Début du module d'estimation.
17. `print "Estimation of the parameters of the  
piecewise constant  
hazard distribution", , ;`  
On rappelle la distribution estimée.
18. `c=t({&threshold}); call sort(c,1);`  
On range les seuils dans un vecteur colonne (d'où la transposition) et on trie les seuils par ordre croissant, en cas d'erreur de l'utilisateur.
19. `nc=nrow(c); m=nc+1;`  
 $nc$  est le nombre de seuils entrés par l'utilisateur,  $m$  est le nombre d'intervalles. On a  $nc=m-1$ .
20. `dc=j(nc,1,0); dc[1]=c[1];`  
Initialisation du calcul du vecteur :
 
$$\underline{g}_{M-1} = (c_1 - c_0, \dots, c_{M-1} - c_{M-2})'$$
 le premier élément de ce vecteur  $c_1 - c_0 = c_1$  car  $c_0 = 0$ .
21. `do j=2 to nc; dc[j]=c[j]-c[j-1]; end;`  
Calcul du vecteur  $\underline{g}_{M-1}$  qui servira pour l'obtention de tous les vecteurs  $\underline{g}_k, k=2, \dots, M-1$ .
22. `n=nrow(yi);`  
Nombre d'observations,  $N$ .
23. `inter=j(n,1,0);`  
Vecteur indiquant à quel intervalle l'individu  $i$  appartient.
24. `do i=1 to n;`  
Calcul de l'intervalle auquel l'individu appartient. On compare sa durée (complète ou non) aux seuils.

```

25. if yi[i] < =c[1] then inter[i]=1;
 else do; if yi[i] > c[nc] then inter[i]=m;
 else do;
 do j=2 to nc;
 if (c[j-1] < yi[i])&(yi[i] < =c[j])
 then inter[i]=j;
 end;
 end;
 end;
end;

```

On utilise une structure arborescente pour diminuer le temps de calcul.

26.  $mi=yi \parallel di \parallel inter$ ;  
Matrice contenant la durée  $y_i$ , l'indicatrice de censure  $d_i$  et le numéro d'intervalle auquel appartient l'individu.
27. `call sort(mi,1);`  
On trie toutes les données par rapport à la durée (la première colonne), par ordre croissant. Ainsi, on pourra traiter les données par bloc d'individus appartenant au même intervalle.
28.  $yi=mi[,1]; di=mi[,2]; inter=mi[,3];$   
On récupère les données triées.
29. `range=j(m,10,0);`  
Cette matrice contient des informations sur chacun des  $m$  intervalles. Elle contient 10 colonnes. On l'initialise à 0.
30. `range[1,1]=1; range[1,2]=0; range[1,3]=c[1];`  
On commence par calculer les trois premières colonnes. Elles contiennent respectivement : le numéro d'intervalle ( $m$ ), la borne inférieure de l'intervalle ( $c_{m-1}$ ) et la borne supérieure de l'intervalle ( $c_m$ ). Pour le premier intervalle on obtient  $(1,0,c_1)$ .
31. `do j=2 to nc;`  
`range[j,1]=j;`  
`range[j,2]=range[j-1,3];`  
`range[j,3]=c[j];`  
`end;`  
Calcul pour les intervalles intermédiaires. Le vecteur  $c$  contient les seuils  $(c_1, \dots, c_{M-1})$ .
32. `range[m,1]=m; range[m,2]=c[nc]; range[m,3]=.;`  
Pour le dernier intervalle, on met la borne supérieure en valeur manquante parce qu'elle est égale à  $\{+\infty\}$ .

33. do i=1 to n;  
 Boucle de calcul du dernier indice de l'intervalle  $m : i_m^+$ . On met cet indice dans la cinquième colonne de la matrice range.  
 do j=1 to m;  
 if inter[i]=j then range[j,5]=i;  
 end;
34. end;  
 L'indice d'un individu écrase les précédents, il ne reste donc que le dernier indice à la fin de la boucle.
35. range[1,4]=1;  
 Calcul du premier indice de l'intervalle  $m : i_m^-$  que l'on met dans la quatrième colonne de la matrice range. Par définition  $i_1^- = 1$  et  $i_m^- = i_{m-1}^+ + 1$ .
36. do j=2 to m;  
 range[j,4]=range[j-1,5]+1;  
 end;  
 range[,6]=range[,5]-range[,4]+j(m,1,1);  
 Calcul du nombre total d'observations dans chaque intervalle que l'on met dans la sixième colonne :

$$N_m = i_m^+ - i_m^- + 1,$$

comme le calcul est vectoriel, on fait :

$$\begin{pmatrix} N_1 \\ \vdots \\ N_M \end{pmatrix} = \begin{pmatrix} i_1^+ \\ \vdots \\ i_M^+ \end{pmatrix} - \begin{pmatrix} i_1^- \\ \vdots \\ i_M^- \end{pmatrix} + \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}$$

en remarquant que :

$$j(m,1,1) = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}_{(M,1)}$$

37. range[,7]=range[,6]/range[m,5]#100;  
 Calcul du pourcentage d'observations dans chaque intervalle :

$$\frac{N_m}{N} \times 100,$$

on remarque que :

$$N = i_M^+ = \text{range}[m,5]$$

38. do j=1 to m;  
 Calcul du nombre d'observations complètes  $D_m$  (colonne 8) et de la somme des durées  $S_m$  (colonne 10).

39. rgj=range[j, 4]:range[j, 5];  
 Liste des indices des observations présentes dans l'intervalle j, donc :

$$\text{rgj} = (i_m^-, i_m^- + 1, \dots, i_m^+).$$

40. dj=di[rgj];  
 Vecteur des indicatrices de présences complètes des individus de l'intervalle j.

41. range[j, 8]=sum(dj);  
 Somme des indicatrices, donne le nombre de présences complètes :

$$D_m = \sum_{i=i_m^-}^{i_m^+} d_i.$$

42. yj=yi[rgj];  
 Vecteur des durées des individus de l'intervalle j.

43. range[j, 10]=sum(yj);  
 Somme des durées :

$$S_m = \sum_{i=i_m^-}^{i_m^+} y_i.$$

44. end;  
 Fin de la boucle

45. range[, 9]=range[, 8]/range[, 6]#100;  
 Calcul du pourcentage d'observations complètes :

$$\frac{D_m}{N_m} \times 100.$$

46. labrange="Intervall" || "gt\_Lower" || "le\_Upper"  
 || "Block\_start" || "Block\_end" || "Number\_Obs"  
 || "Percentage" || "Complete\_obs" ||  
 "Complete\_Pct" || "Sum\_Durations";  
 Noms de colonne de la matrice range.

47. `rangelab=compress(j(m,1,"Int_")+char(range[,1]));`  
 Noms de ligne de la matrice range, il s'agit du vecteur de caractères :

```
Int_1
 ⋮
Int_M
```

l'instruction *char* convertit les nombres en caractères.

48. `print range[colname=labrange rowname=rangelab];`  
 Affichage de la matrice range.
49. `h0=j(1,m,nd/sum(yi));`  
 Valeur initiale de l'algorithme de Newton-Raphson, on prend le cas exponentiel :

$$h_{(M,1)}^{(0)} = \begin{pmatrix} N_d / \sum y_i \\ \vdots \\ N_d / \sum y_i \end{pmatrix}.$$

50. `theta0=log(h0);`  
 Valeur initiale du paramètre d'optimisation :

$$\theta_j^{(0)} = \ln h_j^{(0)}, j = 1, \dots, M.$$

51. `opt=j(11,1,0); opt[1]=1; opt[2]=5;`  
 Initialisation de la routine `nlpnra`, on demande une maximisation (`opt[1]=1`) et un affichage maximal (`opt[2]=5`).
52. `call nlpnra(rc,theta,"lv",theta0,opt);`  
 Lancement de l'algorithme d'optimisation.
53. `call nlpfdd(lv1,g,he,"lv",theta);`  
 Calcul de la log-vraisemblance, du gradient et du hessien au maximum.
54. `theta=t(theta); g=t(g);`  
`elas=abs(theta#g/lv1); vp=eigval(he);`  
 On remet le paramètre et le gradient en colonne, puis on calcule les élasticités de la log-vraisemblance par rapport aux paramètres et les valeurs propres du hessien
55. `print "Maximum log-likelihood =" lv1,`  
`"Gradient =" g "Elasticities =" elas ,`  
`"Hessian eigenvalues =" vp;`  
 Affichage des résultats de l'optimisation.

56. `vtheta=inv(-he)` ;  
Estimation de la matrice de covariance asymptotique de l'estimateur du paramètre d'optimisation.
57. `h=exp(theta)` ;  
Retour aux paramètres d'intérêt :  $\hat{h}_j = \exp(\hat{\theta}_j)$ .
58. `dg=diag(h)` ;  
Calcul de la dérivée du paramètre d'intérêt  $h$  par rapport au paramètre d'optimisation :

$$\frac{\partial h}{\partial \theta'} = \begin{pmatrix} \exp(\theta_1) & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \exp(\theta_M) \end{pmatrix} = \begin{pmatrix} h_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & h_M \end{pmatrix}$$

59. `vh=dg*vtheta*t(dg)` ;  
Application de la formule de Slutsky :

$$\widehat{\text{Vas}}(\hat{h}) = \frac{\partial h}{\partial \theta'}(\hat{\theta}) \widehat{\text{Vas}}(\hat{\theta}) \frac{\partial h}{\partial \theta'}(\hat{\theta})'$$

60. `sh=sqrt(vecdiag(vh))` ;  
`th=abs(h)/sh` ;  
`ph=1-probchi(th##2,1)` ;  
Statistiques associées aux estimations des hasards constants par morceaux (écart-types,  $t$  de Student et probabilités critiques asymptotiques).
61. `res_h=h|sh|th|pb` ;  
Stockage des résultats dans une matrice.
62. `labc="Hasard"|"Std_Dev"|"Student"|"Pvalue"` ;  
Noms des colonnes du tableau de résultat.
63. `labr=compress(j(m,1,"h")+char(range[,1]))` ;  
Noms des lignes du tableau de résultat.
64. `print res_h[colname=labc rowname=labr]` ;  
Affichage des résultats.
65. `finish estim` ;  
Fin du module d'estimation. La moyenne et la médiane sont calculées dans le module de prévision.
66. `start ih(h,t) global(c,dc,sh)` ;  
Fonction calculant le hasard, le hasard intégré et l'estimation de l'écart-type asymptotique de l'estimateur du hasard. En sortie on obtient un vecteur ligne donnant :

$$\left( t, \hat{h}_T(t), \hat{H}_T(t), \sqrt{\widehat{\text{Var}}(\hat{h}_T(t))} \right).$$

67. `m=nrow(h)`; `nc=nrow(c)`;  
 m donne le nombre d'intervalle et nc le nombre de seuils (`nc=m-1`). Comme h et c sont des vecteurs colonne, leur nombre de ligne est égal à leur nombre d'éléments. On applique la formule :

$$H_m(t) = \begin{cases} h_1(t - c_0) & \text{si } m = 1 \\ \sum_{j=1}^{m-1} h_j(c_j - c_{j-1}) + h_m(t - c_{m-1}) & \text{si } m \geq 2 \end{cases}$$

68. `do j=1 to m`;  
   `if t<=c[1] then f=(t||h[1]|| (h[1]#t) ||sh[1]);`  
   `else do; if t>c[nc] then do;`  
     `rg=1:nc;`  
     `f=(t||h[m]|| (t(h[rg])*dc[rg]+h[m]#(t-c[nc])) ||`  
     `sh[m]);`  
     `end;`  
   `else do;`  
     `do j=2 to nc;`  
       `if (c[j-1] < t)&(t <= c[j])`  
       `then do;`  
         `rg=1:(j-1);`  
         `f=(t||h[j]|| (t(h[rg])*dc[rg]+h[j]#(t-c[j-1]))`  
         `||sh[j]);`  
       `end;`  
     `end;`  
     `end;`  
     `end;`  
   `return (f);`  
   `finish;`

La structure arborescente vise à gagner du temps de calcul.

69. `start prev`;  
 Début du module de prévision.
70. `do time=0.1 to 50 by 0.01`;  
 L'utilisateur choisit librement les bornes et le pas.
71. `prev=prev//ih(h,time)`;  
 On stocke la durée, le hasard, le hasard cumulé et l'estimation de l'écart-type asymptotique de l'estimateur du hasard.

72. end ;  
Fin de la boucle de calcul.
73. tt=prev[ , 1 ] ;  
La première colonne de prev contient les durées pour lesquelles les calculs ont été effectués.
74. ht=prev[ , 2 ] ;  
La deuxième colonne contient  $\hat{h}_T(t)$ .
75. iht=prev[ , 3 ] ;  
La troisième colonne contient  $\hat{H}_T(t)$ .
76. st=exp(-iht) ;  
On calcule la survie comme  $\hat{S}_T(t) = \exp(-\hat{H}_T(t))$ .
77. ct=1-st ;  
Calcul de la fonction de répartition,  $\hat{F}_T(t) = 1 - \hat{S}_T(t)$ .
78. ft=ht#st ;  
Calcul de la densité,  $\hat{f}_T(t) = \hat{h}_T(t) \hat{S}_T(t)$ .
79. l\_ht=ht-1.96#prev[ , 4 ] ;  
La quatrième colonne de prev contient  $\sqrt{\widehat{\text{Var}}(\hat{h}_T(t))} \triangleq \hat{\sigma}_T(\hat{h}_T(t))$ . La quantité l\_ht contient donc la borne inférieure de l'intervalle de confiance asymptotique de niveau 95%.
80. u\_ht=ht+1.96#prev[ , 4 ] ;  
Borne supérieure de l'intervalle de confiance asymptotique de niveau 95%.
81. labi="&y" || "Hasard" || "Density" || "SDF" || "CDF" ||  
"l\_ht" || "u\_ht" ;  
Noms des colonnes de la matrice de sortie.
82. resi=tt || ht || ft || st || ct || l\_ht || u\_ht ;  
Matrice de sortie.
83. ih1=ih(h, range[1, 2]) ; ih2=ih(h, range[1, 3]) ;  
Calcul de la médiane selon la méthode indiquée plus haut. On commence par calculer le tableau suivant :

$$s = \begin{pmatrix} S(c_0) & S(c_1) \\ \vdots & \vdots \\ S(c_{M-2}) & S(c_{M-1}) \end{pmatrix}$$

ih1 contient la valeur du hasard intégré à la borne inférieure de l'intervalle, ih2 contient sa valeur à la borne supérieure.

```

84. s=(exp(-ih1[1,3])|exp(-ih2[1,3]));
 do j=2 to nc;
 ih1=ih(h,range[j,2]); ih2=ih(h,range[j,3]);
 s=s/(exp(-ih1[1,3])|exp(-ih2[1,3]));
 end;

85. if min(s)>0.5 then do;
 Si la survie ne descend jamais en dessous de $\frac{1}{2}$ aux bornes, ceci signifie
 que la médiane se situe sur l'intervalle $[c_{M-1}, +\infty[$, on applique donc la
 formule spécifique à ce cas.

86. rg=1:nc;
 md=c[nc]+(log(2)-t(h[rg])*dc[rg])/h[m];
 end;

87. else do;
 Si la médiane est dans un des intervalles, on sélectionne m tel que $S(c_{m-1}) >$
 $1/2 \geq S(c_m)$. Ensuite, on applique la formule du cours.

88. do j=1 to nc;
 if (s[j,1]>0.5)&(0.5>=s[j,2]) then do;
 rg=1:(j-1);
 md=c[j-1]+(log(2)-t(h[rg])*dc[rg])/h[j];
 end;
end;
end;

89. print "Median duration =" md;
 Affichage du résultat.

90. mt=0; c=0//c;
 Initialisation de la durée moyenne. On applique la formule du cours.

91. do j=1 to m-1;
 mt=mt+(exp(-h[j]#c[j])-exp(-h[j]#c[j+1]))/h[j];
 end;
 mt=mt+exp(-h[m]#c[m])/h[m];
 print "Mean duration =" mt;
 Affichage du résultat.

92. finish prev;
 Fin du module de prévision.

```



Estimation of the parameters of the piecewise constant hazard distribution

|        | range     |          |          |             |           |            |
|--------|-----------|----------|----------|-------------|-----------|------------|
|        | Intervall | gt_Lower | le_Upper | Block_start | Block_end | Number_Obs |
| Int_1  | 1         | 0        | 1        | 1           | 344       | 344        |
| Int_2  | 2         | 1        | 2        | 345         | 1808      | 1464       |
| Int_3  | 3         | 2        | 3        | 1809        | 3541      | 1733       |
| Int_4  | 4         | 3        | 4        | 3542        | 5141      | 1600       |
| Int_5  | 5         | 4        | 5        | 5142        | 6575      | 1434       |
| Int_6  | 6         | 5        | 6        | 6576        | 7663      | 1088       |
| Int_7  | 7         | 6        | 7        | 7664        | 8534      | 871        |
| Int_8  | 8         | 7        | 8        | 8535        | 9181      | 647        |
| Int_9  | 9         | 8        | 9        | 9182        | 9649      | 468        |
| Int_10 | 10        | 9        | 10       | 9650        | 10044     | 395        |
| Int_11 | 11        | 10       | 12       | 10045       | 10654     | 610        |

|        | range      |              |              |               |
|--------|------------|--------------|--------------|---------------|
|        | Percentage | Complete_obs | Complete_Pct | Sum_Durations |
| Int_1  | 2.7097282  | 276          | 80.232558    | 344           |
| Int_2  | 11.532099  | 1337         | 91.325137    | 2928          |
| Int_3  | 13.651044  | 1617         | 93.306405    | 5199          |
| Int_4  | 12.603387  | 1469         | 91.8125      | 6400          |
| Int_5  | 11.295786  | 1298         | 90.516039    | 7170          |
| Int_6  | 8.5703033  | 1008         | 92.647059    | 6528          |
| Int_7  | 6.8609689  | 787          | 90.355913    | 6097          |
| Int_8  | 5.0964947  | 606          | 93.66306     | 5176          |
| Int_9  | 3.6864907  | 433          | 92.521368    | 4212          |
| Int_10 | 3.1114612  | 367          | 92.911392    | 3950          |
| Int_11 | 4.8050414  | 558          | 91.47541     | 6963          |

|        | range     |          |          |             |           |            |
|--------|-----------|----------|----------|-------------|-----------|------------|
|        | Intervall | gt_Lower | le_Upper | Block_start | Block_end | Number_Obs |
| Int_12 | 12        | 12       | 14       | 10655       | 11081     | 427        |
| Int_13 | 13        | 14       | 16       | 11082       | 11343     | 262        |
| Int_14 | 14        | 16       | 18       | 11344       | 11553     | 210        |
| Int_15 | 15        | 18       | 20       | 11554       | 11718     | 165        |
| Int_16 | 16        | 20       | 24       | 11719       | 11966     | 248        |
| Int_17 | 17        | 24       | 28       | 11967       | 12145     | 179        |
| Int_18 | 18        | 28       | 32       | 12146       | 12294     | 149        |
| Int_19 | 19        | 32       | 36       | 12295       | 12460     | 166        |
| Int_20 | 20        | 36       | 40       | 12461       | 12579     | 119        |
| Int_21 | 21        | 40       | 44       | 12580       | 12651     | 72         |
| Int_22 | 22        | 44       | .        | 12652       | 12695     | 44         |

|        | range      |              |              |               |
|--------|------------|--------------|--------------|---------------|
|        | Percentage | Complete_obs | Complete_Pct | Sum_Durations |
| Int_12 | 3.3635289  | 380          | 88.992974    | 5742          |
| Int_13 | 2.0638046  | 224          | 85.496183    | 4051          |
| Int_14 | 1.6541946  | 170          | 80.952381    | 3676          |
| Int_15 | 1.2997243  | 126          | 76.363636    | 3199          |
| Int_16 | 1.953525   | 179          | 72.177419    | 5552          |

|        |           |     |           |      |
|--------|-----------|-----|-----------|------|
| Int_17 | 1.4100039 | 114 | 63.687151 | 4724 |
| Int_18 | 1.1736904 | 64  | 42.95302  | 4516 |
| Int_19 | 1.3076014 | 95  | 57.228916 | 5738 |
| Int_20 | 0.9373769 | 75  | 63.02521  | 4551 |
| Int_21 | 0.5671524 | 57  | 79.166667 | 3043 |
| Int_22 | 0.3465931 | 37  | 84.090909 | 2063 |

La seconde partie de la sortie montre l'estimation proprement dite. Elle permet de voir de manière détaillée comment s'est effectuée l'optimisation.

### Sortie SAS 5.2.

```

Optimization Start
Parameter Estimates

 N Parameter Estimate Gradient
 Objective
 Function
1 X1 -2.200461 -1129.997906
2 X2 -2.200461 -30.899175
3 X3 -2.200461 411.241880
4 X4 -2.200461 455.175218
5 X5 -2.200461 461.378549
6 X6 -2.200461 330.197188
7 X7 -2.200461 229.695460
8 X8 -2.200461 145.160518
9 X9 -2.200461 43.817083
10 X10 -2.200461 29.649050
11 X11 -2.200461 10.330848
12 X12 -2.200461 -45.952548
13 X13 -2.200461 -117.891801
14 X14 -2.200461 -117.955421
15 X15 -2.200461 -115.771898
16 X16 -2.200461 -209.518465
17 X17 -2.200461 -177.056558
18 X18 -2.200461 -151.745110
19 X19 -2.200461 -56.287363
20 X20 -2.200461 -5.959835
21 X21 -2.200461 19.455053
22 X22 -2.200461 22.934471

Value of Objective Function = -36091.59799

Newton-Raphson Optimization with Line Search

Minimum Iterations 0
Maximum Iterations 200
Maximum Function Calls 500
ABSGCONV Gradient Criterion 0.00001
GCONV Gradient Criterion 1E-8
GCONV2 Gradient Criterion 0
ABSFCONV Function Criterion 0
FCONV Function Criterion 2.220446E-16
FCONV2 Function Criterion 0
FSIZE Parameter 0
ABSXCONV Parameter Change Criterion 0
KCONV Parameter Change Criterion 0
XSIZE Parameter 0
ABSCONV Function Criterion 1.340781E154
Line Search Method 2
Starting Alpha for Line Search 1
Line Search Precision LSPRECISION 0.9
DAMPSTEP Parameter for Line Search .
MAXSTEP Parameter for Line Search 0
FD Derivatives: Accurate Digits in Obj.F 15.653559775
Singularity Tolerance (SINGULAR) 1E-8

Newton-Raphson Optimization with Line Search

Without Parameter Scaling
Gradient Computed by Finite Differences
CRP Jacobian Computed by Finite Differences

```

| Parameter Estimates      |              |            |              |                    |                |                          |              |                    |              |
|--------------------------|--------------|------------|--------------|--------------------|----------------|--------------------------|--------------|--------------------|--------------|
| Optimization Start       |              |            |              |                    |                |                          |              |                    |              |
| Active Constraints       |              |            |              | 0                  |                | Objective Function       |              | -36091.59799       |              |
| Max Abs Gradient Element |              |            |              | 1129.9979057       |                |                          |              |                    |              |
| Iter                     | Rest arts    | Func Calls | Act Con      | Objective Function | Obj Fun Change | Max Abs Gradient Element | Step Size    | Slope Search Direc |              |
| 1                        | 0            | 3          | 0            | -34865             | 1226.6         | 353.4                    | 1.000        | -2362.3            |              |
|                          |              | X1         |              | X2                 |                | X3                       |              | X4                 | X5           |
| Parms                    | -3.004132498 |            | -2.223050359 |                    | -1.859410684   |                          | -1.751497189 |                    | -1.648964209 |
| Grad                     | 353.44027113 |            | 0.3455638008 |                    | 78.807076507   |                          | 119.34695525 |                    | 154.24826911 |
|                          |              | X6         |              | X7                 |                | X8                       |              | X9                 | X10          |
| Parms                    | -1.713302398 |            | -1.788342802 |                    | -1.88549928    |                          | -2.087947512 |                    | -2.112576833 |
| Grad                     | 95.249221412 |            | 54.537007032 |                    | 25.445289034   |                          | 2.5295232874 |                    | 1.340485229  |
|                          |              | X11        |              | X12                |                | X13                      |              | X14                | X15          |
| Parms                    | -2.181602479 |            | -2.30834358  |                    | -2.54528028    |                          | -2.610109976 |                    | -2.679103295 |
| Grad                     | 0.0953050101 |            | 2.3915621704 |                    | 18.17846513    |                          | 21.16871426  |                    | 23.8075219   |
|                          |              | X16        |              | X17                |                | X18                      |              | X19                | X20          |
| Parms                    | -2.739659035 |            | -2.808714495 |                    | -2.903720366   |                          | -2.572357559 |                    | -2.273814246 |
| Grad                     | 47.589957503 |            | 44.42218125  |                    | 42.787083188   |                          | 9.3012915019 |                    | 0.2337141526 |
|                          |              |            | X21          |                    | X22            |                          |              |                    |              |
| Parms                    |              |            | -1.682249461 |                    | -0.574525625   |                          |              |                    |              |
| Grad                     |              |            | 6.0386872611 |                    | 34.497307434   |                          |              |                    |              |
| 2                        | 0            | 4          | 0            | -34692             | 172.8          | 83.0177                  | 1.000        | -305.3             |              |
|                          |              | X1         |              | X2                 |                | X3                       |              | X4                 | X5           |
| Parms                    | -3.565591902 |            | -2.22330104  |                    | -1.90587101    |                          | -1.826598055 |                    | -1.755162982 |
| Grad                     | 83.017692014 |            | 0.0103009692 |                    | 1.8214740926   |                          | 4.4297052075 |                    | 7.9284740711 |
|                          |              | X6         |              | X7                 |                | X8                       |              | X9                 | X10          |
| Parms                    | -1.799596717 |            | -1.853128296 |                    | -1.925774355   |                          | -2.09372094  |                    | -2.116065883 |
| Grad                     | 4.037097931  |            | 1.7461302372 |                    | 0.5191934806   |                          | 0.0222539969 |                    | 0.057508161  |
|                          |              | X11        |              | X12                |                | X13                      |              | X14                | X15          |
| Parms                    | -2.181675831 |            | -2.314491453 |                    | -2.620263413   |                          | -2.720637603 |                    | -2.837809503 |
| Grad                     | 0.0543272075 |            | 0.0478780557 |                    | 0.6832742651   |                          | 1.1651123854 |                    | 1.822939295  |
|                          |              | X16        |              | X17                |                | X18                      |              | X19                | X20          |
| Parms                    | -2.949485536 |            | -3.0888664   |                    | -3.304098559   |                          | -2.661116507 |                    | -2.276408626 |
| Grad                     | 4.7020809651 |            | 5.7147132874 |                    | 7.5544591666   |                          | 0.4426533447 |                    | 0.0387476472 |
|                          |              |            | X21          |                    | X22            |                          |              |                    |              |
| Parms                    |              |            | -1.77776493  |                    | -1.056778916   |                          |              |                    |              |
| Grad                     |              |            | 0.2961927761 |                    | 7.1417324836   |                          |              |                    |              |
| 3                        | 0            | 5          | 0            | -34681             | 11.5786        | 8.8993                   | 1.000        | -21.667            |              |
|                          |              | X1         |              | X2                 |                | X3                       |              | X4                 | X5           |
| Parms                    | -3.796827842 |            | -2.223304975 |                    | -1.906995007   |                          | -1.829602431 |                    | -1.76123326  |
| Grad                     | 8.8993138535 |            | 0.0051504783 |                    | 0.0031913862   |                          | 0.0098360218 |                    | 0.0252873308 |
|                          |              | X6         |              | X7                 |                | X8                       |              | X9                 | X10          |
| Parms                    | -1.8035845   |            | -1.855338186 |                    | -1.92663118    |                          | -2.093770002 |                    | -2.116223902 |
| Grad                     | 0.009578976  |            | 0.0051301935 |                    | -0.000333681   |                          | 0.0012626181 |                    | -0.00031338  |
|                          |              | X11        |              | X12                |                | X13                      |              | X14                | X15          |
| Parms                    | -2.181774777 |            | -2.314604817 |                    | -2.623309602   |                          | -2.727438798 |                    | -2.852080384 |
| Grad                     | -0.00076731  |            | 0.0050086099 |                    | 0              |                          | 0.0051088615 |                    | 0.0117884758 |
|                          |              | X16        |              | X17                |                | X18                      |              | X19                | X20          |
| Parms                    | -2.975100522 |            | -3.136610692 |                    | -3.409676044   |                          | -2.66575825  |                    | -2.276925073 |
| Grad                     | 0.0563812394 |            | 0.133502071  |                    | 0.3851172218   |                          | 0.0007992037 |                    | 0.0001490059 |

|       |              | X21          |              | X22          |              |          |       |         |     |
|-------|--------------|--------------|--------------|--------------|--------------|----------|-------|---------|-----|
|       |              | Parms        | -1.782904894 | Parms        | -1.218584126 |          |       |         |     |
|       |              | Grad         | 0.002631861  | Grad         | 0.5473560627 |          |       |         |     |
| Iter  | Rest         | Func         | Act          | Objective    | Obj Fun      | Max Abs  | Step  | Slope   |     |
|       | arts         | Calls        | Con          | Function     | Change       | Gradient | Size  | Search  |     |
|       |              |              |              |              |              | Element  |       | Direc   |     |
| 4     | 0            | 6            | 0            | -34680       | 0.1457       | 0.1383   | 1.000 | -0.288  |     |
|       |              | X1           |              | X2           |              | X3       |       | X4      | X5  |
| Parms | -3.828061942 | -2.223309135 | -1.906997257 | -1.829609126 | -1.761253472 |          |       |         |     |
| Grad  | 0.1382501876 | -0.000454453 | -0.00067187  | -0.000172561 | -0.001237832 |          |       |         |     |
|       |              | X6           |              | X7           |              | X8       |       | X9      | X10 |
| Parms | -1.803594952 | -1.855345284 | -1.926631365 | -2.093771951 | -2.1162242   |          |       |         |     |
| Grad  | -0.001044975 | -0.000513018 | -0.000500522 | 0.0001578272 | -0.00062676  |          |       |         |     |
|       |              | X11          |              | X12          |              | X13      |       | X14     | X15 |
| Parms | -2.181774331 | -2.314621372 | -2.623308008 | -2.727470911 | -2.852177314 |          |       |         |     |
| Grad  | -0.000613848 | -0.001473113 | 0.0001347612 | -0.000523981 | -0.000507018 |          |       |         |     |
|       |              | X16          |              | X17          |              | X18      |       | X19     | X20 |
| Parms | -2.975417207 | -3.137782976 | -3.415652562 | -2.665770409 | -2.276932661 |          |       |         |     |
| Grad  | -0.00024565  | -0.000354017 | 0.0014375353 | -0.000532801 | -0.000447017 |          |       |         |     |
|       |              | X21          |              | X22          |              |          |       |         |     |
|       |              | Parms        | -1.782959574 | Parms        | -1.233164868 |          |       |         |     |
|       |              | Grad         | -0.000701816 | Grad         | 0.0037170481 |          |       |         |     |
| Iter  | Rest         | Func         | Act          | Objective    | Obj Fun      | Max Abs  | Step  | Slope   |     |
|       | arts         | Calls        | Con          | Function     | Change       | Gradient | Size  | Search  |     |
|       |              |              |              |              |              | Element  |       | Direc   |     |
| 5     | 0            | 7            | 0            | -34680       | 0.000035     | 0.000298 | 1.000 | -0.0001 |     |
|       |              | X1           |              | X2           |              | X3       |       | X4      | X5  |
| Parms | -3.828562607 | -2.22330879  | -1.906996837 | -1.829609008 | -1.761252513 |          |       |         |     |
| Grad  | 0            | -0.000151484 | 0            | 0            | 0            |          |       |         |     |
|       |              | X6           |              | X7           |              | X8       |       | X9      | X10 |
| Parms | -1.803593915 | -1.855344632 | -1.926630528 | -2.0937723   | -2.116222492 |          |       |         |     |
| Grad  | 0            | -0.000171006 | 0            | 0            | 0            |          |       |         |     |
|       |              | X11          |              | X12          |              | X13      |       | X14     | X15 |
| Parms | -2.181773231 | -2.314617495 | -2.623308609 | -2.727467829 | -2.852173291 |          |       |         |     |
| Grad  | -0.000153462 | 0            | 0            | 0            | 0            |          |       |         |     |
|       |              | X16          |              | X17          |              | X18      |       | X19     | X20 |
| Parms | -2.975415836 | -3.13777987  | -3.415674955 | -2.665764798 | -2.276926621 |          |       |         |     |
| Grad  | -0.000122825 | -0.000118006 | 0            | 0            | 0            |          |       |         |     |
|       |              | X21          |              | X22          |              |          |       |         |     |
|       |              | Parms        | -1.78294713  | Parms        | -1.23326532  |          |       |         |     |
|       |              | Grad         | 0            | Grad         | 0            |          |       |         |     |

Optimization Results

|                           |              |                          |              |
|---------------------------|--------------|--------------------------|--------------|
| Iterations                | 5            | Function Calls           | 8            |
| Hessian Calls             | 6            | Active Constraints       | 0            |
| Objective Function        | -34680.47134 | Max Abs Gradient Element | 0.0002980117 |
| Slope of Search Direction | -0.000069653 | Ridge                    | 0            |

GCONV convergence criterion satisfied.

