VALÉRIE ROY

Régression non paramétrique des percentiles pour données censurées

Mémoire présenté
à la Faculté des études supérieures de l'Université Laval
dans le cadre du programme de maîtrise en statistique
pour l'obtention du grade de Maître ès sciences (M.Sc.)

FACULTÉ DES SCIENCES ET DE GÉNIE
UNIVERSITÉ LAVAL
QUÉBEC

Février 2007

Résumé

L'utilisation de la régression non paramétrique est fréquente en analyse de données, puisque les postulats associés à la régression paramétrique ne sont pas toujours vérifiés, mais également parce qu'elle laisse aux données la décision de la forme de la relation entre une variable dépendante Y et une variable explicative X.

Dans ce mémoire, l'intérêt est porté sur l'estimation de percentiles conditionnels. Plus précisément, comme il arrive parfois que la variable réponse soit censurée, les méthodes d'estimation non paramétrique lisse de régression des percentiles dans le cas où la variable réponse est censurée à droite sont abordées. Ainsi, trois estimateurs sont considérés : un employant l'estimateur de Kaplan-Meier généralisé, un utilisant une optimisation pondérée par les poids Stute et un employant l'estimateur de Bowman et Wright. Ces méthodes sont appliquées à un jeu de données et leurs propriétés sont étudiées par voie de simulations.

Avant-propos

J'aimerais remercier toutes les personnes ayant contribué à la réalisation de ce mémoire, que ce soit par leur implication directe ou simplement par leurs encouragements. En premier lieu, je tiens à remercier mon directeur, Thierry Duchesne, professeur au Département de mathématiques et de statistique de l'Université Laval. Il a fait preuve de dévouement et s'est toujours montré très disponible, ce que j'ai fortement apprécié. Son aide et ses judicieux conseils m'ont été indispensables pendant toute la durée de la progression de ce mémoire. J'aimerais également remercier mon co-directeur, Belkacem Abdous, professeur au Département de médecine sociale et préventive de l'Université Laval. Tous deux m'ont fourni un sujet de recherche et de la documentation sur lesquels travailler, ils m'ont offert un support financier et ils m'ont donné l'opportunité de faire une présentation de ce mémoire au congrès de la Société Statistique du Canada de 2006, à London.

Par ailleurs, je souhaite aussi remercier Sophie Baillargeon, professionnelle de recherche au Département de mathématiques et de statistique de l'Université Laval. Elle a pour sa part généreusement accepté de participer à l'avancement de ce mémoire en ce qui concerne l'incorporation du code C dans un programme R.

Sur un plan plus personnel, je désire remercier mes amis, qui m'ont soutenue et divertie tout au long de l'accomplissement de ce travail. Merci en particulier à Marie-France Joanis, ma meilleure amie depuis le début de l'école primaire, pour avoir toujours été présente pour moi, pour tous ses encouragements et son éternel optimisme, mais également pour tous les bons moments que j'ai eu la chance de partager avec elle. Bien entendu, tous ces remerciements ne sauraient être complets sans que j'aie fait référence à mes parents. Je leur dis merci pour m'avoir offert la possibilité d'avancer dans mes études grâce à leur amour et à leur foi en moi, mais aussi pour leur soutien financier. Pour terminer, il y a une dernière personne que j'aimerais remercier, il s'agit de mon amoureux, Yann. Je le remercie simplement pour sa présence et son amour.

Table des matières

Liste des tableaux

Table des figures

Chapitre 1

La régression non paramétrique

L'utilisation des modèles paramétriques est très fréquente lorsque l'on fait appel à la régression afin d'analyser un jeu de données. Or, il y a certaines situations où ces modèles ne sont pas appropriés et où le choix d'un modèle non paramétrique est préférable. Dans ce chapitre, nous étudierons certaines méthodes de régression non paramétrique, plus particulièrement la méthode du noyau et la méthode par polynômes locaux. Ces méthodes sont très pratiques lorsque l'on s'intéresse à la relation entre une variable réponse Y et une variable explicative X, mais que l'on ne veut supposer aucune forme particulière pour la relation entre ces deux variables, laissant ainsi aux données le choix exclusif de cette forme.

La section 1.1 explique, à l'aide d'une démonstration, le lien qui existe entre la régression et la minimisation d'une espérance conditionnelle. La section 1.2 est entièrement consacrée à la méthode du noyau. Une description détaillée de la méthode (section 1.2.1), certaines propriétés (section 1.2.2), dont le biais et la variance, ainsi que les problèmes reliés à cette méthode (section 1.2.3) sont les sujets qui y sont traités. La section 1.3 porte, quant à elle, sur la méthode par polynômes locaux. Finalement, la section 1.4 présente une courte illustration permettant de constater le résultat, sur un jeu de données, des deux méthodes de régression non paramétrique dont il sera question tout au long de ce chapitre.

1.1 Lien entre la régression et la minimisation d'une espérance conditionnelle

Soient les observations bivariées $(x_i, y_i), i = 1, ..., n$, où les x_i représentent les valeurs observées de la variable aléatoire explicative Xet les y_i représentent celles de la variable aléatoire dépendante Y. La méthode la plus communément utilisée pour étudier la relation entre ces deux variables est la régression linéaire simple, qui suppose un modèle de la forme

$$Y_i = \beta_0 + \beta_1 X_i + \epsilon_i, \qquad i = 1, ..., n,$$

où les erreurs aléatoires ϵ_i sont non corrélées, de moyenne nulle et de variance σ^2. Cette méthode possède l'avantage d'être facile à interpréter et, lorsque les postulats sur les résidus ϵ_i sont vérifiés, elle permet de faire des tests d'hypothèses statistiques formels sur les paramètres. Par contre, il arrive que la linéarité de la relation ne soit pas toujours respectée. Dans ce cas, il est préférable de choisir un modèle plus flexible qui reflète mieux la relation entre X et Y. Le modèle de régression non paramétrique suivant peut alors être employé :

$$y_i = m(x_i) + \epsilon_i, \qquad i = 1, ..., n,$$

où $m(x_i)$ représente la moyenne conditionnelle de la courbe de régression, c'est-à-dire $m(x) = E(Y \mid X = x)$, et où les résidus ϵ_i représentent la variation de Y autour de $m(x)$. Les postulats sur les termes d'erreur ϵ_i sont les mêmes que ceux du modèle linéaire et, à part certaines hypothèses de continuité et de lissage (Simonoff, 1996, p. 42), il n'y a habituellement aucune contrainte associée à $m(x)$.

Il serait intéressant de faire le parallèle avec la définition formelle de la moyenne conditionnelle d'une variable aléatoire, puisque qu'elle fournit une nouvelle expression pour la moyenne conditionnelle de la courbe de régression. À cette fin, la démonstration de l'égalité de ces deux moyennes conditionnelles doit être présentée. Ainsi, soit la définition

$$\begin{aligned} m(x) &= \arg \min_a E[(Y-a)^2 | X = x] \\ &= E(Y \mid X = x). \end{aligned} \tag{1.1}$$

La preuve de cette égalité est trouvée en différenciant l'espérance $E[(Y-a)^2|X = x]$

par rapport à a, en égalant le résultat à 0 et, finalement, en isolant a :

$$\begin{aligned}\frac{\partial}{\partial a}E[(Y-a)^2 \mid X=x] &= -2E[(Y-a) \mid X=x] \\ &= -2E[Y \mid X=x] + 2a \\ &= 0\end{aligned}$$

$$\Rightarrow a = E(Y \mid X=x).$$

Le fait que la dérivée seconde, qui se chiffre à 2, soit positive mène à la conclusion que cette valeur a est bel et bien un minimum, et non un maximum. Ce lien entre la fonction de régression $m(x)$ et l'optimisation d'une espérance conditionnelle sera exploité à maintes reprises au cours de ce mémoire.

1.2 Méthode du noyau

1.2.1 Description de la méthode

La méthode du noyau est une méthode qui est communément utilisée pour faire de la régression non paramétrique. Elle donne pour estimateur de $E(Y \mid X=x)$ une moyenne pondérée des valeurs y_i pour les i dont le point x_i est près du point d'estimation. Pour appliquer cette méthode, il faut suivre les six étapes ci-dessous.

1. Tout d'abord, il est évident que le choix d'un point d'estimation x_0, une valeur de x pour laquelle on veut estimer $m(x_0)$, doit être fait.

2. Dans un autre temps, une fonction de noyau symétrique autour de 0 et unimodale doit être choisie. Cette fonction est maximale en 0 et, à l'exception du noyau gaussien, est non nulle uniquement dans la région [-1,1]. En fait, à mesure que cette fonction se rapproche de 0, sa valeur augmente ou reste la même, mais elle ne peut diminuer. Le tableau 1.1 donne un bref aperçu de certaines fonctions de noyau pouvant être employées (Härdle, 1990, section 2.1).

Il semble que le choix de la fonction de noyau n'est pas critique pour les estimations (Schimek, 2000, section 9.3). La fenêtre de lissage décrite ci-dessous au point 3 serait en effet beaucoup plus déterminante pour l'obtention de bonnes estimations. Le noyau gaussien sera utilisé au chapitre qui traite des simulations, c'est-à-dire le chapitre 4.

TAB. 1.1 – *Définition de certains noyaux*

Noyau	$K(u)$
Uniforme	$\frac{1}{2}I(\|u\| \leq 1)$
Gaussien	$\frac{1}{\sqrt{2\pi}}\ exp(-\frac{1}{2}u^2)I(-\infty < u < \infty)$
Epanechnikov	$\frac{3}{4}(1-u^2)I(\|u\| \leq 1)$

3. Le choix d'un paramètre de lissage h, qui peut uniquement prendre des valeurs positives, s'avère par la suite être indispensable. Sur un graphique des valeurs de Y en fonction des valeurs de X, deux lignes verticales distancées d'une valeur h et dont le milieu est situé en x_0 sont tracées. En d'autres mots, une fenêtre de lissage est créée. Cette situation est illustrée, à la figure 1.1 (Fox, 2004), par deux lignes pointillées qui délimitent la fenêtre de lissage et par une ligne continue qui représente le point d'estimation x_0. Tel qu'il a été vu à la section 1.1, la ligne horizontale représente la moyenne, qui peut être pondérée ou non, de la variable dépendante Y pour les observations à l'intérieur de la fenêtre. Il n'est pas toujours évident de déterminer la fenêtre produi-

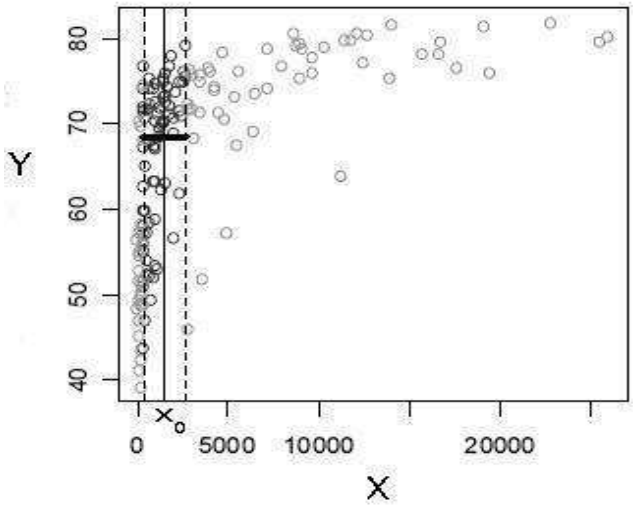


FIG. 1.1 – *Graphique des valeurs de Y en fonction des valeurs de X montrant une estimation $\hat{m}(x)$ (ligne horizontale) dans la fenêtre de lissage.*

sant de bonnes estimations, car plus h est grand, plus l'estimé est lisse et vice-versa. En fait, le paramètre de lissage h permet de contrôler le choix entre un plus petit biais ou une plus petite variance. Dans Simonoff (1996, section 5.3), une manière théorique basée sur la validation croisée est présentée afin de déterminer la valeur que devrait prendre le paramètre h. Or, lorsque cette méthode est utilisée et si on se concentre

uniquement sur les estimations calculées sur un court intervalle des valeurs de X, les estimés trouvés peuvent fortement varier. D'un point de vue plus pratique, on peut dire que cette méthode a tendance à entraîner du sous-lissage, c'est-à-dire que les courbes d'estimés obtenues sont trop "ondulées". La notion de sous-lissage est d'ailleurs illustrée plus loin dans ce chapitre, à la figure 1.4. En raison de cet inconvénient, une autre façon de déterminer la valeur de h est expliquée à la section 1.2.2 et, selon cette version, h dépend généralement de la taille échantillonnale n, plus spécifiquement, il est en général inversement proportionnel à une certaine puissance de n. Au chapitre 4, de multiples simulations seront faites afin d'être en mesure de faire le meilleur choix possible quant à ce paramètre de lissage.