Optimization Results  
Parameter Estimates

| N | Parameter | Estimate  | Gradient Objective Function |
|---|-----------|-----------|-----------------------------|
| 1 | X1        | -3.828563 | 0                           |
| 2 | X2        | -2.223309 | 0.000151                    |
| 3 | X3        | -1.906997 | 0                           |
| 4 | X4        | -1.829609 | 0                           |
| 5 | X5        | -1.761253 | 0                           |
| 6 | X6        | -1.803594 | 0                           |
| 7 | X7        | -1.855345 | 0.000171                    |
| 8 | X8        | -1.926631 | 0                           |
| 9 | X9        | -2.093772 | 0                           |

|    |     |           |          |
|----|-----|-----------|----------|
| 10 | X10 | -2.116222 | 0        |
| 11 | X11 | -2.181773 | 0.000153 |
| 12 | X12 | -2.314617 | 0        |
| 13 | X13 | -2.623309 | 0        |
| 14 | X14 | -2.727468 | 0        |
| 15 | X15 | -2.852173 | 0        |
| 16 | X16 | -2.975416 | 0.000123 |
| 17 | X17 | -3.137780 | 0.000118 |
| 18 | X18 | -3.415675 | 0        |
| 19 | X19 | -2.665765 | 0        |
| 20 | X20 | -2.276927 | 0.000298 |
| 21 | X21 | -1.782947 | 0        |
| 22 | X22 | -1.233265 | 0        |

Value of Objective Function = -34680.47134

lv1  
Maximum log-likelihood = -34680.47

| Gradient = | g         | Elasticities = | elas      |
|------------|-----------|----------------|-----------|
|            | 0.0001515 |                | 9.7114E-9 |
|            | 0         |                | 0         |
|            | 0         |                | 0         |
|            | 0         |                | 0         |
|            | 0         |                | 0         |
|            | 0.000171  |                | 9.1485E-9 |
|            | 0         |                | 0         |
|            | 0         |                | 0         |
|            | 0         |                | 0         |
|            | 0.0001535 |                | 9.6544E-9 |
|            | 0         |                | 0         |
|            | 0         |                | 0         |
|            | 0         |                | 0         |
|            | 0         |                | 0         |
|            | 0.0001228 |                | 1.0538E-8 |
|            | 0.000118  |                | 1.0677E-8 |
|            | 0         |                | 0         |
|            | 0         |                | 0         |
|            | 0.000298  |                | 1.9566E-8 |
|            | 0         |                | 0         |
|            | 0         |                | 0         |

Hessian eigenvalues =

| vp        |  |
|-----------|--|
| -36.99977 |  |
| -56.95417 |  |
| -63.99046 |  |
| -74.98517 |  |
| -94.99069 |  |
| -113.9936 |  |
| -125.9877 |  |
| -169.9765 |  |
| -178.9902 |  |
| -223.9778 |  |
| -275.9822 |  |
| -366.9627 |  |
| -379.9749 |  |
| -432.946  |  |
| -557.9756 |  |
| -605.9668 |  |
| -786.9353 |  |
| -1007.962 |  |
| -1297.926 |  |
| -1336.971 |  |
| -1468.978 |  |
| -1616.939 |  |

res\_h

|    | Hasard    | Std_Dev   | Student   |
|----|-----------|-----------|-----------|
| h1 | 0.0217408 | 0.0013087 | 16.612712 |
| h2 | 0.1082503 | 0.0029605 | 36.564611 |
| h3 | 0.1485258 | 0.0036936 | 40.211173 |
| h4 | 0.1604763 | 0.004187  | 38.327252 |
| h5 | 0.1718295 | 0.0047695 | 36.026739 |
| h6 | 0.1647059 | 0.0051878 | 31.748411 |
| h7 | 0.156399  | 0.0055753 | 28.052367 |
| h8 | 0.1456381 | 0.0059163 | 24.616392 |
| h9 | 0.1232214 | 0.005922  | 20.807351 |

```

h10 0.1204859 0.0062896 19.15627
h11 0.1128413 0.0047771 23.621506
h12 0.098804 0.0050687 19.492945
h13 0.0725624 0.0048485 14.965886
h14 0.0653846 0.0050151 13.037502
h15 0.0577187 0.0051422 11.224423
h16 0.0510262 0.003814 13.378722
h17 0.043379 0.0040629 10.676778
h18 0.0328542 0.0041071 7.9994012
h19 0.0695461 0.0071356 9.7463157
h20 0.102599 0.0118483 8.6593959
h21 0.1681419 0.0222799 7.546803
h22 0.2913397 0.0478961 6.0827433

```

```

 md
Median duration = 5.4998269
 mt
Mean duration = 13.026894

```

Etant donné le grand nombre d'intervalles, le plus simple est d'effectuer une représentation graphique. Le graphique 5.1 montre la fonction de hasard en escaliers ainsi que son intervalle de confiance asymptotique de niveau 95%. La première partie de la fonction montre un profil similaire à ceux obtenus par les modèles paramétriques du chapitre précédent. Le hasard est d'abord croissant puis décroissant. Mais ce graphique montre également que le hasard se remet à augmenter après une durée d'emploi de l'ordre de 30 ans. En effet, si une personne reste aussi longtemps dans la même entreprise, elle finira par la quitter soit pour raisons de santé, soit une fois atteint l'âge de la retraite ou de la pré-retraite. C'est la raison pour laquelle le hasard doit augmenter après une longue durée passée dans le premier emploi. Le deuxième intervalle de hausse correspond aux départs à la retraite des personnes qui ont toujours travaillé dans la même entreprise. Ici, on remarque qu'il faut représenter la fonction en escaliers, ce que l'on peut faire par les instructions :

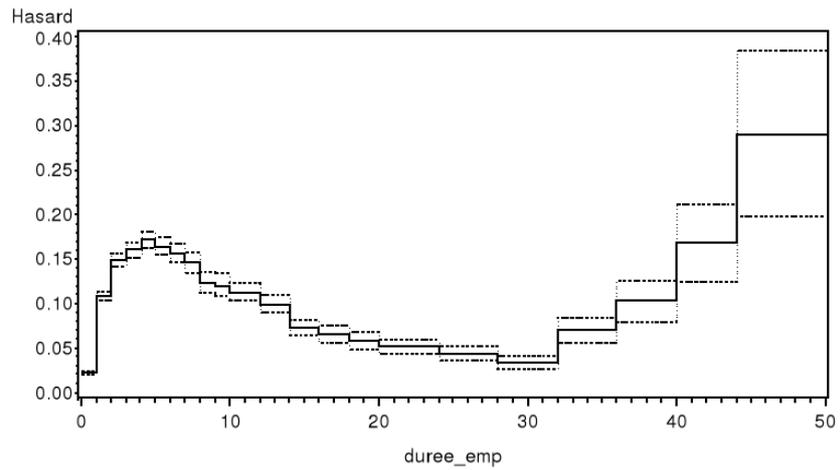
```

symbol1 c=black i=stepjL line=1;
symbol2 c=black i=stepjL line=33;
symbol3 c=black i=stepjL line=33;
proc gplot data=_outi;
plot (hasard l_ht u_ht)*&y/overlay;
run;

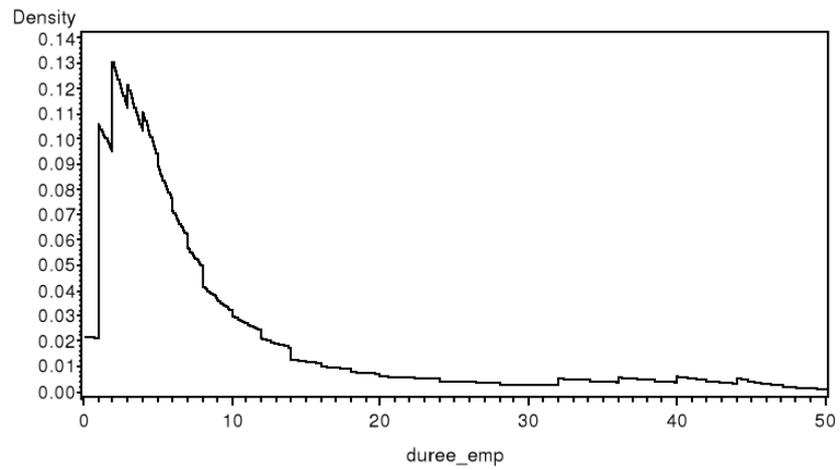
```

La représentation en escalier est obtenue par l'option `i=stepjL` qui trace un trait vertical à gauche de la marche. On pourrait tracer un trait à droite grâce à l'option `i=stepjR`. La densité est représentée sur le graphique 5.2. Elle possède un profil particulier en dents de scie. Ce point est attendu par elle est obtenue par la formule  $\hat{f}_T(t) = \hat{h}_T(t) \hat{S}_T(t)$ . Or, sur un intervalle donné, le hasard  $\hat{h}_T(t)$  est constant et la survie  $\hat{S}_T(t)$  est décroissante. On obtient donc un profil avec tout d'abord un saut vers le bas ou vers le haut de la fonction de

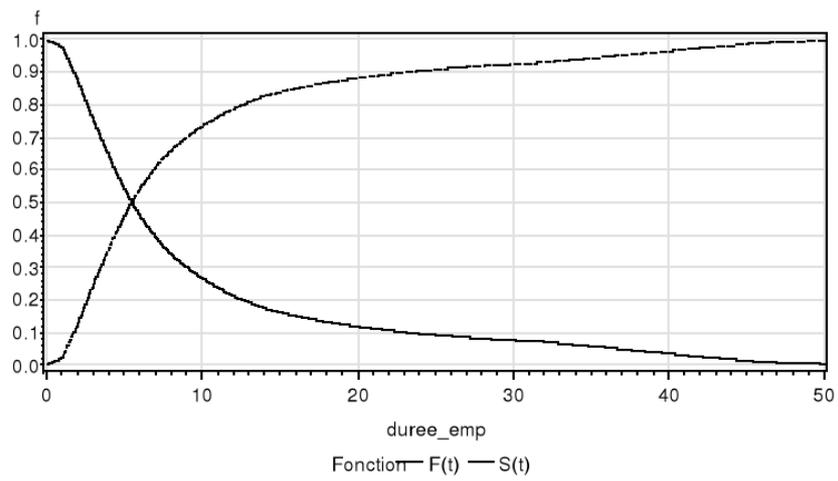
hasard, puis une fonction décroissante jusqu'au début de l'intervalle suivant. Ceci donne le profil en dents de scie que l'on observe. Les fonctions de survie et de répartition sont données par le graphique 5.3.



GRAPHIQUE 5.1 – DISTRIBUTION À HASARD CONSTANT PAR MORCEAUX,  $\hat{h}_T(t)$



GRAPHIQUE 5.2 – DISTRIBUTION À HASARD CONSTANT PAR MORCEAU,  $\hat{f}_T(t)$



GRAPHIQUE 5.3 – DISTRIBUTION À HASARD CONSTANT PAR MORCEAUX,  $\hat{S}_T(t)$  ET  $\hat{F}_T(t)$



## CHAPITRE 6

# Estimation non paramétrique

L'ESTIMATION PARAMÉTRIQUE oblige à faire des hypothèses fortes sur la distribution de la variable de durée. On montre que pour estimer la fonction de survie ces hypothèses ne sont pas nécessaires. Dans cette section, nous ne considérerons que le cas des données censurées, le cas des données complètes s'en déduisant immédiatement.

### 6.1 Estimateur de Kaplan-Meier

On appelle également cet estimateur "product-limit". Le cas non censuré donne un estimateur très simple de la fonction de survie. Pour un échantillon  $(t_1, \dots, t_N)$  de durées complètes, on peut prendre :

$$\hat{S}(t) = \frac{1}{N} \sum_{i=1}^N \delta(t_i > t),$$

avec

$$\delta(t_i > t) = \begin{cases} 1 & \text{si } t_i > t \\ 0 & \text{sinon} \end{cases}$$

l'estimateur de la fonction de survie est le complément à 1 de la fonction de répartition empirique. Cet estimateur est toutefois très peu utilisé parce que la quasi totalité des bases de données disponibles contiennent des données censurées.

Kaplan et Meier ont étendu cet estimateur au cas des données censurées. Pour simplifier la présentation, on considère une distribution discrète ou dont les durées ont été regroupées par intervalles. Les durées observables sont notées  $t_{(1)} < t_{(2)} < \dots < t_{(m)}$  où  $m \leq N$  est le nombre de valeurs que prennent les durées, inférieur ou égal au nombre d'individus. Les probabilités associées à

chacun de ces points sont notées,  $f_1, f_2, \dots, f_m$ . Ce type de représentation sera correct si l'on dispose de données avec des durées en mois, par exemple. On peut alors décrire la distribution de la variable de durée de la manière suivante :

$$f(t) = f_j \delta(t - t_{(j)}), \quad j = 1, \dots, m$$

avec

$$\delta(t - t_{(j)}) = \begin{cases} 1 & \text{si } t = t_{(j)} \\ 0 & \text{sinon} \end{cases}$$

ce qui permet de définir le hasard comme :

$$h(t) = h_j \delta(t - t_{(j)}),$$

et la survie comme :

$$S(t_{(j)}) = \Pr(T > t_{(j)}) = f_{j+1} + \dots + f_m$$

On en déduit que :

$$\begin{aligned} S(t_{(j-1)}) - S(t_{(j)}) &= f_j + f_{j+1} + \dots + f_m - (f_{j+1} + \dots + f_m) \\ &= f_j \end{aligned}$$

Soit  $d_j$  le nombre de sorties (i.e. décès) après une durée de vie  $t_{(j)}$  et  $n_j$  le nombre d'individus encore en vie juste avant la durée  $t_{(j)}$ . Le nombre  $n_i$  est appelé population à risque pour la durée  $t_{(i)}$ . L'estimateur de Kaplan-Meier est défini par :

$$\hat{S}(t) = \prod_{i: t_{(i)} < t} \left(1 - \frac{d_i}{n_i}\right)$$

On peut obtenir cet estimateur par le maximum de vraisemblance non paramétrique. Soit  $c_j$  le nombre d'individus censurés entre  $t_{(j)}$  et  $t_{(j+1)}$ , et  $d_j$  le nombre d'individus qui sortent après une durée  $t_{(j)}$ . La vraisemblance prend la forme suivante :

$$L = \prod_{j=1}^m \underbrace{\left(S(t_{(j-1)}) - S(t_{(j)})\right)}_{f_j}^{d_j} S(t_{(j)})^{c_j}$$

les durées complètes correspondent à  $d_i$  individus et leur contribution à la vraisemblance n'est autre que leur densité. Les durées incomplètes contribuent à

la vraisemblance à hauteur de  $S(t_{(j)}) = \Pr(T > t_{(j)})$ . Pour réécrire cette vraisemblance, on pose une première définition, la population à risque juste avant la durée  $t_{(j)}$  :

$$n_j = \sum_{k \geq j} (d_k + c_k),$$

elle inclut les individus qui sortiront à partir de cette durée et les individus censurés à partir de cette durée. On pose également une deuxième définition :

$$\pi_j = \frac{S(t_{(j)})}{S(t_{(j-1)})}, S(t_{(0)}) = 1, t_{(0)} = 0$$

la probabilité conditionnelle de survivre pendant une durée  $t_{(j)}$  sachant que l'on a déjà survécu pendant une durée  $t_{(j-1)}$ . On obtient alors :

$$S(t_{(j)}) = \frac{S(t_{(1)})}{S(t_{(0)})} \frac{S(t_{(2)})}{S(t_{(1)})} \dots \frac{S(t_{(j)})}{S(t_{(j-1)})} = \pi_1 \pi_2 \dots \pi_j$$

donc

$$S(t_{(j-1)}) - S(t_{(j)}) = \pi_1 \pi_2 \dots \pi_{j-1} (1 - \pi_j),$$

ce qui permet d'obtenir le premier terme de la vraisemblance :

$$S(t_{(0)}) - S(t_{(1)}) = 1 - \pi_1$$

$$S(t_{(1)}) - S(t_{(2)}) = \pi_1 (1 - \pi_2)$$

⋮

$$S(t_{(m-1)}) - S(t_{(m)}) = \pi_1 \dots \pi_{m-1} (1 - \pi_m)$$

tandis que les second termes sont égaux à :

$$S(t_{(1)}) = \pi_1$$

$$S(t_{(2)}) = \pi_1 \pi_2$$

⋮

$$S(t_{(m)}) = \pi_1 \pi_2 \dots \pi_m$$

on doit donc effectuer le produit :

$$(1 - \pi_1)^{d_1} \pi_1^{c_1}$$

$$(\pi_1 (1 - \pi_2))^{d_2} (\pi_1 \pi_2)^{c_2}$$

⋮

$$(\pi_1 \dots \pi_{m-1} (1 - \pi_m))^{d_m} (\pi_1 \pi_2 \dots \pi_m)^{c_m}$$

soit :

$$\begin{aligned}
 & (1 - \pi_1)^{d_1} (1 - \pi_2)^{d_2} \dots (1 - \pi_m)^{d_m} \\
 & \pi_1^{c_1 + c_2 + d_2 + \dots + c_m + d_m} \\
 & \pi_2^{c_2 + \dots + c_m + d_m} \\
 & \vdots \\
 & \pi_{m-1}^{c_{m-1} + c_m + d_m} \\
 & \pi_m^{c_m}
 \end{aligned}$$

on remarque alors que :

$$\begin{aligned}
 c_1 + c_2 + d_2 + \dots + c_m + d_m &= c_1 + d_1 + c_2 + d_2 + \dots + c_m + d_m - d_1 \\
 &= n_1 - d_1 \\
 c_2 + c_3 + d_3 + \dots + c_m + d_m &= c_2 + d_2 + c_3 + d_3 + \dots + c_m + d_m - d_2 \\
 &= n_2 - d_2 \\
 &\vdots \\
 c_{m-1} + c_m + d_m &= c_{m-1} + d_{m-1} + c_m + d_m - d_{m-1} \\
 &= n_{m-1} - d_{m-1} \\
 c_m &= c_m + d_m - d_m \\
 &= n_m - d_m
 \end{aligned}$$

ce qui permet de réécrire le produit :

$$\begin{aligned}
 & (1 - \pi_1)^{d_1} (1 - \pi_2)^{d_2} \dots (1 - \pi_m)^{d_m} \\
 & \pi_1^{n_1 - d_1} \pi_2^{n_2 - d_2} \dots \pi_m^{n_m - d_m}
 \end{aligned}$$

d'où la vraisemblance :

$$L = \prod_{j=1}^m (1 - \pi_j)^{d_j} \pi_j^{n_j - d_j},$$

elle correspond à la vraisemblance d'une loi binomiale, ce qui permet d'interpréter l'estimateur de la manière suivante :  $d_j$  individus ont une probabilité  $1 - \pi_j$  de sortir à la date  $t_{(j)}$  et  $n_j - d_j$  individus ont une probabilité  $\pi_j$  de survivre après cette date. La probabilité de sortie  $1 - \pi_j$  est donc associée à la sous-population qui va effectivement sortir en  $t_{(j)}$ , et la probabilité de survie  $\pi_j$  est associée à la partie de la population à risque qui ne décèdera pas à la date  $t_{(j)}$ .

Il est important de remarquer que l'on n'enlève pas toutes les censures mais seulement celles qui portent sur la durée dont on veut estimer la survie. Si la censure à lieu à une date ultérieure, on garde la donnée censurée car elle fournit des informations sur la probabilité de survivre à la durée  $t_{(j)}$ .

On peut approfondir la forme binomiale de la manière suivante. Considérons une seule durée  $t_{(j)}$ , on part de la population à risque  $n_j$ , car seule cette population est susceptible de sortir à la date  $t_{(j)}$ . Sur cette population, on considère une indicatrice de sortie à la date  $t_{(j)}$  que l'on note  $d_{sj}$ ,  $s = 1, \dots, n_j$ . Donc  $d_{sj} = 1$  si l'individu  $s$  sort à la date  $t_{(j)}$ ,  $d_{sj} = 0$  s'il sort à une date ultérieure. Il s'agit d'une variable de Bernoulli de paramètre  $1 - \pi_j$ . La vraisemblance est donnée par :

$$L_j = \prod_{s=1}^{n_j} (1 - \pi_j)^{d_{sj}} \pi_j^{1-d_{sj}} = (1 - \pi_j)^{\sum_{s=1}^{n_j} d_{sj}} \pi_j^{n_j - \sum_{s=1}^{n_j} d_{sj}},$$

il suffit alors de remarquer que le nombre de sorties à la date  $t_{(j)}$  est égal à :

$$\sum_{s=1}^{n_j} d_{sj} = d_j$$

on obtient donc, pour la durée  $t_{(j)}$  :

$$(1 - \pi_j)^{d_j} \pi_j^{n_j - d_j},$$

en utilisant le même argument pour toutes les durées, on obtient la vraisemblance qui mène à l'estimateur de Kaplan-Meier. De plus, la quantité  $d_{sj}$  est une variable aléatoire de Bernoulli d'espérance  $1 - \pi_j$ , en conséquence :

$$E(d_j) = E\left(\sum_{s=1}^{n_j} d_{sj}\right) = \sum_{s=1}^{n_j} E(d_{sj}) = n_j(1 - \pi_j)$$

Il nous reste à estimer les  $\pi_j$  pour obtenir un estimateur de la survie. La log-vraisemblance globale est donnée par :

$$\ell = \sum_{j=1}^m d_j \ln(1 - \pi_j) + (n_j - d_j) \ln \pi_j,$$

la condition du premier ordre est donnée par :

$$\frac{\partial \ell}{\partial \pi_j}(\hat{\pi}_j) = -\frac{d_j}{1 - \hat{\pi}_j} + \frac{n_j - d_j}{\hat{\pi}_j} = 0,$$

ce qui donne :

$$\hat{\pi}_j = 1 - \frac{d_j}{n_j},$$

d'où l'estimateur de Kaplan-Meier :

$$\hat{S}(t_{(j)}) = \hat{\pi}_1 \hat{\pi}_2 \dots \hat{\pi}_j = \prod_{k=1}^j \left(1 - \frac{d_k}{n_k}\right),$$

que l'on peut réécrire sous la forme d'une fonction en escalier :

$$\hat{S}(t) = \prod_{j: t_{(j)} < t} \left(1 - \frac{d_j}{n_j}\right)$$

Le maximum de vraisemblance permet également de déterminer la variance asymptotique des estimateurs  $\hat{\pi}_j$  qui est utile pour la suite. La dérivée seconde est nulle :

$$\frac{\partial^2 \ell}{\partial \pi_j \partial \pi_k} = 0,$$

donc la covariance asymptotique est nulle entre les estimateurs des probabilités. On a également :

$$\frac{\partial^2 \ell}{\partial \pi_j \partial \pi_j} = \frac{d_j}{(1 - \pi_j)^2} - \frac{(n_j - d_j)}{\pi_j^2},$$

d'où

$$\text{Vas}(\hat{\pi}_j) = E \left[ -\frac{\partial^2 \ell}{\partial \pi_j \partial \pi_j} \right]^{-1} = \frac{\pi_j (1 - \pi_j)}{n_j}$$

et

$$\text{Covas}(\hat{\pi}_j, \hat{\pi}_\ell) = 0, \forall j \neq \ell$$

### 6.1.1 Formule de Greenwood

La formule de Greenwood permet de calculer la variance asymptotique de l'estimateur de Kaplan-Meier. On applique deux fois la méthode de Slutsky. Rappelons ici que si un estimateur  $\hat{\theta}$  est convergent et asymptotiquement normal,  $g$  une fonction de classe  $C^1$ ,  $g(\hat{\theta})$  est asymptotiquement normal de variance :

$$\text{Vas}(g(\hat{\theta})) = g'(\theta)^2 \text{Vas}(\hat{\theta})$$

On a :

$$\hat{S}(t_{(j)}) = \prod_{k=1}^j \hat{\pi}_k$$

ce qui implique :

$$\ln \widehat{S}(t_{(j)}) = \sum_{k=1}^j \ln \widehat{\pi}_k,$$

comme les  $\widehat{\pi}_j$  ne sont pas corrélés, on a :

$$\text{Vas}(\ln \widehat{S}(t_{(j)})) = \sum_{k=1}^j \text{Vas}(\ln \widehat{\pi}_k)$$

et d'après le théorème de Slutsky :

$$\text{Vas}(\ln \widehat{\pi}_k) = \frac{1}{\widehat{\pi}_k^2} \text{Vas}(\widehat{\pi}_k) = \frac{1 - \pi_k}{n_k \pi_k}$$

ce qui implique :

$$\text{Vas}(\ln \widehat{S}(t_{(j)})) = \sum_{k=1}^j \frac{1 - \pi_k}{n_k \pi_k},$$

que l'on estime en remplaçant  $\pi_k$  par  $\widehat{\pi}_k = d_k / n_k$  :

$$\widehat{\text{Vas}}(\ln \widehat{S}(t_{(j)})) = \sum_{k=1}^j \frac{d_k}{n_k (n_k - d_k)},$$

pour obtenir la variance de  $\widehat{S}(t_{(j)})$  on applique le théorème de Slutsky en prenant la variance de l'exponentielle de  $\ln \widehat{S}(t_{(j)})$  ce qui donne la formule de Greenwood :

$$\widehat{\text{Vas}}(\widehat{S}(t_{(j)})) = [\widehat{S}(t_{(j)})]^2 \sum_{k=1}^j \frac{d_k}{n_k (n_k - d_k)},$$

on démontre également que l'estimateur de Kaplan-Meier est convergent et asymptotiquement normal, et que sa variance asymptotique peut être estimée par la formule de Greenwood.

### 6.1.2 Estimateur de Breslow

L'estimateur de Breslow repose sur la relation suivante :

$$S(t) = \exp(-H(t))$$

où  $H(t)$  est le hasard cumulé

$$H(t) = \int_0^t h(x) dx.$$

Il consiste à choisir d'abord un estimateur du hasard cumulé  $\widehat{H}(t_{(j)})$  puis à appliquer la formule :

$$\widehat{S}_B(t_{(j)}) = \exp(-\widehat{H}(t_{(j)})).$$

Pour trouver un estimateur du hasard cumulé, on utilise sa définition en fonction des hasard à chaque période :

$$\widehat{H}(t_{(j)}) = \sum_{k=1}^j \widehat{h}_k.$$

Pour trouver ces estimateurs  $\widehat{h}_j$  on utilise :

$$h_j = \frac{f_j}{S(t_{(j-1)})} = 1 - \frac{S(t_{(j)})}{S(t_{(j-1)})}$$

dont on déduit que :

$$1 - h_j = \frac{S(t_{(j)})}{S(t_{(j-1)})},$$

ce qui implique par récurrence, avec la convention  $S(t_{(0)}) = S(0) = 1$ , que l'on a :

$$S(t_{(j)}) = \prod_{k=1}^j (1 - h_k)$$

En rapprochant cet estimateur de celui de Kaplan-Meier :

$$\widehat{S}(t_{(j)}) = \prod_{k=1}^j \left(1 - \frac{d_k}{n_k}\right),$$

on voit que l'on peut estimer le hasard par :

$$\widehat{h}_k = \frac{d_k}{n_k},$$

ce qui implique que l'on peut estimer le hasard cumulé par l'estimateur de Nelson-Aalen :

$$\widehat{H}(t_{(j)}) = \sum_{k=1}^j \frac{d_k}{n_k},$$

avec

$$\text{Vas}(\widehat{H}(t_{(j)})) = \text{Vas}(-\ln \widehat{S}(t_{(j)})) = \sum_{k=1}^j \frac{d_k}{n_k(n_k - d_k)}$$

d'où l'estimateur de Breslow de la fonction de survie :

$$\widehat{S}_B(t_{(j)}) = \exp(-\widehat{H}(t_{(j)})) = \exp\left(-\sum_{k=1}^j \frac{d_k}{n_k}\right)$$

et la variance asymptotique de cet estimateur est donnée par la formule de Greenwood :

$$\text{Var}\left(\widehat{S}_B(t_{(j)})\right) = \left[\widehat{S}_B(t_{(j)})\right]^2 \sum_{k=1}^j \frac{d_k}{n_k(n_k - d_k)}$$

### 6.1.3 Statistiques utiles

On peut également estimer l'espérance mathématique de la variable de durée par :

$$\widehat{E}(T) = \int_0^{+\infty} \widehat{S}(t) = \sum_{j=1}^M (t_{(j)} - t_{(j-1)}) \widehat{S}(t_{(j)}),$$

en utilisant soit l'estimateur de Kaplan-Meier soit celui de Breslow pour la fonction de survie. Ici, il faut toutefois faire attention au point suivant : la fonction de survie doit être suffisamment proche de 0 en  $t_{(M)}$  pour que l'espérance ne soit pas tronquée. Si la valeur de  $\widehat{S}(t_{(M)})$  est relativement élevée, il vaut mieux faire une hypothèse de hasard constant par morceaux et estimer le modèle du chapitre précédent.

Pour estimer la médiane, il faut distinguer le cas où la survie descend en dessous de 1/2 à la durée maximale du cas inverse. Dans le premier cas, on cherchera la valeur  $t_{(j)}$  telle que :

$$S(t_{(j)}) \leq \frac{1}{2} \leq S(t_{(j+1)}),$$

et on en déduit que :

$$\text{Me} \in [t_{(j)}, t_{(j+1)}],$$

comme la fonction de survie est en escaliers, et repose sur l'hypothèse que la durée est discrète, on peut juste donner un intervalle. Une convention simple consiste à donner le centre de cet intervalle. Dans le second cas, le plus simple est d'estimer le modèle à hasard constant par morceaux.

### 6.1.4 Programmation XIII : Estimateurs de Kaplan Meier et de Breslow

Cette section présente un résumé de l'estimation des fonctions et présente le calcul des intervalles de confiance asymptotique. On commence par estimer

$$\hat{\pi}_j = 1 - \frac{d_j}{n_j}, \quad j = 1, \dots, M$$

dont la variance asymptotique peut être estimée par :

$$\widehat{\text{Vas}}(\hat{\pi}_j) = \frac{\hat{\pi}_j(1 - \hat{\pi}_j)}{n_j}, \quad \widehat{\text{Covas}}(\hat{\pi}_j, \hat{\pi}_k) = 0.$$

On en déduit un estimateur du hasard :

$$\hat{h}(t_{(j)}) = 1 - \hat{\pi}_j = \frac{d_j}{n_j}$$

dont la variance est simplement égale à :

$$\widehat{\text{Vas}}(\hat{h}(t_{(j)})) = \widehat{\text{Vas}}(\hat{\pi}_j).$$

Pour la fonction de survie, on utilise :

$$\hat{S}(t_{(j)}) = \prod_{k=1}^j \hat{\pi}_k$$

et la formule de Greenwood :

$$\widehat{\text{Vas}}(\hat{S}(t_{(j)})) = [\hat{S}(t_{(j)})]^2 \sum_{k=1}^j \frac{1 - \hat{\pi}_k}{n_k \hat{\pi}_k}.$$

Pour la fonction de répartition, on peut donc prendre :

$$\hat{F}(t_{(j)}) = 1 - \hat{S}(t_{(j)}),$$

et la même variance asymptotique que pour la survie. Pour la densité, on utilisera :

$$\hat{f}(t_{(j)}) = \hat{S}(t_{(j-1)}) - \hat{S}(t_{(j)}),$$

et il nous faut calculer la quantité :

$$\text{Vas}(\hat{f}(t_{(j)})) = \text{Vas}(\hat{S}(t_{(j-1)})) + \text{Vas}(\hat{S}(t_{(j)})) - 2\text{Covas}(\hat{S}(t_{(j-1)}), \hat{S}(t_{(j)})),$$

pour cela, nous allons utiliser le théorème de Slutsky. On pose

$$s_j = \ln S(t_{(j)}) \triangleq \ln S_j,$$

et

$$\theta = \begin{pmatrix} s_{j-1} \\ s_j \end{pmatrix}, \quad g(\theta) = \begin{pmatrix} \exp(s_{j-1}) \\ \exp(s_j) \end{pmatrix},$$

avec :

$$\frac{\partial g}{\partial \theta'} = \begin{pmatrix} \exp(s_{j-1}) & 0 \\ 0 & \exp(s_j) \end{pmatrix} = \begin{pmatrix} S_{j-1} & 0 \\ 0 & S_j \end{pmatrix}$$

et :

$$\text{Vas}(\hat{\theta}) = \begin{pmatrix} \text{Vas}(\hat{s}_{j-1}) & \text{Covas}(\hat{s}_{j-1}, \hat{s}_j) \\ \text{Covas}(\hat{s}_{j-1}, \hat{s}_j) & \text{Vas}(\hat{s}_j) \end{pmatrix}$$

on aura donc :

$$\text{Vas}(g(\hat{\theta})) = \frac{\partial g}{\partial \theta'}(\theta) \text{Vas}(\hat{\theta}) \frac{\partial g}{\partial \theta'}(\theta)'$$

ce qui est équivalent à :

$$\text{Vas} \begin{pmatrix} \hat{S}(t_{(j-1)}) \\ \hat{S}(t_{(j)}) \end{pmatrix} = \begin{pmatrix} S_{j-1}^2 \text{Vas}(\hat{s}_{j-1}) & S_{j-1} S_j \text{Covas}(\hat{s}_{j-1}, \hat{s}_j) \\ S_{j-1} S_j \text{Covas}(\hat{s}_{j-1}, \hat{s}_j) & S_j^2 \text{Vas}(\hat{s}_j) \end{pmatrix}$$

on en déduit que :

$$\text{Covas}(\hat{S}(t_{(j-1)}), \hat{S}(t_{(j)})) = S_{j-1} S_j \text{Covas}(\hat{s}_{j-1}, \hat{s}_j)$$

or nous avons :

$$\begin{aligned} \text{Covas}(\hat{s}_{j-1}, \hat{s}_j) &= \text{Covas} \left( \sum_{k=1}^{j-1} \ln \hat{\pi}_k, \sum_{k=1}^j \ln \hat{\pi}_k \right) \\ &= \text{Covas} \left( \sum_{k=1}^{j-1} \ln \hat{\pi}_k, \ln \hat{\pi}_j + \sum_{k=1}^{j-1} \ln \hat{\pi}_k \right) \\ &= \text{Vas} \left( \sum_{k=1}^{j-1} \ln \hat{\pi}_k \right) + \underbrace{\text{Covas} \left( \ln \hat{\pi}_j, \sum_{k=1}^{j-1} \ln \hat{\pi}_k \right)}_0 \\ &= \text{Vas}(\hat{s}_{j-1}), \end{aligned}$$

car les  $\hat{\pi}_j$  sont décorrélés et asymptotiquement normaux. On en déduit que :

$$\text{Covas}(\hat{S}(t_{(j-1)}), \hat{S}(t_{(j)})) = S_{j-1} S_j \text{Vas}(\hat{s}_{j-1}),$$

ce qui implique :

$$\widehat{\text{Vas}}\left(\widehat{f}\left(t_{(j)}\right)\right)=\widehat{\text{Vas}}\left(\widehat{S}\left(t_{(j-1)}\right)\right)+\widehat{\text{Vas}}\left(\widehat{S}\left(t_{(j)}\right)\right)-2\widehat{S}\left(t_{(j-1)}\right)\widehat{S}\left(t_{(j)}\right)\sum_{k=1}^{j-1}\frac{1-\widehat{\pi}_k}{n_k\widehat{\pi}_k}.$$

Pour calculer les intervalles de confiance, on utilise la normalité asymptotique. Pour un intervalle de confiance de niveau 95%, on ajoute et on enlève 1.96 fois l'écart-type à la valeur de la statistique désirée.

Le programme d'estimation est le suivant :

1. %let tab=duree; %let y=duree\_emp;  
%let d=presence;  
Paramètres habituels du programme : nom de la base de données (&tab), nom de la variable de durée (&y) et nom de l'indicatrice de durée complète (&d).
2. %let outkm=\_Kaplan\_Meier; %let outb=\_Breslow;  
Paramètres donnant les noms des tableaux de sortie (&outkm pour Kaplan-Meier et &outb pour Breslow). Les noms par défaut sont respectivement \_Kaplan\_Meier et \_Breslow. Ces fichiers seront créés automatiquement dans le répertoire de travail ("Work").
3. %let outs=\_stats;  
Paramètre définissant le nom du tableau contenant les statistiques par intervalle (voir plus loin).
4. proc freq noprint data=&tab;  
tables &y\*&d/out=\_total(keep=&y &d count);  
proc transpose data=\_total  
out=\_total(drop=\_name\_ \_label\_)  
prefix=n; by &y; id &d;  
Lancement d'une procédure *freq* calculant, pour chaque réalisation de la durée, le nombre d'observations complètes et censurées. La procédure transpose crée deux colonnes. La colonne n0 contient le nombre d'observations complètes et n1 le nombre d'observations censurées, pour chaque réalisation de la durée. Sur les procédures *freq* et *transpose*, voir Duguet (2004).
5. data \_total; set \_total;  
n0=max(n0,0); n1=max(n1,0); ntot=n0+n1;  
Quand une réalisation ne contient pas de données complètes, le nombre n0 est manquant, et quand elle ne contient pas de donnée censurée, le nombre n1 est manquant. La ligne précédente met ces nombres à 0 en utilisant le fait que SAS code les valeurs manquantes comme  $\{-\infty\}$ , ainsi

le maximum d'une valeur manquante et de 0 est toujours égal à 0. Ensuite, on calcule le nombre total d'observations pour une durée donnée, que l'on range dans la variable `ntot`.

6. `proc iml;`  
Début de la procédure IML.
7. `start lecture;`  
Début du module de lecture.
8. `nom_y={&y};`  
Stockage du nom de la variable de durée dans une variable de type caractère.
9. `use _total;`  
`read all var {n0} into ncens;`  
`read all var {n1} into ncomp;`  
`read all var {ntot} into ntot;`  
`read all var (nom_y) into dur;`  
Lecture des données. On range la valeur de la durée dans un vecteur colonne `dur`, les nombres d'observations censurées sont rangés dans le vecteur colonne `ncens`, le nombre d'observations complètes est rangé dans `ncom`, et le nombre total d'observations de chaque réalisation de la durée dans le vecteur `ntot`.
10. `tab=dur || ncens || ncomp || ntot;`  
Tableau de statistiques résumant la distribution de la variable de durée.
11. `labtab=nom_y || "CENSORED" || "FAILED" || "TOTAL";`  
Noms des colonnes pour l'affichage. L'expression "failed" désigne, en anglais, une observation dont la durée est complète.
12. `print "DATA SUMMARY", "*****", , ,`  
`tab[colname=labtab];`  
Affichage du tableau.
13. `finish lecture;`  
Fin du module de lecture.
14. `start estim;`  
Début du module d'estimation.
15. `ndur=nrow(tab);`  
Nombre de durées différentes. Il ne s'agit pas du nombre d'observations.
16. `c_ncens=j(ndur,1,0); c_ncomp=j(ndur,1,0);`  
Initialisation des vecteurs  $[\sum_{k \geq j} c_k]_{j=1, \dots, M}$  et  $[\sum_{k \geq j} d_k]_{j=1, \dots, M}$ .  
La commande `j(ndur,1,0)` crée un vecteur nul de `ndur` lignes et une

colonne. Plus généralement, la commande  $j(a, b, c)$  crée une matrice de  $a$  lignes et  $b$  colonnes, dont tous les éléments sont égaux à  $c$ .

17.  $c\_ncens[ndur]=ncens[ndur];$   
 $c\_ncomp[ndur]=ncomp[ndur];$   
 On commence par la dernière ligne du tableau car  $\sum_{k \geq M} c_k = c_M$  et  $\sum_{k \geq M} d_k = d_M$ .
18.  $do\ j=ndur-1\ to\ 1\ by\ -1;$   
 Boucle à incrément négatif. Le compteur prend successivement les valeurs  $M-1, M-2, \dots, 1$ .
19.  $c\_ncens[j]=c\_ncens[j+1]+ncens[j];$   
 On utilise :  $\sum_{k \geq j} c_k = \sum_{k \geq j-1} c_k + c_j$
20.  $c\_ncomp[j]=c\_ncomp[j+1]+ncomp[j];$   
 On utilise :  $\sum_{k \geq j} d_k = \sum_{k \geq j-1} d_k + d_j$
21.  $end;$   
 Fin de la boucle.
22.  $c\_AtRisk=c\_ncens+c\_ncomp;$   
 Vecteur des nombres d'individus à risque :

$$\sum_{k \geq j} (c_k + d_k) = \sum_{k \geq j} c_k + \sum_{k \geq j} d_k;$$

23.  $tab2=dur\ |\ c\_ncens\ |\ c\_ncomp\ |\ c\_AtRisk;$   
 Matrice d'affichage intermédiaire.
24.  $labtab2=nom_y\ |\ "Cum\_Censored"\ |\ "Cum\_Failure"$   
 $\ |\ "At\_Risk";$   
 Noms des colonnes de la matrice d'affichage intermédiaire.
25.  $print\ "Population\ At\ Risk",$   
 $"*****",\ ,\ ,\ tab2\ [colname=labtab2];$   
 Affichage des nombres d'individus à risque.
26.  $haz=ncomp/c\_AtRisk;$   
 Estimation du hasard :  $\hat{h}_j = d_j/n_j$ .
27.  $pi=1-haz;$   
 Estimation de  $\hat{\pi}_j = 1 - \hat{h}_j$ .
28.  $sdf\_km=j(ndur, 1, 0);$   
 Initialisation du vecteur des estimations de la survie par la méthode de Kaplan-Meier.

29.  $\text{sdf\_km}[1]=\text{pi}[1];$   
 $\widehat{S}(t_{(1)}) = \widehat{\pi}_1.$
30. do  $j=2$  to  $\text{ndur};$   
 Boucle de calcul de l'estimation de la survie.
31.  $\text{sdf\_km}[j]=\text{pi}[j]\#\text{sdf\_km}[j-1];$   
 $\widehat{S}(t_{(j)}) = \widehat{\pi}_j \widehat{S}(t_{(j-1)}) = \prod_{k=1}^j \widehat{\pi}_k.$
32. end;  
 Fin de la boucle.
33.  $\text{vas\_pi}=\text{pi}\#(1-\text{pi})/\text{c\_AtRisk};$   
 $\widehat{\text{Vas}}(\widehat{\pi}_j) = \widehat{\pi}_j(1-\widehat{\pi}_j)/n_j.$
34.  $\text{vas\_ln\_pi}=(1-\text{pi})/(\text{pi}\#\text{c\_AtRisk});$   
 $\widehat{\text{Vas}}(\ln \widehat{\pi}_j) = (1-\widehat{\pi}_j)/(n_j \widehat{\pi}_j).$
35.  $\text{vas\_ln\_sdf\_km}=j(\text{ndur}, 1, 0);$   
 Initialisation du vecteur d'estimations de  $\widehat{\text{Vas}}(\ln \widehat{S}(t_{(j)})).$
36.  $\text{vas\_ln\_sdf\_km}[1]=\text{vas\_ln\_pi}[1];$   
 $\widehat{\text{Vas}}(\ln \widehat{S}(t_{(1)})) = \widehat{\text{Vas}}(\ln \widehat{\pi}_1).$
37. do  $j=2$  to  $\text{ndur};$   
 Début de la boucle d'estimation de  $\widehat{\text{Vas}}(\ln \widehat{S}(t_{(j)})).$
38.  $\text{vas\_ln\_sdf\_km}[j]=\text{vas\_ln\_sdf\_km}[j-1]+\text{vas\_ln\_pi}[j];$   
 $\widehat{\text{Vas}}(\ln \widehat{S}(t_{(j)})) = \widehat{\text{Vas}}(\ln \widehat{S}(t_{(j-1)})) + \widehat{\text{Vas}}(\ln \widehat{\pi}_j) = \sum_{k=1}^j (1-\widehat{\pi}_k)/(n_k \widehat{\pi}_k).$
39. end;  
 Fin de la boucle.
40.  $\text{vas\_sdf\_km}=(\text{sdf\_km}\#\#2)\#\text{vas\_ln\_sdf\_km};$   
 Estimation de la variance de l'estimateur de la fonction de survie par la formule de Greenwood :

$$\widehat{\text{Vas}}(\widehat{S}(t_{(j)})) = [\widehat{S}(t_{(j)})]^2 \sum_{k=1}^j \frac{1-\widehat{\pi}_k}{n_k \widehat{\pi}_k}$$

41.  $\text{Aalen}=j(\text{ndur}, 1, 0);$   
 Initialisation de l'estimation du hasard cumulé par la méthode d'Aalen.
42.  $\text{Aalen}[1]=\text{haz}[1];$   
 $\widehat{H}(t_{(1)}) = \widehat{h}_1.$
43. do  $j=2$  to  $\text{ndur};$   
 Boucle de calcul du hasard cumulé.
44.  $\text{Aalen}[j]=\text{Aalen}[j-1]+\text{haz}[j];$   
 $\widehat{H}(t_{(j)}) = \widehat{H}(t_{(j-1)}) + \widehat{h}_j = \sum_{k=1}^j \widehat{h}_k.$

45. end ;  
Fin de la boucle.
46. vas\_aa=j(ndur,1,0) ;  
Initialisation de l'estimateur de la variance asymptotique du hasard cumulé.
47. vas\_aa[1]=vas\_pi[1] ;  
 $\widehat{\text{Vas}}(\widehat{H}(t_{(1)})) = \widehat{\text{Vas}}(\widehat{\pi}_1)$  car  $\widehat{\pi}_1 = 1 - \widehat{h}_1 \Rightarrow \widehat{\text{Vas}}(\widehat{\pi}_1) = \widehat{\text{Vas}}(\widehat{h}_1)$ .
48. do j=2 to ndur ;  
Boucle de calcul de  $\widehat{\text{Vas}}(\widehat{H}(t_{(j)}))$ .
49. vas\_aa[j]=vas\_aa[j-1]+vas\_pi[j] ;  
 $\widehat{\text{Vas}}(\widehat{H}(t_{(j)})) = \widehat{\text{Vas}}(\widehat{H}(t_{(j-1)})) + \widehat{\text{Vas}}(\widehat{h}_j) = \sum_{k=1}^j \widehat{\text{Vas}}(\widehat{h}_k) = \sum_{k=1}^j \widehat{\text{Vas}}(\widehat{\pi}_k)$ .
50. end ;  
Fin de la boucle.
51. sdf\_b=exp(-Aalen) ;  
Estimation de Breslow de la survie :  $\widehat{S}_B(t_{(j)}) = \exp(-\widehat{H}(t_{(j)}))$ .
52. vas\_sdf\_b=(sdf\_b##2)#vas\_ln\_sdf\_km ;  
Estimation de la variance asymptotique correspondante :

$$\widehat{\text{Vas}}(\widehat{S}_B(t_{(j)})) = [\widehat{S}_B(t_{(j)})]^2 \sum_{k=1}^j \frac{1 - \widehat{\pi}_k}{n_k \widehat{\pi}_k}.$$

53. cdf\_km=1-sdf\_km ; Estimation de la fonction de répartition par la méthode de Kaplan-Meier :

$$\widehat{F}(t_{(j)}) = 1 - \widehat{S}(t_{(j)}).$$

54. cdf\_b=1-sdf\_b ; Estimation de la fonction de répartition par la méthode de Breslow :

$$\widehat{F}_B(t_{(j)}) = 1 - \widehat{S}_B(t_{(j)}).$$

55. pdf\_km=j(ndur,1,0) ; pdf\_b=j(ndur,1,0) ;  
Initialisation des estimations des densités.
56. pdf\_km[1]=1-sdf\_km[1] ; pdf\_b[1]=1-sdf\_b[1] ;  
 $\widehat{f}(t_{(1)}) = S(0) - \widehat{S}(t_{(1)}) = 1 - \widehat{S}(t_{(1)})$  et  $\widehat{f}_B(t_{(1)}) = 1 - \widehat{S}_B(t_{(1)})$ .
57. do j=2 to ndur ;  
Boucle d'estimation des densités ;
58. pdf\_km[j]=sdf\_km[j-1]-sdf\_km[j] ;  
 $\widehat{f}(t_{(j)}) = \widehat{S}(t_{(j-1)}) - \widehat{S}(t_{(j)})$ .

59. pdf\_b[j]=sdf\_b[j-1]-sdf\_b[j];  
 $\hat{f}_B(t_{(j)}) = \hat{S}_B(t_{(j-1)}) - \hat{S}_B(t_{(j)})$ .
60. end;  
 Fin de la boucle.
61. vas\_pdf\_km=j(ndur,1,0); vas\_pdf\_b=j(ndur,1,0);  
 Initialisation des estimations des variances asymptotiques des estimateur des densités.
62. vas\_pdf\_km[1]=vas\_sdf\_km[1];  
 $\widehat{\text{Vas}}(\hat{f}(t_{(1)})) = \widehat{\text{Vas}}(1 - \hat{S}(t_{(1)})) = \widehat{\text{Vas}}(\hat{S}(t_{(1)}))$ .
63. vas\_pdf\_b[1]=vas\_sdf\_b[1];  
 $\widehat{\text{Vas}}(\hat{f}_B(t_{(1)})) = \widehat{\text{Vas}}(\hat{S}_B(t_{(1)}))$ .
64. do j=2 to ndur;  
 Boucle d'estimation de  $\widehat{\text{Vas}}(\hat{f}(t_{(j)}))$  et de  $\widehat{\text{Vas}}(\hat{f}_B(t_{(j)}))$ .
65. vas\_pdf\_km[j]=vas\_sdf\_km[j-1]+vas\_sdf\_km[j]  
 -2#sdf\_km[j-1]#sdf\_km[j]#vas\_ln\_sdf\_km[j-1];  

$$\widehat{\text{Vas}}(\hat{f}(t_{(j)})) = \widehat{\text{Vas}}(\hat{S}(t_{(j-1)})) + \widehat{\text{Vas}}(\hat{S}(t_{(j)}))$$

$$- 2\hat{S}(t_{(j-1)})\hat{S}(t_{(j)})\sum_{k=1}^{j-1} \frac{1-\hat{\pi}_k}{n_k\hat{\pi}_k}$$
66. vas\_pdf\_b[j]=vas\_sdf\_b[j-1]+vas\_sdf\_b[j]  
 -2#sdf\_b[j-1]#sdf\_b[j]#vas\_ln\_sdf\_km[j-1];  

$$\widehat{\text{Vas}}(\hat{f}_B(t_{(j)})) = \widehat{\text{Vas}}(\hat{S}_B(t_{(j-1)})) + \widehat{\text{Vas}}(\hat{S}_B(t_{(j)}))$$

$$- 2\hat{S}_B(t_{(j-1)})\hat{S}_B(t_{(j)})\sum_{k=1}^{j-1} \frac{1-\hat{\pi}_k}{n_k\hat{\pi}_k}$$
67. end;  
 Fin de la boucle.
68. std\_sdf\_km=sqrt(vas\_sdf\_km);  
 std\_haz=sqrt(vas\_pi);  
 std\_pdf\_km=sqrt(vas\_pdf\_km);  
 Calcul des estimations des écarts-types asymptotiques des fonctions.
69. mat\_km\_1=sdf\_km||std\_sdf\_km||cdf\_km;  
 Première matrice d'affichage des résultats pour l'estimation de Kaplan-Meier.

70. `mat_km_2=haz | | std_haz | | pdf_km | | std_pdf_km;`  
 Seconde matrice d'affichage des résultats pour l'estimation de Kaplan-Meier.
71. `char_dur=char(dur);`  
 Conversion des valeurs prises par la variables de durée au format caractère. Lors de l'affichage, elles remplaceront les numéros de ligne.
72. `labkm1="sdf_km" | | "std_sdf_km" | | "cdf_km";`  
 Noms des colonnes du premier affichage.
73. `labkm2="hasard" | | "std_hasard" | | "pdf_km" | |  
 "std_pdf_km";`  
 Noms des colonnes du second affichage.
74. `print "Kaplan Meier estimates", ,  
 mat_km_1[colname=labkm1  
 rowname=char_dur],`
- `mat_km_2[colname=labkm2 rowname=char_dur];`  
 Affichage des résultats de l'estimation par la méthode de Kaplan-Meier.
75. `std_sdf_b=sqrt(vas_sdf_b);`  
`std_pdf_b=sqrt(vas_pdf_b);`  
 Calcul des estimations des écarts-types asymptotiques des fonctions.
76. `mat_b_1=sdf_b | | std_sdf_b | | cdf_b;`  
 Première matrice d'affichage des résultats pour l'estimation de Breslow.
77. `mat_b_2=haz | | std_haz | | pdf_b | | std_pdf_b;`  
 Seconde matrice d'affichage des résultats pour l'estimation de Breslow.
78. `labb1="sdf_Breslow" | | "std_sdf_Breslow" | |  
 "cdf_Breslow";`  
 Noms des colonnes du premier affichage.
79. `labb2="hasard" | | "std_hasard" | | "pdf_Breslow" | |  
 "std_pdf_Breslow";`  
 Noms des colonnes du second affichage.
80. `print "Breslow estimates", , mat_b_1[colname=labb1  
 rowname=char_dur],  
 mat_b_2[colname=labb2 rowname=char_dur];`  
 Affichage des résultats de l'estimation par la méthode de Breslow.
81. `finish estim;`  
 Fin du module d'estimation.

82. `start sortie;`  
Début du module de sortie.
83. `outkm=dur || mat_km_1 || mat_km_2;`  
Tableau de sortie de l'estimation par la méthode de Kaplan-Meier.
84. `labckm=nom_y || labkm1 || labkm2;`  
Noms des colonnes du tableau de sortie.
85. `create &outkm from outkm[colname=labckm];`  
`append from outkm;`  
Création du tableau SAS.
86. `outb=dur || mat_b_1 || mat_b_2;`  
`labcb=nom_y || labb1 || labb2;`  
`create &outb from outb[colname=labcb];`  
`append from outb;`  
Création du tableau de sortie de l'estimation par la méthode de Breslow.
87. `tab=tab || c_atrisk; labtab=labtab || "AT_RISK";`  
Tableau de sortie des statistiques résumant la distribution.
88. `create &outs from tab[colname=labtab];`  
`append from tab;`  
Création du tableau SAS.
89. `finish sortie;`  
Fin du module de sortie.
90. `run lecture;`  
`run estim;`  
`run sortie;`  
Lancement des modules.
91. `quit;`  
`run;`  
Sortie de SAS-IML et lancement du programme.

### 6.1.5 Application XIII

#### Sortie SAS 6.1.

```

 DATA SUMMARY

 tab
DUREE_EMP CENSORED FAILED TOTAL

```

|    |     |      |      |
|----|-----|------|------|
| 1  | 68  | 276  | 344  |
| 2  | 127 | 1337 | 1464 |
| 3  | 116 | 1617 | 1733 |
| 4  | 131 | 1469 | 1600 |
| 5  | 136 | 1298 | 1434 |
| 6  | 80  | 1008 | 1088 |
| 7  | 84  | 787  | 871  |
| 8  | 41  | 606  | 647  |
| 9  | 35  | 433  | 468  |
| 10 | 28  | 367  | 395  |
| 11 | 31  | 326  | 357  |
| 12 | 21  | 232  | 253  |
| 13 | 26  | 210  | 236  |
| 14 | 21  | 170  | 191  |
| 15 | 25  | 116  | 141  |
| 16 | 13  | 108  | 121  |
| 17 | 17  | 87   | 104  |
| 18 | 23  | 83   | 106  |
| 19 | 19  | 82   | 101  |
| 20 | 20  | 44   | 64   |
| 21 | 19  | 62   | 81   |
| 22 | 17  | 34   | 51   |
| 23 | 17  | 38   | 55   |
| 24 | 16  | 45   | 61   |
| 25 | 17  | 35   | 52   |
| 26 | 19  | 28   | 47   |
| 27 | 13  | 25   | 38   |
| 28 | 16  | 26   | 42   |
| 29 | 31  | 18   | 49   |
| 30 | 17  | 19   | 36   |
| 31 | 18  | 15   | 33   |
| 32 | 19  | 12   | 31   |
| 33 | 12  | 23   | 35   |
| 34 | 22  | 23   | 45   |
| 35 | 18  | 25   | 43   |
| 36 | 19  | 24   | 43   |
| 37 | 18  | 22   | 40   |
| 38 | 8   | 24   | 32   |
| 39 | 10  | 15   | 25   |
| 40 | 8   | 14   | 22   |
| 41 | 5   | 16   | 21   |
| 42 | 4   | 17   | 21   |
| 43 | 3   | 17   | 20   |
| 44 | 3   | 7    | 10   |
| 45 | 1   | 4    | 5    |
| 46 | 2   | 10   | 12   |
| 47 | 1   | 18   | 19   |
| 48 | 0   | 5    | 5    |
| 49 | 1   | 0    | 1    |
| 50 | 1   | 0    | 1    |
| 54 | 1   | 0    | 1    |

Population At Risk

\*\*\*\*\*

| tab2      |              |             |         |
|-----------|--------------|-------------|---------|
| DUREE_EMP | Cum_Censored | Cum_Failure | At_Risk |
| 1         | 1418         | 11277       | 12695   |
| 2         | 1350         | 11001       | 12351   |
| 3         | 1223         | 9664        | 10887   |
| 4         | 1107         | 8047        | 9154    |
| 5         | 976          | 6578        | 7554    |
| 6         | 840          | 5280        | 6120    |
| 7         | 760          | 4272        | 5032    |
| 8         | 676          | 3485        | 4161    |
| 9         | 635          | 2879        | 3514    |
| 10        | 600          | 2446        | 3046    |
| 11        | 572          | 2079        | 2651    |
| 12        | 541          | 1753        | 2294    |
| 13        | 520          | 1521        | 2041    |
| 14        | 494          | 1311        | 1805    |
| 15        | 473          | 1141        | 1614    |
| 16        | 448          | 1025        | 1473    |
| 17        | 435          | 917         | 1352    |
| 18        | 418          | 830         | 1248    |
| 19        | 395          | 747         | 1142    |
| 20        | 376          | 665         | 1041    |
| 21        | 356          | 621         | 977     |
| 22        | 337          | 559         | 896     |
| 23        | 320          | 525         | 845     |
| 24        | 303          | 487         | 790     |
| 25        | 287          | 442         | 729     |
| 26        | 270          | 407         | 677     |
| 27        | 251          | 379         | 630     |
| 28        | 238          | 354         | 592     |
| 29        | 222          | 328         | 550     |
| 30        | 191          | 310         | 501     |
| 31        | 174          | 291         | 465     |
| 32        | 156          | 276         | 432     |
| 33        | 137          | 264         | 401     |
| 34        | 125          | 241         | 366     |
| 35        | 103          | 218         | 321     |
| 36        | 85           | 193         | 278     |
| 37        | 66           | 169         | 235     |
| 38        | 48           | 147         | 195     |
| 39        | 40           | 123         | 163     |
| 40        | 30           | 108         | 138     |
| 41        | 22           | 94          | 116     |
| 42        | 17           | 78          | 95      |
| 43        | 13           | 61          | 74      |
| 44        | 10           | 44          | 54      |
| 45        | 7            | 37          | 44      |
| 46        | 6            | 33          | 39      |
| 47        | 4            | 23          | 27      |
| 48        | 3            | 5           | 8       |
| 49        | 3            | 0           | 3       |
| 50        | 2            | 0           | 2       |
| 54        | 1            | 0           | 1       |

## Sortie SAS 6.2.

Kaplan Meier estimates

|    | mat_km_1  | sdf_km    | std_sdf_km | cdf_km |
|----|-----------|-----------|------------|--------|
| 1  | 0.9782592 | 0.0012943 | 0.0217408  |        |
| 2  | 0.8723623 | 0.0029685 | 0.1276377  |        |
| 3  | 0.742794  | 0.0039024 | 0.257206   |        |
| 4  | 0.6235932 | 0.0043421 | 0.3764068  |        |
| 5  | 0.5164415 | 0.0045007 | 0.4835585  |        |
| 6  | 0.4313805 | 0.0044865 | 0.5686195  |        |
| 7  | 0.363913  | 0.0043823 | 0.636087   |        |
| 8  | 0.3109134 | 0.0042401 | 0.6890866  |        |
| 9  | 0.2726022 | 0.0040979 | 0.7273978  |        |
| 10 | 0.2397575 | 0.0039465 | 0.7602425  |        |
| 11 | 0.2102739 | 0.003784  | 0.7897261  |        |
| 12 | 0.1890082 | 0.0036498 | 0.8109918  |        |
| 13 | 0.169561  | 0.0035123 | 0.830439   |        |
| 14 | 0.1535913 | 0.0033884 | 0.8464087  |        |
| 15 | 0.1425525 | 0.0032962 | 0.8574475  |        |
| 16 | 0.1321006 | 0.0032043 | 0.8678994  |        |
| 17 | 0.1236    | 0.003125  | 0.8764     |        |
| 18 | 0.1153798 | 0.0030447 | 0.8846202  |        |
| 19 | 0.1070951 | 0.0029603 | 0.8929049  |        |
| 20 | 0.1025685 | 0.0029128 | 0.8974315  |        |
| 21 | 0.0960596 | 0.0028428 | 0.9039404  |        |
| 22 | 0.0924145 | 0.0028028 | 0.9075855  |        |
| 23 | 0.0882585 | 0.0027567 | 0.9117415  |        |
| 24 | 0.0832312 | 0.0026996 | 0.9167688  |        |
| 25 | 0.0792352 | 0.0026531 | 0.9207648  |        |
| 26 | 0.0759581 | 0.0026147 | 0.9240419  |        |
| 27 | 0.0729439 | 0.0025795 | 0.9270561  |        |
| 28 | 0.0697402 | 0.0025416 | 0.9302598  |        |
| 29 | 0.0674578 | 0.0025147 | 0.9325422  |        |
| 30 | 0.0648996 | 0.0024869 | 0.9351004  |        |
| 31 | 0.062806  | 0.0024647 | 0.937194   |        |
| 32 | 0.0610614 | 0.0024471 | 0.9389386  |        |
| 33 | 0.0575591 | 0.0024133 | 0.9424409  |        |
| 34 | 0.053942  | 0.0023766 | 0.946058   |        |
| 35 | 0.0497409 | 0.0023353 | 0.9502591  |        |
| 36 | 0.0454468 | 0.0022923 | 0.9545532  |        |
| 37 | 0.0411922 | 0.00225   | 0.9588078  |        |
| 38 | 0.0361224 | 0.0021982 | 0.9638776  |        |
| 39 | 0.0327982 | 0.002157  | 0.9672018  |        |
| 40 | 0.0294709 | 0.0021136 | 0.9705291  |        |
| 41 | 0.0254059 | 0.0020518 | 0.9745941  |        |
| 42 | 0.0208596 | 0.0019587 | 0.9791404  |        |
| 43 | 0.0160675 | 0.0018212 | 0.9839325  |        |
| 44 | 0.0139847 | 0.001747  | 0.9860153  |        |
| 45 | 0.0127134 | 0.0016999 | 0.9872866  |        |
| 46 | 0.0094535 | 0.0015453 | 0.9905465  |        |
| 47 | 0.0031512 | 0.0010004 | 0.9968488  |        |
| 48 | 0.0011817 | 0.000657  | 0.9988183  |        |
| 49 | 0.0011817 | 0.000657  | 0.9988183  |        |

```

50 0.0011817 0.000657 0.9988183
54 0.0011817 0.000657 0.9988183
 mat_km_2
 hasard std_hasard pdf_km std_pdf_km
1 0.0217408 0.0012943 0.0217408 0.0012943
2 0.1082503 0.0027957 0.1058969 0.0027385
3 0.1485258 0.0034083 0.1295683 0.0030057
4 0.1604763 0.0038363 0.1192008 0.0029176
5 0.1718295 0.0043403 0.1071517 0.0028075
6 0.1647059 0.0047413 0.0850609 0.0025584
7 0.156399 0.0051205 0.0674675 0.0023177
8 0.1456381 0.0054684 0.0529996 0.0020899
9 0.1232214 0.0055448 0.0383112 0.0018014
10 0.1204859 0.0058983 0.0328447 0.001682
11 0.1229725 0.0063783 0.0294836 0.0016044
12 0.1011334 0.006295 0.0212657 0.0013779
13 0.1028907 0.006725 0.0194472 0.0013254
14 0.0941828 0.0068749 0.0159697 0.0012117
15 0.0718711 0.0064288 0.0110388 0.001017
16 0.0733198 0.0067916 0.0104519 0.0009979
17 0.0643491 0.0066733 0.0085006 0.0009053
18 0.0665064 0.0070531 0.0082202 0.0008962
19 0.0718039 0.0076394 0.0082847 0.0009081
20 0.0422671 0.0062359 0.0045266 0.0006795
21 0.0634596 0.0077995 0.006509 0.0008211
22 0.0379464 0.0063831 0.0036451 0.0006226
23 0.0449704 0.0071292 0.0041559 0.0006708
24 0.056962 0.008246 0.0050274 0.0007445
25 0.048011 0.0079181 0.003996 0.0006717
26 0.0413589 0.0076528 0.0032771 0.0006162
27 0.0396825 0.0077774 0.0030142 0.0005998
28 0.0439189 0.0084219 0.0032036 0.0006247
29 0.0327273 0.0075866 0.0022824 0.0005356
30 0.0379242 0.0085338 0.0025583 0.0005835
31 0.0322581 0.0081936 0.0020935 0.0005378
32 0.0277778 0.0079066 0.0017446 0.0005013
33 0.0573566 0.0116116 0.0035023 0.0007228
34 0.0628415 0.012685 0.0036171 0.0007457
35 0.0778816 0.0149575 0.0042011 0.0008278
36 0.0863309 0.0168444 0.0042942 0.0008618
37 0.093617 0.019002 0.0042546 0.0008898
38 0.1230769 0.0235262 0.0050698 0.0010079
39 0.0920245 0.022641 0.0033241 0.0008425
40 0.1014493 0.0257014 0.0033274 0.0008709
41 0.137931 0.0320164 0.0040649 0.0009876
42 0.1789474 0.0393266 0.0045463 0.0010645
43 0.2297297 0.0489006 0.0047921 0.0011149
44 0.1296296 0.0457096 0.0020828 0.0007714
45 0.0909091 0.0433392 0.0012713 0.0006265
46 0.2564103 0.0699201 0.0032598 0.00099
47 0.6666667 0.0907218 0.0063024 0.0013405
48 0.625 0.1711633 0.0019695 0.0008258
49 0 0 0 0
50 0 0 0 0
54 0 0 0 0

```

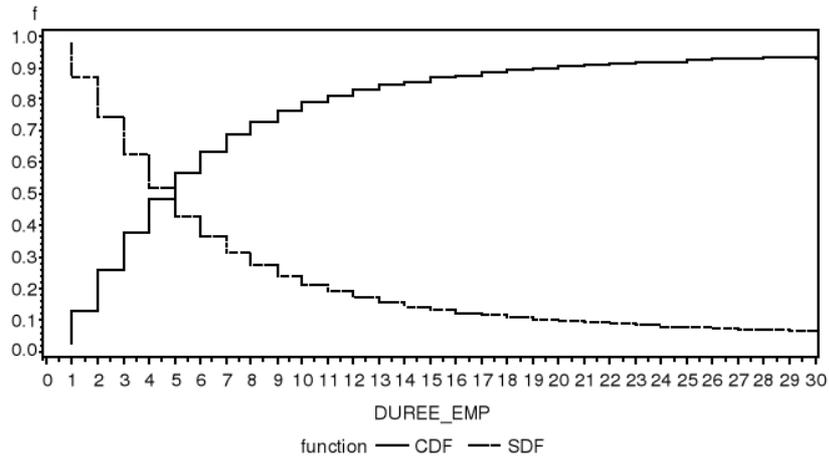
Breslow estimates

```

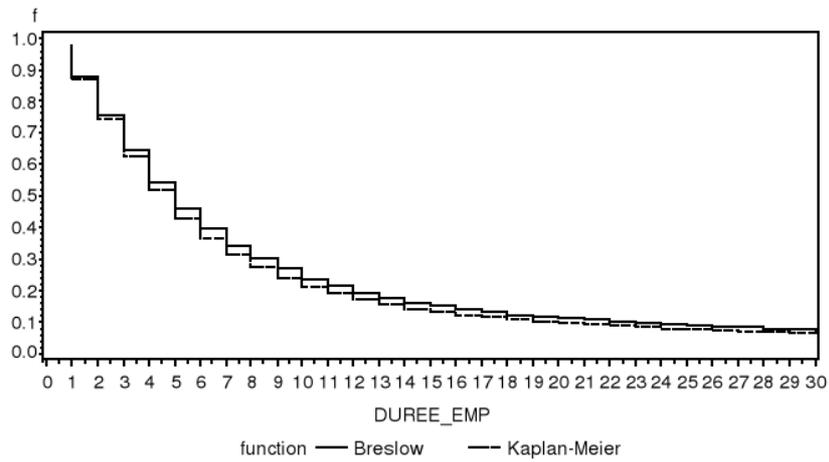
 mat_b_1
sdf_Breslow std_sdf_Breslow cdf_Breslow
1 0.9784938 0.0012947 0.0215062
2 0.8781032 0.0029888 0.1218968
3 0.7569054 0.0039765 0.2430946
4 0.6446851 0.0044889 0.3553149
5 0.542904 0.0047314 0.457096
6 0.4604603 0.004789 0.5395397
7 0.3937939 0.0047421 0.6062061
8 0.3404232 0.0046425 0.6595768
9 0.3009572 0.0045241 0.6990428
10 0.2667954 0.0043916 0.7332046
11 0.235924 0.0042456 0.764076
12 0.2132311 0.0041175 0.7867689
13 0.1923825 0.0039851 0.8076175
14 0.1750905 0.0038627 0.8249095
15 0.1629481 0.0037678 0.8370519
16 0.1514283 0.0036731 0.8485717
17 0.1419909 0.00359 0.8580091
18 0.1328548 0.0035058 0.8671452
19 0.1236497 0.0034179 0.8763503
20 0.1185323 0.0033661 0.8814677
21 0.111244 0.0032922 0.888756
22 0.1071018 0.0032483 0.8928982
23 0.1023921 0.0031981 0.8976079
24 0.0967226 0.0031372 0.9032774
25 0.0921886 0.0030869 0.9078114
26 0.0884535 0.0030448 0.9115465
27 0.0850122 0.0030063 0.9149878
28 0.0813593 0.002965 0.9186407
29 0.0787398 0.0029352 0.9212602
30 0.0758096 0.0029049 0.9241904
31 0.0734031 0.0028805 0.9265969
32 0.0713922 0.0028612 0.9286078
33 0.0674126 0.0028264 0.9325874
34 0.0633066 0.0027891 0.9366934
35 0.0585633 0.0027495 0.9414367
36 0.0537196 0.0027096 0.9462804
37 0.0489187 0.0026721 0.9510813
38 0.0432537 0.0026322 0.9567463
39 0.039451 0.0025945 0.960549
40 0.035645 0.0025563 0.964355
41 0.0310525 0.0025079 0.9689475
42 0.0259645 0.002438 0.9740355
43 0.0206353 0.0023389 0.9793647
44 0.0181265 0.0022644 0.9818735
45 0.0165513 0.002213 0.9834487
46 0.0128078 0.0020936 0.9871922
47 0.0065757 0.0020877 0.9934243
48 0.0035197 0.0019569 0.9964803
49 0.0035197 0.0019569 0.9964803
50 0.0035197 0.0019569 0.9964803

```

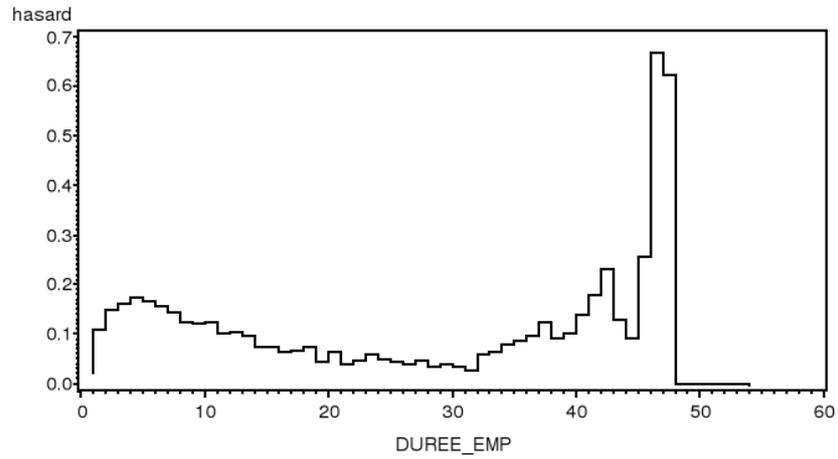
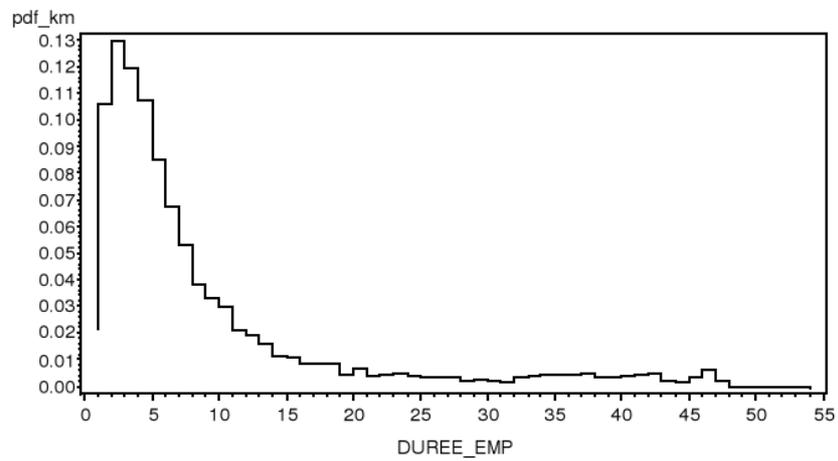
|    | 54        | 0.0035197  | 0.0019569   | 0.9964803       |
|----|-----------|------------|-------------|-----------------|
|    |           | mat_b_2    |             |                 |
|    | hasard    | std_hasard | pdf_Breslow | std_pdf_Breslow |
| 1  | 0.0217408 | 0.0012943  | 0.0215062   | 0.0012947       |
| 2  | 0.1082503 | 0.0027957  | 0.1003906   | 0.0027561       |
| 3  | 0.1485258 | 0.0034083  | 0.1211977   | 0.0030577       |
| 4  | 0.1604763 | 0.0038363  | 0.1122203   | 0.0030044       |
| 5  | 0.1718295 | 0.0043403  | 0.1017811   | 0.0029322       |
| 6  | 0.1647059 | 0.0047413  | 0.0824437   | 0.0027106       |
| 7  | 0.156399  | 0.0051205  | 0.0666664   | 0.0024888       |
| 8  | 0.1456381 | 0.0054684  | 0.0533707   | 0.0022717       |
| 9  | 0.1232214 | 0.0055448  | 0.039466    | 0.0019779       |
| 10 | 0.1204859 | 0.0058983  | 0.0341618   | 0.0018614       |
| 11 | 0.1229725 | 0.0063783  | 0.0308714   | 0.0017895       |
| 12 | 0.1011334 | 0.006295   | 0.0226929   | 0.0015482       |
| 13 | 0.1028907 | 0.006725   | 0.0208486   | 0.0014973       |
| 14 | 0.0941828 | 0.0068749  | 0.017292    | 0.0013763       |
| 15 | 0.0718711 | 0.0064288  | 0.0121424   | 0.00116         |
| 16 | 0.0733198 | 0.0067916  | 0.0115198   | 0.0011413       |
| 17 | 0.0643491 | 0.0066733  | 0.0094374   | 0.0010383       |
| 18 | 0.0665064 | 0.0070531  | 0.0091361   | 0.00103         |
| 19 | 0.0718039 | 0.0076394  | 0.009205    | 0.0010463       |
| 20 | 0.0422671 | 0.0062359  | 0.0051174   | 0.0007846       |
| 21 | 0.0634596 | 0.0077995  | 0.0072883   | 0.0009493       |
| 22 | 0.0379464 | 0.0063831  | 0.0041422   | 0.0007211       |
| 23 | 0.0449704 | 0.0071292  | 0.0047097   | 0.0007776       |
| 24 | 0.056962  | 0.008246   | 0.0056695   | 0.0008641       |
| 25 | 0.048011  | 0.0079181  | 0.004534    | 0.0007807       |
| 26 | 0.0413589 | 0.0076528  | 0.003735    | 0.0007171       |
| 27 | 0.0396825 | 0.0077774  | 0.0034413   | 0.0006986       |
| 28 | 0.0439189 | 0.0084219  | 0.0036528   | 0.0007282       |
| 29 | 0.0327273 | 0.0075866  | 0.0026196   | 0.0006249       |
| 30 | 0.0379242 | 0.0085338  | 0.0029302   | 0.0006813       |
| 31 | 0.0322581 | 0.0081936  | 0.0024064   | 0.0006283       |
| 32 | 0.0277778 | 0.0079066  | 0.0020109   | 0.0005859       |
| 33 | 0.0573566 | 0.0116116  | 0.0039796   | 0.0008456       |
| 34 | 0.0628415 | 0.012685   | 0.0041059   | 0.000874        |
| 35 | 0.0778816 | 0.0149575  | 0.0047433   | 0.0009727       |
| 36 | 0.0863309 | 0.0168444  | 0.0048437   | 0.0010161       |
| 37 | 0.093617  | 0.019002   | 0.0048008   | 0.0010538       |
| 38 | 0.1230769 | 0.0235262  | 0.005665    | 0.001201        |
| 39 | 0.0920245 | 0.022641   | 0.0038027   | 0.0010106       |
| 40 | 0.1014493 | 0.0257014  | 0.003806    | 0.0010498       |
| 41 | 0.137931  | 0.0320164  | 0.0045926   | 0.0011994       |
| 42 | 0.1789474 | 0.0393266  | 0.005088    | 0.0013098       |
| 43 | 0.2297297 | 0.0489006  | 0.0053293   | 0.0014024       |
| 44 | 0.1296296 | 0.0457096  | 0.0025088   | 0.0009935       |
| 45 | 0.0909091 | 0.0433392  | 0.0015752   | 0.0008132       |
| 46 | 0.2564103 | 0.0699201  | 0.0037435   | 0.0013042       |
| 47 | 0.6666667 | 0.0907218  | 0.006232    | 0.0020593       |
| 48 | 0.625     | 0.1711633  | 0.003056    | 0.0018768       |
| 49 | 0         | 0          | 0           | 0               |
| 50 | 0         | 0          | 0           | 0               |
| 54 | 0         | 0          | 0           | 0               |



GRAPHIQUE 6.1 – KAPLAN MEIER, FONCTIONS DE RÉPARTITION  $\hat{F}_T(t)$  ET DE SURVIE  $\hat{S}_T(t)$



GRAPHIQUE 6.2 – ESTIMATIONS DE BRESLOW ET DE KAPLAN-MEIER,  $\hat{S}_T(t)$

GRAPHIQUE 6.3 – KAPLAN-MEIER, FONCTION DE HASARD  $\hat{h}_T(t)$ GRAPHIQUE 6.4 – KAPLAN-MEIER, FONCTION DE DENSITÉ  $\hat{f}_T(t)$

## 6.2 Estimateur actuariel

### 6.2.1 Dérivation

Cet estimateur est également appelé "life table". Pour cet estimateur, on considère que la distribution est continue plutôt que discrète. On fait juste l'hypothèse que le hasard est constant par morceaux, de sorte que au sein de chaque intervalle :

$$h(t) = \eta_j, \quad t_{(j-1)} \leq t < t_{(j)},$$

avec la convention  $t_{(0)} = 0$ . Cette forme est beaucoup plus souple qu'il n'y paraît car elle autorise n'importe quel profil de variation, avec un degré de détails qui augmente avec le nombre de seuils. Le coefficient  $h_j$  représente la valeur théorique du hasard. On suppose de plus que la censure est déterminée par un processus aléatoire résumé par sa propre fonction de hasard constante par morceaux :

$$h_c(t) = \kappa_j, \quad t_{(j-1)} \leq t < t_{(j)}$$

Pour simplifier les notations, la largeur de l'intervalle  $j$  est notée :

$$b_j = t_{(j)} - t_{(j-1)},$$

et l'on dispose de  $n_j$  individus à risques juste après la durée  $t_{(j-1)}$ , dont  $d_j$  sortent pendant  $[t_{(j-1)}, t_{(j)}[$  et  $c_j$  sont censurés pendant  $[t_{(j-1)}, t_{(j)}[$ . Le fait que les hasards soient constants au sein de chaque intervalle permet d'utiliser des lois exponentielles dont la densité est de la forme  $\gamma \exp(-\gamma t)$  et la fonction de survie de la forme  $\exp(-\gamma t)$ . Pour déterminer les estimateurs des paramètres  $h_j$  et  $\kappa_j$ , on doit considérer trois cas possibles :

- les individus qui survivent à l'intervalle  $[t_{(j-1)}, t_{(j)}[$ . Ils sont en nombre  $n_j - c_j - d_j$ . Sur l'intervalle  $[t_{(j-1)}, t_{(j)}[$  le hasard est constant, on est donc en présence d'une loi exponentielle. Le processus de sortie suit une loi exponentielle de paramètre  $\eta_j$  et le processus de censure une autre loi exponentielle de paramètre  $\kappa_j$ . La probabilité que l'individu dépasse la durée  $b_j$  est donc égale à :

$$\begin{aligned} \Pr[(T > b_j) \cap (T_c > b_j)] &= S(b_j) S_c(b_j) \\ &= \exp(-\eta_j b_j) \exp(-\kappa_j b_j) \\ &= \exp\{-(\eta_j + \kappa_j) b_j\}, \end{aligned}$$