4. Par la suite, le poids associé à chacune des observations doit être calculé. Ces poids sont d'ailleurs obtenus comme suit :

$$w_i(x_0) = K_h(X_i - x_0) = \frac{1}{h} K\Big(\frac{X_i - x_0}{h}\Big), \qquad i = 1, ..., n.$$

Ainsi, plus une donnée est rapprochée du point d'estimation x_0, plus le poids qui lui correspond est élevé, car $\mid x_i - x_0 \mid$ est plus près de 0. De même, plus une donnée est éloignée de x_0, plus le poids lui étant accordé est minime. Cette situation est d'ailleurs illustrée à la figure 1.2. De cette façon, excepté pour le noyau gaussien, lorsqu'une donnée est située en dehors de la fenêtre de lissage, elle obtient un poids de 0 et cette donnée n'a donc aucune influence sur l'estimation effectuée au point d'estimation.

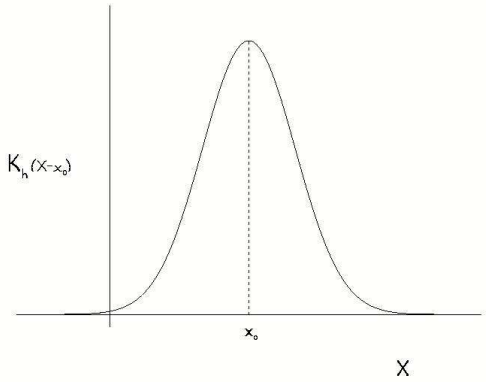


FIG. 1.2 – *Graphique illustrant la densité du noyau gaussien.*

5. L'estimateur de $m(x_0)$ est la moyenne pondérée des valeurs de Y :

$$\hat{m}(x_0) = \frac{\sum_{i=1}^{n} w_i(x_0) y_i}{\sum_{i=1}^{n} w_i(x_0)}. \tag{1.2}$$

Sur le graphique à la figure 1.1, cette estimation est représentée par une ligne horizontale dans la fenêtre de lissage.

On peut voir l'estimateur (1.2) comme une solution au problème d'optimisation (1.1) lorsque l'espérance conditionnelle est remplacée par une version empirique, c'est-à-dire par la moyenne pondérée par les $W_i(x_0)$, qui se définissent comme suit :

$$W_i(x_0) = \frac{w_i(x_0)}{\sum_{i=1}^n w_i(x_0)} = \frac{K_h(X_i - x_0)}{\sum_{i=1}^n K_h(X_i - x_0)}, \tag{1.3}$$

où

$$w_i(x_0) = K_h(X_i - x_0) = \frac{1}{h} K\Big(\frac{X_i - x_0}{h}\Big).$$

L'équation qui est finalement obtenue après avoir effectué ce remplacement dans le problème d'optimisation (1.1) est

$$\begin{aligned}
\hat{m}(x_0) &= \arg\min_a \hat{E}[(Y-a)^2 | X = x_0] \\
&= \arg\min_a \sum_{i=1}^n \frac{(Y_i - a)^2 K_h(X_i - x_0)}{\sum_{i=1}^n K_h(X_i - x_0)} \\
&= \arg\min_a \sum_{i=1}^n W_i(x_0)(Y_i - a)^2.
\end{aligned}$$

De manière analogue à la démonstration de l'équation (1.1), il est possible de trouver l'estimateur du noyau. En effet, cet estimateur est obtenu en dérivant la version empirique $\hat{E}[(Y-a)^2|X = x_0]$ par rapport à a, comme suit :

$$\begin{aligned}
\frac{\partial}{\partial a} \sum_{i=1}^n \frac{(Y_i - a)^2 K_h(X_i - x_0)}{\sum_{i=1}^n K_h(X_i - x_0)} &= -2 \sum_{i=1}^n \frac{(Y_i - a) K_h(X_i - x_0)}{\sum_{i=1}^n K_h(X_i - x_0)} \\
&= -2 \sum_{i=1}^n \frac{[Y_i K_h(X_i - x_0) - a K_h(X_i - x_0)]}{\sum_{i=1}^n K_h(X_i - x_0)} \\
&= -2 \sum_{i=1}^n \frac{Y_i K_h(X_i - x_0)}{\sum_{i=1}^n K_h(X_i - x_0)} \\
&\quad + 2a \sum_{i=1}^n \frac{K_h(X_i - x_0)}{\sum_{i=1}^n K_h(X_i - x_0)} \\
&= 0
\end{aligned}$$

$$\Rightarrow a = \frac{\sum_{i=1}^n Y_i K_h(X_i - x_0)}{\sum_{i=1}^n K_h(X_i - x_0)}.$$

Pour sa part, la dérivée seconde de cette même version empirique, qui prend la valeur $2\sum_{i=1}^n \frac{K_h(X_i - x_0)}{\sum_{j=1}^n K_h(X_j - x_0)}$, est positive. Il est donc possible d'affirmer avec certitude que cette valeur a est bien une valeur minimale.

L'estimateur du noyau qui est ainsi obtenu est l'estimateur du noyau de Nadaraya-Watson :

$$\hat{m}_{NW}(x_0) = \frac{\sum_{i=1}^{n} K_h(X_i - x_0) Y_i}{\sum_{i=1}^{n} K_h(X_i - x_0)}. \tag{1.4}$$

6. En général, $\hat{m}_{NW}(x_0)$ est estimé pour plusieurs valeurs de x_0 sur une fine grille afin d'obtenir "une courbe" de $\hat{m}(x)$ en fonction de x.

1.2.2 Les propriétés

Lorsque l'on veut comparer plusieurs estimateurs, il faut calculer certaines mesures permettant d'évaluer leurs qualités, telles le biais et la variance. L'erreur quadratique moyenne (EQM) peut aussi être calculée. Cette dernière est en fait une mesure de la différence quadratique espérée entre l'estimateur et sa valeur théorique. Bien entendu, l'estimateur par la méthode du noyau ne donne pas exactement la même valeur que la valeur théorique. Il serait donc intéressant de voir comment se comportent le biais et la variance pour cet estimateur du noyau $\hat{m}_{NW}(x)$. Toutefois, les calculs effectués pour l'obtention des résultats ne sont pas démontrés dans ce mémoire ; seuls les résultats (Simonoff, 1996, chap.5) y sont présentés.

Comme il sera vu à la section 1.2.3, le biais est accentué aux bornes des données. Pour cette raison, deux types de formules pour le biais et la variance de $\hat{m}_{NW}(x)$ existent : celles pour les estimateurs qui sont évalués à des points x se trouvant à l'intérieur des bornes du support de $f_X(x)$ et celles pour les estimateurs évalués à des points x situés à l'extérieur de ces bornes. Avant de s'intéresser à ces mesures, il est tout d'abord indispensable d'avoir une bonne compréhension de la signification de ces bornes. Pour donner un exemple très simple (Schimek, 2000, section 9.2.3), supposons que le support de $f_X(x)$ soit l'intervalle [0,1] et que l'on veuille estimer la fonction de régression aux points problématiques situés à gauche de $x = ch$ et aux points problématiques localisés à droite de $x = 1 - ch$, où $c > 0$. Si la fonction de noyau K a un support [-1,1], alors les points qui sont réellement aux bornes sont ceux pour lesquels $c < 1$; autrement, pour $c > 1$ les points sont considérés comme étant à l'intérieur des bornes. Un exemple de ce phénomène d'un biais plus élevé pour les estimateurs évalués à des points x se trouvant aux extrémités est d'ailleurs illustré ci-dessous. En effet, la figure 1.3 illustre le problème survenant à gauche du point $x = ch$, pour $c < 1$. Plus particulièrement, on voit sur cette figure que l'estimation de la courbe de régression, à gauche de ch, est moins élevée que les données le suggèrent réellement. Cette sous-estimation est due au fait qu'aux points d'estimation situés à gauche du point $x = ch$, on ne retrouve pas des données sur

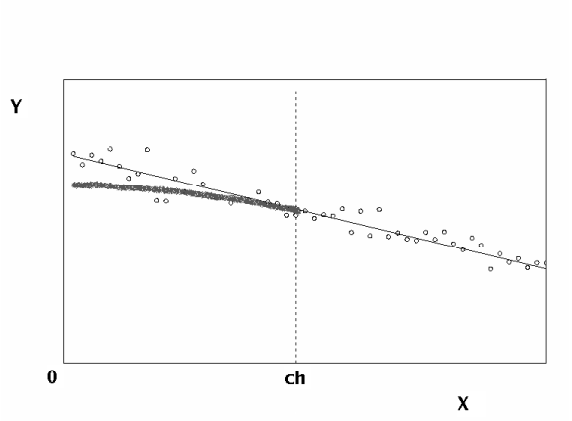


FIG. 1.3 – *Graphique des valeurs de Y en fonction des valeurs de X montrant un biais plus élevé pour les estimations évaluées à gauche de $x = ch$, pour $c < 1$.*

la totalité de la fenêtre de lissage, mais seulement sur une portion. En effet, pour ces points d'estimation, la borne inférieure de la fenêtre de lissage théorique devrait avoir une valeur négative, mais puisque la valeur minimale pour la variable explicative X est de 0, la borne inférieure de la fenêtre de lissage devient le point $x = 0$. Ainsi, puisque les valeurs de la variable Y diminuent en fonction des valeurs de la variable X et que la majorité des données qui servent à l'estimation des points problématiques sont celles qui se trouvent à droite de ces points, l'estimation obtenue est biaisée vers le bas.

Les formules pour les estimateurs évalués à des points qui se trouvent à l'intérieur des bornes seront tout d'abord abordées. Ainsi, pour ce type d'observations, le biais est

$$\begin{aligned} Biais[\hat{m}_{NW}(x) \mid x_1, ...x_n] &= E[\hat{m}(x) - m(x)] \\ &= h^2 \left[\frac{m'(x) f'_X(x)}{f_X(x)} + \frac{m''(x)}{2} \right] \mu_2(K_{(0)}) + o_p(h^2), \quad (1.5) \end{aligned}$$

où

$$\mu_q(K_{(p)}) = \int u^q K_{(p)}(u) du,$$

$f_X(x)$ est la fonction de densité des données exogènes x_i et $K_{(p)}$ est le noyau d'ordre $(p+1)$ lorsque p est impair et le noyau d'ordre $(p+2)$ lorsque p est pair. La définition du noyau d'ordre p peut être trouvée dans Simonoff (1996, p. 60). Ainsi, il devient clair que cette fonction $K_{(p)}$ est la même pour la méthode du noyau et pour la méthode localement linéaire, qui sera présentée à la section 1.3, c'est-à-dire que $K_{(0)}(u) = K_{(1)}(u)$, ce qui correspond plus précisément au noyau d'ordre 2. Il faut par ailleurs préciser que cette égalité n'est valable que dans le cas où les noyaux sont générés par la méthode des polynômes locaux. Ainsi, on comprend que la même situation se produit avec le terme

$R(K)$, qui sera vu ci-dessous et également plus loin dans ce mémoire, dans la section portant sur la méthode localement linéaire. En effet, puisque $R(K_{(p)}) = \int K_{(p)}(u)^2 du$ et $K_{(0)}(u) = K_{(1)}(u)$, l'égalité $R(K_{(0)}) = R(K_{(1)}) = R(K)$ est finalement obtenue. De plus, une définition formelle pour $o_p(h^2)$ est donnée à l'annexe A.1, mais ce qu'il faut absolument retenir à propos de ce terme est qu'il peut être négligé si la taille échantillonnale n est suffisamment grande. La formule pour la variance est, quant à elle,

$$\begin{aligned} Var[\hat{m}_{(NW)}(x) \mid x_1, ...x_n] &= E\left[\left(\hat{m}(x) - E[\hat{m}(x)]\right)^2\right] \\ &= \frac{R(K_{(0)})\sigma^2(x)}{nhf_X(x)} + o_p[(nh)^{-1}], \end{aligned}$$

où

$$R(K_{(p)}) = \int K_{(p)}(u)^2 du.$$

La formule pour l'EQM peut ainsi être obtenue :

$$\begin{aligned} EQM[\hat{m}(x)] &= E[\hat{m}(x) - m(x)]^2 \\ &= Var[\hat{m}(x)] + Biais^2[\hat{m}(x)]. \end{aligned}$$

Il peut parfois également être utile de calculer les valeurs intégrées de ces mesures, car elles donnent une idée de la qualité globale de l'estimateur. Par exemple, l'erreur quadratique moyenne intégrée ($EQMI$) est

$$\begin{aligned} EQMI[\hat{m}(x)] &= \int EQM[\hat{m}(x)]dx \\ &= \int E[\hat{m}(x) - m(x)]^2 dx. \end{aligned}$$

Comme l'allure du graphique de $\hat{m}(x)$ en fonction de x dépend beaucoup de la fenêtre de lissage h, il serait souhaitable d'avoir une formule permettant de décider de ce paramètre. À cette fin, l'$EQMI$ doit tout d'abord être obtenue :

$$\begin{aligned} EQMI[\hat{m}_{NW}(x) \mid x_1, ...x_n] = \int \Bigg\{ & h^4 \left[\frac{m'(x)f'_X(x)}{f_X(x)} + \frac{m''(x)}{2}\right]^2 \mu_2(K_{(0)})^2 \\ & + \frac{R(K_{(0)})\sigma^2(x)}{nhf_X(x)} + o_p[h^4 + (nh)^{-1}] \Bigg\} dx. \end{aligned}$$

En dérivant cette formule par rapport à h, la fenêtre de lissage optimale peut être trouvée.

Bien entendu, comme on l'a déjà précisé précédemment, il y a également des formules pour les estimateurs évalués à des points x qui se trouvent aux extrémités. Par exemple,

dans une telle situation, la formule pour le biais de ces estimateurs devient (Simonoff, 1996, section 5.2.3)

$$Biais[\hat{m}_{NW}(x) \mid x_1, ...x_n] = -\frac{m'(x)s_{1,c}h}{s_{0,c}} + o_p(h),$$

où

$$s_{l,c} = \int_{-\infty}^{c} u^l K(u)du, \qquad l = 0, 1, 2, 3.$$

En comparant ce biais à celui pour les estimateurs qui sont évalués à des points x se situant à l'intérieur des bornes, on remarque que la forme de ces deux biais est en fait la somme d'un facteur multipliant h à une certaine puissance et d'un terme négligeable. Pour les estimateurs qui sont évalués à des points x près des bornes, cette puissance de h est 1, alors qu'elle est de 2 pour les estimateurs qui sont évalués à des points à l'intérieur des bornes. Ainsi, quand h tend vers 0, le terme h^2 tend plus vite vers 0 que h. L'estimateur est donc moins biaisé lorsqu'il est évalué à des points x situés à l'intérieur des bornes.

1.2.3 Problèmes reliés à la méthode du noyau

Évidemment, en dépit du fait que l'estimateur par la méthode du noyau soit très simple et facile à comprendre, cette méthode n'est pas parfaite et certains problèmes y sont rattachés. En effet, il y a des problèmes de biais aux extrémité des données, de manque de variation locale du lissage et également d'aplatissement des pics et des vallées (Simonoff, 1996, section 3.2).

Tout d'abord, l'estimation par la méthode du noyau peut échouer dramatiquement aux extrémités des données car, comme on l'a vu à la section précédente, le biais y est plus élevé qu'à l'intérieur des données. En effet, le biais est d'ordre $O(h)$ aux bornes des données, alors qu'il est d'ordre $O(h^2)$ à l'intérieur de ces bornes. Cette réalité se manifeste d'ailleurs dans le fait qu'aux points d'estimation x situés aux extrémités, la méthode du noyau fournit soit une sous-estimation de la valeur réelle, ou encore une sur-estimation de celle-ci.

Par ailleurs, un problème de manque de variation locale du lissage est aussi lié à la méthode du noyau. En effet, étant donné qu'une seule fenêtre de lissage est utilisée, l'estimateur du noyau ne tient pas compte des différents niveaux de lissage des multiples parties de la fonction. En fait, pour réduire l'EQM, h devrait augmenter avec $m(x)$, ce

qui réduirait la variance, et diminuer avec $\mid m''(x) \mid$, ce qui réduirait le biais. D'un point de vue plus pratique, ce manque d'adaptation se manifeste par du surlissage (figure 1.4 a)) dans les régions où $\mid m'' \mid$ est grande et par du sous-lissage (figure 1.4 b)) dans les régions où $\mid m'' \mid$ est petite. Les deux graphiques de la figure 1.4 ont été tirés du site web de Duong (2001).

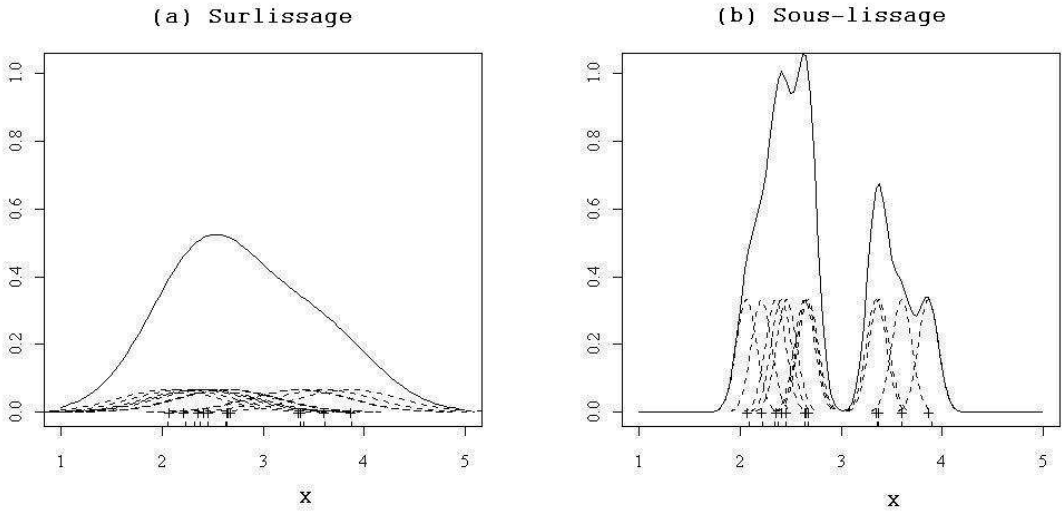


FIG. 1.4 – *Graphiques de surlissage et de sous-lissage.*

De plus, le biais de l'estimateur du noyau conduit souvent à un problème d'aplatissement des pics et des vallées. En réalité, ceci s'explique par le fait que le premier terme du biais asymptotique de $\hat{m}(x)$ est de la forme $h^2\sigma_K^2 m''(x)/2$. Or, cette valeur est habituellement plus grande, en valeur absolue, aux maximums et aux minimums locaux, où $m'(x)$ change rapidement de signe.

1.3 Polynômes locaux

La méthode du noyau n'est évidemment pas la seule méthode de régression qui existe. Il arrive également parfois que la méthode par polynômes locaux soit employée et elle est même préférée à la méthode du noyau. En effet, certains des problèmes que comporte la méthode du noyau, qui ont été discutés à la section précédente, peuvent être diminués lorsque la méthode par polynômes locaux est utilisée. Il est entre autres possible d'affirmer que la méthode par polynômes locaux est préférable, car la variance de cette méthode et celle de la méthode du noyau sont les mêmes, mais le biais aux bornes des données de la méthode par polynômes locaux est d'ordre $O(h^{p+1})$, alors qu'il

était d'ordre $O(h)$ pour la méthode du noyau. Les impacts de cette dernière méthode sur les mesures permettant d'évaluer la qualité d'un estimateur seront d'ailleurs expliqués de façon plus détaillée plus loin dans cette section.

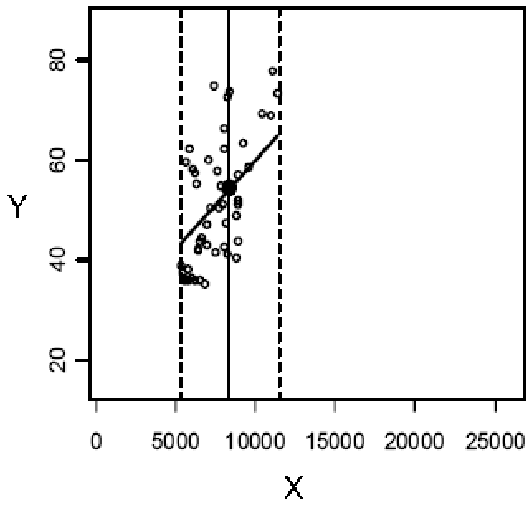


FIG. 1.5 – *Graphique des valeurs de Y en fonction des valeurs de X montrant une estimation $\hat{m}(x)$ (localement linéaire) dans la fenêtre de lissage.*

La régression par polynômes locaux est très similaire à l'estimation du noyau. Par contre, les valeurs obtenues sont produites par régression polynomiale pondérée par la distance au lieu de la moyenne pondérée par la distance. En fait, ce qui est différent de la méthode du noyau est que l'estimation de $m(x)$ est obtenue par régression polynomiale de y sur x. Cette nouvelle estimation de $m(x)$ est d'ailleurs illustrée à la figure 1.5 (Fox, 2002). Il faut maintenant minimiser la somme des carrés résiduels pondérée suivante :

$$\sum_{i=1}^{n} W_i(x)\{Y_i - \beta_0 - ... - \beta_p(X_i - x)^p\}^2, \tag{1.6}$$

où $W_i(x)$ est défini en (1.3).

Voici maintenant la solution à ce problème de minimisation (Simonoff, 1996, section 5.2.1). Soit la matrice

$$\mathbf{X}_x = \begin{pmatrix} 1 & x - x_1 & & (x - x_1)^p \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x - x_n & \dots & (x - x_n)^p \end{pmatrix}$$

et soit

$$W_x = h^{-1}\text{diag}\left[K\Big(\frac{x-x_1}{h}\Big), ..., K\Big(\frac{x-x_n}{h}\Big)\right],$$

la matrice des poids. Alors, si $X_x' W_x X_x$ est inversible,

$$\hat{\beta}_{\mathbf{x}} = (X_x' W_x X_x)^{-1} X_x' W_x \mathbf{y},$$

où

$$\mathbf{y} = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix} \quad \text{et} \quad \hat{\beta}_{\mathbf{x}} = \begin{pmatrix} \hat{\beta}_x^{(0)} \\ \vdots \\ \hat{\beta}_x^{(p)} \end{pmatrix}.$$

Ensuite, on pose $\hat{m}_p(x) = \hat{\beta}_x^{(0)}$.