- les  $d_j$  individus qui sortent dans l'intervalle  $[t_{(j-1)}, t_{(j)}[$  sans être censurés. La probabilité instantanée de cet événement pour une durée  $v$  est

égale à  $\eta_j \exp(-\eta_j v)$ , et la durée de censure doit dépasser  $v$ , avec une probabilité  $S_c(v)$ , donc la probabilité associée à ce groupe est égale à :

$$\begin{aligned} \int_0^{b_j} \eta_j \exp(-\eta_j v) \exp(-\kappa_j v) dv &= \int_0^{b_j} \eta_j \exp(-(\eta_j + \kappa_j) v) dv \\ &= \left[ -\frac{\eta_j}{\eta_j + \kappa_j} \exp(-(\eta_j + \kappa_j) v) \right]_0^{b_j} \\ &= \frac{\eta_j}{\eta_j + \kappa_j} \{1 - \exp(-(\eta_j + \kappa_j) b_j)\}, \end{aligned}$$

- les  $c_j$  individus qui sont censurés dans l'intervalle. Pour une durée de censure  $v$ , intervenant avec la densité  $\kappa_j \exp(-\kappa_j v)$ , il faut que la durée de l'évènement soit supérieure à  $v$ , avec une probabilité  $S(v)$ , donc la probabilité associée à ce groupe est égale à :

$$\begin{aligned} \int_0^{b_j} \kappa_j \exp(-\kappa_j v) \exp(-\eta_j v) dv &= \int_0^{b_j} \kappa_j \exp(-(\eta_j + \kappa_j) v) dv \\ &= \left[ -\frac{\kappa_j}{\eta_j + \kappa_j} \exp(-(\eta_j + \kappa_j) v) \right]_0^{b_j} \\ &= \frac{\kappa_j}{\eta_j + \kappa_j} \{1 - \exp(-(\eta_j + \kappa_j) b_j)\}, \end{aligned}$$

La log-vraisemblance associée à l'intervalle  $[t_{(j-1)}, t_{(j)}[$  est donc égale à :

$$\begin{aligned} \ell_j(\eta_j, \kappa_j) &= -(n_j - c_j - d_j) b_j (\eta_j + \kappa_j) + d_j \ln \left( \frac{\eta_j}{\eta_j + \kappa_j} \right) \\ &\quad + c_j \ln \left( \frac{\kappa_j}{\eta_j + \kappa_j} \right) + (d_j + c_j) \ln \{1 - \exp(-(\eta_j + \kappa_j) b_j)\} \end{aligned}$$

Comme les log-vraisemblances sont séparées par intervalle, grâce à la définition de  $n_j$ , on obtient les conditions du premier ordre suivantes :

$$\begin{aligned} \frac{\partial \ell_j}{\partial \eta_j} &= -(n_j - c_j - d_j) b_j + \frac{d_j}{\hat{\eta}_j} - \frac{c_j + d_j}{\hat{\eta}_j + \hat{\kappa}_j} \\ &\quad + \frac{(d_j + c_j) b_j \exp\{-b_j (\hat{\eta}_j + \hat{\kappa}_j)\}}{1 - \exp\{-b_j (\hat{\eta}_j + \hat{\kappa}_j)\}} = 0 \end{aligned}$$

$$\begin{aligned} \frac{\partial \ell_j}{\partial \kappa_j} &= -(n_j - c_j - d_j) b_j + \frac{c_j}{\hat{\kappa}_j} - \frac{c_j + d_j}{\hat{\eta}_j + \hat{\kappa}_j} \\ &\quad + \frac{(d_j + c_j) b_j \exp\{-b_j (\hat{\eta}_j + \hat{\kappa}_j)\}}{1 - \exp\{-b_j (\hat{\eta}_j + \hat{\kappa}_j)\}} = 0 \end{aligned}$$

En faisant la différence de ces deux équations, on trouve :

$$\begin{aligned}\frac{d_j}{\widehat{\eta}_j} &= \frac{c_j}{\widehat{\kappa}_j} \\ \Rightarrow \frac{d_j}{\widehat{\eta}_j} &= \frac{c_j}{\widehat{\kappa}_j} = \frac{c_j + d_j}{\widehat{\eta}_j + \widehat{\kappa}_j},\end{aligned}$$

en reportant ce résultat dans les conditions du premier ordre, on obtient :

$$\begin{aligned}\widehat{\eta}_j &= -\frac{d_j}{b_j(c_j + d_j)} \ln\left(1 - \frac{c_j + d_j}{n_j}\right), \\ \widehat{\kappa}_j &= -\frac{c_j}{b_j(c_j + d_j)} \ln\left(1 - \frac{c_j + d_j}{n_j}\right),\end{aligned}$$

pour obtenir l'expression usuelle de l'estimateur actuariel, on approxime le logarithme par :

$$-\ln(1-x) \stackrel{0}{\approx} x + \frac{1}{2}x^2,$$

ce qui permet d'écrire :

$$\widehat{\eta}_j \approx \frac{d_j}{b_j n_j} \left(1 + \frac{(c_j + d_j)/2}{n_j}\right),$$

on utilise ensuite l'approximation :

$$\frac{1}{1-x} \stackrel{0}{\approx} 1+x,$$

ce qui donne :

$$\widehat{\eta}_j \approx \frac{d_j}{b_j} \left(\frac{1}{n_j - (c_j + d_j)/2}\right).$$

Arrivé à ce stade, il nous reste à estimer le taux de hasard sur la période  $[t_{(j-1)}, t_{(j)}]$ . On utilise la relation :

$$h_j = \int_0^{b_j} \eta_j \exp(-\eta_j v) dv = 1 - \exp(-\eta_j b_j),$$

en utilisant l'approximation :

$$1 - \exp(-x) \stackrel{0}{\approx} x,$$

on obtient :

$$h_j \approx \eta_j b_j,$$

ce qui permet d'estimer  $h_j$  par :

$$\hat{h}_j = \hat{\eta}_j b_j = \frac{d_j}{n_j - (c_j + d_j) / 2}.$$

On définit alors le nombre d'individus à risque corrigé par :

$$n'_j = n_j - (c_j + d_j) / 2,$$

ce qui permet d'écrire :

$$\hat{h}_j = \frac{d_j}{n'_j}$$

d'où l'estimateur actuariel de la fonction de survie :

$$\hat{S}_A(t_{(j)}) = \prod_{k=1}^j \left( 1 - \frac{d_k}{n'_k} \right),$$

en appliquant la méthode de Greenwood, on obtient la variance asymptotique suivante :

$$\text{Vas}(\hat{S}_A(t_{(j)})) = \left[ \hat{S}_A(t_{(j)}) \right]^2 \sum_{k=1}^j \frac{d_k}{n'_k (n'_k - d_k)}.$$

### 6.2.2 Programmation XIV : la procédure lifetest

Nous pourrions programmer les estimateurs de cette section comme ceux de la section précédente, mais il est plus intéressant, sur le plan pédagogique, de voir comment fonctionne la procédure SAS qui effectue ces estimations. Il s'agit de la procédure *lifetest*. Nous avons choisi l'estimateur actuariel car sa matrice de sortie ne présente pas de duplications, contrairement à celle de l'estimateur de Kaplan-Meier. Voici le programme que nous avons utilisé :

```
1. %let tab=duree;
 %let y=duree_emp;
 %let d=presence;
```

La macro-variable &tab contient le nom du tableau SAS sur lequel on réalise l'estimation, &y le nom de la variable de durée et &d le nom de l'indicatrice de présence. Cette indicatrice vaut 1 si la durée &y est complète, et vaut 0 si la durée &y est censurée.

```
2. proc lifetest data=&tab method=ACT outsurv=s_act;
 On appelle la procédure lifetest que l'on appliquera au tableau &tab. On
```

choisit d'estimer les fonctions avec l'estimateur actuariel (`method=ACT`) et de sauvegarder les résultats d'estimation dans un tableau SAS nommé `s_act` (`outsurv=s_act`) qui sera situé sous le répertoire de travail par défaut.

3. `time &y*&d(0);`  
L'instruction `time` indique les variables de durée et de censure selon la syntaxe :  
`time variable*censure(valeurs)`  
où `valeurs` indique les valeurs de la variable censure qui correspondent à la censure. On sépare ces valeurs par des virgules. Ici il n'y a qu'une valeur : 0. Notre programme sera donc équivalent à :  
`time duree_emp*presence(0); run;`  
Lancement de la procédure avec les intervalles par défaut.
4. `proc lifetest data=&tab method=ACT outsurv=s_act`  
`intervals=1 to 50 by 1;`  
`time &y*&d(0);`  
`run;`  
Lancement de la procédure avec des intervalles choisis par l'utilisateur. L'instruction `intervals` permet de choisir ici une instruction point par point de 1 an à 50 ans par pas de 1 an, selon la syntaxe `intervals=début to fin by pas;`
5. `data s_act; set s_act;`  
`label`  
`survival="S(t)"`  
`pdf="f(t)"`  
`haz_lcl="h(t)"`  
`hazard="h(t)"`  
`&y="T";`  
Le fichier `s_act` contient les estimations réalisées par la procédure `lifetest`. Ici, on ajoute des étiquettes pour les graphiques via l'instruction `label`.
6. On trace les graphiques des fonctions de survie, de la densité et du hasard. Dans la base de données en sortie, les fonctions sont identifiées par les variables `survival`, `pdf` (pour *probability density function*) et `hazard`.  
`symbol1 c=black i=stepJR v=none;`  
L'instruction `i=stepJR` permet de tracer une fonction en escalier.
7. `proc gplot data=s_act;`  
`plot survival*&y/haxis=0 to 50 by 5;`  
`proc gplot data=s_act;`

```

plot pdf*&y/haxis=0 to 50 by 5;
proc gplot data=s_act;
plot hazard*&y/haxis=0 to 50 by 5;
run;

```

8. Voici une manière de tracer la fonction de hasard avec son intervalle de confiance pour les durées inférieures à 45 ans :

```

data s_act; set s_act;
if &y<=45;
symbol1 c=black i=stepJR v=none line=3;
symbol2 c=black i=stepJR v=none line=1;
symbol3 c=black i=stepJR v=none line=3;
proc gplot data=s_act;
plot (haz_lcl hazard haz_ucl)*&y/haxis=0 to 45
by 5 overlay;
run;

```

### 6.2.3 Application XIV

Par défaut, la procédure choisit les intervalles elle-même. Ici, il s'agit d'intervalles de 10 ans qui sont trop larges par rapport aux données disponibles. Nous pouvons effectuer des estimations point par point sur une grande partie de la fonction de survie. C'est la raison pour laquelle nous utilisons l'instruction *interval*.

#### Sortie SAS 6.3.

The LIFETEST Procedure

Life Table Survival Estimates

| Interval |        | Number | Number   | Effective | Conditional |
|----------|--------|--------|----------|-----------|-------------|
| [Lower,  | Upper) | Failed | Censored | Sample    | Probability |
|          |        |        |          | Size      | of Failure  |
| 0        | 10     | 8831   | 818      | 12286.0   | 0.7188      |
| 10       | 20     | 1781   | 224      | 2934.0    | 0.6070      |
| 20       | 30     | 355    | 185      | 948.5     | 0.3743      |
| 30       | 40     | 202    | 161      | 420.5     | 0.4804      |
| 40       | 50     | 108    | 28       | 124.0     | 0.8710      |
| 50       | 60     | 0      | 2        | 1.0       | 0           |
| 60       | .      | 0      | 0        | 0.0       | 0           |

| Interval |        | Conditional |          |         | Survival |
|----------|--------|-------------|----------|---------|----------|
| [Lower,  | Upper) | Probability | Survival | Failure | Standard |
|          |        | Standard    |          |         | Error    |
|          |        | Error       |          |         |          |
| 0        | 10     | 0.0000      | 0.7188   | 0.0000  | 0.0000   |
| 10       | 20     | 0.0000      | 0.6070   | 0.0000  | 0.0000   |
| 20       | 30     | 0.0000      | 0.3743   | 0.0000  | 0.0000   |
| 30       | 40     | 0.0000      | 0.4804   | 0.0000  | 0.0000   |
| 40       | 50     | 0.0000      | 0.8710   | 0.0000  | 0.0000   |
| 50       | 60     | 0.0000      | 0.0000   | 0.0000  | 0.0000   |
| 60       | .      | 0.0000      | 0.0000   | 0.0000  | 0.0000   |

|    |    |         |         |        |         |
|----|----|---------|---------|--------|---------|
| 0  | 10 | 0.00406 | 1.0000  | 0      | 0       |
| 10 | 20 | 0.00902 | 0.2812  | 0.7188 | 0.00406 |
| 20 | 30 | 0.0157  | 0.1105  | 0.8895 | 0.00300 |
| 30 | 40 | 0.0244  | 0.0691  | 0.9309 | 0.00255 |
| 40 | 50 | 0.0301  | 0.0359  | 0.9641 | 0.00214 |
| 50 | 60 | 0       | 0.00464 | 0.9954 | 0.00112 |
| 60 | .  | 0       | 0.00464 | 0.9954 | 0.00112 |

| Interval |        | Median   | Median   |
|----------|--------|----------|----------|
| [Lower,  | Upper) | Residual | Standard |
|          |        | Lifetime | Error    |
| 0        | 10     | 6.9562   | 0.0628   |
| 10       | 20     | 8.2369   | 0.1521   |
| 20       | 30     | 14.1827  | 0.5401   |
| 30       | 40     | 10.4335  | 0.5388   |
| 40       | 50     | 5.7407   | 0.5155   |
| 50       | 60     | .        | .        |
| 60       | .      | .        | .        |

Evaluated at the Midpoint of the Interval

| Interval |        | PDF     | PDF      | Hazard   | Hazard   |
|----------|--------|---------|----------|----------|----------|
| [Lower,  | Upper) | PDF     | Standard | Hazard   | Standard |
|          |        |         | Error    |          | Error    |
| 0        | 10     | 0.0719  | 0.000406 | 0.112204 | 0.000988 |
| 10       | 20     | 0.0171  | 0.000353 | 0.087154 | 0.001859 |
| 20       | 30     | 0.00414 | 0.000207 | 0.046044 | 0.002378 |
| 30       | 40     | 0.00332 | 0.000208 | 0.063224 | 0.00422  |
| 40       | 50     | 0.00313 | 0.000216 | 0.154286 | 0.009447 |
| 50       | 60     | 0       | .        | 0        | .        |
| 60       | .      | .       | .        | .        | .        |

The LIFETEST Procedure

Summary of the Number of Censored and Uncensored Values

| Total | Failed | Censored | Percent  |
|-------|--------|----------|----------|
|       |        |          | Censored |
| 12695 | 11277  | 1418     | 11.17    |

La sortie suivante montre le début de l'estimation, pour les 25 premières années. Ces estimations se trouvent dans le tableau indiqué par l'option *outsurv*. Les abréviations viennent de l'anglais :

- SURVIVAL : estimateur actuariel de la fonction de survie (graphique 6.5) ;
- SDF\_LCL : borne inférieure de l'estimateur de la fonction de survie, pour un intervalle de confiance asymptotique de niveau 95% (par défaut, peut être modifié) ;
- SDF\_UCL : borne supérieure de l'estimateur de la fonction de survie (à 95%) ;
- PDF : estimateur de la densité (graphique 6.6) ;

- PDF\_LCL : borne inférieure de l'estimateur de la densité (à 95%) ;
- PDF\_UCL : borne supérieure de l'estimateur de la densité (à 95%) ;
- HAZARD : estimateur du hazard (graphique 6.7) ;
- HAZ\_LCL : borne inférieure de l'estimateur du hazard (à 95%) ;
- HAZ\_UCL : borne supérieure de l'estimateur du hazard (à 95%).

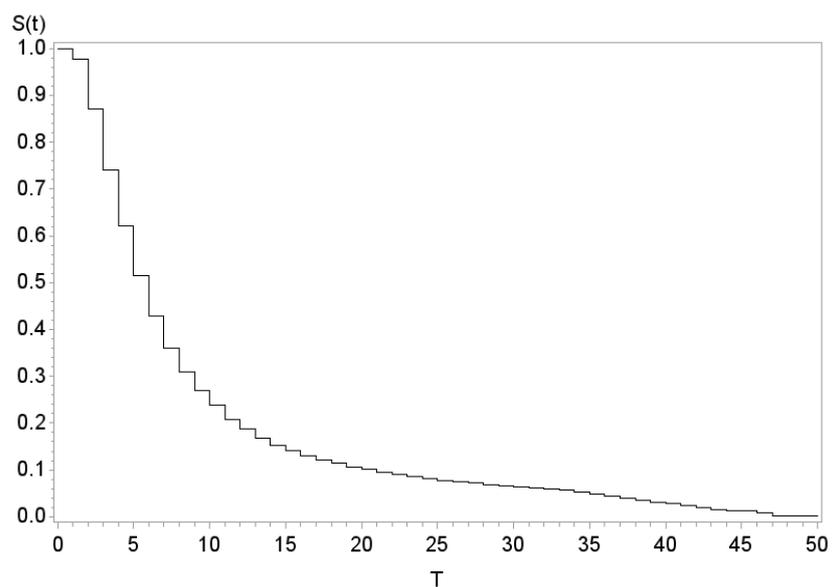
Le graphique 6.8 superpose le graphique de l'estimateur du hazard avec son intervalle de confiance asymptotique de niveau 95%. On voit que cet intervalle s'élargit quand le nombre d'observations diminue, ce qui est flagrant après 30 ans.

Il existe de nombreuses options associées à la procédure *lifetest*. Nous ne pouvons qu'encourager le lecteur à consulter la documentation qui est intégralement disponible en ligne.

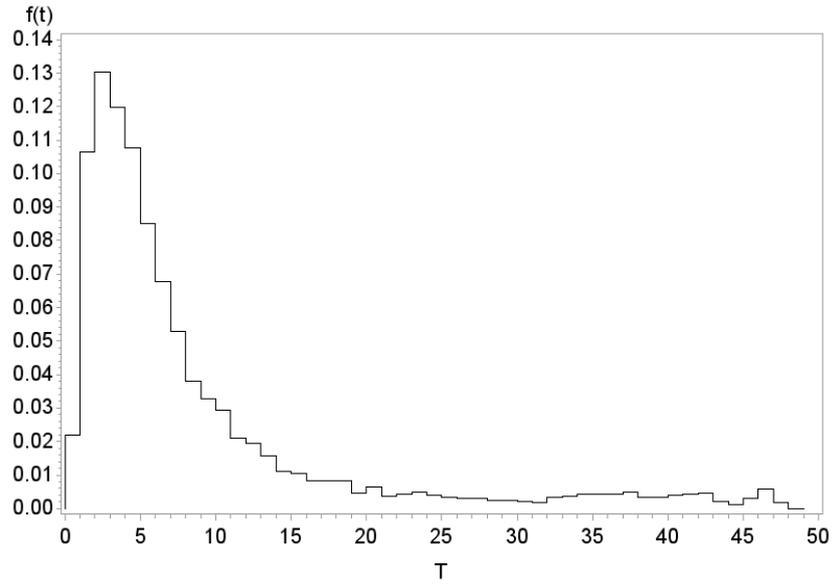
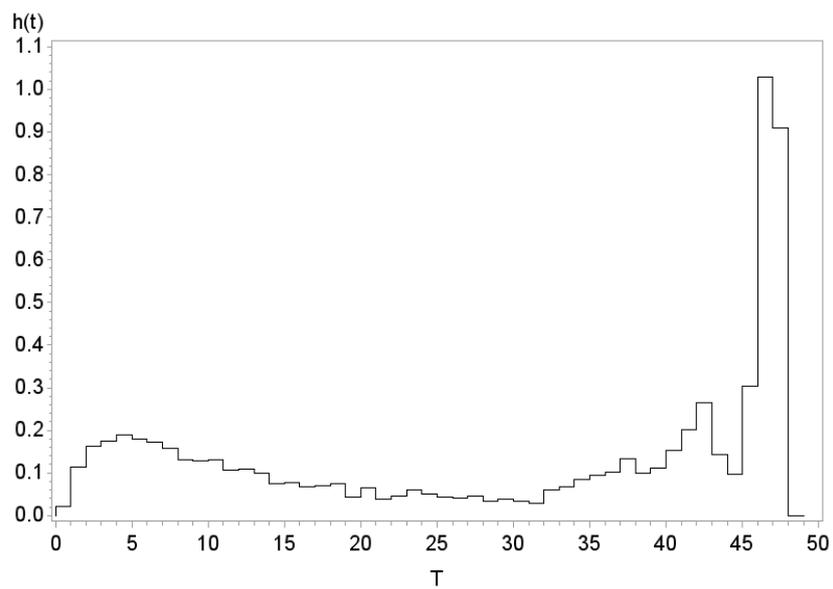
#### Sortie SAS 6.4.

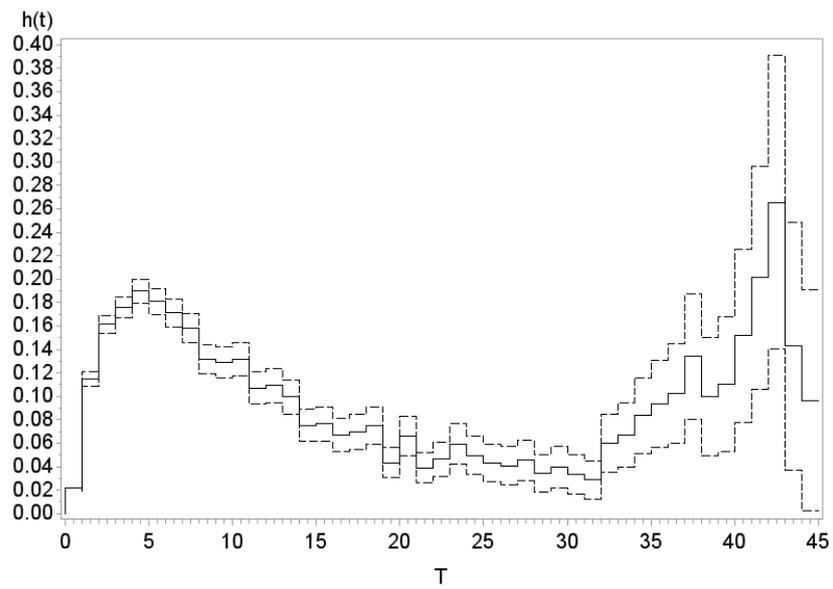
| Obs. | duree_<br>emp | SURVIVAL | SDF_LCL | SDF_UCL | MIDPOINT | PDF     |
|------|---------------|----------|---------|---------|----------|---------|
| 1    | 0             | 1.00000  | .       | .       | 0.5      | 0.00000 |
| 2    | 1             | 1.00000  | .       | .       | 1.5      | 0.02180 |
| 3    | 2             | 0.97820  | 0.97550 | 0.98060 | 2.5      | 0.10644 |
| 4    | 3             | 0.87176  | 0.86579 | 0.87748 | 3.5      | 0.13017 |
| 5    | 4             | 0.74159  | 0.73382 | 0.74918 | 4.5      | 0.11987 |
| 6    | 5             | 0.62172  | 0.61312 | 0.63020 | 5.5      | 0.10780 |
| 7    | 6             | 0.51392  | 0.50504 | 0.52273 | 6.5      | 0.08520 |
| 8    | 7             | 0.42872  | 0.41990 | 0.43751 | 7.5      | 0.06762 |
| 9    | 8             | 0.36110  | 0.35252 | 0.36970 | 8.5      | 0.05285 |
| 10   | 9             | 0.30825  | 0.29997 | 0.31657 | 9.5      | 0.03817 |
| 11   | 10            | 0.27008  | 0.26210 | 0.27812 | 10.5     | 0.03269 |
| 12   | 11            | 0.23739  | 0.22972 | 0.24514 | 11.5     | 0.02936 |
| 13   | 12            | 0.20803  | 0.20069 | 0.21546 | 12.5     | 0.02114 |
| 14   | 13            | 0.18689  | 0.17983 | 0.19406 | 13.5     | 0.01935 |
| 15   | 14            | 0.16754  | 0.16075 | 0.17444 | 14.5     | 0.01587 |
| 16   | 15            | 0.15167  | 0.14513 | 0.15833 | 15.5     | 0.01099 |
| 17   | 16            | 0.14068  | 0.13434 | 0.14717 | 16.5     | 0.01036 |
| 18   | 17            | 0.13032  | 0.12416 | 0.13663 | 17.5     | 0.00844 |
| 19   | 18            | 0.12188  | 0.11588 | 0.12804 | 18.5     | 0.00818 |
| 20   | 19            | 0.11370  | 0.10786 | 0.11970 | 19.5     | 0.00823 |
| 21   | 20            | 0.10547  | 0.09980 | 0.11131 | 20.5     | 0.00450 |
| 22   | 21            | 0.10097  | 0.09540 | 0.10671 | 21.5     | 0.00647 |
| 23   | 22            | 0.09450  | 0.08907 | 0.10011 | 22.5     | 0.00362 |
| 24   | 23            | 0.09088  | 0.08553 | 0.09641 | 23.5     | 0.00413 |
| 25   | 24            | 0.08675  | 0.08149 | 0.09220 | 24.5     | 0.00499 |

| Obs. | PDF_LCL | PDF_UCL | HAZARD  | HAZ_LCL | HAZ_UCL |
|------|---------|---------|---------|---------|---------|
| 1    | .       | .       | 0.00000 | .       | .       |
| 2    | 0.01926 | 0.02434 | 0.02204 | 0.01944 | 0.02464 |
| 3    | 0.10104 | 0.11183 | 0.11507 | 0.10891 | 0.12123 |

GRAPHIQUE 6.5 – ESTIMATEUR ACTUARIEL, FONCTION DE SURVIE  $\hat{S}_T(t)$ 

|    |         |         |         |         |         |
|----|---------|---------|---------|---------|---------|
| 4  | 0.12426 | 0.13609 | 0.16137 | 0.15353 | 0.16921 |
| 5  | 0.11412 | 0.12561 | 0.17584 | 0.16689 | 0.18480 |
| 6  | 0.10227 | 0.11333 | 0.18985 | 0.17957 | 0.20013 |
| 7  | 0.08018 | 0.09023 | 0.18077 | 0.16966 | 0.19189 |
| 8  | 0.06306 | 0.07217 | 0.17122 | 0.15930 | 0.18314 |
| 9  | 0.04876 | 0.05694 | 0.15792 | 0.14538 | 0.17045 |
| 10 | 0.03465 | 0.04169 | 0.13201 | 0.11961 | 0.14442 |
| 11 | 0.02941 | 0.03597 | 0.12884 | 0.11569 | 0.14199 |
| 12 | 0.02623 | 0.03250 | 0.13185 | 0.11757 | 0.14613 |
| 13 | 0.01845 | 0.02382 | 0.10704 | 0.09328 | 0.12079 |
| 14 | 0.01677 | 0.02194 | 0.10920 | 0.09446 | 0.12395 |
| 15 | 0.01351 | 0.01823 | 0.09944 | 0.08451 | 0.11437 |
| 16 | 0.00900 | 0.01297 | 0.07515 | 0.06149 | 0.08882 |
| 17 | 0.00842 | 0.01230 | 0.07646 | 0.06205 | 0.09087 |
| 18 | 0.00668 | 0.01020 | 0.06692 | 0.05287 | 0.08098 |
| 19 | 0.00643 | 0.00993 | 0.06946 | 0.05452 | 0.08439 |
| 20 | 0.00646 | 0.01000 | 0.07513 | 0.05888 | 0.09137 |
| 21 | 0.00318 | 0.00583 | 0.04361 | 0.03073 | 0.05649 |
| 22 | 0.00487 | 0.00807 | 0.06620 | 0.04973 | 0.08267 |
| 23 | 0.00241 | 0.00483 | 0.03906 | 0.02593 | 0.05218 |
| 24 | 0.00282 | 0.00543 | 0.04648 | 0.03171 | 0.06126 |
| 25 | 0.00354 | 0.00644 | 0.05925 | 0.04195 | 0.07655 |

GRAPHIQUE 6.6 – ESTIMATEUR ACTUARIEL, FONCTION DE DENSITÉ  $\hat{f}_T(t)$ GRAPHIQUE 6.7 – ESTIMATEUR ACTUARIEL, FONCTION DE HASARD  $\hat{h}_T(t)$



GRAPHIQUE 6.8 – ESTIMATEUR ACTUARIEL, HASARD AVEC INTERVALLE DE CONFIANCE

## CHAPITRE 7

# Introduction des variables explicatives

**N**OUS VENONS DE TERMINER la partie du cours qui relève des statistiques descriptives. Or ce type d'analyse est souvent insuffisante en économie. On recherche plutôt une liste de variables qui expliquent la durée, qui la causent. Pour cela, il faut introduire un modèle économétrique. Le modèle économétrique le plus simple avec variables explicatives est le modèle exponentiel. Il existe de nombreuses manières d'introduire l'effet des variables explicatives dans un modèle de durée. Deux modélisations très répandues sont les modèles à hasard proportionnels et les modèles à durée accélérée. Il se trouve que les modèles exponentiel et de Weibull appartiennent à ces deux modélisations, ce qui motive leur choix pour ce chapitre.

Pour le modèle exponentiel, nous utiliserons l'algorithme du score que nous n'avons pas employé jusqu'à maintenant. Cette section nous permettra également d'illustrer le fait que l'on doit faire attention aux espérances mathématiques quand on travaille sur des données censurées. La seconde section sera consacrée au modèle de Weibull, et nous utiliserons la procédure *lifereg* de SAS, qui est consacrée aux modèles paramétriques à durée accélérée.

### 7.1 L'effet des variables explicatives

#### 7.1.1 Modèles à hasards proportionnels

Dans un modèle à hasard proportionnels, le ratio des hasards de deux individus est indépendant du temps, on a :

$$h_i(t) = h_0(t) \psi(X_i), \psi(x) > 0 \forall x$$

on appelle  $h_i(t)$  le hasard individuel, qui dépend des variables explicatives, et  $h_0(t)$  le hasard de base, qui ne dépend que de la durée. Cette modélisation implique que pour deux individus ( $i$  et  $j$ ) :

$$\frac{h_i(t)}{h_j(t)} = \frac{h_0(t) \psi(X_i)}{h_0(t) \psi(X_j)} = \frac{\psi(X_i)}{\psi(X_j)} \text{ indépendant de } t,$$

les hasards sont proportionnels à la fonction des variables explicatives qui caractérise chaque individu. Il nous reste à choisir la fonction  $\psi$ . On suppose que le modèle comporte  $P$  variables explicatives indicées par  $p = 1, \dots, P$  et notées :

$$X = (X_1, \dots, X_P),$$

les coefficients de ces variables sont notés  $b_p$  ( $p = 1, \dots, P$ ) et rangés dans un vecteur colonne :

$$b = \begin{pmatrix} b_1 \\ \vdots \\ b_P \end{pmatrix},$$

de sorte que :

$$Xb = (X_1, \dots, X_P) \begin{pmatrix} b_1 \\ \vdots \\ b_P \end{pmatrix} = \sum_{p=1}^P X_p b_p,$$

la modélisation la plus répandue consiste alors à prendre l'exponentielle de cette combinaison linéaire, ce qui garantit la positivité de  $\psi$  pour toutes les valeurs possibles de  $X$  et de  $b$  :

$$\psi(X) = \exp(Xb). \quad (7.1)$$

Avec cette convention, on obtient le modèle à hasard proportionnel suivant :

$$h_i(t) = h_0(t) \exp(X_i b) \Rightarrow \frac{h_i(t)}{h_j(t)} = \exp((X_i - X_j) b).$$

Pour écrire la log-vraisemblance de ce modèle, il suffit d'utiliser l'hypothèse sur la fonction de hasard puisque l'on a, dans le cas général, la propriété suivante :

$$\ell_i = d_i \ln h_i(y_i) - H_i(y_i),$$

où  $H_i(y_i)$  est le hasard cumulé individuel. On le calcule directement :

$$\begin{aligned} H_i(t) &= \int_0^t h_i(x) dx \\ &= \int_0^t h_0(x) \exp(X_i b) dx \\ &= \exp(X_i b) \int_0^t h_0(x) dx \\ &= \exp(X_i b) H_0(t) \end{aligned}$$

où  $H_0(t)$  est le hasard cumulé de base. Ceci permet d'écrire la log-vraisemblance :

$$\ell = \sum_{i=1}^N d_i (\ln h_0(y_i) + X_i b) - \exp(X_i b) H_0(y_i).$$

Pour réaliser une estimation paramétrique, il suffit donc de choisir une distribution de base, c'est à dire de hasard  $h_0(t)$ .

Dans le cas particulier du modèle exponentiel, il suffit de postuler que la distribution de base est exponentielle. En prenant une loi de paramètre  $h_0$ , on a :

$$\begin{aligned} h_0(t) &= h_0, \\ H_0(t) &= h_0 t \end{aligned}$$

on obtient alors la log-vraisemblance suivante :

$$\ell(h_0, b) = \sum_{i=1}^N d_i (\ln h_0 + X_i b) - y_i \exp(\ln h_0 + X_i b)$$

ce qui suggère de poser :

$$\underline{X} = (1, X), \quad \underline{b} = \begin{pmatrix} \ln h_0 \\ b \end{pmatrix},$$

de sorte que :

$$\ell(\underline{b}) = \sum_{i=1}^N d_i \underline{X}_i \underline{b} - y_i \exp(\underline{X}_i \underline{b}).$$

### 7.1.2 Modèles à durée accélérée

Dans un modèle à durée accélérée, on suppose que l'effet des variables explicatives porte sur l'unité de temps. On dispose d'une distribution de base et, pour obtenir la distribution individuelle, tout se passe comme si l'action des variables explicatives compressait ou détendait l'unité de temps. Plus précisément, on postule que :

$$S_i(t) = S_0(\xi(X_i) t)$$

où  $S_i(t)$  est la survie individuelle et  $S_0(t)$  la survie de base, ce qui est équivalent à :

$$\begin{aligned} \Pr[T_i > t] &= \Pr[T_0 > \xi(X_i) t] \\ \Leftrightarrow \Pr[T_i > t] &= \Pr\left[\frac{T_0}{\xi(X_i)} > t\right], \end{aligned}$$

on en déduit que  $T$  et  $T_0/\xi(X_i)$  ont la même distribution :

$$T_i = \frac{T_0}{\xi(X_i)},$$

en conséquence, si  $\xi(X_i) > 1$  tout se passe comme si la durée  $T_i$  passait plus vite que la durée  $T_0$ ; dans ce cas on doit s'attendre à des durées plus courtes, donc à un taux de sortie plus élevé. Inversement, si  $\xi(X_i) < 1$  les durées seront plus longues et donc les taux de sortie seront plus faibles. Ces coefficients  $\xi(X_i)$  peuvent varier avec chaque individu car ils dépendent des variables explicatives. Si on compare deux individus  $i$  et  $j$ , on obtient :

$$\frac{T_i}{T_j} = \frac{\xi(X_j)}{\xi(X_i)},$$

les durées sont proportionnelles aux fonctions des variables individuelles. Si  $\xi(X_i) > \xi(X_j)$ , l'individu  $i$  connaîtra des durées plus courtes (sortira plus vite) que l'individu  $j$ . Pour déterminer la vraisemblance, il suffit d'utiliser la relation :

$$S_i(t) = S_0(\xi(X_i) t)$$

pour calculer le hasard. On a :

$$\begin{aligned}
 h_i(t) &= -\frac{d \ln S_i(t)}{dt} \\
 &= -\frac{1}{S_i(t)} \frac{d S_i(t)}{dt} \\
 &= \xi(X_i) \times \left( -\frac{1}{S_0(\xi(X_i) t)} \frac{d S_0(\xi(X_i) t)}{dt} \right) \\
 &= \xi(X_i) \times \left( -\frac{d \ln S_0(\xi(X_i) t)}{dt} \right) \\
 &= \xi(X_i) h_0(\xi(X_i) t),
 \end{aligned}$$

d'où le hasard cumulé :

$$H_i(t) = H_0(\xi(X_i) t),$$

ce que l'on vérifie par dérivation. En prenant une forme exponentielle pour les variables explicatives :

$$\xi(X_i) = \exp(X_i c)$$

on obtient la log-vraisemblance suivante :

$$\ell = \sum_{i=1}^N d_i (X_i c + \ln h_0(\exp(X_i c) y_i)) - H_0(\exp(X_i c) y_i).$$

Pour un modèle exponentiel, on pose  $h_0(t) = h_0$  donc  $H_0(t) = h_0 t$ , ce qui donne la log-vraisemblance utilisée habituellement :

$$\begin{aligned}
 \ell &= \sum_{i=1}^N d_i (X_i c + \ln h_0) - h_0 \exp(X_i c) y_i \\
 &= \sum_{i=1}^N d_i (\ln h_0 + X_i c) - y_i \exp(\ln h_0 + X_i c).
 \end{aligned}$$

Ceci suggère de poser :

$$\underline{X} = (1, X), \quad \underline{b} = \begin{pmatrix} \ln h_0 \\ c \end{pmatrix},$$

de sorte que :

$$\ell(\underline{b}) = \sum_{i=1}^N d_i \underline{X}_i \underline{b} - y_i \exp(\underline{X}_i \underline{b}),$$

on obtient exactement la même log-vraisemblance que dans le cas à hasard proportionnel. Il s'agit bien entendu du cas particulier de la loi exponentielle, et cette propriété n'est pas valable dans le cas général.

### 7.1.3 Ecriture en logarithmes

Tous les modèles de durée peuvent se mettre sous une forme logarithmique, puisque les durées sont positives. Prenons le cas particulier d'un modèle à durée accélérée. On a :

$$T_i = \frac{T_0}{\xi(X_i)}$$

donc :

$$\ln T_i = \ln T_0 - \ln \xi(X_i),$$

en posant  $\xi(X_i) = \exp(X_i c)$  et en centrant la variable aléatoire  $\ln T_0$ , on obtient :

$$\begin{aligned} \ln T_i &= E(\ln T_0) - X_i c + u_i, \\ u_i &= \ln T_0 - E(\ln T_0), \end{aligned}$$

avec

$$E(u_i | X_i) = 0, \quad u_i \text{ indépendant de } X_i$$

on peut alors poser :

$$a = \begin{pmatrix} E(\ln T_0) \\ -c \end{pmatrix}, \quad \underline{X}_i = (1, X_i)$$

et le modèle s'écrit :

$$\ln T_i = \underline{X}_i a + u_i, \quad E(u_i | X_i) = 0. \quad (7.2)$$

On retombe sur le cas d'un modèle linéaire standard *quand il n'y a pas de censure*. Dans ce dernier cas, on peut donc obtenir un estimateur sans biais et convergent de  $c$  en utilisant la méthode des moindres carrés ordinaires. Mais cette méthode est de peu d'intérêt dans la pratique car la quasi totalité des données de durée est censurée. On notera que la loi de  $u_i$  dépend de la distribution de la durée de base  $T_0$ . Si elle est log-normale, on retrouvera le modèle linéaire normal, mais avec des données censurées. Dans le cas exponentiel, on montre que  $u_i$  suit une loi de Gumbel (voir l'annexe C.1.1). L'annexe montre comment calculer la densité de la loi de  $u_i$ .

## 7.2 Aspects appliqués

### 7.2.1 Interprétation des coefficients

La forme exponentielle (7.1) permet d'interpréter les paramètres facilement. Si la variable explicative  $p$  est continue, on obtient :

$$\frac{\partial \ln h_i}{\partial X_{pi}} = b_p,$$

donc, si la variable  $X_p$  est exprimée en logarithmes,  $X_{pi} = \ln x_{pi}$ , le coefficient  $b_p = \partial \ln h_i / \partial \ln x_{pi}$  représente l'élasticité du hasard  $h$  à la variable  $x_p$ . Une hausse de 1% de la variable  $x_p$  entraîne une hausse de  $b_p$  du hasard. De la même manière, on voit que :

$$\frac{\partial E(\ln T_i | X_i)}{\partial X_{pi}} = c_p = -b_p,$$

lorsque la variable  $x_p$  augmente de 1%, la durée moyenne *diminue* de  $b_p$ %.

Si la variable explicative est dichotomique,  $X_{pi} \in \{0, 1\}$ , on peut interpréter  $b_p$  facilement à la condition de faire l'écart entre le hasard du groupe  $X_{pi} = 1$  et le hasard du groupe  $X_{pi} = 0$  toutes choses égales par ailleurs. La conditions "toutes choses égales par ailleurs" s'obtient en égalisant les variables  $X_{ki}$  entre les deux groupes pour  $k \neq p$ . On obtient :

$$\begin{aligned} \ln h_i |_{X_{pi}=1} - \ln h_i |_{X_{pi}=0} &= 1 \times b_p + \sum_{k \neq p}^p X_{ki} \times b_k - \left( 0 \times b_p + \sum_{k \neq p}^p X_{ki} \times b_k \right) \\ &= b_p \end{aligned}$$

or, d'autre part, pour  $b_p \approx 0$  :

$$b_p = \ln h_i |_{X_{pi}=1} - \ln h_i |_{X_{pi}=0} \approx \frac{h_i |_{X_{pi}=1} - h_i |_{X_{pi}=0}}{h_i |_{X_{pi}=0}},$$

on obtient l'écart relatif entre les hasard des deux groupes définis par  $X_{pi}$ . En multipliant le coefficient  $b_p$  par 100, on obtient l'écart en pourcentages. Par exemple, si  $X_{pi}$  est le sexe,  $b_p$  représentera l'écart entre les taux conditionnels de sortie des femmes et des hommes. La modalité de référence est celle qui figure au dénominateur, ici  $X_{pi} = 0$ . Il faut garder cette valeur en tête quand on commente les résultats. Pour l'espérance de la durée, on obtient le même résultat mais avec un signe opposé :

$$c_p = -b_p \approx \frac{E(\ln T_i | X_i) |_{X_{pi}=1} - E(\ln T_i | X_i) |_{X_{pi}=0}}{E(\ln T_i | X_i) |_{X_{pi}=0}}$$

### 7.2.2 Terme constant et centrage

Pour le modèle exponentiel, il faut inclure un terme constant dans la liste des variables explicatives. En effet, si aucune variable n'était significative dans le modèle, on aurait  $b = 0 \Rightarrow h_T(t) = 1$ , ce qui est arbitraire. Dans le cas où il n'y a pas de variable significative ( $b = 0$ ), nous devrions retrouver le cas des statistiques descriptives où la valeur du hasard est libre :  $h_T(t) = h_0$ . Sans perte de

généralité, on impose que la première variable explicative est le terme constant de sorte que :

$$\underline{X}_i \underline{b} = b_0 + X_i b, \text{ avec } b_0 = \ln h_0$$

on aura donc :

$$\begin{aligned} h_i(t) &= \exp(b_0 + X_i b) \\ &= \exp(\ln h_0 + X_i b) \\ &= h_0 \exp(X_i b), \end{aligned}$$

ainsi :

$$b = 0 \Rightarrow h_i(t) = h_0,$$

ce qui correspond au modèle descriptif (i.e. sans variable explicative). On en déduit que le test global de significativité ne devra pas tenir compte du terme constant, comme pour le test de Fisher dans le modèle linéaire standard. Le test portera sur  $b$ , pas sur  $\underline{b}$ .

D'autres conventions d'écriture du modèle sont possibles. Nous présentons maintenant une des plus pratiques : le modèle centré, qui s'écrit obligatoirement avec un terme constant. Ce modèle consiste à remplacer chaque variable explicative autre que le terme constant par sa valeur centrée :

On remplace  $X_{pi}$  par  $X_{pi} - \bar{X}_p$ ,

$$\text{avec } \bar{X}_p = \frac{1}{N} \sum_{i=1}^N X_{pi},$$

ceci implique que les coefficients du modèle seront différents avec cette convention. On les note  $\underline{\beta}$  au lieu de  $\underline{b}$ . On pose également la convention :

$$\underline{\beta} = \begin{pmatrix} \beta_0 \\ \beta \end{pmatrix},$$

où  $\beta_0$  est le terme constant et  $\beta$  le vecteur des coefficients des variables explicatives. Le modèle s'écrit :

$$\underline{X}_i \underline{\beta} = \beta_0 + (X_i - \bar{X}) \beta,$$

avec :

$$X_i = (X_{1i}, \dots, X_{pi}) \text{ et } \bar{X} = (\bar{X}_1, \dots, \bar{X}_p).$$

donc

$$\begin{aligned} h_i(t) &= \exp(\beta_0 + (X_i - \bar{X}) \beta) \\ &= \exp(\beta_0) \exp((X_i - \bar{X}) \beta). \end{aligned}$$

Le hasard au point moyen de l'échantillon, noté  $\bar{h}$ , est obtenu en posant :

$$X_i = \bar{X},$$

ce qui donne :

$$\bar{h} = \exp(\beta_0),$$

donc on peut écrire :

$$h_i(t) = \bar{h} \exp\left((X_i - \bar{X})\beta\right)$$

En général  $\bar{h} \neq h_0$ , le hasard au point moyen de l'échantillon est différent du hasard en  $b = 0$ , mais il existe une relation entre les deux conventions. On a :

$$\underline{X}_i \underline{b} = b_0 + X_i b$$

et

$$\underline{X}_i \underline{\beta} = \beta_0 + (X_i - \bar{X})\beta = \beta_0 - \bar{X}\beta + X_i\beta,$$

comme on doit avoir  $\underline{X}_i \underline{b} = \underline{X}_i \underline{\beta} \forall i$  et que les colonnes de  $\underline{X}$  sont linéairement indépendantes, on obtient (par identification) :

$$b_0 = \beta_0 - \bar{X}\beta \text{ et } b = \beta,$$

on en déduit que :

$$\begin{aligned} \exp(b_0) &= \exp(\beta_0 - \bar{X}\beta) \\ \Leftrightarrow h_0 &= \bar{h} \exp(-\bar{X}\beta) \\ \Leftrightarrow \bar{h} &= h_0 \exp(\bar{X}\beta) \end{aligned}$$

ainsi les deux conventions ne sont équivalentes que quand aucune variable explicative n'est significative  $\beta = 0$ . Dans tous les autres cas ( $\exists p : \beta_p \neq 0$ ), on aura  $\bar{h} \neq h_0$ .

Notons que quelque soit la convention retenue, on peut toujours faire le test de significativité globale des coefficients car l'hypothèse nulle est identique dans les deux modélisations vu que  $\beta = b$ .

Pour notre applications, nous prendrons la version centrée, car la moyenne des variables de l'échantillon est un point de référence plus naturel que la référence abstraite à ce que serait un modèle sans effet des variables explicatives.

### 7.3 Estimation du modèle exponentiel

On prend des variables centrées, que l'on note :

$$\tilde{X}_i = (1, X_i - \bar{X}), \quad i = 1, \dots, N$$

La log-vraisemblance est donnée par :

$$\ell(\underline{\beta}) = \sum_{i=1}^N d_i \tilde{X}_i \underline{\beta} - y_i \exp(\tilde{X}_i \underline{\beta}).$$

On en déduit le vecteur du score :

$$\begin{aligned} s(\underline{\beta}) &= \frac{\partial \ell(\underline{\beta})}{\partial \underline{\beta}} \\ &= \sum_{i=1}^N \tilde{X}_i' (d_i - y_i \exp(\tilde{X}_i \underline{\beta})), \end{aligned}$$

et le hessien :

$$\begin{aligned} H(\underline{\beta}) &= \frac{\partial^2 \ell(\underline{\beta})}{\partial \underline{\beta} \partial \underline{\beta}'} \\ &= - \sum_{i=1}^N \tilde{X}_i' \tilde{X}_i y_i \exp(\tilde{X}_i \underline{\beta}) \end{aligned}$$

Arrivé à ce point, s'il on souhaite calculer la matrice d'information, il faut faire attention au fait que les données sont censurées et que l'on ne peut pas prendre l'espérance mathématique de  $y_i$  directement. En effet,

$$E(y_i) \neq \exp(-\tilde{X}_i \underline{\beta})$$

car  $y_i$  est une variable censurée de durée, pas une durée complète. Nous montrons en annexe que :<sup>1</sup>

$$\begin{aligned} J(\underline{\beta}) &= E_y \left[ - \frac{\partial^2 \ell(\underline{\beta})}{\partial \underline{\beta} \partial \underline{\beta}'} \Big| \tilde{X} \right] \\ &= \sum_{i=1}^N \tilde{X}_i' \tilde{X}_i (1 - S_T(y_i)) \\ &= \sum_{i=1}^N \tilde{X}_i' \tilde{X}_i (1 - \exp(-\exp(\tilde{X}_i \underline{\beta}) y_i)), \end{aligned}$$

alors que nous aurions trouvé  $\sum_{i=1}^N \tilde{X}_i' \tilde{X}_i$  avec des données non censurées.

1. Ce point motive l'utilisation de l'algorithme du score, qui permet de bien voir le problème posé par la censure des données

### 7.3.1 Programmation XV : Algorithme du score

Techniquement, la programmation ressemble à celle de Berndt-Hall-Hall-Hausman. Nous allons donc reprendre ce programme. La seule différence que nous allons introduire dans ce programme, est que nous remplacerons le produit croisé des dérivées premières par l'espérance mathématique des dérivées secondes. La matrice que nous allons employer pour remplacer le hessien est donc égale à :

$$J = - \sum_{i=1}^N \tilde{X}'_i \tilde{X}_i \left( 1 - \exp \left( - \exp \left( \tilde{X}_i \underline{\beta} \right) y_i \right) \right),$$

cette matrice est toujours définie négative, de sorte qu'elle est toujours admissible pour cet algorithme. Nous ferons la programmation directement, sans utiliser les routines disponibles. Le lecteur est invité à utiliser les routines que nous avons déjà vues, elles donnent les mêmes résultats. L'algorithme du score est le suivant pour la  $k$ -ième itération :

$$\underline{\beta}^{(k)} = \underline{\beta}^{(k-1)} - \lambda_k J^{-1} s \left( \underline{\beta}^{(k-1)} \right), \quad k \geq 1, \lambda > 0$$

un premier module calculera le score  $s$  et un second module l'information de Fisher  $J$ . Comme valeur initiale nous prenons l'estimateur des MCO qui serait convergent si les données n'étaient pas censurées. Il n'est pas convergent dans le cas général mais suffira à mettre l'algorithme sur les bons rails. En nous basant sur l'annexe C.1.1 :

$$\ln T_i = - (\ln h_i + \gamma_E), \quad \gamma_E = -0.57721$$

avec :

$$h_i = \bar{h} \exp \left( (X_i - \bar{X}) \beta \right),$$

soit :

$$\ln T_i = - \ln \bar{h} - (X_i - \bar{X}) \beta - \gamma_E, \quad \beta_0 = \ln \bar{h}$$

donc en posant

$$\ln T_i = d_0 + (X_i - \bar{X}) d = \tilde{X}_i \underline{d} \text{ avec } \underline{d} = \begin{pmatrix} d_0 \\ d \end{pmatrix}$$

on peut prendre :

$$\hat{\underline{d}} = \left( \sum_{i=1}^N \tilde{X}'_i \tilde{X}_i \right)^{-1} \sum_{i=1}^N \tilde{X}'_i \ln y_i$$

$$\underline{\beta}^{(0)} = \begin{pmatrix} -\hat{d}_0 + 0.57721 \\ -\hat{d} \end{pmatrix}$$

En sortie, on pourra présenter le hasard au point moyen de l'échantillon :

$$\widehat{h} = \exp(\widehat{\beta}_0),$$

et l'écart type asymptotique de cet estimateur pourra être estimé par :

$$\sqrt{\text{Vas}(\widehat{h})} = \exp(\widehat{\beta}_0) \sqrt{\text{Vas}(\widehat{\beta}_0)}.$$

et effectuer le test de Wald de significativité des coefficients ( $H_0 : b = 0$ ), dont la statistique est donnée par :

$$W = \widehat{b}' \widehat{\text{Vas}}(\widehat{b})^{-1} \widehat{b},$$

et suit asymptotiquement une distribution du  $\chi_p^2$  sous l'hypothèse nulle.

Nous pourrions également calculer les différentes fonctions associées à la distribution de base, ou examiner l'hétérogénéité des taux de sortie puisqu'ils sont constants dans le temps. Nous préférons la seconde approche et estimons donc :

$$\widehat{h}_i = \exp(\widetilde{X}_i \widehat{\beta}),$$

il sera alors possible de représenter la distribution des hasards individuels estimés avec un estimateur de densité à noyau.

### 7.3.2 Application XV

L'application porte sur la durée du premier emploi. Nous expliquons cette durée par l'année de naissance, le niveau de formation et le genre. L'estimation est réalisée par un programme similaires aux précédents à l'exception de trois opérations : le centrage des variables qui sera effectué par la procédure *standard*, la gestion du vecteur des variables explicatives, le test de significativité global et les statistiques de sortie dans la dimension individuelle.

```
1. data duree; set dur.base_duree;
 c_nais1=(anaisenq<=1948);
 c_nais2=(1948<anaisenq<=1958);
 c_nais3=(1958<anaisenq<=1968);
 c_nais4=(1968<anaisenq<=1978);
 c_nais5=(1978<anaisenq<=1988);
 run;
```

On ajoute cinq indicatrices de dates de naissance à la base de données. On n'utilisera que les quatre dernières, de sorte que les personnes nées avant 1948 serviront de référence. La variable *anaisenq* contient l'année de naissance.

2. 

```
%let tab=duree;
%let e=1e-6;
%let maxiter=300;
%let maxc=10;
```

Les paramètres habituels du programme.
3. 

```
%let y=duree_emp;
%let d=presence;
%let x=/*dip_pri*/ dip_sec dip_sup /*gen_h*/ gen_f
/* c_nais1 */c_nais2 c_nais3 c_nais4 c_nais5;
```

Variables intervenant dans le modèle. On laisse la modalité de référence entre commentaires (*/\* et \*/* en SAS). Les premières variables explicatives portent sur le niveau de formation (primaire, secondaire ou supérieur représenté par les indicatrices respectives *dip\_pri*, *dip\_sec* et *dip\_sup*), le genre (*gen\_h* pour les hommes, *gen\_f* pour les femmes) et l'année de naissance dont les indicatrices sont définies plus haut. Mettre les modalités de référence entre commentaires permet de s'en souvenir même longtemps après avoir utilisé le programme.
4. 

```
proc standard data=&tab out=_tab mean=0;
var &x; run;
```

La procédure *standard* avec l'option *mean=0* centre les variables indiquées par l'instruction *var*. Ici *&x* désigne toutes les variables explicatives du modèle. Le tableau en entrée est indiqué par l'option *data=* et le tableau en sortie par l'option *out=*. Ici, on évite d'écraser les données sources, rangées dans *&tab*, et on crée un nouveau tableau en sortie, de nom *\_tab*, qui contient les variables explicatives centrées.
5. 

```
proc iml;
start lecture;
nom_y={&y};
nom_d={&d};
nom_x={&x};
use _tab; read all var(nom_y) into yi;
read all var(nom_d) into di;
read all var(nom_x) into xi;
nd=sum(di); n=nrow(yi); nc=n-nd;
xi=j(n,1,1)||xi; nx="Intercept"||nom_x;
print "Time variable =" nom_y,
"Censoring variable (=0 if censored) =" nom_d,
"Number of observations =" n,
"Complete =" nd, "Censored =" nc;
```

```
finish lecture;
```

Module de lecture. Par rapport aux cas précédents, on lit la liste des variables explicatives que l'on range dans le vecteur de caractères IML `nom_x`. On lit ensuite ces variables que l'on range dans la matrice IML `xi`. On ajoute ensuite un terme constant à cette matrice par l'instruction :

```
xi=j(n,1,1) || xi.
```

Une fois cette opération effectuée, on n'oublie pas d'ajouter un nom pour le terme constant qui a été ajouté aux variables explicatives. Il s'agit de *intercept*, qui signifie constante ou terme constant, en anglais.

```
6. start lv(theta) global(di,yi,xi);
 mi=xi*theta;
 lv=sum(di#mi-exp(mi)#yi);
 return (lv);
 finish;
```

Calcul de la log-vraisemblance du modèle exponentiel. Ici, il faut faire attention au point suivant : dans tout le programme, *le paramètre est écrit en colonne et non en ligne* car nous n'utilisons pas de routine IML qui impose l'écriture en ligne. Le cas usuel en économétrie est d'écrire le paramètre en colonne, ce que nous faisons ici. On remarque que les variables explicatives `xi` sont passées à la fonction par l'option *global*.

```
7. start sc(theta) global(di,yi,xi);
 mi=xi*theta;
 der=t(xi)*(di-exp(mi)#yi);
 return (der);
 finish;
```

Calcul du vecteur (colonne) du score. Le produit matriciel de la ligne `der` permet d'éviter de faire une somme. En écrivant la matrice des variables explicatives en empilant les individus :

$$X = \begin{pmatrix} X_1 \\ \vdots \\ X_N \end{pmatrix}$$

on obtient le résultat suivant :

$$X' \begin{pmatrix} d_1 - \exp(m_1)y_1 \\ \vdots \\ d_N - \exp(m_N)y_N \end{pmatrix} = \sum_{i=1}^N X'_i (d_i - \exp(m_i)y_i)$$

```

8. start fisher(theta) global(di, yi, xi);
 mi=xi*theta;
 he=t(xi)*(xi#(1-exp(-exp(mi)#yi)));
 return (he);
 finish;

```

Calcul de l'information de Fisher. On multiplie la matrice  $(p, N) X'$  par la matrice d'élément générique :

$$X_i(1 - \exp(-\exp(m_i) y_i))$$

ce qui donne :

$$\sum_{i=1}^N X_i' X_i (1 - \exp(-\exp(m_i) y_i))$$

```

9. start h(theta) global(xi, yi);
 mi=xi*theta;
 h=-t(xi)*(xi#yi#exp(mi));
 return (h);
 finish;

```

Calcul du hessien. On multiplie la matrice  $(p, N) -X'$  par la matrice d'élément générique :

$$X_i'(y_i \exp(m_i))$$

ce qui donne :

$$\sum_{i=1}^N -X_i' X_i (y_i \exp(m_i))$$

```

10. start estim;
 print "Exponential distribution - scoring
 algorithm", , "S(t)=exp(-h.t)", , ;
 theta0=ginv(xi)*(-log(yi));

```

L'inverse généralisée est égale dans ce cas à  $X^+ = (X'X)^{-1} X'$  de sorte que la ligne précédente cacule l'estimateur des moindres carrés ordinaires de  $-\ln y_i$  sur  $X_i$ . Il s'agit de la valeur initiale du paramètre  $\underline{\beta}$  (non convergente).

```

11. lv0=lv(theta0);
 theta=theta0;
 maxiter=&maxiter;
 maxc=&maxc;
 labopt="Iteration" || "Log-likelihood" || "Elasticity"
 || "Step";

```

```

do iter=1 to maxiter;
lambda=1;
sc=sc(theta); fisher=fisher(theta);
step=-inv(fisher)*sc;
theta=theta0-lambda#step;
lv1=lv(theta);
if lv1<lv0 then do;
do c=1 to maxc;
lambda=lambda/2;
theta=theta0-lambda#step;
lv1=lv(theta);
if lv1>=lv0 then c=maxc;
end;
end;
elas=max(abs(theta#sc/lv1));
riter=(iter||lv1||elas||lambda);
print riter[colname=labopt];
theta0=theta; lv0=lv1;
if elas<&e then iter=maxiter;
end;

```

Boucle d'optimisation. On remplace la matrice hessienne par l'opposée de l'information de Fisher.

```

12. gr1=sc(theta);
he=h(theta);
elas1=abs(theta#gr1/lv1);
vp1=eigval(he);
lab1="Gradient"||"Elasticities"||
"Hessian_Eigenvalues";
out1=gr1||elas1||vp1;
print out1[colname=lab1];
vtheta=inv(-he);
sb1=sqrt(vecdiag(vtheta));
tb1=abs(theta)/sb1;
pb1=1-probchi(tb1##2,1);
lab2="Coeff"||"Std_Dev"||"Student"||"P_value";
out2=theta||sb1||tb1||pb1;
print "Estimation of the exponential model",
out2[rowname=nx colname=nx];

```

Impression des résultats habituels.

13. `hm=exp(theta[1]); shm=hm#sb1[1];`  
`hm_L=max(0, hm-1.96#shm); hm_U=hm+1.96#shm;`  
`print "Hasard at the average point of the`  
`sample :" hm,`  
`"95% confidence interval :" hm_L "," hm_U;`  
 Impression des résultats sur le hasard au point moyen.
14. `k=ncol(xi); rg=2:k;`  
`b=theta[rg]; vb=vtheta[rg,rg];`  
 On extrait les composantes 2 à  $k = p + 1$  du vecteur des paramètres, ce qui correspond à  $\hat{\beta}$ . On extrait également l'estimation de la matrice de covariance de l'estimateur.
15. `Wald=t(b)*inv(vb)*b;`  
`pWald=1-probchi(Wald,p);`  
`print "Wald test of global significance", ,`  
`"Statistic =" Wald,`  
`"p value =" pWald;`  
`finish estim;`  
 Calculs associés au test de Wald et affichage des résultats.
16. `start prev;`  
`hi=exp(xi*theta);`  
`create hi from hi[colname="hasard"];`  
`append from hi;`  
`finish prev;`  
 Création d'un tableau SAS hi qui contient les prévisions des hasards individuels.
17. `run lecture;`  
`run estim;`  
`run prev;`  
`quit;`  
`run;`  
 Lancement des modules puis sortie d'IML.
18. `proc kde data=hi; univar hasard/out=densite; run;`  
 Calcul de la densité des prévisions par la méthode du noyau. Le résultat se trouve dans le tableau SAS *densite*. La densité calculée est stockée sous le nom "density", le point auquel elle est calculée est stockée dans la variable "value" et le nom de la série dont on calcule la densité est stockée dans la variable "var".
19. `symbol1 c=black i=join line=1;`  
`proc gplot data=densite;`

```
plot density*value=var/grid;
run;
```

Graphique de la densité des prévisions individuelles des taux de hasard.

Le programme génère la sortie suivante :

**Sortie SAS 7.1.**

```

 nom_y
Time variable = DUREE_EMP
 nom_d
Censoring variable (=0 if censored) = PRESENCE
 n
Number of observations = 12695
 nd
Complete = 11277
 nc
Censored = 1418

Exponential distribution - scoring algorithm

S(t)=exp(-h.t)

 riter
Iteration Log-likelihood Elasticity Step
ROW1 1 -36490.12 0.4069537 1

 riter
Iteration Log-likelihood Elasticity Step
ROW1 2 -35945.47 0.1877526 0.5

 riter
Iteration Log-likelihood Elasticity Step
ROW1 3 -35944.27 0.009012 1

 riter
```

|      | Iteration | Log-likelihood | Elasticity | Step |
|------|-----------|----------------|------------|------|
| ROW1 | 4         | -35943.83      | 0.0063202  | 1    |
|      | riter     |                |            |      |
|      | Iteration | Log-likelihood | Elasticity | Step |
| ROW1 | 5         | -35943.63      | 0.0044362  | 1    |
|      | riter     |                |            |      |
|      | Iteration | Log-likelihood | Elasticity | Step |
| ROW1 | 6         | -35943.52      | 0.0033599  | 1    |
|      | riter     |                |            |      |
|      | Iteration | Log-likelihood | Elasticity | Step |
| ROW1 | 7         | -35943.45      | 0.0024985  | 1    |
|      | riter     |                |            |      |
|      | Iteration | Log-likelihood | Elasticity | Step |
| ROW1 | 8         | -35943.41      | 0.0019229  | 1    |
|      | riter     |                |            |      |
|      | Iteration | Log-likelihood | Elasticity | Step |
| ROW1 | 9         | -35943.39      | 0.0014474  | 1    |
|      | ...       |                |            |      |
|      | riter     |                |            |      |
|      | Iteration | Log-likelihood | Elasticity | Step |
| ROW1 | 163       | -35943.31      | 1.0634E-6  | 1    |
|      | riter     |                |            |      |
|      | Iteration | Log-likelihood | Elasticity | Step |
| ROW1 | 164       | -35943.31      | 1.0203E-6  | 1    |

```

 riter
Iteration Log-likelihood Elasticity Step
ROW1 165 -35943.31 9.7905E-7 1

```

```

 out1
Gradient Elasticities Hessian_Eigenvalues

 0.01544 9.3941E-7 -351.793
-0.001281 1.2451E-9 -703.363
0.0031368 5.2733E-9 -1176.19
-0.000453 8.306E-10 -2117.008
-0.001962 6.2448E-9 -2741.386
-0.004543 3.0503E-8 -2793.242
-0.003482 4.5026E-8 -4468.485
-0.001428 3.4083E-9 -11331.28

```

Estimation of the exponential model

```

 out2
Intercept DIP_SEC DIP_SUP GEN_F
Intercept -2.186886 0.0097259 224.85184 0
DIP_SEC 0.0349296 0.0259159 1.3478017 0.1777222
DIP_SUP -0.060425 0.0265163 2.2787998 0.022679
GEN_F 0.0659647 0.0189677 3.4777319 0.0005057
C_NAIS2 0.1144 0.0258026 4.4336594 9.2647E-6
C_NAIS3 0.2413312 0.0270989 8.905557 0
C_NAIS4 0.4647306 0.0297838 15.603493 0
C_NAIS5 0.0858053 0.0500539 1.7142574 0.0864815

```

hm

```

Hasard at the average point of the sample : 0.1122658
 hm_L hm_U

```

95% confidence interval : 0.1101257 , 0.1144059

Wald test of global significance

Wald

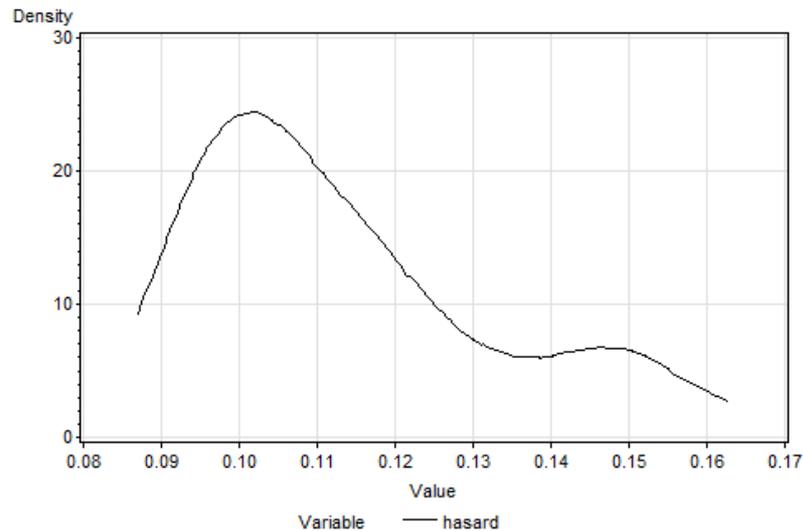
```

Statistic = 307.08773
pWald

```

p value = 0

La convergence avec l'algorithme du score est assez lente (165 itérations), de sorte qu'il n'y a pas vraiment d'avantage à utiliser cette méthode. Globalement les variables sont significatives et, individuellement, seul le coefficient de l'indicatrice de formation de niveau secondaire n'est pas significative au seuil de 5%. Ceci signifie que son coefficient est égal à celui de la modalité de référence : l'éducation primaire. Les taux de sortie individuels sont moins élevés avec les diplômés du supérieur, plus élevés avec les femmes et varient de manière non monotone avec la date de naissance. Le taux de sortie est d'abord croissant avec l'année de naissance jusqu'en 1978 puis décroissant après (courbe en cloche, ou "bell shaped" en anglais). L'effet de l'ensemble de ces variables peut se voir en étudiant les prévisions des taux de sortie individuels. Les prévisions des taux de sortie individuels sont résumés par le graphique 7.1. La densité des prévisions est bimodale. Un premier mode est situé autour de 0.1, ce qui correspond à une durée moyenne de 10 ans, et un second mode est situé autour de 0.15, ce qui correspond à une durée moyenne de 6 ans et 8 mois. Cette différence traduit l'effet des variables explicatives et montre que l'hypothèse de loi Gamma que l'on fait parfois pour les calculs théoriques peut être inadaptée à certains cas réels.



GRAPHIQUE 7.1 – MODÈLE EXPONENTIEL CONDITIONNEL, DISTRIBUTION DES HASARDS INDIVIDUELS

## 7.4 Estimation du modèle de Weibull

Le modèle de Weibull permet de généraliser le modèle exponentiel en autorisant le hasard de base à varier avec la durée. Nous voyons également ce modèle ici parce qu'il s'agit du second type de modèle pour lequel les modélisations à hasard proportionnels et à durée accélérée sont équivalentes. Il existe juste une petite différence de paramétrage entre les deux modélisations. Nous réaliserons l'estimation par la procédure *lifereg*, dont le modèle de Weibull à durée accélérée est la modélisation par défaut. Les hasard et hasard intégré de base sont donnés par :

$$\begin{aligned} h_0(t) &= h_0 \alpha t^{\alpha-1}, \\ H_0(t) &= h_0 t^\alpha. \end{aligned}$$

et nous poserons :

$$\psi_i = \exp(X_i b),$$

pour l'effet des variables explicatives dans la spécification à hasards proportionnels, et

$$\xi_i = \exp(X_i c),$$

pour l'effet des variables explicatives dans la spécification à durée accélérée.

#### 7.4.1 Paramétrage à hasards proportionnels

Pour la modélisation à hasards proportionnels, on utilise :

$$\begin{aligned} h_i(t) &= h_0(t) \exp(X_i b) = h_0 \alpha t^{\alpha-1} \exp(X_i b), \\ H_i(t) &= H_0(t) \exp(X_i b) = h_0 t^\alpha \exp(X_i b). \end{aligned}$$

ainsi que :

$$\begin{aligned} \ell_i &= d_i \ln h_i(y_i) - H_i(y_i) \\ &= d_i (\ln h_0 + \ln \alpha + (\alpha - 1) \ln y_i + X_i b) - h_0 y_i^\alpha \exp(X_i b) \\ &= d_i (\ln \alpha + (\alpha - 1) \ln y_i + \ln h_0 + X_i b) - \exp(\alpha \ln y_i + \ln h_0 + X_i b) \end{aligned}$$

arrivés à ce stade, on pose :

$$\underline{X}_i = (1, X_i), \underline{b} = \begin{pmatrix} \ln h_0 \\ b \end{pmatrix}, \underline{\beta} = \begin{pmatrix} \alpha \\ \underline{b} \end{pmatrix},$$

ce qui donne, la log-vraisemblance :

$$\ell(\underline{\beta}) = \sum_{i=1}^N d_i (\underline{X}_i \underline{b} + (\alpha - 1) \ln y_i + \ln \alpha) - \exp(\underline{X}_i \underline{b} + \alpha \ln y_i).$$

Dans la pratique, on raisonnera sur des variables explicatives centrées afin d'obtenir facilement le hasard de base au point moyen de l'échantillon :

$$\tilde{X}_i = (1, X_i - \bar{X}), \tilde{b} = \begin{pmatrix} \ln \bar{h} \\ b \end{pmatrix}, \tilde{\beta} = \begin{pmatrix} \alpha \\ \tilde{b} \end{pmatrix},$$

d'où le paramétrage à hasards proportionnels :

$$\ell(\tilde{\beta}) = \sum_{i=1}^N d_i (\tilde{X}_i \tilde{b} + (\alpha - 1) \ln y_i + \ln \alpha) - \exp(\tilde{X}_i \tilde{b} + \alpha \ln y_i).$$

#### 7.4.2 Paramétrage à durée accélérée

Cette fois ci, on a :

$$\begin{aligned} h_i(t) &= \exp(X_i c) h_0 (\exp(X_i c) t) \\ &= \exp(X_i c) h_0 \alpha (\exp(X_i c) t)^{\alpha-1} \\ &= h_0 \alpha (\exp(X_i c))^\alpha t^{\alpha-1}, \\ H_i(t) &= H_0 (\exp(X_i c) t) \\ &= h_0 (\exp(X_i c) t)^\alpha. \end{aligned}$$

ainsi que :

$$\begin{aligned}\ell_i &= d_i \ln h_i(y_i) - H_i(y_i) \\ &= d_i (\ln h_0 + \ln \alpha + (\alpha - 1) \ln y_i + X_i c \alpha) - h_0 y_i^\alpha \exp(X_i c)^\alpha \\ &= d_i (\ln \alpha + (\alpha - 1) \ln y_i + \ln h_0 + X_i c) - \exp(\alpha \ln y_i + \ln h_0 + X_i c \alpha)\end{aligned}$$

on pose donc :

$$\underline{X}_i = (1, X_i), \underline{c} = \begin{pmatrix} \ln h_0 \\ c \alpha \end{pmatrix}, \underline{\gamma} = \begin{pmatrix} \alpha \\ \underline{c} \end{pmatrix},$$

ce qui donne, la log-vraisemblance :

$$\ell(\underline{\gamma}) = \sum_{i=1}^N d_i (\underline{X}_i \underline{c} + (\alpha - 1) \ln y_i + \ln \alpha) - \exp(\underline{X}_i \underline{c} + \alpha \ln y_i).$$

Dans la pratique, on raisonnera sur des variables explicatives centrées afin d'obtenir facilement le hasard de base au point moyen de l'échantillon :

$$\tilde{X}_i = (1, X_i - \bar{X}), \tilde{c} = \begin{pmatrix} \ln \bar{h} \\ c \alpha \end{pmatrix}, \tilde{\gamma} = \begin{pmatrix} \alpha \\ \tilde{c} \end{pmatrix},$$

d'où le paramétrage à durée accélérée :

$$\ell(\tilde{\gamma}) = \sum_{i=1}^N d_i (\tilde{X}_i \tilde{c} + (\alpha - 1) \ln y_i + \ln \alpha) - \exp(\tilde{X}_i \tilde{c} + \alpha \ln y_i).$$

On retrouve la même log-vraisemblance que dans la modélisation à hasards proportionnels, avec un simple changement de paramètres : le coefficient des variables explicatives est égal à  $b$  dans le modèle à hasards proportionnels et à  $c\alpha$  dans le modèle à durée accélérée.

La procédure *lifereg* utilise, en fait, une troisième convention. Celle qui consiste à écrire le modèle en logarithmes. En conséquence, il faudra interpréter les coefficients comme s'appliquant à la durée, pas au hasard.

### 7.4.3 Programmation XVI : la procédure *lifereg*

La procédure *lifereg* permet d'obtenir facilement les coefficients avec un paramétrage différent de celui que nous avons utilisé. Quand on centre les variables explicatives avant de lancer la procédure, les coefficients données par le logiciel SAS peuvent être notés  $\theta' = (\sigma, \beta')$ , où  $\sigma$  est appelé *scale* et  $\beta$  est le vecteur des paramètres des coefficients. Le vecteur de paramètres  $\theta$  obéit à la relation :

$$\begin{pmatrix} \tilde{\beta} = \alpha \\ \tilde{b} \end{pmatrix} = \begin{pmatrix} 1/\sigma \\ -\beta/\sigma \end{pmatrix} = g(\theta)$$

On doit donc utiliser le théorème de Slutsky pour revenir aux paramètres originaux. On utilisera :

$$\frac{\partial \mathbf{g}}{\partial \theta'} = \begin{pmatrix} -1/\sigma^2 & 0 \\ \beta/\sigma^2 & -I_{p+1}/\sigma \end{pmatrix}$$

et l'on calculera la matrice de covariance par :

$$\widehat{\text{Vas}}(\hat{\beta}) = \frac{\partial \mathbf{g}}{\partial \theta'}(\hat{\theta}) \widehat{\text{Vas}}(\hat{\theta}) \left( \frac{\partial \mathbf{g}}{\partial \theta'}(\hat{\theta}) \right)'$$

On peut ensuite calculer les fonctions qui caractérisent la distribution de base ; ici ces fonctions représentent la distribution de durée qui prévaut au point moyen de l'échantillon. On peut la représenter graphiquement. On aura :

$$\begin{aligned} h_0(t) &= \bar{h} \alpha t^{\alpha-1} \\ S_0(t) &= \exp(-\bar{h} t^\alpha) \\ f_0(t) &= h_0(t) S_0(t) \\ F_0(t) &= 1 - S_0(t) \end{aligned}$$

où  $\bar{h}$  est donné par l'exponentielle du terme constant inclus dans  $\tilde{\beta}$ .