Il est important de souligner le fait que la méthode du noyau est en réalité un cas particulier de la méthode par polynômes locaux. En effet, pour un choix de $p = 0$, l'équation à minimiser devient exactement celle qui donnait la méthode du noyau :

$$\sum_{i=1}^{n} W_i(x)(Y_i - \beta_0)^2.$$

En général, hormis le cas où $p = 0$, le choix de l'ordre du polynôme local est de $p = 1$. Dans cette situation, on parle d'une régression localement linéaire. C'est en fait ce cas qui est expliqué plus en profondeur dans ce mémoire. Tout d'abord, il faut savoir que l'estimateur localement linéaire ($p = 1$) de $m(x)$ est (Simonoff, 1996, section 5.2.1)

$$\hat{m}_1(x) = \frac{1}{nh}\sum_{i=1}^{n} \frac{[\hat{s}_2(x,h) - \hat{s}_1(x,h)(x - x_i)]K[(x - x_i)/h]y_i}{\hat{s}_2(x,h)\hat{s}_0(x,h) - \hat{s}_1(x,h)^2},$$

où

$$\hat{s}_r(x,h) = \frac{1}{nh}\sum_{i=1}^{n}(x - x_i)^r K\Big(\frac{x - x_i}{h}\Big).$$

Comme pour la méthode du noyau, il y a deux types de formules pour le biais et la variance de $\hat{m}_1(x)$: celles pour les estimateurs évalués à des points x qui se situent à l'intérieur des extrémités des données et celles pour les estimateurs évalués à des points x situés près des extrémités. Celles pour les estimateurs évalués à des points se trouvant à l'intérieur des bornes sont tout d'abord données. La formule pour le biais est donc

$$Biais[\hat{m}_1(x) \mid x_1, ...x_n] = \frac{h^2 m''(x)\mu_2(K_{(1)})}{2} + o_p(h^2),$$

où $\mu_q(K_{(p)})$ est défini de la même façon qu'à la section 1.2.2, c'est-à-dire

$$\mu_q(K_{(p)}) = \int u^q K_{(p)}(u)du.$$

La variance vaut pour sa part

$$Var[\hat{m}_1(x) \mid x_1, ...x_n] = \frac{R(K_{(1)})\sigma^2(x)}{nhf_X(x)} + o_p[(nh)^{-1}].$$

Puisque $K_{(0)} = K_{(1)}$, on remarque que cette variance est exactement la même que celle de la méthode du noyau, mais que le biais est plus faible. En effet, en comparant ce biais avec celui obtenu par la méthode du noyau (section 1.2.2), on voit que ce dernier comporte un terme supplémentaire en facteur de h^2,

$$\frac{m'(x)f'_X(x)}{f_X(x)}\mu_2(K_{(0)}).$$

Ainsi, contrairement au biais trouvé par la méthode du noyau, le biais obtenu par la méthode localement linéaire ne dépend ni de la fonction de densité des valeurs exogènes x_i, ni de sa dérivée $f'_X(x)$.

Tout comme à la section 1.2, il serait intéressant d'obtenir une formule qui permettrait d'obtenir un paramètre de lissage optimisant les estimations. Pour cela, l'$EQMI$ doit d'abord être obtenue :

$$\begin{aligned} EQMI[\hat{m}_1(x) \mid x_1, ...x_n] &= \left[\frac{h^2\mu_2(K_{(1)})}{2}\right]^2 \int m''(u)^2 f_X(u)du + \frac{R(K_{(1)})\sigma^2}{nh} \\ &\quad + o_p[h^4 + (nh)^{-1}]. \end{aligned}$$

En calculant la dérivée de cette équation par rapport à h, la fenêtre de lissage optimale est trouvée :

$$\begin{aligned} \frac{\partial}{\partial h} EQMI[\hat{m}_1(x) \mid x_1, ...x_n] &= h^3\mu_2(K)^2 \int m''(u)^2 f_X(u)du - \frac{R(K)\sigma^2}{nh^2} \\ &= 0 \end{aligned}$$

$$\Rightarrow h^5\mu_2(K)^2 \int m''(u)^2 f_X(u)du - \frac{R(K)\sigma^2}{n} = 0.$$

La fenêtre de lissage optimale est donc

$$h_{opt} = \left[\frac{R(K)\sigma^2}{n\mu_2(K)^2 \int m''(u)^2 f_X(u)du}\right]^{\frac{1}{5}}.$$

Cette dernière formule démontre bien que le point 3 de la section 1.2.1 était exact.

En effet, ce point stipulait que le paramètre de lissage h est généralement inversement proportionnel à une certaine puissance de n, qui a en fait une valeur de $1/5$ dans la présente situation.

En utilisant la même définition qu'à la section 1.2.2 pour le terme $s_{l,c}$, c'est-à-dire

$$s_{l,c} = \int_{-\infty}^{c} u^l K(u) du, \qquad l = 0, 1, 2, 3,$$

de nouvelles formules s'appliquant aux estimateurs évalués à des points x se trouvant aux extrémités des données peuvent être obtenues. Ainsi, le biais est corrigé et il devient alors

$$Biais[\hat{m}_1(x) \mid x_1, ...x_n] = \frac{\alpha_K(c) m''(x) h^2}{2} + o_p(h^2),$$

où

$$\alpha_K(c) = \frac{s_{2,c}^2 - s_{3,c} s_{1,c}}{s_{2,c} s_{0,c} - s_{1,c}^2}.$$

Cette formule permet de constater qu'une fois de plus, pour les estimateurs évalués à des points x situés aux bornes, l'estimateur par la méthode du noyau est plus biaisé que l'estimateur par la méthode localement linéaire. En effet, le biais est d'ordre $O(h)$ dans le cas de la méthode du noyau, alors qu'il est d'ordre $O(h^2)$ pour la méthode localement linéaire.

Évidemment, la correction pour le biais implique nécessairement un prix à payer, c'est-à-dire que la variance devient plus élevée. Cette variance asymptotique conditionnelle de $\hat{m}_1$ près des bornes est

$$Var[\hat{m}_1(x) \mid x_1, ...x_n] = \frac{\beta_K(c) \sigma^2(x)}{n h f_X(x)} + o_p[(nh)^{-1}],$$

où

$$\beta_K(c) = \frac{\int_{-\infty}^{c} (s_{2,c} - u s_{1,c})^2 K^2(u) du}{(s_{2,c} s_{0,c} - s_{1,c}^2)^2}.$$

Aux extrémités des données, si le noyau gaussien est utilisé et que la même fenêtre de lissage est choisie, la variance asymptotique conditionnelle de $\hat{m}_1$ est environ 3.17 fois celle de $\hat{m}_{NW}$ (Simonoff, 1996, section 5.2.3).

1.4 Exemple

Tout au long de ce premier chapitre, la théorie entourant la régression non paramétrique a été discutée. Afin de mieux comprendre les explications qui ont été fournies

jusqu'à présent, cette section présente l'application de cette nouvelle théorie à l'analyse d'un jeu de données (Johnson, 1995). Cet ensemble de données provient en fait d'une étude qui a été menée sur 252 hommes et pour laquelle une multitude de mesures ont été recueillies sur les individus en question. Le pourcentage de graisse, l'âge, le poids, la taille, la circonférence au niveau de la poitrine ont entre autres été mesurés sur ces individus. Dans cette section, seules deux de ces variables seront conservées et étudiées à partir d'une régression non paramétrique par la méthode localement linéaire. Pour être plus précis, l'effet du poids de l'individu, en livres, sur le pourcentage de graisse calculé selon l'équation de Siri (1956) fera l'objet de l'analyse qui suit. Il est à noter que les programmes en langage C et en langage R ayant été nécessaires à cette analyse apparaissent à l'annexe C.

Afin de procéder à la régression, il faut tout d'abord trouver la fenêtre de lissage permettant de minimiser l'équation (1.6). Les fenêtres optimales calculées dans ce mémoire ne sont pas utilisées dans cet exemple, car elles impliquent des quantités inconnues qui doivent être estimées, comme par exemple le calcul de $m''(u)$, ce qui implique donc un processus itératif fastidieux pour le choix de la fenêtre. La méthode du double noyau, qui est décrite ci-dessous, est donc préférée. À cette fin, les estimateurs en chaque point de la grille par la méthode localement linéaire, qui sont obtenus en résolvant le problème de minimisation

$$\arg \min_{\beta_0,\ \beta_1} \sum_{i=1}^{n} W_i(x)\{Y_i - \beta_0 - \beta_1(X_i - x)\}^2$$

et en posant $\hat{m}_1(x) = \hat{\beta}_0$, ainsi que par la méthode localement quadratique, qui sont quant à eux trouvés en résolvant

$$\arg \min_{\beta_0,\ \beta_1} \sum_{i=1}^{n} W_i(x)\{Y_i - \beta_0 - \beta_1(X_i - x) - \beta_2(X_i - x)^2\}^2$$

et en posant $\hat{m}_2(x) = \hat{\beta}_0$, sont d'abord calculés pour plusieurs valeurs de h. Par la suite, pour chacun des paramètres h ayant été utilisés, la somme, sur tous les points de la grille d'évaluation de l'estimateur ($\mathcal{G}$), du carré de la différence entre ces deux estimateurs a été trouvée :

$$\sum_{x \in \mathcal{G}} [\hat{m}_1(x) - \hat{m}_2(x)]^2.$$

La fenêtre de lissage h qui minimisait cette dernière équation est alors celle qui a été conservée pour pratiquer la régression. Dans cet exemple, la valeur pour h qui a été trouvée s'avère être de 5. Il faut toutefois préciser le fait que plusieurs méthodes existent afin de trouver la fenêtre de lissage optimale, par exemple, la méthode basée

sur la validation croisée (Simonoff, 1996, section 5.3). Or, dans ce mémoire, c'est la méthode discutée ci-dessus qui a été retenue.

La valeur de h étant maintenant choisie, la régression non paramétrique par la méthode localement linéaire du pourcentage de graisse en fonction du poids peut être exécutée. Les estimateurs en chacun des points de la grille sont donc obtenus à partir du paramètre de lissage $h = 5$ et la courbe de régression qui émane de la jonction de ces derniers est reproduite à la figure 1.6.

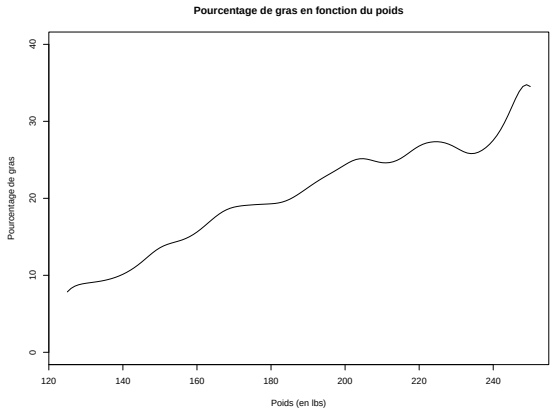


FIG. 1.6 – *Régression non paramétrique par la méthode localement linéaire du pourcentage de graisse en fonction du poids obtenue avec une fenêtre de lissage de* 5.

Cette figure suggère qu'une relation linéaire pourrait être appropriée et elle démontre également très clairement que le pourcentage de graisse augmente en fonction du poids de l'individu. Par ailleurs, une façon de vérifier si la fenêtre de lissage choisie est appropriée est de refaire la même régression avec deux autres paramètres de lissage : un valant une demie fois la fenêtre utilisée ci-dessus, c'est-à-dire 2.5, et un se chiffrant à une fois et demie la valeur de cette même fenêtre, plus précisément 7.5. La figure 1.7 illustre donc la première situation, soit une régression non paramétrique localement linéaire effectuée avec une fenêtre de lissage de 2.5. On voit aisément que la courbe de régression sur cette figure admet beaucoup plus de bruit que la précédente, ce qui suggère donc naturellement que le paramètre de lissage $h = 2.5$ est trop petit pour convenir à ce jeu de données.