Enfin, nous ferons un peu de prévision. En économétrie des durées, les prévisions individuelles peuvent être utilisées pour mettre en évidence l'hétérogénéité des individus, ou pour réaliser diverses illustrations graphiques (par exemple, Duguet, Goujard et L'horty, 2008). L'option `output=` de la procédure `lifereg` ne fournit que des prévisions pour une durée égale à la valeur de la durée observée pour chaque individu, qu'elle soit censurée ou pas. En conséquence, les prévisions de différents individus sont calculées pour des valeurs différentes de la durée et ne peuvent pas être comparées entre elles. Une exception est la sortie `xbeta` que l'on peut utiliser comme un ingrédient de prévisions calées sur les mêmes valeurs des durées. A titre d'illustration, nous allons calculer les fonctions de survies individuelles pour les cinq premières années en emploi. Chaque ligne représentera donc la fonction de survie d'un individu, et chaque colonne la probabilité de survivre la même durée pour tous les individus. Enfin, nous calculerons une estimation de l'espérance de durée pour tous les individus. En utilisant la modélisation à hasard proportionnel, la probabilité de survivre à une durée  $t$  pour l'individu  $i$  est égale à :

$$S_i(t) = \exp(-t^\alpha \exp(X_i \tilde{b})),$$

cette quantité varie avec la durée et il faudra donc faire une boucle. La seule différence avec les statistiques descriptives est que, maintenant, il faut faire

une boucle pour chaque individu (en fait, leur équivalent en langage matriciel). L'espérance mathématique de la variable de durée pour l'individu  $i$  peut se trouver facilement en utilisant la modélisation en logarithmes :

$$T_i = \frac{T_0}{\exp(X_i c)},$$

ce qui implique :

$$\begin{aligned} E(T_i) &= E(T_0) \exp(-\tilde{X}_i \tilde{c}) \\ &= \Gamma(1 + 1/\alpha) (\bar{h})^{-1/\alpha} \exp(-(X_i - \bar{X})c) \end{aligned}$$

et l'on estime cette quantité en remplaçant les paramètres théoriques par les contreparties empiriques correspondantes.

#### 7.4.4 Application XVI

Nous reprenons l'application sur la durée du premier emploi. Voici le programme de départ :

```
%let tab=duree;
%let y=duree_emp;
%let x=dip_sec dip_sup gen_f c_nais2 c_nais3
c_nais4 c_nais5;
%let d=presence;
proc standard data=&tab out=_tab mean=0; var &x;
run;
proc lifereg data=_tab outest=b covout;
model &y*&d(0)=&x;
run;
```

On lance la procédure *lifereg* à partir du tableau contenant les variables explicatives centrées. Par défaut, elle estimera un modèle de Weibull. On demande à ce que les coefficients soient sortis dans un tableau de nom *b* (option *outest=b*), et que ce tableau contienne la matrice de covariance estimée est estimateurs (option *covout*). La variable explicative est donnée par la macro variable *&y* et celle-ci est censurée quand la variable donnée par la macro variable *&d* est égale à 0. La liste des variables explicatives est donnée par la macro variable *&x*. L'instruction *model* utilise la même syntaxe que dans la procédure *lifetest*. Cette procédure fournit la sortie suivante :

**Sortie SAS 7.2.**

The LIFEREG Procedure

## Model Information

|                             |                |
|-----------------------------|----------------|
| Data Set                    | WORK._TAB      |
| Dependent Variable          | Log(duree_emp) |
| Censoring Variable          | presence       |
| Censoring Value(s)          | 0              |
| Number of Observations      | 12695          |
| Noncensored Values          | 11277          |
| Right Censored Values       | 1418           |
| Left Censored Values        | 0              |
| Interval Censored Values    | 0              |
| Number of Parameters        | 9              |
| Name of Distribution        | Weibull        |
| Log Likelihood              | -16447.86117   |
| Number of Observations Read | 12695          |
| Number of Observations Used | 12695          |

## Fit Statistics

|                          |          |
|--------------------------|----------|
| -2 Log Likelihood        | 32895.72 |
| AIC (smaller is better)  | 32913.72 |
| AICC (smaller is better) | 32913.74 |
| BIC (smaller is better)  | 32980.76 |

## Fit Statistics (Unlogged Response)

|                                  |          |
|----------------------------------|----------|
| -2 Log Likelihood                | 71203.96 |
| Weibull AIC (smaller is better)  | 71221.96 |
| Weibull AICC (smaller is better) | 71221.97 |
| Weibull BIC (smaller is better)  | 71289.00 |

Algorithm converged.

## Type III Analysis of Effects

| Effect  | DF | Wald       |            |
|---------|----|------------|------------|
|         |    | Chi-Square | Pr > ChiSq |
| dip_sec | 1  | 0.8392     | 0.3596     |
| dip_sup | 1  | 8.4501     | 0.0037     |
| gen_f   | 1  | 18.2778    | <.0001     |
| c_nais2 | 1  | 34.0892    | <.0001     |
| c_nais3 | 1  | 144.6691   | <.0001     |
| c_nais4 | 1  | 456.1524   | <.0001     |
| c_nais5 | 1  | 48.9451    | <.0001     |

## Analysis of Maximum Likelihood Parameter Estimates

| Parameter     | DF | Estimate | Standard Error | 95% Confidence Limits |         | Chi-Square |
|---------------|----|----------|----------------|-----------------------|---------|------------|
| Intercept     | 1  | 2.2248   | 0.0082         | 2.2087                | 2.2408  | 73808.8    |
| dip_sec       | 1  | -0.0196  | 0.0214         | -0.0614               | 0.0223  | 0.84       |
| dip_sup       | 1  | 0.0633   | 0.0218         | 0.0206                | 0.1060  | 8.45       |
| gen_f         | 1  | -0.0669  | 0.0157         | -0.0976               | -0.0362 | 18.28      |
| c_nais2       | 1  | -0.1241  | 0.0213         | -0.1658               | -0.0825 | 34.09      |
| c_nais3       | 1  | -0.2685  | 0.0223         | -0.3123               | -0.2248 | 144.67     |
| c_nais4       | 1  | -0.5251  | 0.0246         | -0.5732               | -0.4769 | 456.15     |
| c_nais5       | 1  | -0.2929  | 0.0419         | -0.3749               | -0.2108 | 48.95      |
| Scale         | 1  | 0.8246   | 0.0058         | 0.8134                | 0.8360  |            |
| Weibull Shape | 1  | 1.2127   | 0.0085         | 1.1961                | 1.2294  |            |

## Analysis of Maximum Likelihood Parameter Estimates

| Parameter     | Pr > ChiSq |
|---------------|------------|
| Intercept     | <.0001     |
| dip_sec       | 0.3596     |
| dip_sup       | 0.0037     |
| gen_f         | <.0001     |
| c_nais2       | <.0001     |
| c_nais3       | <.0001     |
| c_nais4       | <.0001     |
| c_nais5       | <.0001     |
| Scale         |            |
| Weibull Shape |            |

Ici, il ne faut pas oublier de changer de signe par rapport à l'estimation du modèle exponentiel de la section précédente. L'estimation donne l'effet des variables sur l'espérance de la durée, pas sur le hasard. On trouve que la durée passée dans le premier emploi est plus élevée pour les diplômés de l'enseignement supérieur, les hommes, et que la durée est d'abord décroissante puis croissante avec la date de naissance (courbe en U). Tous les paramètres sont significativement différents de 0, à l'exception du coefficient de la variable *dip\_sec* (l'indicatrice d'enseignement secondaire) ce qui signifie que son coefficient est identique à celui de la modalité de référence (l'enseignement primaire). Le paramètre *scale* est égal à  $1/\alpha$  et le paramètre *shape* est égal à  $\alpha$ . Son estimation  $\hat{\alpha} = 1.2127$  indique un hasard de base qui croît à taux décroissant.

Les effets de type III donnés en préambule de l'estimation donnent des effets "toutes choses égales par ailleurs", et n'apportent donc rien aux tests de Wald individuels présentés plus bas. On constate que les probabilités critiques sont les mêmes.

Pour aller plus loin, nous allons exploiter les sorties de cette procédure sous SAS-IML. Tout d'abord, nous préparons les données.

```
data b; set b;
if compress(upcase(_name_))="SCALE"
then _name_="_SCALE_";
data parms; set b; if _type_="PARMS";
data cov; set b; if _type_="COV";
```

Les estimations ont été mises dans le tableau *b* par l'option *outest=b covout* de la procédure *lifereg*. Dans ce tableau, il y a une variable *\_SCALE\_* qui est indiquée sous le nom *Scale*, sans les caractères de soulignement, ce qui va poser problème lors de la lecture de la matrice de covariance car les noms doivent être les mêmes en ligne et en colonne. On renomme donc *Scale* en *\_SCALE\_*. Ensuite, on met le paramètre estimé dans le tableau *Parms* et la matrice de covariance asymptotique estimée dans le tableau *Cov*. Voici le résultat :

### Sortie SAS 7.3.

| Obs | _MODEL_ | _NAME_    | _TYPE_ | _DIST_  | _STATUS_    | _LNLIKE_  |
|-----|---------|-----------|--------|---------|-------------|-----------|
| 1   |         | duree_emp | PARMS  | Weibull | 0 Converged | -16447.86 |

| Obs | emp | Intercept | dip_sec   | dip_sup  | gen_f     | c_nais2  |
|-----|-----|-----------|-----------|----------|-----------|----------|
| 1   | -1  | 2.22477   | -0.019565 | 0.063311 | -0.066913 | -0.12415 |

| Obs | c_nais3  | c_nais4  | c_nais5  | _SCALE_ |
|-----|----------|----------|----------|---------|
| 1   | -0.26854 | -0.52505 | -0.29286 | 0.82463 |

### Sortie SAS 7.4.

| Obs | _MODEL_ | _NAME_    | _TYPE_ | _DIST_  | _STATUS_    | _LNLIKE_  |
|-----|---------|-----------|--------|---------|-------------|-----------|
| 1   |         | Intercept | COV    | Weibull | 0 Converged | -16447.86 |
| 2   |         | dip_sec   | COV    | Weibull | 0 Converged | -16447.86 |
| 3   |         | dip_sup   | COV    | Weibull | 0 Converged | -16447.86 |
| 4   |         | gen_f     | COV    | Weibull | 0 Converged | -16447.86 |
| 5   |         | c_nais2   | COV    | Weibull | 0 Converged | -16447.86 |

|   |         |     |         |   |           |           |
|---|---------|-----|---------|---|-----------|-----------|
| 6 | c_nais3 | COV | Weibull | 0 | Converged | -16447.86 |
| 7 | c_nais4 | COV | Weibull | 0 | Converged | -16447.86 |
| 8 | c_nais5 | COV | Weibull | 0 | Converged | -16447.86 |
| 9 | _SCALE_ | COV | Weibull | 0 | Converged | -16447.86 |

|     | duree_   |             |             |             |             |
|-----|----------|-------------|-------------|-------------|-------------|
| Obs | emp      | Intercept   | dip_sec     | dip_sup     | gen_f       |
| 1   | 2.22477  | 0.000067060 | -.000001114 | 0.000003470 | -.000000535 |
| 2   | -0.01956 | -.000001114 | 0.000456109 | 0.000306379 | 0.000001657 |
| 3   | 0.06331  | 0.000003470 | 0.000306379 | 0.000474344 | -.000023478 |
| 4   | -0.06691 | -.000000535 | 0.000001657 | -.000023478 | 0.000244960 |
| 5   | -0.12415 | 0.000001908 | -.000078561 | -.000072315 | -.000006996 |
| 6   | -0.26854 | 0.000001845 | -.000122576 | -.000114419 | -.000009649 |
| 7   | -0.52505 | 0.000003618 | -.000110525 | -.000142315 | -.000007875 |
| 8   | -0.29286 | 0.000067195 | -.000080886 | -.000108350 | 0.000001267 |
| 9   | 0.82463  | -.000009529 | -.000003175 | -.000000597 | 0.000000195 |

| Obs | c_nais2     | c_nais3     | c_nais4     | c_nais5     | _SCALE_     |
|-----|-------------|-------------|-------------|-------------|-------------|
| 1   | 0.000001908 | 0.000001845 | 0.000003618 | 0.000067195 | -.000009529 |
| 2   | -.000078561 | -.000122576 | -.000110525 | -.000080886 | -.000003175 |
| 3   | -.000072315 | -.000114419 | -.000142315 | -.000108350 | -.000000597 |
| 4   | -.000006996 | -.000009649 | -.000007875 | 0.000001267 | 0.000000195 |
| 5   | 0.000452131 | 0.000225535 | 0.000226461 | 0.000221255 | 0.000001920 |
| 6   | 0.000225535 | 0.000498479 | 0.000241895 | 0.000236185 | 0.000005843 |
| 7   | 0.000226461 | 0.000241895 | 0.000604357 | 0.000250278 | 0.000012830 |
| 8   | 0.000221255 | 0.000236185 | 0.000250278 | 0.001752302 | 0.000041242 |
| 9   | 0.000001920 | 0.000005843 | 0.000012830 | 0.000041242 | 0.000033297 |

La procédure IML suivante réalise l'ensemble des opérations présentées plus haut : écriture des paramètres de type hasard proportionnel, graphique des fonctions qui caractérisent la distribution de base, prévision d'un ensemble de survies individuelles pour les mêmes durées, et calcul des estimations de durée moyenne pour chaque individu. Le programme est le suivant :

```
1. proc iml;
 use cov;
 read all var{_name_} into nx;
 nx=compress(nx);
 read all var(nx) into vtheta;
 use parms;
 read all var (nx) into theta;
```

Lecture des noms des paramètres dans la colonne `_name_` du tableau `cov`; on enlève les espaces en trop grâce à la commande `compress`. On utilise ensuite ces noms pour lire la matrice de covariance, que l'on range dans `vtheta`, puis le paramètre, que l'on range dans `theta`.

```

2. theta=t(theta); p=nrow(theta);
 rgb=1:(p-1); rgs=p;
 beta=theta[rgb]; sigma=theta[rgs];
 b=(1/sigma)/(-beta/sigma);
 nb="Alpha"/nx[rgb];
 dg=(j(1,p-1,0)||(-1/sigma##2))//
 ((-i(p-1)/sigma)|| (beta/sigma##2));
 vb=dg*vtheta*t(dg);
 sb=sqrt(vecdiag(vb));
 tb=abs(b)/sb;
 pb=1-probchi(tb##2,1);
 print "Coefficients for the hasard function",
 "with parametrisation ",
 "h0(t)=alpha.h.t^(alpha-1)", ,
 b[rowname=nb] sb tb pb;
 stat=b[1]/sb[1];
 pstat=1-probchi(stat##2,1);
 print "Test H0 : exponential distribution",
 "Statistic =" stat,
 "p value =" pstat;

```

On transpose le paramètre  $\theta$  fournit par la procédure *lifereg* car c'est plus pratique pour l'affichage. Il possède  $p$  composantes. La variable *rgb* contient tous les entiers de 1 à  $p-1$ , il s'agit de la position du paramètre  $\beta$ , la variable *rgs* contient la position du paramètre  $\sigma$  (en dernière position). On calcule ensuite le paramètre  $\tilde{b}$  en fonction de  $\theta$  et on applique le théorème de Slutsky pour obtenir une estimation de la matrice de covariance de  $\hat{b}$ . On affiche le résultat obtenu, puis on calcule le test d'adéquation à la distribution exponentielle ( $H_0 : \alpha = 1$ ). Le  $t$  de Student élevé au carré donne la statistique de Wald pour ce test. On donne le  $t$  de Student puis la probabilité critique du test de Wald. On obtient la sortie suivante.

#### Sortie SAS 7.5.

```

 Coefficients for the hasard function
 with parametrisation
 h0(t)=alpha.h.t^ (alpha-1)

 b sb tb pb
Alpha 1.2126582 0.0084856 142.90806 0
Intercept -2.697888 0.0230351 117.12051 0
dip_sec 0.0237252 0.0258947 0.9162215 0.3595508
dip_sup -0.076774 0.026419 2.9060214 0.0036606

```

```

gen_f 0.0811424 0.0189893 4.2730652 0.0000193
c_nais2 0.1505492 0.0258232 5.8300052 5.5426E-9
c_nais3 0.3256489 0.0272731 11.940302 0
c_nais4 0.6367078 0.0305386 20.84928 0
c_nais5 0.3551381 0.0512453 6.930158 4.204E-12

```

```

Test H0 : exponential distribution
 stat
Statistic = 142.90806
 pstat
p value = 0

```

```

3. alpha=b[1]; h=exp(b[2]);
do t=0 to 1 by 0.01;
 valt=valt//t;
 hbase=hbase//(alpha#h#t##(alpha-1));
 sbase=sbase//(exp(-h#t##alpha));
end;
do t=1 to 30 by 0.1;
 valt=valt//t;
 hbase=hbase//(alpha#h#t##(alpha-1));
 sbase=sbase//(exp(-h#t##alpha));
end;
fbase=hbase#sbase;
cbase=j(nrow(sbase),1,1)-sbase;
base=valt || hbase || fbase || sbase || cbase;
labase="Duration" || "Hasard" || "PDF" || "SDF"
 || "CDF";
create base from base[colname=labase];
append from base;

```

Le paramètre  $\alpha$  correspond au premier élément de  $b$  (ici  $\tilde{\beta}$ ), et le paramètre  $\bar{h}$  correspond à l'exponentielle du terme constant, le deuxième élément du vecteur  $b$ . Pour améliorer le graphique, on prend un pas fin sur l'intervalle  $[0, 1]$  et un pas plus grand pour  $t > 1$ . Une première boucle calcule le hasard et la survie de base pour  $t = 0$  à 1 avec un pas de 0.01, et la seconde boucle prend la suite de 1 à 30 avec un pas de 0.1. L'ensemble des résultats est empilé. A la fin de la seconde boucle, on calcule la densité globalement par le produit terme à terme (#) suivant :

$$\hat{f}_i(t) = \hat{h}_i(t)\hat{S}_i(t),$$

de même, on calcule la fonction de répartition comme :

$$\hat{F}_i(t) = 1 - \hat{S}_i(t),$$

en n'oubliant pas que *sbase* est un vecteur. On retranche donc le vecteur unité à *sbase*. On crée ensuite une base en sortie pour réaliser les graphiques après la sortie de la procédure IML. Le hasard et la densité sont donnés par le graphique 7.2 tandis que la survie et la fonction de répartition sont donnés par le graphique 7.3.

```
4. rgc=2:(p-1); nxc=nx[rgc]; rg1=2:p;
 use _tab; read all var(nxc) into xc;
 n=nrow(xc);
 xt=j(n,1,1) || xc;
 do t=1 to 5;
 s0=s0 || exp(-t##alpha)##exp(xt*b[rg1]);
 ns0=ns0 || compress("S"+char(t));
 end;
 create prev from s0[colname=ns0];
 append from s0;
```

On effectue les prévisions des survies individuelles pour  $t$  variant de 1 à 5. On commence par lire les variables explicatives centrées. Leurs noms sont situés entre les positions 2 et  $p - 1$  de la matrice de sortie SAS (*intercept* figure en première position et *\_Scale\_* en  $p$ -ième position). On ajoute un terme constant à cet ensemble de variables et l'on applique la formule du cours. La boucle effectue le calcul pour des durées variant de 1 an à 5 ans. On peut la modifier librement (tant que les valeurs sont entières, à cause de la formation des noms de variables qui suit). L'instruction `char(t)` transforme la durée en caractère, et `"S"+char(t)` concatène les deux caractères ensemble (pour  $t=3$ , cela donne la chaîne de caractères S3). Le noms de variables sont donc construits en même temps que les prévisions. On sauvegarde les prévisions dans le tableau *prev*. Le tableau complet, comporte 12695 lignes (individus) et 5 colonnes. En voici un petit extrait. Les fonctions se lisent en ligne et chaque ligne correspond à une fonction différente (si les variables explicatives sont différentes). On peut résumer l'hétérogénéité des fonctions de survie en traçant leur densité. Le graphique 7.4 représente la distribution des survies individuelles pour une durée de 5 ans,  $\hat{S}_i(5)$ .

#### Sortie SAS 7.6.

| Obs | S1 | S2 | S3 | S4 | S5 |
|-----|----|----|----|----|----|
|-----|----|----|----|----|----|

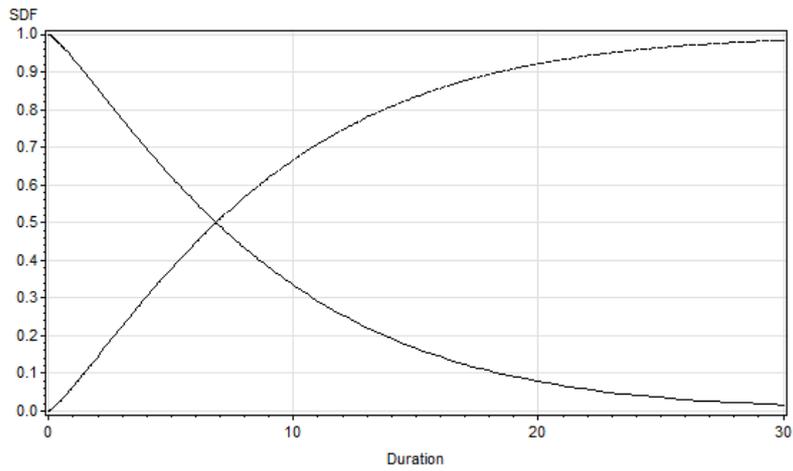
|    |         |         |         |         |         |
|----|---------|---------|---------|---------|---------|
| 1  | 0.90568 | 0.79484 | 0.68699 | 0.58733 | 0.49781 |
| 2  | 0.93128 | 0.84789 | 0.76353 | 0.68220 | 0.60576 |
| 3  | 0.94989 | 0.88768 | 0.82299 | 0.75871 | 0.69632 |
| 4  | 0.95010 | 0.88814 | 0.82369 | 0.75962 | 0.69742 |
| 5  | 0.95370 | 0.89596 | 0.83558 | 0.77522 | 0.71624 |
| 6  | 0.93646 | 0.85886 | 0.77976 | 0.70284 | 0.62990 |
| 7  | 0.94989 | 0.88768 | 0.82299 | 0.75871 | 0.69632 |
| 8  | 0.94989 | 0.88768 | 0.82299 | 0.75871 | 0.69632 |
| 9  | 0.91430 | 0.81248 | 0.71210 | 0.61799 | 0.53214 |
| 10 | 0.94600 | 0.87928 | 0.81029 | 0.74217 | 0.67650 |
| 11 | 0.93606 | 0.85800 | 0.77848 | 0.70122 | 0.62799 |
| 12 | 0.94989 | 0.88768 | 0.82299 | 0.75871 | 0.69632 |
| 13 | 0.95010 | 0.88814 | 0.82369 | 0.75962 | 0.69742 |
| 14 | 0.94199 | 0.87066 | 0.79735 | 0.72542 | 0.65655 |
| 15 | 0.94474 | 0.87657 | 0.80621 | 0.73688 | 0.67018 |
| 16 | 0.94199 | 0.87066 | 0.79735 | 0.72542 | 0.65655 |
| 17 | 0.94199 | 0.87066 | 0.79735 | 0.72542 | 0.65655 |
| 18 | 0.94989 | 0.88768 | 0.82299 | 0.75871 | 0.69632 |
| 19 | 0.94639 | 0.88011 | 0.81155 | 0.74380 | 0.67844 |
| 20 | 0.94199 | 0.87066 | 0.79735 | 0.72542 | 0.65655 |

```

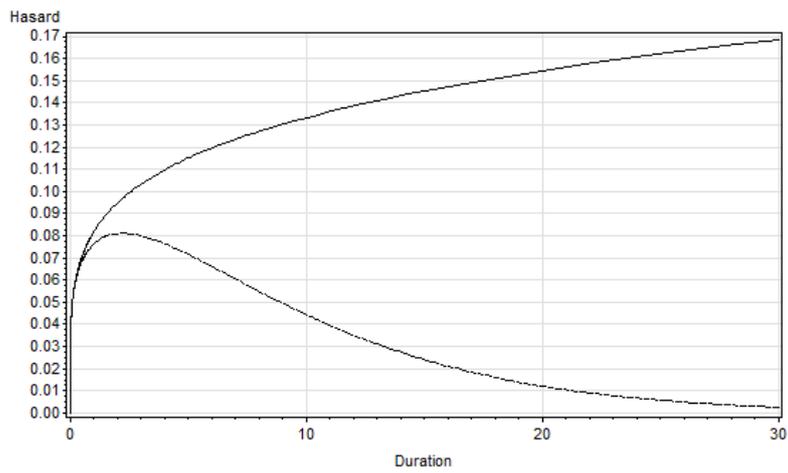
5. et0=gamma(1+1/alpha)/(h##(1/alpha));
print "Durée moyenne au point moyen", et0;
rg2=3:p;
eti=et0/exp(xc*b[rg2]/alpha);
create eti from eti[colname="Mean"];
append from eti;
quit;
run;

```

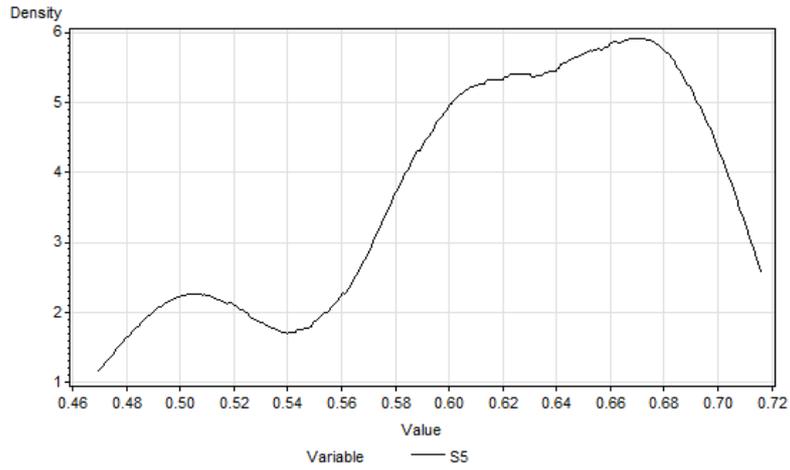
Nous calculons maintenant, la durée moyenne d'emploi...au point moyen de l'échantillon, car cette durée varie avec chaque individu (tout le monde n'a pas la même moyenne, qui dépend des variables explicatives). La quantité  $et0$  contient cette espérance au point moyen. On calcule ensuite les durées moyennes de tous les individus. Ces durées sont corrigées de la censure. La variable  $rg2$  contient la position des variables dans le vecteur  $\tilde{\beta}$  (la première position est occupée par  $\alpha$  et la deuxième par le terme constant  $\ln \bar{h}$ , on commence donc en 3). Le tout est rangé dans un tableau SAS  $eti$ . Le graphique 7.5 représente la densité des moyennes individuelles. On voit qu'il existe trois modes, autour de 6 ans  $\frac{1}{2}$ , 8 ans  $\frac{1}{2}$  et 10 ans. Cette répartition est à rapprocher des répartition des variables explicatives.



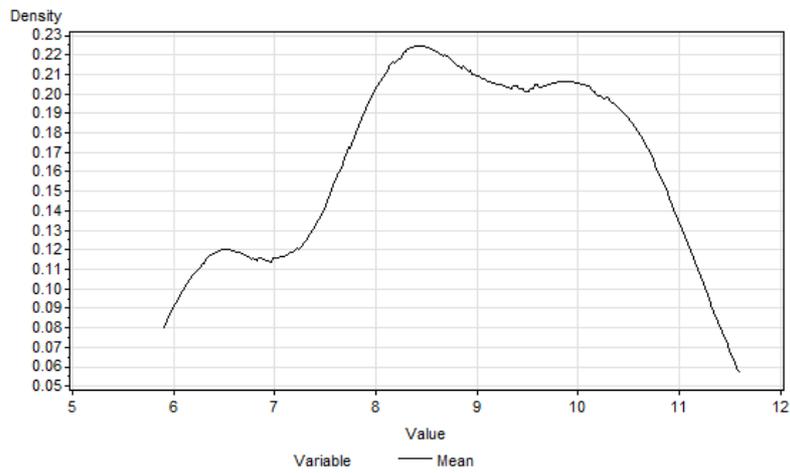
GRAPHIQUE 7.2 – WEIBULL, DISTRIBUTION DE BASE,  $\hat{S}_0(t)$  ET  $\hat{F}_0(t)$



GRAPHIQUE 7.3 – WEIBULL, DISTRIBUTION DE BASE,  $\hat{h}_0(t)$  ET  $\hat{f}_0(t)$



GRAPHIQUE 7.4 – WEIBULL, DISTRIBUTION DES SURVIES INDIVIDUELLES À 5 ANS



GRAPHIQUE 7.5 – WEIBULL, DISTRIBUTION DES MOYENNES INDIVIDUELLES

## Annexe A

# Rappels de statistique

### A.1 Espérance mathématique et fonction de survie

La propriété suivante peut s'avérer utile pour calculer l'espérance de la durée (i.e. l'espérance de vie à la naissance) :

$$\begin{aligned} E(T) &= \int_0^{+\infty} x f(x) dx \\ &= \int_0^{+\infty} -x \frac{dS(x)}{dx} dx, \end{aligned}$$

en intégrant par partie ( $u = -x$ ,  $v' = S'(x)$ ), on obtient :

$$\begin{aligned} E(T) &= [-xS(x)]_0^{+\infty} + \int_0^{+\infty} S(x) dx \\ &= \int_0^{+\infty} S(x) dx, \end{aligned}$$

sous l'hypothèse que :

$$\lim_{x \rightarrow +\infty} xS(x) = 0,$$

et cette hypothèse est généralement bien vérifiée car les fonctions de survie contiennent souvent des exponentielles et des puissances qui annulent la limite ci-dessus. Cette formule peut être utile, mais il faut garder à l'esprit que dans certains cas la méthode la plus simple est celle de la fonction génératrice des moments que nous présenterons plus loin.

### A.2 Le changement de variable

Supposons que l'on dispose d'une variable de durée  $T$  de densité  $f_T(t)$  et que l'on effectue un changement de variable  $U = g(T)$ . La fonction de réparti-

tion de la variable est donnée par :

$$F_U(u) = \Pr[g(T) \leq u],$$

à ce stade, on doit distinguer le cas des fonctions  $g$  croissantes de celui des fonctions décroissantes. Si la fonction  $g$  est croissante,  $g^{-1}$  est croissante :

$$F_U(u) = \Pr[T \leq g^{-1}(u)] = F_T(g^{-1}(u)),$$

ce qui implique que la densité de la variable  $u$  est donnée par :

$$f_U(u) = \frac{dg^{-1}(u)}{du} f_T(g^{-1}(u)) > 0.$$

Dans le cas où la fonction  $g$  est décroissante (donc  $g^{-1}$  est décroissante, on a :

$$F_U(u) = \Pr[T \geq g^{-1}(u)] = 1 - F_T(g^{-1}(u)),$$

ce qui implique que la densité de la variable  $u$  est donnée par :

$$f_U(u) = -\frac{dg^{-1}(u)}{du} f_T(g^{-1}(u)) > 0.$$

Globalement, on a donc :

$$f_U(u) = \left| \frac{dg^{-1}(u)}{du} \right| f_T(g^{-1}(u)) > 0.$$

### A.3 Constante d'Euler

La constante d'Euler est définie par :

$$\gamma = \sum_{k=1}^{+\infty} \left( \frac{1}{k} - \ln \left( 1 + \frac{1}{k} \right) \right) \simeq 0,5772156649\dots$$

Elle vérifie les deux propriétés suivantes :

$$\gamma_E = - \int_0^{+\infty} (\ln x) e^{-x} dx,$$

et

$$\gamma_E^2 + \frac{\pi^2}{6} = \int_0^{+\infty} (\ln x)^2 e^{-x} dx.$$

## A.4 Fonctions Gamma, Beta et Polygamma

### A.4.1 Fonction Gamma

La fonction Gamma est définie par :

$$\Gamma(p) = \int_0^{+\infty} x^{p-1} e^{-x} dx, \quad p > 0.$$

On a donc :

$$\Gamma(p+1) = \int_0^{+\infty} x^p e^{-x} dx,$$

en intégrant par parties avec  $u = x^p$  et  $v' = e^{-x}$ , on obtient :

$$\begin{aligned} \Gamma(p+1) &= \underbrace{[x^p (-e^{-x})]_0^{+\infty}}_0 + \int_0^{+\infty} p x^{p-1} e^{-x} dx \\ &= p \int_0^{+\infty} x^{p-1} e^{-x} dx \\ &= p \Gamma(p), \end{aligned}$$

Les propriétés suivantes sont également utiles :

- $\Gamma(1) = 1$ . En effet :

$$\Gamma(1) = \int_0^{+\infty} e^{-x} dx = [-e^{-x}]_0^{+\infty} = 1$$

- Si  $p$  est un nombre entier, on a, par récurrence :

$$\begin{aligned} \Gamma(1) &= 1, \quad \Gamma(p+1) = p \Gamma(p) \\ \Rightarrow \Gamma(p) &= (p-1)!, \quad p \geq 1 \end{aligned}$$

- $\Gamma\left(\frac{1}{2}\right) = \sqrt{\pi}$

- La dérivée de la fonction Gamma est égale à :

$$\Gamma'(p) = \int_0^{+\infty} (\ln x) x^{p-1} e^{-x} dx,$$

ce qui implique :

$$\Gamma'(1) = -\gamma_E,$$

où  $\gamma_E$  est la constante d'Euler

- La dérivée seconde de la fonction Gamma est égale à :

$$\Gamma''(p) = \int_0^{+\infty} (\ln x)^2 x^{p-1} e^{-x} dx,$$

ce qui implique :

$$\Gamma''(1) = \gamma_E^2 + \frac{\pi^2}{6}.$$

#### A.4.2 La fonction Beta

- La fonction Beta est définie par :

$$B(p, s) = \int_0^1 u^{p-1} (1-u)^{s-1} du,$$

- La fonction Beta vérifie :

$$B(p, s) = \frac{\Gamma(p)\Gamma(s)}{\Gamma(p+s)}.$$

#### A.4.3 La fonction Polygamma

La fonction polygamma d'ordre  $m$  est définie comme la dérivée d'ordre  $m+1$  du logarithme de la fonction Gamma :

$$\psi^{(m)} = \frac{d^{m+1}}{dz^{m+1}} [\ln \Gamma(z)]$$

on a souvent besoin, pour les estimations par le maximum de vraisemblance, des fonctions *digamma* :

$$\psi^{(0)} = \frac{d}{dz} [\ln \Gamma(z)]$$

et *trigamma* :

$$\psi^{(1)} = \frac{d^2}{dz^2} [\ln \Gamma(z)]$$

#### A.4.4 La fonction Gamma tronquée

Les fonctions Gamma tronquées régularisées ("regularized incomplete Gamma functions") se définissent par :

$$\gamma(a, x) = \frac{1}{\Gamma(a)} \int_0^x u^{a-1} e^{-u} du,$$

$$\Gamma(a, x) = \frac{1}{\Gamma(a)} \int_x^{+\infty} u^{a-1} e^{-u} du,$$

et l'on remarque que :

$$\gamma(a, x) + \Gamma(a, x) = 1, \quad \lim_{x \rightarrow +\infty} \gamma(a, x) = 1, \quad \lim_{x \rightarrow 0} \Gamma(a, x) = 1.$$

Pour évaluer les fonctions Gamma tronquées à partir des logiciels usuels, il faut utiliser la fonction de répartition du khi-deux, qui est généralement disponible. La somme des carrés de  $k$  variables aléatoires normales centrées-réduites

indépendantes suit une loi du khi-deux de paramètre  $k$ . Sa densité est donnée par :

$$f(z) = \frac{1}{2^{k/2}} \frac{z^{k/2-1} e^{-z/2}}{\Gamma(k/2)}, \quad z > 0,$$

sa fonction de répartition est donnée par :

$$F_k(z) = \frac{1}{2^{k/2} \Gamma(k/2)} \int_0^z x^{k/2-1} e^{-x/2} dx,$$

en faisant le changement de variables  $u = x/2$ , on obtient :

$$\begin{aligned} F_k(z) &= \frac{1}{2^{k/2} \Gamma(k/2)} \int_0^{z/2} (2u)^{k/2-1} e^{-u} \times 2 du \\ &= \frac{\int_0^{z/2} u^{k/2-1} e^{-u} du}{\Gamma(k/2)} \\ &= \gamma(k/2, z/2), \end{aligned}$$

en conséquence, on peut évaluer  $\gamma(a, x)$  à partir de la formule :

$$\gamma(a, x) = F_{\chi^2, 2a}(2x),$$

ce qui est très pratique pour estimer les modèles Gamma sur données censurées.

## A.5 Loi Gamma à deux paramètres

Une variable aléatoire  $X$  suit une loi Gamma de paramètres  $(a, b)$ , notée  $G(a, b)$  si sa densité s'écrit :

$$f_X(x) = \frac{b^a x^{a-1} \exp(-bx)}{\Gamma(a)}, \quad x > 0, \quad a > 0, \quad b > 0,$$

les deux premiers moments sont égaux à  $E(X) = a/b$  et  $V(X) = a/b^2$ . Si  $a = 1$ , on retrouve la loi exponentielle dont la densité est égale à :

$$f_X(x) = b \exp(-bx),$$

et dont les deux premiers moments sont égaux à  $E(X) = 1/b$  et  $V(X) = 1/b^2$ . Si on prend le cas symétrique ( $b = 1$ ), on obtient une loi Gamma à un paramètre de densité :

$$f_X(x) = \frac{x^{a-1} \exp(-x)}{\Gamma(a)},$$

dont les deux premiers moments sont égaux à  $E(X) = V(X) = a$ .



## Annexe B

# Calcul des moments

Lors des estimations, on peut avoir besoin de faire une prévision de la durée moyenne passée dans l'état étudié. La méthode la plus simple avec les distributions qui précèdent est, souvent, d'utiliser les fonctions génératrices des moments. Toutefois, dans certains cas, les fonctions génératrices des moments n'existent pas pour  $t > 0$  donc il faut faire le calcul directement. La première section présente l'approche par la fonction génératrice des moments ; la seconde section utilise l'approche du calcul direct.

### B.1 Approche par la fonction génératrice des moments

#### B.1.1 Moments non centrés

La fonction génératrice des moments non centrés d'une variable aléatoire  $T$  est définie par :

$$M(s) = E(e^{sT}) = \int_0^{+\infty} e^{st} f_T(t) dt,$$

quand cette intégrale converge. On remarque que :

$$M(0) = E(e^0) = 1.$$

On vérifie que :

$$M'(s) = E(Te^{sT}),$$

et, par récurrence, que :

$$\frac{d^k M}{ds^k}(s) = E(T^k e^{sT}).$$

En prenant la quantité précédente en  $s = 0$ , on obtient :

$$\frac{d^k M}{ds^k}(0) = E(T^k).$$

On peut donc obtenir les moments non centrés par dérivation au lieu de procéder par intégration, ce qui est plus rapide.

### B.1.2 Moments centrés

De la même manière, on peut obtenir certains moments centrés par la fonction :

$$K(s) = \ln M(s),$$

on voit que :

$$K'(s) = \frac{M'(s)}{M(s)},$$

en prenant la fonction précédente au point  $s = 0$  on a :

$$K'(0) = E(T),$$

en dérivant la fonction  $K$  une deuxième fois, on obtient :

$$K''(s) = \frac{M''(s)}{M(s)} - \left( \frac{M'(s)}{M(s)} \right)^2,$$

en la prenant au point  $s = 0$  :

$$K''(0) = E(T^2) - E(T)^2 = V(T).$$

### B.1.3 Moments du logarithme

Enfin, en économétrie des durées, on a souvent besoin des fonctions génératrices des moments du logarithme d'une variable de durée, parce que tous les modèles peuvent s'écrire en logarithmes du fait que  $T > 0$  :

$$M_{\ln T}(s) = E(e^{s \ln T}) = E(T^s),$$

dont l'intérêt est ici évident. Pour calculer la variance du logarithme d'une variable de durée, on utilisera la fonction correspondante des moments centrés :

$$K_{\ln T}(s) = \ln M_{\ln T}(s) = \ln E(T^s).$$

### B.1.4 Loi exponentielle

La densité est donnée par :

$$f(t) = h \exp(-ht), \quad t > 0, h > 0,$$

on a donc :

$$\begin{aligned} M(s) &= E(e^{sT}) \\ &= h \int_0^{+\infty} e^{-(h-s)t} dt \\ &= h \left[ -\frac{1}{h-s} e^{-(h-s)t} \right]_0^{+\infty} \\ &= \frac{h}{h-s}, \end{aligned}$$

remarquons bien ici que l'on a choisi de mettre dans l'exponentielle un terme en  $h-s$  car on utilise cette fonction en  $s=0$ , ce qui garantit que  $h-s > 0$  et donc la convergence de l'intégrale. On en déduit :

$$\begin{aligned} M'(s) &= \frac{h}{(h-s)^2} \Rightarrow E(T) = M'(0) = \frac{1}{h}, \\ M''(s) &= \frac{2h}{(h-s)^3} \Rightarrow E(T^2) = M''(0) = \frac{2}{h^2}, \end{aligned}$$

et par récurrence :

$$E(T^k) = \frac{k!}{h^k} = \frac{\Gamma(1+k)}{h^k}.$$

On pourrait en déduire la variance par la formule classique :

$$\begin{aligned} V(T) &= E(T^2) - E(T)^2 \\ &= 2/h^2 - (1/h)^2 \\ &= 1/h^2, \end{aligned}$$

mais on peut l'obtenir plus directement par la fonction génératrice des moments centrés. Elle est définie par :

$$K(s) = \ln h - \ln(h-s).$$

On en déduit :

$$\begin{aligned} K'(s) &= \frac{1}{h-s} \Rightarrow K'(0) = E(T) = \frac{1}{h}, \\ K''(s) &= \frac{1}{(h-s)^2} \Rightarrow K''(0) = V(T) = \frac{1}{h^2}. \end{aligned}$$

Les fonctions génératrices correspondant au logarithme de la durée  $\ln T$  sont celles de la loi de Gumbel données dans la section suivante.

### B.1.5 Loi de Gumbel

On peut calculer la fonction génératrice de la loi de Gumbel en remarquant qu'une variable de ce type s'obtient comme le logarithme d'une variable exponentielle d'espérance 1 et en utilisant la propriété :

$$K_{\ln T}(s) = E(T^s).$$

Il suffit donc de calculer le moment d'ordre  $s$  de la loi exponentielle d'espérance 1. En fait, dans ce cas particulier, il n'y a pas de calcul à faire, puisque l'on a :

$$E(T^s) = \int_0^{+\infty} t^s \exp(-t) dt = \Gamma(1+s),$$

on en déduit la fonction génératrice des moments centrés du logarithme de la loi exponentielle (i.e. de la loi de Gumbel) :

$$K_{\ln T}(s) = \ln \Gamma(1+s)$$

ce qui implique :

$$\begin{aligned} K'_{\ln T}(s) &= \frac{\Gamma'(1+s)}{\Gamma(1+s)} \\ \Rightarrow K'_{\ln T}(0) &= E(\ln T) = \Gamma'(1) = -\gamma_E, \end{aligned}$$

où  $\gamma_E$  est la constante d'Euler. De même, on voit que :

$$K''_{\ln T}(s) = \frac{\Gamma''(1+s)}{\Gamma(1+s)} - \left( \frac{\Gamma'(1+s)}{\Gamma(1+s)} \right)^2,$$

de sorte que :

$$\begin{aligned} K''_{\ln T}(0) &= V(\ln T) \\ &= \Gamma''(1) - \Gamma'(1)^2 \\ &= \gamma_E^2 + \frac{\pi^2}{6} - (-\gamma_E)^2 \\ &= \frac{\pi^2}{6}. \end{aligned}$$

### B.1.6 Loi Gamma à un paramètre

La fonction génératrice de la loi Gamma à un paramètre est définie par :

$$\begin{aligned} M_T(s) &= E(e^{sT}) \\ &= \int_0^{+\infty} e^{st} \frac{t^{\beta-1} e^{-t}}{\Gamma(\beta)} dt \\ &= \int_0^{+\infty} \frac{t^{\beta-1} e^{-(1-s)t}}{\Gamma(\beta)} dt, \end{aligned}$$

on effectue donc le changement de variable  $x = (1-s)t$ , de sorte que les bornes sont inchangées et que  $dt = (1-s)^{-1} dx$ . On obtient donc :

$$\begin{aligned} M_T(s) &= \frac{1}{\Gamma(\beta)} \int_0^{+\infty} \left(\frac{x}{1-s}\right)^{\beta-1} e^{-x} \frac{1}{1-s} dx \\ &= \frac{1}{(1-s)^\beta \Gamma(\beta)} \underbrace{\int_0^{+\infty} x^{\beta-1} e^{-x} dx}_{\Gamma(\beta)} \\ &= (1-s)^{-\beta}. \end{aligned}$$

Pour obtenir l'espérance et la variance de la loi Gamma, on utilise donc :

$$K_T(s) = \ln M(s) = -\beta \ln(1-s).$$

La dérivée première donne l'espérance de la distribution :

$$K'_T(s) = \frac{\beta}{1-s} \Rightarrow K'_T(0) = E(T) = \beta,$$

et la dérivée seconde donne la variance :

$$K''_T(s) = \frac{\beta}{(1-s)^2} \Rightarrow K''_T(0) = V(T) = \beta.$$

La fonction génératrice du logarithme de cette variable s'obtient par :

$$\begin{aligned} M_{\ln T}(s) &= E(T^s) \\ &= \int_0^{+\infty} \frac{t^{s+\beta-1} e^{-t}}{\Gamma(\beta)} dt \\ &= \frac{\Gamma(s+\beta)}{\Gamma(\beta)}, \end{aligned}$$

et pour trouver ses deux premiers moments on utilise :

$$K_{\ln T}(s) = \ln M_{\ln T}(s) = \ln \Gamma(s + \beta) - \ln \Gamma(\beta).$$

Pour le modèle en logarithmes, on utilise donc :

$$K'_{\ln T}(s) = \frac{\Gamma'(s + \beta)}{\Gamma(s + \beta)} \Rightarrow K'_{\ln T}(0) = E(\ln T) = \frac{\Gamma'(\beta)}{\Gamma(\beta)},$$

ainsi que :

$$\begin{aligned} K''_{\ln T}(s) &= \frac{\Gamma''(s + \beta)}{\Gamma(s + \beta)} - \left( \frac{\Gamma'(s + \beta)}{\Gamma(s + \beta)} \right)^2 \\ \Rightarrow K''_{\ln T}(0) &= V(\ln T) = \frac{\Gamma''(\beta)}{\Gamma(\beta)} - \left( \frac{\Gamma'(\beta)}{\Gamma(\beta)} \right)^2. \end{aligned}$$

### B.1.7 Loi Gamma à deux paramètres

Pour trouver la fonction génératrice de la loi Gamma à deux paramètres il suffit de remarquer que  $U = \ln hT$  suit une loi de Gumbel de fonction génératrice des moments égale à  $\Gamma(1 + s)$ . Ceci implique :

$$\begin{aligned} M_U(s) &= E\left(e^{s \ln(hT)}\right) = \Gamma(1 + s) \\ \Leftrightarrow E\left(h^s T^s\right) &= \Gamma(1 + s) \\ \Leftrightarrow E\left(T^s\right) &= h^{-s} \Gamma(1 + s), \end{aligned}$$

on en déduit :

$$E(T) = \frac{1}{h} \text{ et } V(T) = \frac{1}{h^2}$$

### B.1.8 Loi de Weibull

Pour trouver la fonction génératrice des moments de la loi de Weibull, il suffit de remarquer que  $U = \ln hT^\alpha$  suit une loi de Gumbel de fonction génératrice des moments égale à  $\Gamma(1 + s)$ . Ceci implique :

$$\begin{aligned} M_U(s) &= E\left(e^{s \ln(hT^\alpha)}\right) = \Gamma(1 + s) \\ \Leftrightarrow E\left(h^s T^{\alpha s}\right) &= \Gamma(1 + s) \\ \Leftrightarrow E\left(T^{\alpha s}\right) &= h^{-s} \Gamma(1 + s) \\ \Leftrightarrow E\left(T^j\right) &= h^{-j/\alpha} \Gamma(1 + j/\alpha), \end{aligned}$$

avec  $j = \alpha s$ . On en déduit :

$$E(T) = h^{-1/\alpha} \Gamma(1 + 1/\alpha) \text{ et } E(T^2) = h^{-2/\alpha} \Gamma(1 + 2/\alpha),$$

d'où la variance :

$$\begin{aligned} V(T) &= E(T^2) - E(T)^2 \\ &= h^{-2/\alpha} (\Gamma(1 + 2/\alpha) - \Gamma(1 + 1/\alpha)^2). \end{aligned}$$

On remarque que l'espérance mathématique peut également se simplifier par la formule :

$$\begin{aligned} \Gamma(x) &= (x-1) \Gamma(x-1) \\ \Rightarrow \Gamma(1 + 1/\alpha) &= \frac{1}{\alpha} \Gamma\left(\frac{1}{\alpha}\right), \end{aligned}$$

ce qui implique :

$$E(T) = h^{-1/\alpha} \frac{1}{\alpha} \Gamma\left(\frac{1}{\alpha}\right).$$

L'expression des moments du logarithme de la durée est également utilisée dans les applications. D'après ce qui précède :

$$M_{\ln T}(s) = E(T^s) = h^{-s/\alpha} \Gamma(1 + s/\alpha),$$

de sorte que :

$$\begin{aligned} K_{\ln T}(s) &= \ln M_{\ln T}(s) \\ &= -\frac{s}{\alpha} \ln h + \ln \Gamma(1 + s/\alpha), \end{aligned}$$

d'où les dérivées :

$$\begin{aligned} K'_{\ln T}(s) &= \frac{1}{\alpha} \left( -\ln h + \frac{\Gamma'(1 + s/\alpha)}{\Gamma(1 + s/\alpha)} \right) \\ \Rightarrow K'_{\ln T}(0) &= \frac{1}{\alpha} (-\ln h + \Gamma'(1)) \\ \Leftrightarrow E(\ln T) &= -\frac{1}{\alpha} (\ln h + \gamma_E), \end{aligned}$$

et que :

$$\begin{aligned} K''_{\ln T}(s) &= \frac{1}{\alpha^2} (\Gamma''(1) - \Gamma'(1)^2) \\ \Leftrightarrow K''_{\ln T}(0) &= \frac{1}{\alpha^2} (\Gamma''(1) - \Gamma'(1)^2) \\ \Leftrightarrow V(\ln T) &= \frac{\pi^2}{6\alpha^2}. \end{aligned}$$

### B.1.9 Loi Gamma généralisée

Pour définir le modèle Gamma généralisé, on suppose que  $hT^\alpha = V$  suit une loi Gamma à un paramètre, d'espérance  $\beta$ . Les moments de cette loi peuvent être trouvés directement en utilisant :

$$E(V^j) = \frac{\Gamma(j + \beta)}{\Gamma(\beta)},$$

ce qui implique :

$$\begin{aligned} E((hT^\alpha)^j) &= \frac{\Gamma(j + \beta)}{\Gamma(\beta)} \\ \Leftrightarrow E(T^{\alpha j}) &= h^{-j} \frac{\Gamma(j + \beta)}{\Gamma(\beta)}, \end{aligned}$$

il suffit alors de poser  $s = \alpha j$  ( $\Leftrightarrow j = s/\alpha$ ) pour obtenir :

$$E(T^s) = h^{-s/\alpha} \frac{\Gamma(s/\alpha + \beta)}{\Gamma(\beta)},$$

ce qui permet d'obtenir les moments non centrés :

$$E(T) = \frac{h^{-1/\alpha} \Gamma(1/\alpha + \beta)}{\Gamma(\beta)} \text{ et } E(T^2) = \frac{h^{-2/\alpha} \Gamma(2/\alpha + \beta)}{\Gamma(\beta)},$$

dont on déduit la variance par la formule classique :

$$\begin{aligned} V(T) &= E(T^2) - E(T)^2 \\ &= h^{-2/\alpha} \left\{ \frac{\Gamma(2/\alpha + \beta)}{\Gamma(\beta)} - \left( \frac{\Gamma(1/\alpha + \beta)}{\Gamma(\beta)} \right)^2 \right\}. \end{aligned}$$

Pour obtenir les moments du logarithme de la variable de durée, il suffit de remarquer que la fonction  $E(T^s)$  est identique à  $M_{\ln T}(s)$  de sorte que l'on peut écrire la fonction génératrice des moments centrés :

$$\begin{aligned} K_{\ln T}(s) &= \ln M_{\ln T}(s) \\ &= -\frac{s}{\alpha} \ln h + \ln \Gamma(s/\alpha + \beta) - \ln \Gamma(\beta), \end{aligned}$$

d'où l'espérance du logarithme :

$$\begin{aligned} K'_{\ln T}(s) &= -\frac{1}{\alpha} \ln h + \frac{1}{\alpha} \frac{\Gamma'(s/\alpha + \beta)}{\Gamma(s/\alpha + \beta)} \\ \Leftrightarrow K'_{\ln T}(0) &= E(\ln T) = \frac{1}{\alpha} \left( -\ln h + \frac{\Gamma'(\beta)}{\Gamma(\beta)} \right) = \frac{1}{\alpha} (-\ln h + \psi^{(0)}(\beta)), \end{aligned}$$

et sa variance :

$$K''_{\ln T}(s) = \frac{1}{\alpha^2} \left[ \frac{\Gamma''(s/\alpha + \beta)}{\Gamma(s/\alpha + \beta)^2} - \left( \frac{\Gamma'(s/\alpha + \beta)}{\Gamma(s/\alpha + \beta)} \right)^2 \right]$$

$$\Leftrightarrow K''_{\ln T}(0) = V(\ln T) = \frac{1}{\alpha^2} \left[ \frac{\Gamma''(\beta)}{\Gamma(\beta)^2} - \left( \frac{\Gamma'(\beta)}{\Gamma(\beta)} \right)^2 \right] = \frac{\psi^{(1)}}{\alpha^2}.$$

### B.1.10 Loi normale

La loi log-normale n'admet pas de fonction génératrice des moments parce que l'intégrale qui la définit n'est pas convergente :

$$K_T(s) = +\infty,$$

mais on peut calculer tous les moments de cette loi en utilisant la fonction génératrice des moments de loi normale. C'est ce qui explique la présence de cette section. Soit une variable aléatoire  $X$  suivant une loi normale  $N(m, \sigma^2)$ , sa fonction génératrice des moments peut être obtenue de la manière suivante :

$$K_X(s) = E(e^{sX})$$

$$= \int_{-\infty}^{+\infty} e^{sx} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-m}{\sigma}\right)^2} dx,$$

on effectue donc le changement de variable  $z = (x - m) / \sigma$ , ce qui ne change pas la valeur des bornes et implique  $x = m + \sigma z$  et  $dx = \sigma dz$ . On a donc :

$$M_X(s) = \int_{-\infty}^{+\infty} e^{s(m+\sigma z)} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}z^2} \sigma dz$$

$$= e^{sm} \int_{-\infty}^{+\infty} e^{s\sigma z} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}z^2} dz$$

$$= e^{sm} \int_{-\infty}^{+\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}(z^2 - 2s\sigma z)} dz$$

$$= e^{sm + s^2\sigma^2/2} \int_{-\infty}^{+\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}(z^2 - 2s\sigma z + s^2\sigma^2)} dz$$

$$= e^{sm + s^2\sigma^2/2} \int_{-\infty}^{+\infty} \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}(z - s\sigma)^2} dz,$$

et le second terme de l'expression ci-dessus est la somme sur son support de la densité d'une loi normale  $N(s\sigma, 1)$ , qui est égale à 1 par définition. On a donc :

$$M_X(s) = \exp(sm + s^2\sigma^2/2).$$

On peut en déduire tous les moments de la loi normale. Pour obtenir les moments centrés, on prend :

$$\begin{aligned} K_X(s) &= \ln M_X(s) \\ &= sm + s^2\sigma^2/2, \end{aligned}$$

on en déduit :

$$\begin{aligned} K'_X(s) &= m + s\sigma^2 \\ \Rightarrow K'_X(0) &= E(X) = m, \end{aligned}$$

ainsi que :

$$\begin{aligned} K''_X(s) &= \sigma^2 \\ \Rightarrow K''_X(0) &= V(X) = \sigma^2. \end{aligned}$$

### B.1.11 Loi log-normale

Par définition, la fonction génératrice du logarithme de la variable de durée  $\ln T$  est identique à celle de la loi normale  $N(m, \sigma^2)$  donnée par :

$$K_X(s) = K_{\ln T}(s) = E\left(e^{s \ln T}\right) = E(T^s),$$

on en déduit les moments de la loi log-normale :

$$E(T^s) = \exp(sm + s^2\sigma^2/2),$$

ce qui donne pour espérance :

$$E(T) = \exp(m + \sigma^2/2),$$

et pour variance :

$$\begin{aligned} V(T) &= E(T^2) - E(T)^2 \\ &= \exp(2m + 2\sigma^2) - \exp(2m + \sigma^2) \\ &= \exp(2m + \sigma^2)(\exp(\sigma^2) - 1). \end{aligned}$$

TABLEAU B.1 – MODÈLES DE DURÉE EN LOGARITHMES : ÉCRITURE

| Modèle           | Hypothèses                                                                          |
|------------------|-------------------------------------------------------------------------------------|
| Exponentiel      | $k_1 = -\ln h, k_2 = 1, \exp(U) \rightsquigarrow G(1, 1)$                           |
| Weibull          | $k_1 = -\alpha^{-1} \ln h, k_2 = \alpha^{-1}, \exp(U) \rightsquigarrow G(1, 1)$     |
| Gamma            | $k_1 = -\ln h, k_2 = 1, \exp(U) \rightsquigarrow G(\beta, 1)$                       |
| Gamma généralisé | $k_1 = -\alpha^{-1} \ln h, k_2 = \alpha^{-1}, \exp(U) \rightsquigarrow G(\beta, 1)$ |
| Log-Normal       | $k_1 = m, k_2 = \sigma, \exp(U) \rightsquigarrow \text{LN}(0, 1)$                   |

TABLEAU B.2 – MODÈLES DE DURÉE EN LOGARITHMES : MOMENTS DE  $U$ 

| Modèle           | $E(U)$                         | $V(U)$                                                             |
|------------------|--------------------------------|--------------------------------------------------------------------|
| Exponentiel      | $-\gamma_E$                    | $\pi^2/6$                                                          |
| Weibull          | $-\gamma_E$                    | $\pi^2/6$                                                          |
| Gamma            | $\Gamma'(\beta)/\Gamma(\beta)$ | $\Gamma''(\beta)/\Gamma(\beta) - (\Gamma'(\beta)/\Gamma(\beta))^2$ |
| Gamma généralisé | $\Gamma'(\beta)/\Gamma(\beta)$ | $\Gamma''(\beta)/\Gamma(\beta) - (\Gamma'(\beta)/\Gamma(\beta))^2$ |
| Log-Normal       | 0                              | 1                                                                  |

### B.1.12 Résumé

Le tableau B.1 résume les hypothèses qu'il faut effectuer pour retrouver chacun des modèles à partir de la relation suivante :

$$\ln T = k_1 + k_2 U,$$

le lecteur notera que les variables explicatives influençant la fonction de hasard ou la durée moyenne se trouvent dans la partie  $k_1$  de ce modèle. Le tableau B.2 résume l'espérance et la variance du terme d'erreur, et le tableau B.3 la perturbation complète du modèle.

## B.2 Approche par le calcul direct

Dans cette section, nous allons illustrer l'approche par le calcul direct pour les distributions de Pareto, log-logistique et de Burr de type 12 (aussi appelée distribution de Singh-Maddala). Dans le cas de la distribution de Pareto, la fonction génératrice des moments n'est définie que pour des valeurs négatives

TABLEAU B.3 – MODÈLES DE DURÉE EN LOGARITHMES : MOMENTS DE  $k_2U$ 

| Modèle           | $E(k_2U)$                                 | $V(k_2U)$                                                                         |
|------------------|-------------------------------------------|-----------------------------------------------------------------------------------|
| Exponentiel      | $-\gamma_E$                               | $\pi^2/6$                                                                         |
| Weibull          | $-\gamma_E/\alpha$                        | $\pi^2/(6\alpha^2)$                                                               |
| Gamma            | $\Gamma'(\beta)/\Gamma(\beta)$            | $\Gamma''(\beta)/\Gamma(\beta) - (\Gamma'(\beta)/\Gamma(\beta))^2$                |
| Gamma généralisé | $\alpha^{-1}\Gamma'(\beta)/\Gamma(\beta)$ | $\alpha^{-2}\{\Gamma''(\beta)/\Gamma(\beta) - (\Gamma'(\beta)/\Gamma(\beta))^2\}$ |
| Log-Normal       | 0                                         | $\sigma^2$                                                                        |

tives de la variable aléatoire, ce qui ne permet pas de l'utiliser pour les modèles de durée.

### B.2.1 Loi de Pareto

La densité de la loi de Pareto est définie par :

$$f_T(T) = ha^{a+1} (a + ht)^{-(a+1)}.$$

Nous allons calculer  $E(T)$  puis  $E(T^k)$  en fonction de  $E(T^{k-1})$ . En utilisant cette récurrence, nous obtiendrons l'expression de tous les moments non centrés. On a :

$$E(T) = \int_0^{+\infty} ha^{a+1} (a + ht)^{-(a+1)} t dt,$$

on effectue donc une intégration par partie  $\int uv' = [uv] - \int u'v$ , avec :

$$u = t, u' = 1, v' = (a + ht)^{-(a+1)}, v = -\frac{1}{ah} (a + ht)^{-a},$$

ce qui donne :

$$E(T) = ha^{a+1} \left\{ \left[ -\frac{t}{ah} (a + ht)^{-a} \right]_0^{+\infty} + \frac{1}{ah} \int_0^{+\infty} (a + ht)^{-a} dt \right\},$$

le terme entre crochets ne converge que si  $a > 1$ , et dans ce cas il est égal à zéro. L'espérance mathématique n'existe donc que si  $a > 1$ . On obtient (en utilisant  $a > 1$ ) :

$$\begin{aligned} E(T) &= a^a \left[ -\frac{1}{(a-1)h} (a + ht)^{1-a} \right]_0^{+\infty} \\ &= \frac{a}{(a-1)h}, \end{aligned} \quad (\text{B.1})$$

on vérifie que l'on retrouve le cas de la distribution exponentielle lorsque  $a \rightarrow +\infty$ .

En utilisant la même méthode, nous allons exprimer  $E(T^k)$  en fonction de  $E(T^{k-1})$  afin de compléter la récurrence. On a :

$$E(T^k) = \int_0^{+\infty} h a^{a+1} (a+ht)^{-(a+1)} t^k dt,$$

on effectue une intégration par parties avec :

$$u = t^k, u' = k t^{k-1}, v' = (a+ht)^{-(a+1)}, v = -\frac{1}{ah} (a+ht)^{-a},$$

ce qui donne :

$$E(T^k) = h a^{a+1} \left\{ \left[ -\frac{t^k}{ah} (a+ht)^{-a} \right]_0^{+\infty} + \frac{k}{ah} \int_0^{+\infty} t^{k-1} (a+ht)^{-a} dt \right\},$$

le terme entre crochets tend vers 0 si  $a > k$ . Le moment d'ordre  $k$  existe donc quand cette condition est remplie. Il reste :

$$E(T^k) = \frac{k h a^{a+1}}{h a} \int_0^{+\infty} t^{k-1} (a+ht)^{-a} dt,$$

en développant le terme dans l'intégrale et en réarrangeant, on obtient :

$$\begin{aligned} E(T^k) &= \frac{k}{h a} \left\{ \int_0^{+\infty} h a^{a+1} (a+ht)^{-(a+1)} (a+ht) t^{k-1} dt \right\} \\ &= \frac{k}{h a} \left\{ a \int_0^{+\infty} f_T(t) t^{k-1} dt + h \int_0^{+\infty} f_T(t) t^k dt \right\} \\ &= \frac{k}{h} E(T^{k-1}) + \frac{k}{a} E(T^k) \end{aligned}$$

dont on déduit :

$$E(T^k) = \frac{k}{h} \frac{a}{a-k} E(T^{k-1}),$$

en combinant cette relation avec la relation (B.1), on obtient :

$$E(T^k) = \left(\frac{a}{h}\right)^k \frac{k!}{\prod_{j=1}^k (a-j)}, \quad a > k.$$

Il existe une écriture plus pratique pour la programmation. On remarque que :

$$k! = \Gamma(k+1)$$

et que :

$$(a-1)! = (a-1)(a-2)\dots(a-s)(a-k-1)!$$

donc :

$$\prod_{j=1}^k (a-j) = \frac{(a-1)!}{(a-k-1)!} = \frac{\Gamma(a)}{\Gamma(a-k)},$$

d'où :

$$E(T^k) = \left(\frac{a}{h}\right)^k \frac{\Gamma(a-k)\Gamma(k+1)}{\Gamma(a)}, \quad a > k.$$

### B.2.2 Loi log-logistique

La densité de la loi logistique est donnée par :

$$f_T(t) = \frac{\alpha h t^{\alpha-1}}{(1+ht^\alpha)^2},$$

et nous allons procéder directement par changement de variable car il n'est pas nécessaire d'utiliser une récurrence. Par contre, il est nécessaire de se ramener à la fonction Beta. Nous cherchons donc :

$$\begin{aligned} E(T^k) &= \int_0^{+\infty} \frac{\alpha h t^{\alpha-1}}{(1+ht^\alpha)^2} t^k dt \\ &= \alpha \int_0^{+\infty} t^{k-1} \frac{1}{1+ht^\alpha} \frac{ht^\alpha}{1+ht^\alpha} dt \end{aligned}$$

et l'on fait le changement de variable :

$$u = \frac{1}{1+ht^\alpha}, \quad t = \left(\frac{1-u}{hu}\right)^{1/\alpha}$$

donc :

$$dt = -\frac{1}{\alpha} \left(\frac{1}{h}\right)^{1/\alpha} \frac{1}{u^2} \left(\frac{1-u}{u}\right)^{1/\alpha-1} du$$

et :

$$\lim_{t \rightarrow 0} u = 1, \quad \lim_{t \rightarrow +\infty} u = 0,$$

les bornes de l'intégrale sont donc inversées, ce qui change son signe :

$$\begin{aligned} E(T^k) &= \int_0^1 \left(\frac{1}{h} \times \frac{1-u}{u}\right)^{(k-1)/\alpha} u(1-u) \left(\frac{1}{h}\right)^{1/\alpha} \frac{1}{u^2} \left(\frac{1-u}{u}\right)^{1/\alpha-1} du \\ &= \left(\frac{1}{h}\right)^{k/\alpha} \int_0^1 u^{-k/\alpha} (1-u)^{k/\alpha} dt \end{aligned}$$

donc :

$$E(T^k) = \left(\frac{1}{h}\right)^{k/\alpha} B(1-k/\alpha, 1+k/\alpha), \quad \alpha > k.$$

### B.2.3 Loi de Burr de type 12

On applique la même méthode que pour la loi log-logistique. La densité est donnée par :

$$f_T(t) = h\alpha t^{\alpha-1} \left( \frac{a}{a+ht^\alpha} \right)^{a+1},$$

donc :

$$\begin{aligned} E(T^k) &= \alpha a^{a+1} \int_0^{+\infty} t^{k-1} \left( \frac{1}{a+ht^\alpha} \right)^a \frac{ht^\alpha}{a+ht^\alpha} dt \\ &= \alpha a \int_0^{+\infty} t^{k-1} \left( \frac{1}{1+(h/a)t^\alpha} \right)^a \frac{(h/a)t^\alpha}{1+(h/a)t^\alpha} dt \end{aligned}$$

et l'on effectue le changement de variable :

$$u = \frac{1}{1+(h/a)t^\alpha}, \quad t = \left( \frac{a}{h} \frac{1-u}{u} \right)^{1/\alpha}$$

donc :

$$dt = -\frac{1}{\alpha} \left( \frac{a}{h} \right)^{1/\alpha} \left( \frac{1}{u^2} \right) \left( \frac{1-u}{u} \right)^{1/\alpha-1} du$$

et :

$$\lim_{t \rightarrow 0} u = 1, \quad \lim_{t \rightarrow +\infty} u = 0$$

les bornes de l'intégrale sont donc inversées, ce qui change son signe :

$$\begin{aligned} E(T^k) &= a \int_0^{+\infty} \left( \frac{a}{h} \times \frac{1-u}{u} \right)^{(k-1)/\alpha} u^a (1-u) \left( \frac{a}{h} \right)^{1/\alpha} \left( \frac{1}{u^2} \right) \left( \frac{1-u}{u} \right)^{1/\alpha-1} dt \\ &= a \left( \frac{a}{h} \right)^{k/\alpha} \int_0^{+\infty} u^{-k/\alpha+a-1} (1-u)^{k/\alpha} dt \end{aligned}$$

donc :

$$E(T^k) = a \left( \frac{a}{h} \right)^{k/\alpha} B(a - k/\alpha, 1 + k/\alpha), \quad a\alpha > k$$

On doit retrouver le cas log-logistique quand  $a = 1$ , on obtient :

$$E(T^k) = \left( \frac{1}{h} \right)^{k/\alpha} B(1 - k/\alpha, 1 + k/\alpha),$$

et l'on doit retrouver le cas Weibull quand  $a \rightarrow +\infty$ . Pour cela, on utilise la propriété suivante de la fonction Beta (Konnully, 1968, p. 37) :

$$\lim_{q \rightarrow +\infty} q^p B(p, q) = \lim_{q \rightarrow +\infty} q^p B(q, p) = \Gamma(p),$$

on fait le changement de paramètres :

$$q = a - k/\alpha, \quad p = 1 + k/\alpha$$

ce qui donne :

$$a = p + q - 1, \quad \alpha = \frac{k}{p-1}$$

d'où :

$$\begin{aligned} \lim_{a \rightarrow +\infty} E(T^k) &= \lim_{a \rightarrow +\infty} h^{-k/\alpha} a^{1+k/\alpha} B(a - k/\alpha, 1 + k/\alpha) \\ &= h^{1-p} \lim_{q \rightarrow +\infty} (p + q - 1)^p B(q, p) \\ &= h^{1-p} \lim_{q \rightarrow +\infty} \left( \frac{p + q - 1}{q} \right)^p q^p B(q, p) \\ &= h^{1-p} \underbrace{\lim_{q \rightarrow +\infty} \left( 1 + \frac{p-1}{q} \right)^p}_1 \lim_{q \rightarrow +\infty} q^p B(q, p) \\ &= h^{1-p} \Gamma(p) \\ &= h^{-k/\alpha} \Gamma(1 + k/\alpha). \end{aligned}$$

## Annexe C

# Modélisation en logarithmes

Les variables de durée peuvent toujours être prises en logarithmes, ce qui facilite l'interprétation des résultats quand les variables explicatives sont elles-mêmes en logarithmes ou sous forme d'incatrices. Mais cette modélisation peut également être utilisée pour mieux comprendre les relations entre les différentes loi usuelles, et notamment les loi exponentielles, de Weibull, Gamma et Gamma généralisée.

Deux types d'écriture coexistent. Dans la première version, on écrit le modèle sous une forme logarithmique,  $\ln T$ , et on cherche la loi de la perturbation qui donne la distribution souhaitée pour  $T$ . La perturbation qui intervient sur le logarithme sera notée  $U$ . Dans la deuxième version, on écrit le modèle en niveau, sur  $T$ , et on cherche la distribution de la perturbation qui donne la loi désirée. La perturbation qui intervient sur le niveau sera notée  $V$ . On peut donc définir le même modèle de deux manières différentes. Ainsi, le modèle exponentiel peut se définir par rapport à la loi de Gumbel (par rapport à  $U$ ) ou directement par rapport à une loi exponentielle  $G(1, 1)$ , par rapport à  $V$ .

### C.1 Distributions à un paramètre

#### C.1.1 Modèle exponentiel et loi de Gumbel

Considérons le modèle suivant :

$$\ln T = -\ln h + U,$$

où  $U$  est une variable aléatoire dont on cherche la loi. On sait seulement que la durée  $T$  suit une loi exponentielle  $G(1, h)$  de densité :

$$f_T(t) = h \exp(-ht).$$

Pour trouver la densité de la loi de  $U$ , on remarque que :

$$U = \ln(hT) = g(T),$$

de sorte que :

$$T = \frac{1}{h} \exp(U) = g^{-1}(U)$$

ce qui implique :

$$\frac{dg^{-1}(u)}{du} = \frac{1}{h} \exp(u),$$

d'où la densité :

$$\begin{aligned} f_U(u) &= \frac{1}{h} \exp(u) h \exp \left[ -h \left( \frac{1}{h} \exp(u) \right) \right] \\ &= \exp(u) \exp(-\exp(u)), \end{aligned}$$

qui n'est autre que la densité d'une loi de Gumbel (i.e. valeur extrême de type I, minimum) d'espérance égale à l'opposée de la constante d'Euler,  $E(U) = -\gamma_E$ , avec  $\gamma_E \approx 0,57721$ , et de variance  $\pi^2/6$ . Pour trouver directement ces résultats on peut utiliser les deux propriétés suivantes de la constante d'Euler :<sup>1</sup>

$$\gamma_E = - \int_0^{+\infty} (\ln x) e^{-x} dx,$$

et

$$\gamma_E^2 + \frac{\pi^2}{6} = \int_0^{+\infty} (\ln x)^2 e^{-x} dx.$$

On utilise également la propriété suivante de la fonction Gamma :

$$\Gamma(p) = \int_0^{+\infty} x^{p-1} e^{-x} dx \Rightarrow \Gamma'(p) = \int_0^{+\infty} (\ln x) x^{p-1} e^{-x} dx, \quad (\text{C.1})$$

car on dérive par rapport à  $p$  et non par rapport à  $x$ .<sup>2</sup> Ceci implique :

$$\Gamma'(1) = \int_0^{+\infty} (\ln x) e^{-x} dx = -\gamma_E.$$

1. Pour évaluer la constante d'Euler, on peut utiliser la définition sous forme de série :

$$\gamma_E = \sum_{k=1}^{+\infty} \left[ \frac{1}{k} - \ln \left( 1 + \frac{1}{k} \right) \right].$$

2. On a :

$$\frac{d}{dp} (x^{p-1}) = \frac{d}{dp} (e^{(p-1)\ln x}) = (\ln x) e^{(p-1)\ln x} = (\ln x) x^{p-1}.$$

En dérivant une nouvelle fois la relation (C.1) par rapport à  $p$ , on obtient :

$$\Gamma''(p) = \int_0^{+\infty} (\ln x)^2 x^{p-1} e^{-x} dx,$$

ce qui implique :

$$\Gamma''(1) = \gamma_E^2 + \frac{\pi^2}{6},$$

On retrouve les valeurs  $\Gamma'(1)$  et  $\Gamma''(1)$  lors de l'étude de la fonction génératrice des moments de la loi de Gumbel. Finalement, on peut réécrire le modèle exponentiel sous la forme :

$$\begin{aligned} E(\ln T) &= -\ln h + E(U) \\ &= -(\ln h + \gamma_E), \end{aligned}$$

de sorte qu'en mettant des variables explicatives dans la fonction de hasard, on peut aboutir à un modèle log-linéaire avec une simple correction pour le terme constant du modèle. On remarque également que plus le taux de hasard  $h$  est élevé, plus l'espérance de durée est faible.

### C.1.2 Modèle exponentiel et loi exponentielle

On peut également définir le modèle exponentiel directement en niveaux et non en logarithmes. C'est l'approche qui est suivie habituellement pour généraliser ce modèle vers les modèles de Weibull, Gamma à deux paramètres et Gamma généralisé. De manière cohérente avec la section précédente, on pose :

$$T = g(V) = \frac{V}{h} \Leftrightarrow \ln T = -\ln h + \ln V \quad (\text{C.2})$$

où  $V$  suit une loi exponentielle de paramètre 1, notée  $G(1, 1)$ , de densité :

$$f_V(v) = \exp(-v).$$

La densité de la variable de durée tirée de ce modèle est donnée par la transformation :

$$\begin{aligned} T = \frac{V}{h} \Leftrightarrow V = hT = g^{-1}(T) \\ \Rightarrow \frac{dg^{-1}(t)}{dt} = h, \end{aligned}$$

d'où la densité :

$$f_T(t) = \left| \frac{dg^{-1}(t)}{dt} \right| f_V(g^{-1}(t)) = h \exp(-ht),$$

qui correspond à la densité de la loi exponentielle de hasard  $h$ .