Évidemment, le cas où le paramètre de lissage vaut $h = 7.5$ a également été examiné et il est présenté à la figure 1.8. On remarque que la courbe de régression sur cette figure est plus lisse que celle obtenue à partir d'un paramètre de lissage $h = 5$. Or, on ne peut pas nécessairement affirmer qu'elle est trop lisse, car elle semble tout de même bien décrire les données. On arrive donc à la conclusion que cette fenêtre $h = 7.5$ pourrait tout aussi bien être choisie pour décrire la relation existante entre le pourcentage de

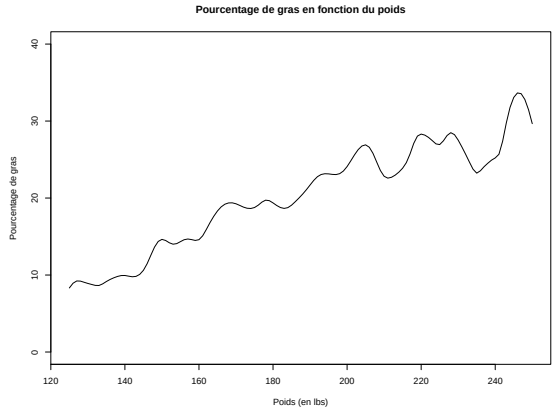


FIG. 1.7 – *Régression non paramétrique par la méthode localement linéaire du pourcentage de graisse en fonction du poids trouvée avec une fenêtre de lissage de* 2.5.

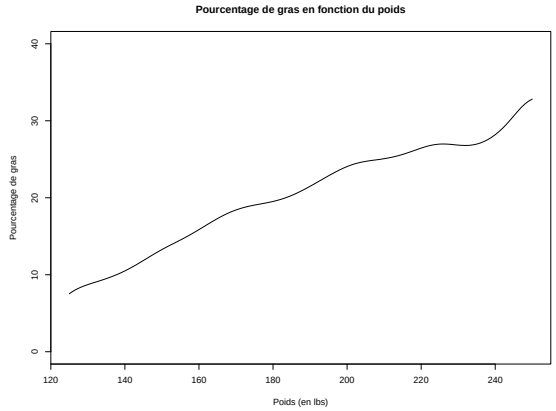


FIG. 1.8 – *Régression non paramétrique par la méthode localement linéaire du pourcentage de graisse en fonction du poids trouvée avec une fenêtre de lissage de* 7.5.

graisse et le poids des individus à partir de cet ensemble de données.

Chapitre 2

La régression non paramétrique des percentiles

Il arrive souvent qu'une estimation des percentiles des valeurs de Y en fonction de celles de X soit désirée. En effet, cela est très populaire dans plusieurs domaines, tels l'économique et la pédiatrie, où des courbes de croissance sont voulues. Un exemple de courbe de croissance pour les percentiles du poids des filles âgées entre 3 et 18 ans tiré du site web de l'Université de Toronto est donné à la figure 2.1. Dans le présent chapitre, la méthode de régression non paramétrique des percentiles, qui permet de tracer de telles courbes sans modèle paramétrique, sera étudiée en détails.

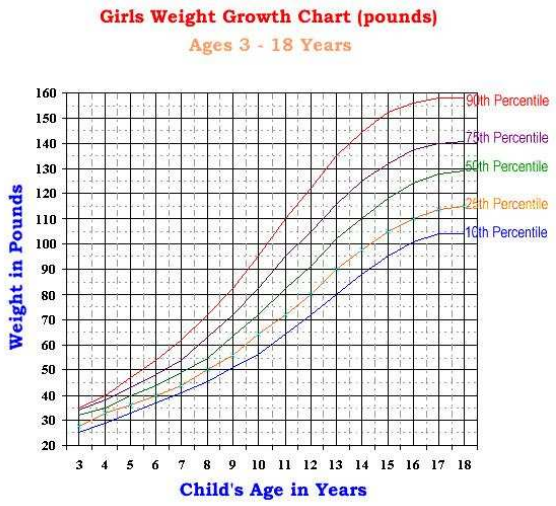


FIG. 2.1 – *Courbe de croissance pour le* 10^e, 25^e, 50^e, 75^e *et* 90^e *percentile du poids des filles âgées entre 3 et 18 ans.*

La section 2.1 permet de mieux comprendre ce qu'est un percentile conditionnel, car une définition y est donnée. La section 2.2 porte exclusivement sur la méthode du noyau avec les percentiles conditionnels. À la section 2.3, on retrouve la méthode par polynômes locaux, également utilisée avec les percentiles conditionnels. Quant à elle, la section 2.4 est consacrée à un petit exemple servant à voir le résultat de ces 2 méthodes appliquées à un jeu de données.

2.1 Les percentiles conditionnels

Avant d'introduire les estimations par la méthode de régression non paramétrique des percentiles, il faut d'abord s'assurer de bien comprendre ce qu'est un percentile conditionnel. Soient $\alpha \in]0,1[$ fixé et $g_\alpha(x)$ le quantile conditionnel d'ordre α de Y sachant $\{X = x\}$, qui vérifie la relation

$$P(Y \leq g_\alpha(x) \mid X = x) = \alpha. \tag{2.1}$$

Lorsque plusieurs valeurs de $g_\alpha(x)$ satisfont à l'équation (2.1), la plus petite d'entre elles doit être choisie, de sorte que la fonction des percentiles soit continue à gauche. De plus, on a que le quantile $g_\alpha(x)$ peut également être défini de manière équivalente comme solution au problème de minimisation énoncé au théorème 2.1.1 ci-dessous.

Théorème 2.1.1. $g_\alpha(x) = \arg\ \min_a\ E[\rho_\alpha(Y - a) \mid X = x]$,

où

$$\begin{aligned} \rho_\alpha(z) &= \alpha\ z\ I_{[0,\infty)}(z) - (1-\alpha)\ z\ I_{(-\infty,0)}(z) \\ &= z\ [\alpha - I_{(-\infty,0)}(z)] \\ &= \frac{|z| + (2\alpha - 1)z}{2}. \end{aligned}$$

Preuve : (Koenker, 2005, section 1.3)

On cherche à minimiser

$$\begin{aligned} E[\rho_\alpha(Y - a) \mid X = x] &= (\alpha - 1) \int_{-\infty}^{a} (y - a) dF(y \mid X = x) \\ &\quad + \alpha \int_{a}^{\infty} (y - a) dF(y \mid X = x). \end{aligned}$$

Afin de différencier cette expression par rapport à a, on doit utiliser la formule de Leibnitz (Casella et Berger, 2001, p. 69), qui stipule que si les fonctions $f(x,\theta)$, $a(\theta)$ et $b(\theta)$ sont dérivables par rapport à θ, alors

$$\frac{d}{d\theta}\int_{a(\theta)}^{b(\theta)} f(x,\theta)dx = f(b(\theta),\theta)\frac{d}{d\theta}b(\theta) - f(a(\theta),\theta)\frac{d}{d\theta}a(\theta) + \int_{a(\theta)}^{b(\theta)} \frac{\partial}{\partial\theta}f(x,\theta)dx.$$

De cette façon, on obtient la solution à la dérivation qui suit :

$$\begin{aligned} 0 &= (1-\alpha)\int_{-\infty}^{a} dF(y \mid X=x) - \alpha\int_{a}^{\infty} dF(y \mid X=x) \\ &= (1-\alpha)\left[F(a \mid X=x) - F(-\infty)\right] - \alpha\left[F(\infty) - F(a \mid X=x)\right] \\ &= F(a \mid X=x) - \alpha. \end{aligned}$$

Étant donné que F est monotone, tout élément de $\{x : F(x) = \alpha\}$ minimise l'espérance de la perte. Lorsque la solution est unique, $a = F^{-1}(\alpha) = g_\alpha(x)$; autrement, on est en présence d'un "intervalle de percentiles d'ordre α" duquel le plus petit élément doit être choisi. En effet, il faut que la fonction empirique des percentiles soit continue à gauche.

□

Il serait intéressant de voir l'allure d'un graphique de $\rho_\alpha(z)$ en fonction de z. En guise d'exemple, les graphiques de $\rho_{0.5}(z)$ vs z et $\rho_{0.75}(z)$ vs z sont montrés à la figure 2.2. On remarque, à partir de cette figure, que ces graphiques sont tous deux en forme de "V" et que plus l'ordre du percentile est élevé, moins la pente de la droite pour les valeurs inférieures à $z = 0$ est abrupte, alors que plus celle de la droite pour les valeurs supérieures à $z = 0$ le devient. En fait, plus la valeur de α augmente, plus les grandes valeurs de z sont pénalisées, comparativement aux petites qui le sont moins. Puisque le théorème 2.1.1 a démontré que l'on cherche à minimiser la fonction de pénalité, on désire donc obtenir de petites valeurs de z et, par le fait même, avoir de grandes valeurs de a, car $z = y - a$.

2.2 La méthode du noyau

La régression non paramétrique par la méthode du noyau vue au chapitre précédent peut aussi être utilisée pour estimer les percentiles conditionnels. En effet, à l'exception d'une étape, la procédure servant à obtenir les estimateurs est exactement la même que celle mentionnée à la section 1.2.1. En réalité, au lieu de chercher à résoudre le problème

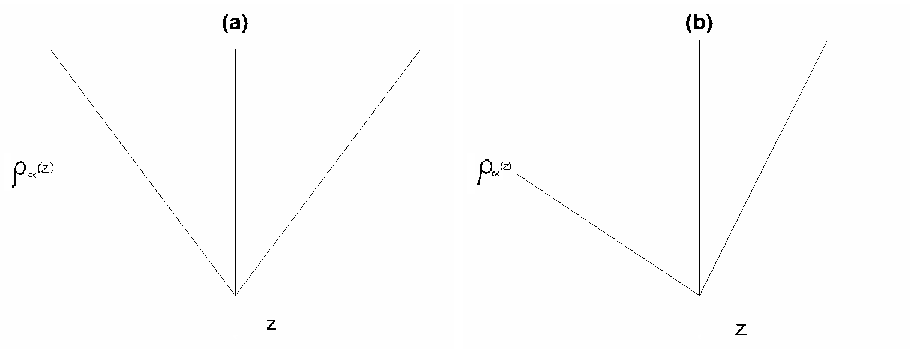


FIG. 2.2 – *Graphiques de $\rho_\alpha(z)$ en fonction de z pour : (a) $\alpha = 0.5$ et (b) $\alpha = 0.75$.*

de minimisation

$$m(x) \quad = \quad \arg \min_a \; E[(Y-a)^2 | X = x],$$

on cherche maintenant à résoudre

$$g_\alpha(x) \quad = \quad \arg \min_a \; E[\rho_\alpha(Y-a) | X = x]. \tag{2.2}$$

Ensuite, avant d'obtenir un estimateur pour $g_\alpha(x)$, il ne reste plus qu'à remplacer $E[\rho_\alpha(Y_i - a) \mid X = x]$ par une version estimée, $\hat{E}[\rho_\alpha(Y_i - a) \mid X = x]$, et à résoudre le problème de minimisation qui suit :

$$\begin{aligned} \hat{g_\alpha}(x) &= \arg \min_a \hat{E}[\rho_\alpha(Y_i - a) \mid X = x] \\ &= \arg \min_a \frac{\sum_{i=1}^n \rho_\alpha(Y_i - a) K_h(X_i - x)}{\sum_{i=1}^n K_h(X_i - x)} \\ &= \arg \min_a \sum_{i=1}^n W_i(x) \rho_\alpha(Y_i - a). \end{aligned}$$

On cherche donc à résoudre l'équation

$$\hat{g_\alpha}(x) = \arg \min_a \sum_{i=1}^n W_i(x)[|Y_i - a| - (2\alpha - 1)(Y_i - a)].$$

Puisque cette dernière équation ne peut être résolue analytiquement, elle doit l'être de façon numérique. Au chapitre 4, toutes les simulations sont faites à l'aide d'incorporation de code C dans un programme R et, afin de minimiser $\hat{g_\alpha}(x)$, la méthode de Nelder-Mead (Nelder et Mead, 1965) de la procédure `optim()` de R y est utilisée. Cette méthode est robuste et relativement lente, mais elle est choisie car elle fonctionne raisonnablement bien pour les fonctions non dérivables.

Comme au chapitre précédent, les propriétés de l'estimateur par la méthode du noyau $\hat{g_\alpha}(x)$ sont énumérées (Jones et Hall, 1990). Bien entendu, il serait intéressant

d'avoir ces propriétés dans les deux cas qui ont été discutés dans le chapitre 1, c'est-à-dire le cas où les estimateurs sont évalués à des points x qui sont à l'intérieur des bornes et le cas où ces points d'estimation sont situés aux extrémités. Or, Jones et Hall (1990) ne fournissent que les propriétés pour les données faisant partie du premier cas. Ce ne seront donc que celles-ci qui seront décrites dans ce mémoire. Ainsi, soient les observations bivariées $(x_i, y_i), i = 1, ..., n$, où les y_i sont les n réalisations indépendantes de la distribution conditionnelle de Y sachant $\{X = x\}$. Dénotons la fonction de distribution correspondante $F_X(y)$ et la fonction de densité $f_X(y)$ et définissons

$$F_X^{ab}(g_\alpha(x)) = \frac{\partial^{a+b}}{\partial z^a \partial y^b} F_Z(y)|_{x,g_\alpha(x)}, \qquad t(x) = E\sum_{i=1}^{n} W_i(x),$$

$\sigma_K^2 = \int u^2 K(u)du$ et $R(K) = \int K^2(u)du$, tout comme à la section 1.2.2. Lorsque les estimateurs sont évalués à des points qui se situent aux bornes, le biais associé à cet estimateur est

$$Biais[\hat{g_\alpha}(x)] = \frac{1}{2}\sigma_K^2 h_\alpha^2 \left[\frac{F_X^{20}(g_\alpha(x))}{f_X(g_\alpha(x))} - \frac{2t'(x)g'_\alpha(x)}{t(x)}\right] + o(h_\alpha^2),$$

alors que sa variance correspond à

$$Var[\hat{g_\alpha}(x)] = \frac{R(K)}{nh_\alpha}\frac{1}{t(x)}\frac{\alpha(1-\alpha)}{[f_X(g_\alpha(x))]^2} + o\left(\frac{1}{nh_\alpha}\right),$$

Ainsi, on trouve une erreur quadratique moyenne

$$EQM[\hat{g_\alpha}(x)] = \frac{1}{4}\sigma_K^4 h_\alpha^4 \left[\frac{F_X^{20}(g_\alpha(x))}{f_X(g_\alpha(x))} - \frac{2t'(x)g'_\alpha(x)}{t(x)}\right]^2 + \frac{R(K)}{nh_\alpha}\frac{1}{t(x)}\frac{\alpha(1-\alpha)}{[f_X(g_\alpha(x))]^2},$$

La fenêtre de lissage optimale peut alors être obtenue :

$$h_{opt} = \left[\frac{R(K)}{\sigma_K^4}\frac{\alpha(1-\alpha)t(x)}{\{t(x)F_X^{20}(g_\alpha(x)) - 2t'(x)g'_\alpha(x)f_X(g_\alpha(x))\}^2}\right]^{1/5} n^{-1/5}.$$

En analysant plus en profondeur les formules citées ci-dessus, on remarque que lorsqu'une valeur de 1/2 est attribuée au paramètre α et que tous les autres paramètres de la variance de l'estimateur obtenu par la méthode du noyau sont fixés, cette dernière est maximale. Par le fait même, l'EQM et la fenêtre de lissage optimale deviennent à leur tour plus grandes. Une deuxième observation qui découle de cette analyse est que, tout comme il en était le cas pour les fenêtres de lissage optimales décrites au chapitre 1, la fenêtre de lissage obtenue dans cette section est proportionnelle à $n^{-1/5}$. Ceci démontre donc une fois de plus la véracité du point 3 de la section 1.2.1, qui spécifiait que la fenêtre de lissage optimale est en général inversement proportionnelle à une certaine puissance de n, correspondant à une valeur de 1/5 dans le cas présent.

2.3 Polynômes locaux

La régression par polynômes locaux peut également être employée pour l'estimation de percentiles conditionnels. Or, contrairement à la section 1.3 dans laquelle il fallait minimiser

$$\sum_{i=1}^{n} W_i(x)\{Y_i - \beta_0 - ... - \beta_p(X_i - x)^p\}^2,$$

il faut maintenant minimiser

$$\sum_{i=1}^{n} W_i(x)\rho_\alpha\{Y_i - \beta_0 - ... - \beta_p(X_i - x)^p\}. \tag{2.3}$$

Comme à la section 1.3, nous prenons comme estimateur $\hat{g_\alpha}(x) = \hat{\beta}_0$, où $\hat{\beta}_0$ est un des éléments de la solution au problème de minimisation (2.3). Bien entendu, ce problème de minimisation doit également être résolu de façon numérique et, encore une fois, l'algorithme de Nelder-Mead de la fonction `optim`() de R est utilisé.

Il serait maintenant intéressant de découvrir les propriétés, plus spécifiquement, le biais, la variance, l'EQM et la fenêtre de lissage optimale, de cet estimateur des percentiles par la méthode localement linéaire, c'est-à-dire dans le cas où $p = 1$ dans l'équation (2.3). Notons que les résultats qui seront fournis dans cette section ont tous été tirés de Fan *et al.* (1994). Encore une fois, les propriétés des estimateurs évalués à des points x se trouvant l'intérieur des bornes diffèrent de ceux évalués à des points x qui se situent à l'extérieur de ces bornes. Dans les lignes qui suivent, les estimateurs évalués à des points x situés à l'intérieur des bornes seront tout d'abord examinés et ceux qui sont évalués à des points x se trouvant aux extrémités suivront par la suite.

À l'intérieur des bornes, le biais est

$$Biais[\hat{g_l}(x) \mid x_1, ...x_n] = \beta(x)h^2,$$

où

$$\beta(x) = (1/2)g''_\alpha(x)\int u^2K(u)du,$$

alors que la variance est

$$Var[\hat{g_l}(x) \mid x_1, ...x_n] = \frac{\tau^2(x)}{nh},$$

où

$$\tau^2(x) = \int \frac{K^2(u)du}{f_X(x)}\frac{\alpha(1-\alpha)}{[r(g_\alpha(x)|x)]^2},$$

avec $r(y|x)$ qui désigne la densité conditionnelle de Y sachant $\{X = x\}$ et où $\mu(y) = y$.

Maintenant que les formules du biais et de la variance de cet estimateur des percentiles par la méthode localement linéaire sont fournies, il est facile de déduire la valeur de l'erreur quadratique moyenne. En effet, en additionnant le carré du biais à la variance, on trouve que l'EQM est

$$EQM = \beta^2(x)h^4 + \frac{\tau^2(x)}{nh}.$$

La fenêtre de lissage optimale associée à cet estimateur peut également être facilement obtenue. Comme il en a déjà été fait mention à maintes reprises dans ce mémoire, on arrive à l'obtention de cette fenêtre en dérivant l'EQM par rapport au paramètre h et en égalant le résultat à 0. Dans le cas présent, la fenêtre de lissage optimale obtenue est

$$h_{opt} = \left(\frac{\tau^2(x)}{4n\beta^2(x)} \right)^{\frac{1}{5}}.$$

Mais que deviennent les propriétés associées à cet estimateur lorsqu'il est évalué à des points qui se situent aux extrémités ? Afin d'être en mesure de répondre à cette question, il est d'abord nécessaire de fournir la définition des termes qui suivent :

$$\alpha(c) = \frac{c_2^2 - c_1 c_3}{c_0 c_2 - c_1^2} \qquad et \qquad \beta(c) = \frac{\int_{-\infty}^{c} (c_2 - c_1 u)^2 K^2(u) du}{(c_0 c_2 - c_1^2)^2},$$

où

$$c_j = \int_{-\infty}^{c} u^j K(u) du, \qquad j = 1, 2, 3.$$

Le biais asymptotique aux extrémités est donc

$$Biais[\hat{g}_l(x) \mid x_1, ...x_n] = \beta h^2,$$

où

$$\beta = (1/2)\alpha(c) g''_\alpha(0)$$

et la variance asymptotique est

$$Var[\hat{g}_l(x) \mid x_1, ...x_n] = \frac{\tau^2}{nh},$$

où

$$\tau^2 = \frac{\beta(c)\alpha(1-\alpha)}{[r(g_\alpha(0)|0)]^2 f_X(0)}.$$

L'EQM peut être facilement trouvée à partir du biais et de la variance :

$$EQM = \beta^2 h^4 + \frac{\tau^2}{nh}.$$

Ainsi, la fenêtre optimale est obtenue :

$$h_{opt} = \left(\frac{\tau^2}{4n\beta^2}\right)^{\frac{1}{5}}.$$

À la vue de ces résultats, certaines remarques, qui s'appliquent autant au cas où les estimateurs sont évalués à des points situés à l'intérieur des bornes que dans le cas où ils sont situés aux extrémités, peuvent être formulées. En effet, tout comme à la section précédente (section 2.2), on remarque que, dans les deux cas et avec les autres paramètres fixés, la variance de l'estimateur est maximale lorsque le paramètre α vaut 1/2. Cela signifie donc que les EQM et les fenêtres de lissage optimales seront plus grandes. De plus, les deux fenêtres de lissage optimales trouvées sont encore une fois inversement proportionnelles à $n^{1/5}$.

Par ailleurs, le principal inconvénient des méthodes localement polynomiales est, dans le cas de la régression non paramétrique des percentiles, que rien ne garantit que les courbes des percentiles ne s'entrecouperont pas. Les figures présentées à la fin de ce chapitre montrent en effet des courbes qui se croisent en au moins un point.

2.4 Exemple

Tout comme au chapitre 1, la théorie du chapitre actuel sera examinée à l'aide de l'application au même jeu de données qu'à la section 1.4, c'est-à-dire celui provenant du site web de Johnson (1995). Les deux variables ayant servies à l'analyse du chapitre précédent seront également celles qui feront l'objet de la modélisation effectuée ci-dessous. En effet, une régression non paramétrique des percentiles par la méthode localement linéaire du pourcentage de graisse calculé selon l'équation de Siri (1956) en fonction du poids, en livres, de l'individu sera produite dans la présente section. Tout comme au chapitre précédent, il est à noter que les programmes en langage C et en langage R ayant été utilisés pour cette analyse apparaissent à l'annexe C.

Tel qu'à l'exemple du chapitre 1, la première étape qui s'impose est de déterminer la fenêtre de lissage qui servira à produire les estimateurs d'ordre α. La méthode du double noyau utilisée au chapitre antérieur, pour laquelle la somme du carré de la différence des estimateurs obtenus par la méthode localement linéaire et la méthode localement quadratique, est aussi celle qui a été adoptée dans ce chapitre. En effet, comme il a déjà été expliqué au chapitre 1, les fenêtres optimales calculées dans ce chapitre ne sont pas utilisées dans cet exemple, car elles impliquent des quantités inconnues qui doivent être estimées, comme par exemple le calcul de $g''_\alpha(x)$. Ainsi, au lieu d'utiliser des méthodes adaptatives, nous préférons procéder par la méthode du double noyau.