### C.1.3 Modèle Gamma à un paramètre

On prend simplement :

$$T = V \Leftrightarrow \ln T = \ln V,$$

où  $V$  suit une loi Gamma d'espérance  $\beta$  dont la densité est donnée par :

$$f_V(v) = \frac{v^{\beta-1} \exp(-v)}{\Gamma(\beta)},$$

et comme  $T = V$  il s'agit de la loi de la durée. Si l'on souhaite connaître la loi du logarithme, on utilise :

$$U = g(V) = \ln V, \quad g^{-1}(V) = \exp(U),$$

ce qui donne

$$\frac{dg^{-1}}{du}(u) = \exp(u),$$

et la densité est donnée par :

$$\begin{aligned} f_U(u) &= \left| \frac{dg^{-1}(u)}{du} \right| f_V(g^{-1}(u)) \\ &= \frac{\exp(\beta u) \exp(-\exp(u))}{\Gamma(\beta)}. \end{aligned}$$

Cette densité généralise la loi de Gumbel, que l'on obtient pour  $\beta = 1$ . Toutefois, nous n'avons pas de paramètre  $h$  dans ce modèle, donc il ne généralise pas le modèle exponentiel. Pour obtenir une généralisation, il faut une loi Gamma à deux paramètres qui combine la présence d'un terme en  $-\ln h$  et la distribution ci-dessus.

## C.2 Distributions à deux paramètres

### C.2.1 Modèle de Weibull

Il existe différentes manières de généraliser le modèle exponentiel (C.2). Une première manière consiste à introduire un paramètre d'échelle  $\alpha > 0$  dans la définition de la variable de durée :

$$T = g(V) = \left(\frac{V}{h}\right)^{1/\alpha} \Leftrightarrow \ln T = -\frac{1}{\alpha} \ln h + \frac{1}{\alpha} \ln V \quad (\text{C.3})$$

on retrouve le modèle exponentiel de hasard  $h$  pour  $\alpha = 1$ . On suppose toujours que  $V$  suit une loi exponentielle de hasard 1. La loi suivie par  $T$  a donc changé puisque l'on a :

$$\begin{aligned} T &= \left(\frac{V}{h}\right)^{1/\alpha} \Leftrightarrow V = hT^\alpha = g^{-1}(T) \\ &\Rightarrow \frac{dg^{-1}(t)}{dt} = h\alpha t^{\alpha-1}, \end{aligned}$$

de sorte que la densité de  $T$  s'écrit :

$$\begin{aligned} f_T(t) &= \left| \frac{dg^{-1}(t)}{dt} \right| f_V(g^{-1}(t)) \\ &= \alpha h t^{\alpha-1} \exp(-ht^\alpha), \end{aligned}$$

qui correspond à la densité d'une variable de Weibull. On remarque qu'en logarithme la relation peut s'écrire :

$$\begin{aligned} \ln T &= \frac{1}{\alpha} (-\ln h + \ln V) \\ &= \frac{1}{\alpha} (-\ln h + U), \end{aligned}$$

de sorte qu'avec nos notations  $\alpha$  est un paramètre d'échelle qui porte sur l'ensemble du modèle. L'espérance mathématique correspondante s'écrit :

$$E(\ln T) = -\frac{1}{\alpha} (\ln h + \gamma_E),$$

car  $U$  suit toujours une loi exponentielle de hasard 1 comme dans le modèle exponentiel.

### C.2.2 Modèle Gamma à deux paramètres

Le modèle Gamma généralise le modèle exponentiel (C.2) en changeant la distribution du terme d'erreur  $V$  au lieu d'introduire un paramètre d'échelle. On suppose, comme dans le modèle exponentiel, que :

$$T = g(V) = \frac{V}{h},$$

mais cette fois-ci  $V$  suit une loi Gamma d'espérance  $\beta$ . La densité de  $V$  est donc donnée par :

$$f_V(v) = \frac{v^{\beta-1} e^{-v}}{\Gamma(\beta)}, \quad \beta > 0, \quad v > 0,$$

et l'on retrouve le modèle exponentiel en posant  $\beta = 1$  et le modèle Gamma à un paramètre en posant  $h = 1$ . En utilisant  $g^{-1}(t) = ht$ , on obtient :

$$\begin{aligned} f_T(t) &= \left| \frac{dg^{-1}(t)}{dt} \right| f_V(g^{-1}(t)) \\ &= h \frac{(ht)^{\beta-1} e^{-ht}}{\Gamma(\beta)} \\ &= \frac{h^\beta t^{\beta-1} e^{-ht}}{\Gamma(\beta)}, \end{aligned}$$

qui correspond à la densité d'une loi Gamma à deux paramètres. Écrit en espérance le modèle log linéaire donne :

$$E(\ln T) = -\ln h + E(\ln V).$$

### C.2.3 Modèle log-normal

On peut également utiliser la méthode du changement de variable pour le modèle log-normal, mais ici la modélisation ne porte pas sur le hasard mais sur l'espérance mathématique de la variable de durée, quantité qui est décroissante avec le taux de hasard. Dans un modèle avec des variables explicatives, un modèle basé sur l'espérance de la variable de durée implique généralement un changement de signe des coefficients par rapport à un modèle basé sur la fonction de hasard. On pose :

$$\ln T = m + \sigma U$$

où  $U$  suit une loi normale centrée et réduite, de sorte que  $\ln T$  suit une loi normale  $N(m, \sigma^2)$ . On a donc la transformation suivante :

$$T = \exp(m + \sigma U) = g(U) \Leftrightarrow U = \frac{\ln T - m}{\sigma} = g^{-1}(T),$$

ce qui implique :

$$\frac{dg^{-1}(t)}{dt} = \frac{1}{\sigma t}.$$

D'autre part,  $U$  suit une loi normale de densité  $\varphi(u)$ , ce qui implique que la densité de  $T$  (i.e. de la loi log-normale) est donnée par :

$$f_T(t) = \left| \frac{dg^{-1}(t)}{dt} \right| \varphi(g^{-1}(t)) = \frac{1}{\sigma t} \varphi\left(\frac{\ln t - m}{\sigma}\right).$$

Ici, par construction, l'espérance du modèle log linéaire se réduit à sa plus simple expression :

$$E(\ln T) = m + \sigma E(U) = m.$$

**C.2.4 Modèle log-logistique**

On reprend l'idée du modèle log-normal et on pose :

$$\ln T = m + \delta U,$$

où  $U$  suit une loi logistique  $L(0, 1)$  de densité :

$$f_U(u) = \frac{\exp(-u)}{(1 + \exp(-u))^2},$$

Cette loi est centrée mais pas réduite (car  $V(U) = \pi^2/3$ ). On utilise la même transformation que pour la loi normale :

$$g^{-1}(t) = \frac{\ln t - m}{\delta} \Rightarrow \frac{dg^{-1}}{dt}(t) = \frac{1}{\delta t}$$

on a donc :

$$\begin{aligned} f_T(t) &= \left| \frac{dg^{-1}(t)}{dt} \right| f_U(g^{-1}(t)) \\ &= \frac{1}{\delta t} \frac{\exp(-(\ln t - m)/\delta)}{(1 + \exp(-(\ln t - m)/\delta))^2}, \end{aligned}$$

à ce stade, on remarque que :

$$\exp(-(\ln t - m)/\delta) = \frac{\exp(m/\delta)}{t^{1/\delta}},$$

et l'on pose :

$$h = \exp(-m/\delta), \quad \alpha = 1/\delta$$

ce qui donne :

$$\exp(-(\ln t - m)/\delta) = \frac{1}{ht^\alpha},$$

d'où la densité de la variable de durée log-logistique :

$$f_T(t) = \frac{\alpha h t^{\alpha-1}}{(1 + ht^\alpha)^2}.$$

### C.2.5 Modèle de Pareto

Pour représenter le modèle de Pareto en logarithmes, le plus simple est de considérer sa fonction de survie. Soit la distribution :

$$S_T(t) = \left( \frac{a}{a+ht} \right)^a.$$

On a :

$$\begin{aligned} \Pr \left[ \ln \left( 1 + \frac{hT}{a} \right) > z \right] &= \Pr \left[ T > \frac{a}{h} (e^z - 1) \right] \\ &= \exp(-az) \end{aligned}$$

la fonction de survie de la loi exponentielle de paramètre  $a$ . Donc  $Z = \ln(1 + hT/a)$  suit une loi exponentielle de hasard  $a$ . On peut donc écrire :

$$\ln(1 + hT/a) = \frac{V}{a},$$

où  $V$  suit une loi exponentielle  $G(1, 1)$ . On voit que si  $a \rightarrow +\infty$ ,  $h/a \rightarrow 0$ , donc :

$$\ln(1 + hT/a) \simeq \frac{hT}{a}$$

et l'on se trouve dans la situation où :

$$T \simeq \frac{V}{h},$$

le cas exponentiel. La loi de Pareto généralise la loi exponentielle de la manière suivante :

$$T = \frac{a}{h} \left\{ \exp \left( \frac{V}{a} \right) - 1 \right\} > 0$$

car  $V/a > 0 \Rightarrow \exp(V/a) > 1$ .

### C.3 Modèle Gamma à trois paramètres

Il s'agit d'une troisième généralisation du modèle exponentiel (C.2). Cette fois-ci, nous allons combiner les deux généralisations du modèle de Weibull et du modèle Gamma. On suppose, comme dans le modèle de Weibull, que la variable de durée est définie par la relation :

$$T = g(V) = \left( \frac{V}{h} \right)^{1/\alpha},$$

et, comme dans le modèle Gamma, que  $V$  suit une loi Gamma d'espérance  $\beta$  de densité :

$$f_V(v) = \frac{v^{\beta-1} e^{-v}}{\Gamma(\beta)}, \quad \beta > 0, v > 0.$$

On trouve directement la nouvelle densité :

$$\begin{aligned} f_T(t) &= \left| \frac{dg^{-1}(t)}{dt} \right| f_V(g^{-1}(t)) \\ &= h\alpha t^{\alpha-1} \frac{(ht^\alpha)^{\beta-1} e^{-(ht^\alpha)}}{\Gamma(\beta)} \\ &= \frac{\alpha h^\beta t^{\alpha\beta-1} \exp(-ht^\alpha)}{\Gamma(\beta)}. \end{aligned}$$

On obtient les cas particuliers suivants :

- $\alpha = 1$  : loi Gamma ;
- $\beta = 1$  : loi de Weibull ;
- $\alpha = 1$  et  $\beta = 1$  : loi exponentielle ;
- $\beta \rightarrow +\infty$  : loi log-normale.

Le modèle log linéaire en espérance s'écrit maintenant :

$$E(\ln T) = \frac{1}{\alpha} (-\ln h + E(\ln V)),$$

où  $E(\ln V)$  prend la même valeur que pour le modèle Gamma de la section précédente.



## Annexe D

# Propriétés du modèle exponentiel

### D.1 Propriété utile pour la statistique du score

Pour interpréter la statistique du score, on a besoin de la propriété suivante de la loi exponentielle :

$$\text{Cov}(T, \ln T) = E(T).$$

Commençons par calculer :

$$E(T \ln T) = \int_0^{+\infty} t (\ln t) h \exp(-ht) dt,$$

en utilisant une intégration par parties,  $u = t \ln t$ ,  $v' = h \exp(-ht)$ , on obtient :

$$\begin{aligned} E(T \ln T) &= \underbrace{[t \ln t \times (-\exp(-ht))]_0^{+\infty}}_0 + \int_0^{+\infty} (1 + \ln t) \exp(-ht) dt \\ &= \int_0^{+\infty} \underbrace{\exp(-ht)}_{S_T(t)} dt + \frac{1}{h} \int_0^{+\infty} \ln t \times h \exp(-ht) dt \\ &= E(T) + \frac{1}{h} E(\ln T) \\ &= E(T) + E(T) E(\ln T), \end{aligned}$$

ce qui est équivalent à :

$$\underbrace{E(T \ln T) - E(T) E(\ln T)}_{\text{Cov}(T, \ln T)} = E(T).$$

## D.2 Matrice d'information

Pour calculer la matrice d'information du modèle exponentiel conditionnel, il faut faire attention au point suivant : les durées sont censurées, on ne peut donc pas supposer que  $y_i$  est la réalisation d'une variable aléatoire de loi exponentielle. Plus précisément,

$$Y_i = \min(T_i, \bar{t}_i) = \begin{cases} T_i & \text{si } T_i \leq \bar{t}_i \text{ (cas } d_i = 1) \\ \bar{t}_i & \text{si } T_i > \bar{t}_i \text{ (cas } d_i = 0) \end{cases},$$

où  $T_i$  suit une loi exponentielle de paramètre  $h_i$ . Pour calculer l'espérance conditionnelle de  $Y_i$ , il nous faut la quantité suivante :

$$E(T|T \leq c)$$

où  $c$  est un seuil connu et  $T$  une variable exponentielle de paramètre  $h$ . On a :

$$E(T|T \leq c) = \int_0^{+\infty} t f_T(t|t \leq c) dt \text{ avec } f_T(t|t \leq c) = \frac{f_T(t) \mathbf{1}_{(t \leq c)}}{\Pr[t \leq c]},$$

on a donc :

$$\begin{aligned} E(T|T \leq c) &= \int_0^{+\infty} \frac{t f_T(t) \mathbf{1}_{(t \leq c)}}{\Pr[t \leq c]} dt \\ &= \frac{1}{F_T(c)} \int_0^c t f_T(t) dt \\ &= \frac{1}{F_T(c)} \int_0^c h t \exp(-ht) dt, \end{aligned}$$

il suffit alors d'intégrer par parties :

$$\begin{aligned} E(T|T \leq c) &= \frac{1}{F_T(c)} \left\{ [-t \exp(-ht)]_0^c + \int_0^c \exp(-ht) dt \right\} \\ &= \frac{1}{F_T(c)} \left\{ -c \exp(-hc) + \left[ -\frac{1}{h} \exp(-ht) \right]_0^c \right\} \\ &= \frac{1}{F_T(c)} \left\{ -c \exp(-hc) - \frac{1}{h} \exp(-hc) + \frac{1}{h} \right\}, \end{aligned}$$

on remarque alors que  $S_T(c) = \exp(-hc)$  et que  $F_T(c) = 1 - S_T(c)$ , ce qui donne :

$$E(T|T \leq c) = \frac{1}{h} - \frac{c S_T(c)}{1 - S_T(c)},$$

dans notre cas, nous aurons besoin de l'espérance de notre variable de durée observée sachant qu'elle est complète ( $T_i \leq y_i$ ) :

$$E(Y_i | T_i \leq y_i) = E(T_i | T_i \leq y_i) = \frac{1}{h_i} - \frac{y_i S_T(y_i)}{1 - S_T(y_i)},$$

avec :

$$h_i = \exp(\tilde{X}_i \tilde{\beta}).$$

Dans l'ensemble, on a :

$$\begin{aligned} E(Y_i) &= E(T_i | T_i \leq y_i) \Pr[T_i \leq y_i] + \underbrace{E(Y_i | T_i > y_i)}_{y_i} \Pr[T_i > y_i] \\ &= \left( \frac{1}{h_i} - \frac{y_i S_T(y_i)}{1 - S_T(y_i)} \right) (1 - S_T(y_i)) + y_i S_T(y_i) \\ &= \frac{1 - S_T(y_i)}{h_i}, \end{aligned}$$

où  $S_T(y_i)$  représente la probabilité que l'observation  $i$  soit censurée. Quand les données ne sont pas censurées, on a  $\Pr[T_i > y_i] = 0 = S_T(y_i)$  et on retrouve l'espérance habituelle  $1/h_i$ . Pour une observation  $y_i$ , la probabilité est égale à :

$$\begin{aligned} S_T(y_i) &= \exp(-h_i y_i) \\ &= \exp(-\exp(\tilde{X}_i \tilde{\beta}) y_i), \end{aligned}$$

de sorte que :

$$\begin{aligned} E(Y_i) &= \frac{1 - S_T(y_i)}{h_i} \\ &= \frac{1}{\exp(\tilde{X}_i \tilde{\beta})} (1 - \exp(-\exp(\tilde{X}_i \tilde{\beta}) y_i)) \end{aligned}$$

On en déduit que :

$$E_y [\exp(\tilde{X}_i \tilde{\beta}) y_i] = 1 - \exp(-\exp(\tilde{X}_i \tilde{\beta}) y_i),$$

et l'information de Fisher que nous allons utiliser est donc égale à :

$$J(\tilde{\beta}) = \sum_{i=1}^N X_i' X_i (1 - \exp(-\exp(\tilde{X}_i \tilde{\beta}) y_i)).$$



## Annexe E

# Les algorithmes d'optimisation

### E.1 Présentation des algorithmes

Le problème que nous devons résoudre est de trouver une valeur numérique  $\hat{\theta}$  qui résout un système d'équations de la forme :<sup>1</sup> :

$$s(\hat{\theta}) = 0$$

où  $s(\cdot)$  est une fonction connue des observations et du paramètre  $\theta$ . Elle dépend aussi bien des données que du modèle postulé (distribution, paramètres). Dans le cas du maximum de vraisemblance  $s(\theta)$  est appelé le score, dans le cas du pseudo maximum de vraisemblance, on l'appelle le pseudo score. Quand la fonction à maximiser est concave, cette condition du premier ordre est suffisante pour un maximum global. La technique que l'on utilise pour parvenir à la valeur  $\hat{\theta}$  s'appelle un algorithme. On peut décomposer un algorithme en quatre grandes étapes.

1. *Une valeur initiale.* Le choix de la valeur initiale n'est pas problématique quand l'objectif est concave. Dans ce cas, avec un algorithme croissant (voir plus loin), tout point de départ doit mener au maximum. Par contre, quand l'objectif n'est pas concave, ou pour accélérer la procédure quand l'objectif est concave, on prendra un estimateur convergent comme point de départ. Il est en effet possible, dans certains cas, de trouver un estimateur en deux étapes relativement facile à calculer. Il n'est généralement pas efficace, et c'est la raison pour laquelle on réalise une estimation supplémentaire. On note cette valeur initiale  $\theta_{(0)}$ .

---

1. Un système d'équations car  $\hat{\theta}$  est un vecteur.

2. *Une règle d'itération.* Une fois la valeur initiale fixée, il faut utiliser une règle qui permette de trouver une nouvelle valeur plus proche du maximum. Le pas de l'itération, défini comme la différence entre deux valeurs successives du paramètre, est déterminé selon différentes méthodes et constitue le cœur de l'algorithme. Nous utiliserons des méthodes dites de gradient et plus particulièrement les algorithmes de Newton-Raphson, du Score et de Berndt-Hall-Hall-Hausman. La pratique montre qu'ils permettent de traiter la plupart des cas. On résume cette étape par la relation :

$$\theta_{(p+1)} = \theta_{(p)} + M(\theta_{(p)}),$$

où  $p$  est l'itération,  $p \in \mathbb{N}$ . La valeur du pas  $M(\theta_{(p)})$  doit dépendre uniquement de la valeur du paramètre à l'étape précédente.

3. *Une règle d'arrêt de l'algorithme.* Au voisinage du maximum la fonction objectif ne doit plus varier, on peut donc baser l'arrêt de la procédure sur la différence entre deux valeurs successives de la fonction objectif. Une seconde condition porte sur le gradient, qui doit être nul (condition du premier ordre). On peut aussi utiliser des variantes, comme l'élasticité de l'objectif aux paramètres du modèle, qui est également nulle à l'optimum et possède l'avantage d'être insensible à un changement d'unité des variables (contrairement au gradient). Dans l'ensemble, tous ces critères sont équivalents à l'optimum.
4. *Vérifier que l'on a bien atteint un maximum local.* Ce problème ne se pose que lorsque l'objectif n'est pas globalement concave. La condition du second ordre pour un optimum local précise que le hessien doit être défini négatif au point en question. Il faudra donc le vérifier systématiquement. Ceci est d'autant plus important que dans le cas du maximum de vraisemblance l'inverse du hessien n'est autre qu'un estimateur convergent de la matrice de covariance de l'estimateur du paramètre. En conséquence, si cette propriété n'était pas vérifiée, on obtiendrait un estimateur de la matrice de covariance qui n'est pas défini positif et, pour cette raison, inutilisable.

## E.2 Les méthodes de gradient

Nous voulons maximiser une fonction  $\ell(\theta)$ , de gradient  $s(\theta) = \partial\ell(\theta)/\partial\theta$  et de hessien  $H(\theta) = \partial^2\ell(\theta)/\partial\theta\partial\theta'$ . Nous disposons également d'une valeur initiale notée  $\theta_{(0)}$ . De même, on note  $\theta_{(p)}$  la valeur du paramètre à la  $p$ -ième itéra-

tion. Un algorithme du gradient est une règle d'itération de la forme suivante :

$$\theta_{(p+1)} = \theta_{(p)} + W_{(p)} s(\theta_{(p)}),$$

où  $s(\theta_{(p)})$  est le gradient de la fonction que l'on cherche à maximiser et  $W_{(p)}$  une matrice qui dépend de l'algorithme particulier que l'on emploie. On vérifie que lorsque l'on a atteint le maximum,  $s(\hat{\theta}) = 0$ , et le pas de l'itération est nul. Toutefois, dans la pratique, il peut arriver que le pas d'une itération soit trop fort et dépasse le point qui donne le maximum, on modifie donc la règle précédente en l'écrivant :

$$\theta_{(p+1)} = \theta_{(p)} + \lambda_p W_{(p)} s(\theta_{(p)}),$$

où  $\lambda_p \geq 0$ . Il est possible de prendre  $\lambda_p > 1$  au début de l'algorithme pour accélérer la convergence. Le lecteur intéressé par ce dernier cas peut consulter Gouriéroux et Monfort (1989, chap. XIII). La valeur de  $\lambda_p$  n'est réduite que lorsque  $\ell(\theta_{(p+1)}) < \ell(\theta_{(p)})$ .

Les algorithmes de gradient sont très employés car ils possèdent une propriété intéressante : ils sont croissants. Cette propriété signifie que, si  $W_{(p)}$  est symétrique et définie positive, alors pour de petits accroissements du pas de l'itération (i.e.,  $\lambda_p$  petit), l'algorithme mène toujours à une valeur supérieure ou égale de l'objectif soit  $\ell(\theta_{(p+1)}) \geq \ell(\theta_{(p)})$ . Quand la fonction est concave, ceci garantit que l'on parvient au maximum. Les trois algorithmes de gradient les plus utilisés sont ceux de Newton-Raphson, de Berndt-Hall-Hall-Hausman et du score.

### E.3 Algorithme de Newton-Raphson

Il consiste à effectuer une approximation quadratique de la fonction à maximiser, en chacun des points de l'itération. Dans ce cas, si le hessien est défini négatif, on obtient le maximum de l'approximation par la condition du premier ordre sur une forme quadratique, dont on peut calculer facilement l'expression analytique parce qu'elle est linéaire. La succession de maxima ainsi obtenue donne la solution du problème. Le développement limité au second ordre de  $\ell(\theta)$  au voisinage de  $\theta_{(p)}$  est égal à :

$$\ell(\theta) \simeq \ell(\theta_{(p)}) + s'(\theta_{(p)})(\theta - \theta_{(p)}) + \frac{1}{2}(\theta - \theta_{(p)})' H(\theta_{(p)}) (\theta - \theta_{(p)}).$$

La maximisation de cette forme quadratique par rapport à  $\theta$  donne la condition du premier ordre :

$$\begin{aligned} s(\theta_{(p)}) + H(\theta_{(p)})(\theta - \theta_{(p)}) &= 0 \\ \Leftrightarrow \theta - \theta_{(p)} &= -H(\theta_{(p)})^{-1} s(\theta_{(p)}), \end{aligned}$$

de plus la dérivée seconde est égale à  $H(\theta_{(p)})$ , qui est définie négative lorsque l'objectif est concave au point  $\theta_{(p)}$ . Dans ce cas, on a bien un maximum local donné par les conditions du premier ordre. Dans l'ensemble l'itération est donnée par :

$$\theta_{(p+1)} = \theta_{(p)} - \lambda_p H(\theta_{(p)})^{-1} s(\theta_{(p)}).$$

Cet algorithme présente généralement un pas assez fort dans les premières itérations et, dans l'ensemble, s'avère assez rapide. Tout dépend toutefois si l'on travaille sur des dérivées secondes analytiques ou numériques, car ces dernières augmentent fortement le temps de calcul à chaque itération.

#### E.4 Algorithme de Berndt-Hall-Hall-Hausman

Cet algorithme, justifié dans le cas du maximum de vraisemblance, se base sur l'égalité de la matrice d'information :

$$\begin{aligned} I_1(\theta) &= E_x E_y \left[ \frac{\partial \ln f(\theta)}{\partial \theta} \frac{\partial \ln f(\theta)}{\partial \theta'} \right] \\ &= E_x E_y \left[ - \frac{\partial^2 \ln f(\theta)}{\partial \theta \partial \theta'} \right] \\ &= J_1(\theta). \end{aligned}$$

Pour l'estimation, on remplace les moments théoriques par les moments empiriques correspondants, ce qui suggère d'approximer les dérivées secondes par l'opposé des produits croisés des dérivées premières. Pour un échantillon de taille  $N$  on peut écrire la fonction objectif et ses dérivées sous la forme :

$$\begin{aligned} \ell(\theta) &= \sum_{i=1}^N \ln f(y_i | x_i, \theta), \\ s(\theta) &= \sum_{i=1}^N \frac{\partial \ln f(y_i | x_i, \theta)}{\partial \theta} \\ \text{et } H(\theta) &= \sum_{i=1}^N \frac{\partial^2 \ln f(y_i | x_i, \theta)}{\partial \theta \partial \theta'}. \end{aligned}$$

La méthode de Berndt-Hall-Hall-Hausman consiste à approximer :

$$H(\theta) \text{ par } \sum_{i=1}^N \frac{\partial \ln f(y_i | x_i, \theta)}{\partial \theta} \frac{\partial \ln f(y_i | x_i, \theta)}{\partial \theta'},$$

ce qui donne la règle d'itération suivante :

$$\theta_{(p+1)} = \theta_{(p)} - \lambda_p \left[ \sum_{i=1}^N \frac{\partial \ln f(y_i|x_i, \theta)}{\partial \theta} \frac{\partial \ln f(y_i|x_i, \theta)}{\partial \theta'} \right]^{-1} s(\theta_{(p)}).$$

Cet algorithme ne nécessite que les dérivées au premier ordre et est donc facile à mettre en œuvre. Toutefois, il implique généralement plus d'itérations que l'algorithme de Newton-Raphson. Son principal défaut est qu'il ne permet pas de vérifier la négativité du hessien à chaque itération, ce qui peut s'avérer problématique en certains points  $\theta_{(p)}$  lorsque l'objectif n'est pas globalement concave.

## E.5 Algorithme du score

Il s'agit d'un raffinement de l'algorithme de Berndt-Hall-Hausman où l'on prend l'espérance mathématique des produits croisés du score, égale à l'information de Fisher dans le cas du maximum de vraisemblance, à la place de leurs produits croisés. On approxime :

$$H(\theta) \text{ par } \sum_{i=1}^N E_y \left[ \frac{\partial \ln f(y_i|x_i, \theta)}{\partial \theta} \frac{\partial \ln f(y_i|x_i, \theta)}{\partial \theta'} \right],$$

ce qui donne la règle d'itération :

$$\theta_{(p+1)} = \theta_{(p)} - \lambda_p \left\{ \sum_{i=1}^N E_y \left[ \frac{\partial \ln f(y_i|x_i, \theta)}{\partial \theta} \frac{\partial \ln f(y_i|x_i, \theta)}{\partial \theta'} \right] \right\}^{-1} s(\theta_{(p)}).$$

Dans le cas où les dérivées secondes ne dépendent pas de la variable endogène  $y$ , cet algorithme est identique à celui de Newton-Raphson.

## E.6 Méthodologie de programmation

Comment s'assurer que le programme que l'on a écrit ne comporte aucune erreur et qu'il soit pratique à utiliser ? En prenant un certain nombre de précautions présentées dans cette section. Il y a quatre étapes qu'il faut prendre soin de bien effectuer.

1. *Vérifier la fonction objectif.* A la fois par le calcul, mais également en consultant les ouvrages et les articles qui la donnent.

2. *Vérifier le gradient.* Par la même méthode que précédemment, mais également numériquement. Ainsi, on peut détecter des erreurs de recopie aussi bien sur la fonction objectif que sur son gradient. A cette étape, on utilise un algorithme basé sur les dérivées premières, de type Berndt-Hall-Hall-Hausman.
3. *Vérifier le hessien.* On utilise le gradient analytique pour calculer le hessien numérique, afin d'éviter le cumul des erreurs d'approximation. Ce problème est particulièrement sensible ici car les dérivées sont calculées à partir de quantités très petites par définition. On utilise un algorithme de Newton-Raphson. On peut conserver les dérivées secondes numériques si le calcul des dérivées analytiques est trop complexe.
4. *Paramétrer le programme définitif.* Ceci vise à éviter toute intervention sur le programme une fois qu'il a été vérifié, car c'est une source d'erreur potentielle. Par exemple, on peut paramétrer des programmes écrits en SAS-IML par des macro-variables, ce qui permet d'écrire des routines appelées macro-commandes. On peut alors utiliser ces routines sur toutes les bases de données et quel que soit le nombre de variables explicatives.

# Bibliographie

- [1] Konnully A. Relationship between the beta and the gamma functions. Mathematics Magazine, 41(1) :37–39, 1968.
- [2] Lawless J. F. Inference in the generalized gamma and log gamma distributions. Technometrics, 22(3) :409–419, 1980.
- [3] Lawless J. F. Statistical Models and Methods for Lifetime Data. Wiley Series in Probability and Statistics. Wiley, 2003.
- [4] Berndt E. K., Hall B. H., Hall R. E., and Hausman J. A. Estimation and inference in nonlinear structural models. Annals of Economic and Social Measurement, (3-4) :653–665, 1974.
- [5] Prentice R. L. A log gamma model and its maximum likelihood estimation. Biometrika, 61(3) :539–544, 1974.
- [6] Lomax K. M. Business failures : Another example of the analysis of failure data. Journal of the American Statistical Association, 49(268) :847–852, 1954.
- [7] Cox D. R. and Oakes D. Analysis of Survival Data. Monographs on Statistics and Applied Probability No 21. Chapman & Hall/CRC, 1984.
- [8] Tadikamalla P. R. A look at the burr and related distributions. International Statistical Review / Revue Internationale de Statistique, 48(3) :337–344, 1980.
- [9] Burr I. W. Cumulative frequency functions. The Annals of Mathematical Statistics, 13(2) :215–232, 1942.
- [10] Stacy E. W. A generalization of the gamma distribution. The Annals of Mathematical Statistics, 33(3) :1187–1192, 1962.



# Table des Graphiques

|      |                                                                                            |     |
|------|--------------------------------------------------------------------------------------------|-----|
| 2.1  | Relations entre les distributions . . . . .                                                | 12  |
| 2.2  | Fonction de hasard de la loi de Weibull ( $h = 1$ ) . . . . .                              | 17  |
| 2.3  | Loi normale, $\varphi(x)/(1 - \Phi(x))$ . . . . .                                          | 21  |
| 4.1  | Loi exponentielle, $\widehat{S}_T(t)$ et $\widehat{F}_T(t)$ . . . . .                      | 59  |
| 4.2  | Loi exponentielle, $\widehat{h}_T(t)$ et $\widehat{f}_T(t)$ . . . . .                      | 59  |
| 4.3  | Loi Gamma 1, $\widehat{S}_T(t)$ et $\widehat{F}_T(t)$ . . . . .                            | 76  |
| 4.4  | Loi Gamma 1, $\widehat{h}_T(t)$ et $\widehat{f}_T(t)$ . . . . .                            | 76  |
| 4.5  | Loi de Lomax, $\widehat{S}_T(t)$ et $\widehat{F}_T(t)$ . . . . .                           | 87  |
| 4.6  | Loi de Lomax, $\widehat{h}_T(t)$ et $\widehat{f}_T(t)$ . . . . .                           | 87  |
| 4.7  | Loi exponentielle à état absorbant, $\ell^*(\pi)$ . . . . .                                | 98  |
| 4.8  | Loi exponentielle à état absorbant, $\widehat{S}_\pi(t)$ et $\widehat{F}_\pi(t)$ . . . . . | 109 |
| 4.9  | Loi exponentielle à état absorbant, $\widehat{h}_\pi(t)$ et $\widehat{f}_\pi(t)$ . . . . . | 110 |
| 4.10 | Loi de Weibull, $\widehat{S}_T(t)$ et $\widehat{F}_T(t)$ . . . . .                         | 122 |
| 4.11 | Loi de Weibull, $\widehat{h}_T(t)$ et $\widehat{f}_T(t)$ . . . . .                         | 123 |
| 4.12 | Loi Gamma 2, $\widehat{S}_T(t)$ et $\widehat{F}_T(t)$ . . . . .                            | 136 |
| 4.13 | Loi Gamma 2, $\widehat{h}_T(t)$ et $\widehat{f}_T(t)$ . . . . .                            | 136 |
| 4.14 | Loi log-normale, $\widehat{S}_T(t)$ et $\widehat{F}_T(t)$ . . . . .                        | 148 |
| 4.15 | Loi log-normale, $\widehat{h}_T(t)$ et $\widehat{f}_T(t)$ . . . . .                        | 149 |
| 4.16 | Loi log-logistique, $\widehat{S}_T(t)$ et $\widehat{F}_T(t)$ . . . . .                     | 162 |
| 4.17 | Loi log-logistique, $\widehat{h}_T(t)$ et $\widehat{f}_T(t)$ . . . . .                     | 163 |
| 4.18 | Loi de Pareto, $\widehat{S}_T(t)$ et $\widehat{F}_T(t)$ . . . . .                          | 174 |
| 4.19 | Loi de Pareto, $\widehat{h}_T(t)$ et $\widehat{f}_T(t)$ . . . . .                          | 175 |
| 4.20 | Comparaison des fonctions de répartition pour $k = 200$ . . . . .                          | 181 |
| 4.21 | Loi Gamma généralisée, $\ell^*(q)$ . . . . .                                               | 186 |
| 4.22 | Loi Gamma généralisée, $\ell^*(q)$ , $0.005 \leq q \leq 0.110$ . . . . .                   | 186 |
| 4.23 | Loi de Burr, $\ell^*(a)$ , balayage de $a = 0$ à $a = 19$ . . . . .                        | 211 |
| 4.24 | Loi de Burr, $\ell^*(a)$ , balayage de $a = 0.1$ à $a = 1$ . . . . .                       | 212 |
| 4.25 | Loi de Burr, $\widehat{S}_T(t)$ et $\widehat{F}_T(t)$ . . . . .                            | 221 |

|      |                                                                                       |     |
|------|---------------------------------------------------------------------------------------|-----|
| 4.26 | Loi de Burr, $\hat{h}_T(t)$ et $\hat{f}_T(t)$ . . . . .                               | 221 |
| 5.1  | Distribution à hasard constant par morceaux, $\hat{h}_T(t)$ . . . . .                 | 250 |
| 5.2  | Distribution à hasard constant par morceau, $\hat{f}_T(t)$ . . . . .                  | 250 |
| 5.3  | Distribution à hasard constant par morceaux, $\hat{S}_T(t)$ et $\hat{F}_T(t)$ . . .   | 251 |
| 6.1  | Kaplan Meier, Fonctions de répartition $\hat{F}_T(t)$ et de survie $\hat{S}_T(t)$ . . | 278 |
| 6.2  | Estimations de Breslow et de Kaplan-Meier, $\hat{S}_T(t)$ . . . . .                   | 278 |
| 6.3  | Kaplan-Meier, Fonction de hasard $\hat{h}_T(t)$ . . . . .                             | 279 |
| 6.4  | Kaplan-Meier, Fonction de densité $\hat{f}_T(t)$ . . . . .                            | 279 |
| 6.5  | Estimateur actuariel, Fonction de survie $\hat{S}_T(t)$ . . . . .                     | 288 |
| 6.6  | Estimateur actuariel, Fonction de densité $\hat{f}_T(t)$ . . . . .                    | 289 |
| 6.7  | Estimateur actuariel, Fonction de hasard $\hat{h}_T(t)$ . . . . .                     | 289 |
| 6.8  | Estimateur actuariel, hasard avec intervalle de confiance . . . . .                   | 290 |
| 7.1  | Modèle exponentiel conditionnel . . . . .                                             | 312 |
| 7.2  | Weibull, distribution de base, $\hat{S}_0(t)$ et $\hat{F}_0(t)$ . . . . .             | 325 |
| 7.3  | Weibull, distribution de base, $\hat{h}_0(t)$ et $\hat{f}_0(t)$ . . . . .             | 325 |
| 7.4  | Weibull, distribution des survies individuelles à 5 ans . . . . .                     | 326 |
| 7.5  | Weibull, distribution des moyennes individuelles . . . . .                            | 326 |

# Table des Tableaux

|     |                                                               |     |
|-----|---------------------------------------------------------------|-----|
| 4.1 | Techniques d'estimation appliquées . . . . .                  | 48  |
| 4.2 | Instructions et fonctions IML (1/5) . . . . .                 | 61  |
| 4.3 | Instructions et fonctions IML (2/5) . . . . .                 | 62  |
| 4.4 | Instructions et fonctions IML (3/5) . . . . .                 | 63  |
| 4.5 | Instructions et fonctions IML (4/5) . . . . .                 | 64  |
| 4.6 | Instructions et fonctions IML (5/5) . . . . .                 | 65  |
| B.1 | Modèles de durée en logarithmes : écriture . . . . .          | 343 |
| B.2 | Modèles de durée en logarithmes : moments de $U$ . . . . .    | 343 |
| B.3 | Modèles de durée en logarithmes : moments de $k_2U$ . . . . . | 344 |