Les estimateurs qui devront être calculés sont maintenant ceux qui ont été décrits dans ce chapitre. Ainsi, l'estimateur par la méthode localement linéaire est trouvé en résolvant le problème de minimisation

$$\arg \min_{\beta_0,\ \beta_1} \sum_{i=1}^{n} W_i(x)\rho_\alpha\{Y_i - \beta_0 - \beta_1(X_i - x)\}$$

et en posant $\hat{g}^1_\alpha(x) = \hat{\beta}_0$, alors que celui par la méthode localement quadratique est obtenu en résolvant

$$\arg \min_{\beta_0,\ \beta_1} \sum_{i=1}^{n} W_i(x)\rho_\alpha\{Y_i - \beta_0 - \beta_1(X_i - x) - \beta_2(X_i - x)^2\}$$

et en posant $\hat{g}^2_\alpha(x) = \hat{\beta}_0$. La résolution de l'équation ci-dessous doit par la suite être obtenue :

$$\sum_{x\in\mathcal{G}} [\hat{g}^1_\alpha(x) - \hat{g}^2_\alpha(x)]^2.$$

Ainsi, pour chacune des trois valeurs de α, la solution de cette dernière équation est trouvée pour plusieurs valeurs de h. Finalement, pour un α donné, la valeur de h qui a conduit à l'obtention de la quantité la plus minime est celle qui sera conservée. Ce nombre s'avère être approximativement de 6 pour chacune des trois valeurs de α. C'est donc cette valeur qui sera attribuée à la fenêtre de lissage dans les régressions non paramétriques des percentiles du pourcentage de graisse en fonction du poids des hommes et des femmes qui suivent. Ainsi, les estimateurs par la méthode localement linéaire trouvés en chacun des points de la grille, pour chacune des trois valeurs de α et des fenêtres de lissage leur correspondant, sont obtenus et la liaison de tous ces estimateurs permet d'obtenir les trois courbes de percentiles qui se retrouvent à la figure 2.3 ci-dessous.

Comme les résultats trouvés à la section 1.4 le suggéraient, la figure ci-dessus montre qu'une régression linéaire conviendrait à ce jeu de données et elle démontre également

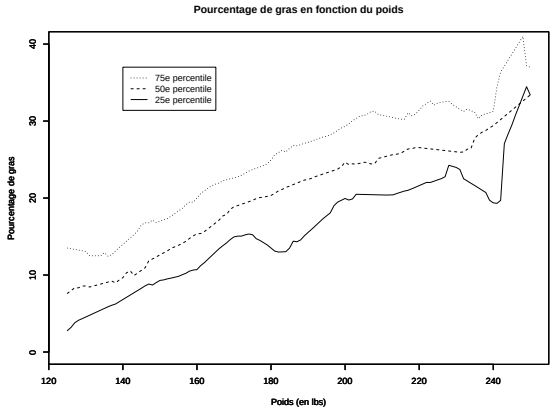


FIG. 2.3 – *Régression non paramétrique des* 25^e, 50^e *et* 75^e *percentiles par la méthode localement linéaire du pourcentage de graisse en fonction du poids obtenue avec une fenêtre de lissage h=6.*

très clairement qu'il existe une relation croissante entre le pourcentage de graisse des individus et le poids de ces derniers. Par ailleurs, la figure 2.4 illustre quant à elle une régression non paramétrique des percentiles produite avec une fenêtre de lissage $h = 3$. Sur cette figure, il est à noter que les fluctuations des courbes aux valeurs les plus élevées du poids, c'est-à-dire approximativement à partir de 225 livres, sont causées par une moins grande quantité d'observations à ces points. Cette figure permet facilement de s'assurer qu'une fenêtre de lissage de 3 est évidemment trop petite étant donné que la fluctuation des courbes y est beaucoup trop élevé.

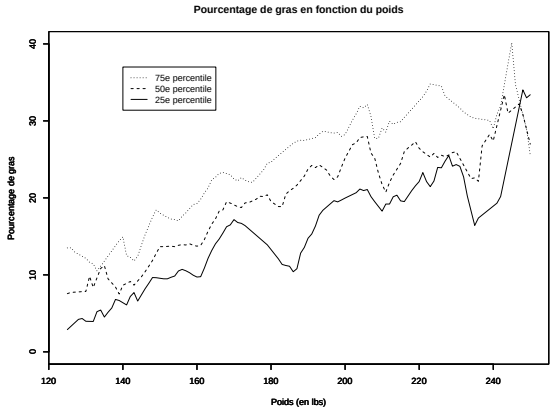


FIG. 2.4 – *Régression non paramétrique des* 25^e, 50^e *et* 75^e *percentiles par la méthode localement linéaire du pourcentage de graisse en fonction du poids trouvée avec une fenêtre de lissage h=3.*

Afin de clore cet exemple, une régression non paramétrique des percentiles effectuée

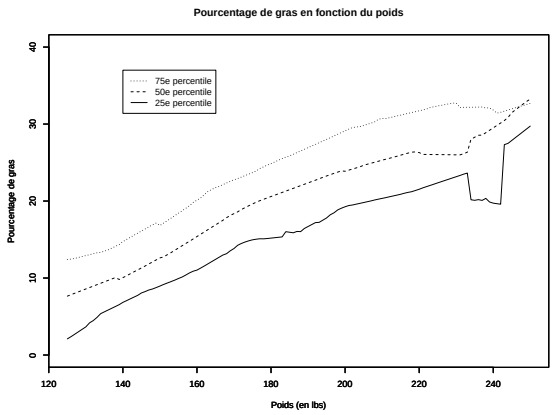


FIG. 2.5 – *Régression non paramétrique des* 25^e, 50^e *et* 75^e *percentiles par la méthode localement linéaire du pourcentage de graisse en fonction du poids obtenue avec une fenêtre de lissage* h*=9.*

à partir d'une fenêtre de lissage $h = 9$ apparaît finalement à la figure 2.5. On remarque que les courbes de régression des percentiles sur cette dernière figure sont plus lisses que celles obtenues à partir d'un paramètre de lissage $h = 6$. Or, on ne peut pas affirmer qu'elles sont excessivement lisses, car elles semblent tout de même très bien prendre en considération la distribution des données. On arrive donc à la conclusion que cette fenêtre de lissage $h = 9$ pourrait également être un bon choix pour décrire la relation existante entre le pourcentage de graisse et le poids des individus à partir de cet ensemble de données.

Chapitre 3

La régression non paramétrique des percentiles avec réponse censurée

Jusqu'à présent, les méthodes de régression non paramétrique pour la moyenne (chap. 1) et pour les percentiles (chap. 2) ont été étudiées dans le cas où toutes les données représentaient la valeur réelle de la variable réponse et non une borne inférieure ou supérieure pour celle-ci. Or, de tels cas ne se présentent pas toujours dans la réalité, car il arrive parfois que certaines observations ne soient disponibles que pour une partie de l'étude. On dit qu'une donnée est censurée à droite si l'information sur cette dernière est en fait une borne inférieure pour la vraie valeur de l'observation. Cette situation survient soit parce que l'étude se termine avant que la valeur de la variable ait été observée ou bien parce que le sujet quitte l'étude ou décède avant la fin de celle-ci. Un exemple typique pour une donnée censurée à droite est celui qui suit. Lorsqu'une étude est menée afin d'analyser les temps de guérison d'un certain remède sur les patients atteints d'une maladie et que le patient meurt avant d'être guéri, le temps auquel l'individu décède devient alors une borne inférieure du temps de guérison, c'est-à-dire une donnée censurée à droite. La régression non paramétrique des percentiles avec variable réponse censurée à droite sera étudiée tout au long de ce chapitre.

La complexité de l'analyse par régression avec variable réponse censurée réside dans le choix des poids à employer. En effet, comme les Y_i censurés sont des bornes inférieures pour la vraie valeur de l'observation, il est logique de s'attendre à ce que ces valeurs censurées n'entrent pas dans le calcul de la moyenne pondérée, car leur inclusion dans la moyenne pondérée ferait en sorte que cette dernière sous-estimerait la moyenne théorique. Il faut donc redistribuer le poids des Y_i censurés à ceux qui ne le sont pas, mais la difficulté est de trouver une bonne façon de le faire. Pour cette raison, la section 3.1 porte sur les nouveaux poids qui seront utilisés afin de remplacer ceux

des chapitres précédents ainsi que les estimateurs leur correspondant. Les poids par la méthode du Kaplan-Meier généralisé (KMG) seront traités à la section 3.1.1, alors que les poids proposés par Stute (Stute, 1993) seront étudiés à la section 3.1.2. Cette dernière section sera la seule, dans ce chapitre, à contenir les propriétés de l'estimateur trouvé, puisqu'un article a tout récemment été publié à ce sujet (Gannoun *et al.*, 2005). La section 3.2 est consacrée à un tout autre type d'estimation, soit la méthode proposée par Bowman et Wright (2000). Afin de conclure ce chapitre, un exemple d'analyse de données par la méthode de régression non paramétrique des percentiles avec variable réponse censurée à droite sera exploré à la section 3.3.

3.1 Les poids

Dans les chapitres précédents, où aucune donnée censurée n'était présente, il a été mentionné que les poids $W_i(x)$ sont inversement proportionnels à la valeur $|X_i - x|$, c'est-à-dire que les poids diminuent au fur et à mesure que cette valeur augmente. Cette affirmation demeure toujours valable dans ce chapitre. Par contre, lorsqu'il y a présence de données censurées, les poids des Y_i censurés, qui n'entrent pas dans les calculs de moyennes pondérées, doivent être redistribués aux poids des données non censurées, à l'opposé des chapitres précédents où un poids était attribué à chaque observation. Ceci démontre bien que, lorsqu'il y a censure, les poids $W_i(x)$ ne dépendent pas uniquement de la i^e observation, mais plutôt de l'échantillon en entier. Afin de mieux refléter ce fait, ces poids s'écriront donc désormais comme suit : $W_{i:n}(x)$. Or, la difficulté mentionnée au début du paragraphe précédent persiste toujours : il est plutôt complexe de faire le choix d'un bon poids. C'est d'ailleurs pour cette raison que deux méthodes de calcul pour les poids seront étudiées dans cette section et que leur performance sera par la suite évaluée par voie de simulation au chapitre 4.

Bien que les méthodes qui seront étudiées dans ce chapitre soient également valables dans le cas où il y a la présence de temps de décès et de temps de censure égaux, seul le cas où ces temps de décès et de censure sont tous distincts sera développé dans ce mémoire. Or, il faut absolument retenir que l'on procède de cette façon simplement dans le but d'alléger le contenu du texte et que cela ne change aucunement le fait que les méthodes qui seront expliquées peuvent aussi bien être employées dans une situation où les temps de décès et de censure sont égaux que dans le cas où ces derniers sont tous distincts. Avant d'entrer dans les détails, il faut tout d'abord définir les termes qui seront ultérieurement utilisés dans ce chapitre, c'est-à-dire les statistiques d'ordre. Ainsi, soit l'échantillon de taille n

$$(Y_i, \delta_i, X_i), \qquad i = 1, 2, ..., n,$$

où δ_i est une variable indicatrice qui vaut 1 si Y_i est observée directement et 0 si Y_i est censurée à droite et, comme définies aux chapitres précédents, X_i est une variable aléatoire explicative et Y_i est une variable aléatoire dépendante. Les notations qui seront utilisées tout au long de ce chapitre sont les trois variables ordonnées $Y_{(i)}, \delta_{(i)}$ et $x_{(i)}$, qui sont respectivement définies

$$\begin{aligned} Y_{(1)} < Y_{(2)} < \ldots < Y_{(n)} \\ \delta_{(i)} = \{\delta_j : Y_j = Y_{(i)}\} \\ X_{(i)} = \{X_j : Y_j = Y_{(i)}\}. \end{aligned}$$

3.1.1 Méthode du Kaplan-Meier généralisé

La méthode avec poids basés sur le Kaplan-Meier généralisé est une méthode localement linéaire et pour laquelle la fonction de survie est utilisée. En fait, les poids conventionnels dont on se servait au chapitre 2 y sont remplacés par les sauts que fait la fonction de survie du KMG aux points Y_i sachant $\{X_i = x\}$.

Avant d'entrer dans le vif du sujet, il faut tout d'abord se souvenir de la formule de l'estimation que l'on désire produire, c'est-à-dire l'expression qui suit :

$$\begin{aligned} \hat{E}[\rho_\alpha\{Y_i - \beta_0 - \beta_1(X_i - x)\}|X = x] &= \sum_{i=1}^{n} \rho_\alpha\{Y_i - \beta_0 - \beta_1(X_i - x)\} \\ &\times \hat{P}(Y = y_i|X = x). \end{aligned} \tag{3.1}$$

On se rappelle que, puisqu'il n'y avait aucune donnée censurée, le terme qui était utilisé au chapitre 2 pour désigner l'estimation de $\hat{P}(Y = y_i|X = x)$ était l'estimateur de Nadaraya-Watson (NW) :

$$\hat{P}(Y = y_i|X = x) = \frac{K_h(X_i - x)}{\sum_{i=1}^{n} K_h(X_i - x)}.$$

Or, dans cette section, on peut estimer $F(y_i|x)$ par l'estimateur du KMG , qui sera défini plus bas. Ainsi, puisque les statistiques d'ordre sont utilisées dans le présent chapitre, on se retrouve conséquemment avec l'équation

$$\hat{P}(Y = y_{(i)}|X = x) = \Delta_{KMG}(y_{(i)}|x),$$

où $\Delta_{KMG}(y_{(i)}|x)$ représente la valeur du saut que fait l'estimateur de la fonction de répartition du Kaplan-Meier généralisé au temps $y_{(i)}$ sachant $\{X = x\}$.

Ainsi, le tout premier poids qui sera étudié dans ce chapitre est celui obtenu par la méthode du KMG. Leconte *et al.* (2002) ont étudié l'estimateur du KMG, originalement

proposé par Beran (1981). L'estimateur de la fonction de survie sachant $\{X_{(i)} = x\}$, c'est-à-dire la probabilité que l'événement n'ait toujours pas eu lieu au temps $Y_{(i)}$, est

$$\hat{S}_{KMG}(Y_{(i)} \mid x) = \prod_{l=1}^{i} \left[1 - \frac{K_h(x - X_{(l)})}{\sum_{k=l}^{n} K_h(x - X_{(k)})} \right]^{\delta_{(l)}}, \tag{3.2}$$

où

$$\delta_{(l)} = \begin{cases} 0, & \text{si la } l^e \text{ observation ordonnée est censurée à droite} \\ 1, & \text{sinon.} \end{cases}$$

Avant de voir l'expression qui permet d'obtenir l'estimateur du quantile conditionnel d'ordre α, une démonstration du calcul servant à trouver la valeur de $\Delta_{KMG}(Y_{(i)}|x)$ sera montrée. Or, pour démontrer ce calcul, il faut tout d'abord trouver la valeur de $\hat{S}_{KMG}(Y_{(i)}^{-} \mid x)$, qui est obtenue simplement en remplaçant i par $i-1$ dans le produit de l'équation (3.2). De cette façon, on obtient l'expression

$$\hat{S}_{KMG}(Y_{(i)}^{-} \mid x) = \prod_{l=1}^{i-1} \left[1 - \frac{K_h(x - X_{(l)})}{\sum_{k=l}^{n} K_h(x - X_{(k)})} \right]^{\delta_{(l)}}.$$

La valeur de $\Delta_{KMG}(Y_{(i)}|x)$ peut par la suite être obtenue en résolvant la soustraction qui suit :

$$\begin{aligned}
\Delta_{KMG}(Y_{(i)}|x) &= \hat{S}_{KMG}(Y_{(i)}^{-} \mid x) - \hat{S}_{KMG}(Y_{(i)} \mid x) \\
&= \prod_{l=1}^{i-1} \left[1 - \frac{K_h(x - X_{(l)})}{\sum_{k=l}^{n} K_h(x - X_{(k)})} \right]^{\delta_{(l)}} - \prod_{l=1}^{i} \left[1 - \frac{K_h(x - X_{(l)})}{\sum_{k=l}^{n} K_h(x - X_{(k)})} \right]^{\delta_{(l)}} \\
&= \prod_{l=1}^{i-1} \left[1 - \frac{K_h(x - X_{(l)})}{\sum_{k=l}^{n} K_h(x - X_{(k)})} \right]^{\delta_{(l)}} \\
&\quad \times \left\{ 1 - \left[1 - \frac{K_h(x - X_{(i)})}{\sum_{k=i}^{n} K_h(x - X_{(k)})} \right]^{\delta_{(i)}} \right\}.
\end{aligned}$$

On remarque que le second terme de la ligne précédente peut être simplifié de la manière qui suit :

$$\begin{aligned}
1 - \left[1 - \frac{K_h(x - X_{(i)})}{\sum_{k=i}^{n} K_h(x - X_{(k)})} \right]^{\delta_{(i)}} &= \begin{cases} 0, & \text{si} \quad \delta_{(i)} = 0 \\ \frac{K_h(x - X_{(i)})}{\sum_{k=i}^{n} K_h(x - X_{(k)})}, & \text{si} \quad \delta_{(i)} = 1 \end{cases} \\
&= \frac{\delta_{(i)} K_h(x - X_{(i)})}{\sum_{k=i}^{n} K_h(x - X_{(k)})}.
\end{aligned}$$

L'expression finale pour la valeur du saut de KMG en $Y_{(i)}$ est donc

$$W_{i:n}(x) = \Delta_{KMG}(Y_{(i)}|x) = \frac{\delta_{(i)} K_h(x - X_{(i)})}{\sum_{k=i}^{n} K_h(x - X_{(k)})} \prod_{l=1}^{i-1} \left[1 - \frac{K_h(x - X_{(l)})}{\sum_{k=l}^{n} K_h(x - X_{(k)})} \right]^{\delta_{(l)}}. \quad (3.3)$$

Tel qu'il a été mentionné précédemment dans cette section, l'estimateur du quantile conditionnel d'ordre α peut être trouvé en remplaçant le terme $\hat{P}(Y = y_{(i)}|X = x)$ de l'équation (3.1) par un autre terme $\Delta_{KMG}(Y_{(i)}|x)$. Cet estimateur est donc obtenu en résolvant

$$\arg \min_{\beta_0,\ \beta_1} \sum_{i=1}^{n} \rho_\alpha \{Y_{(i)} - \beta_0 - \beta_1 (X_{(i)} - x)\}\ \Delta_{KMG}(Y_{(i)}|x). \quad (3.4)$$

En voyant cette dernière expression, on remarque qu'elle ressemble énormément à l'équation (2.3). Pour être plus précis, l'unique différence réside dans le fait que le terme $W_i(x)$ était présent dans l'équation (2.3), alors qu'il a été remplacé par $\Delta_{KMG}(Y_{(i)}|x)$ dans l'équation ci-dessus (3.4) et que la variable explicative, la variable dépendante ainsi que la variable de censure ont été ordonnées selon la variable dépendante.

Afin de clore cette section, il serait intéressant de voir, comme on le faisait aux chapitres antérieurs, comment se comporte le biais et la variance pour l'estimateur du Kaplan-Meier généralisé. Or, aucun document concernant ces deux mesures d'évaluation de la qualité d'un estimateur n'a été publié jusqu'à ce jour. Dans ce cas-ci, il faudra donc uniquement se contenter de l'étude par voie de simulation qui sera effectuée au chapitre 4.

3.1.2 Méthode des poids proposés par Stute

Une autre méthode qui permet également d'estimer les percentiles conditionnels lorsque la variable réponse est censurée à droite est la méthode des poids proposés par Stute (Stute, 1993). Cette méthode a d'ailleurs fait l'objet d'un article qui a tout récemment été publié par Gannoun *et al.* (2005). La définition proprement dite de cet estimateur ainsi que ses propriétés, qui seront développés dans le même ordre dans ce mémoire, y sont traités.

Dans un premier temps, il faut mentionner le fait que la méthode employant une optimisation pondérée par les poids proposés par Stute est très similaire à la méthode du Kaplan-Meier généralisé, explorée à la section précédente. En effet, cette méthode est aussi basée sur la méthode localement linéaire et les calculs s'effectuent de la même

manière que ceux trouvés à l'aide de la méthode du KMG, à l'exception du calcul des poids assignés aux $Y_{(i)}$. En réalité, ces nouveaux sauts sont simplement obtenus en multipliant le facteur $K_h(x - X_{(i)})$ par les sauts obtenus à l'aide de la méthode du Kaplan-Meier ordinaire (KM). Pour être en mesure d'effectuer cette multiplication, il faut bien entendu savoir que les sauts obtenus par la méthode du KM sont pareils à ceux trouvés par la méthode du KMG, sauf que chacun des termes $K_h(x - X_{(i)})$ qui entrent dans la formule (3.3) du $\Delta_{KMG}(Y_{(i)}|x)$ sont remplacés par la valeur 1. Plus précisément, l'estimateur de la fonction de survie du Kaplan-Meier ordinaire est

$$\hat{S}_{KM}(Y_{(i)}) = \prod_{l=1}^{i} \left[\frac{n-l}{n-l+1} \right]^{\delta_{(l)}},$$

où la variable indicatrice des temps de censure, $\delta_{(l)}$, est définie comme à la section précédente. Par la suite, en procédant de la même façon qu'à la section précédente, la valeur de $\Delta_{KM}(Y_{(i)})$, le saut dans le Kaplan-Meier ordinaire au temps $Y_{(i)}$, peut être obtenue :

$$\Delta_{KM}(Y_{(i)}) = \frac{\delta_{(i)}}{n-i+1} \prod_{l=1}^{i-1} \left[\frac{n-l}{n-l+1} \right]^{\delta_{(l)}}.$$

Ainsi, puisqu'il a précédemment été expliqué que les $\Delta_{Stute}(Y_i|x)$ sont donnés par $\Delta_{Stute}(Y_i|x) = K_h(x - X_i)\Delta_{KM}(Y_i)$, l'expression pour ce saut est trouvée de la façon qui suit :

$$W_{i:n}(x) = \Delta_{Stute}(Y_{(i)}|x) = K_h(x - X_{(i)}) \frac{\delta_{(i)}}{n-i+1} \prod_{l=1}^{i-1} \left[\frac{n-l}{n-l+1} \right]^{\delta_{(l)}}.$$

On se retrouve alors avec le problème de minimisation

$$\arg \min_{\beta_0,\, \beta_1} \sum_{i=1}^{n} \rho_\alpha \{ Y_{(i)} - \beta_0 - \beta_1 (X_{(i)} - x) \}\, \Delta_{Stute}(Y_{(i)}|x). \tag{3.5}$$

Une fois de plus, on prend comme estimateur $\hat{q}_{n,\alpha}(x) = \hat{\beta}_0$, où $\hat{\beta}_0$ est un des éléments de la solution au problème de minimisation (3.5). Tel qu'on l'a mentionné au début de cette section, un article traitant des propriétés de l'estimateur utilisant les poids proposés par Stute a tout récemment été publié par Gannoun *et al.* (2005). Pour le moment, voyons la valeur du biais et celle de la variance que Gannoun *et al.* (2005) ont calculées. Soient $f_2(\cdot)$ la fonction de densité marginale des X_i et $G(\cdot)$ la fonction de répartition des variables aléatoires des temps de censure C_i, qui sont définies de sorte que

$$T_i = \min(Y_i, C_i),$$

où T_i est la variable des temps observés. Pour faire un lien avec l'approche qui a été utilisée au début de ce chapitre, T_i représente d'une part, soit la valeur réelle de la variable réponse, c'est-à-dire (Y_i), ou bien, d'autre part, une borne inférieure pour celle-ci. On assiste à la dernière situation lorsque C_i est moins élevé que ce Y_i. De plus, soient $\beta_0(x) = q_\alpha(x)$ et $\beta_1(x) = d(q_\alpha(x))/dx$. Alors, sous certaines conditions, le biais et la variance de l'estimateur $\hat{q}_{n,\alpha}(x)$ de Y_i sachant $\{X_i = x\}$ peuvent être obtenus. En premier lieu, la formule pour le biais est décrite :

$$Biais[\hat{q}_{n,\alpha}(x) \mid x_1, ...x_n] = \frac{h^2\beta_1(x)f_2'(x)\mu_2}{\mu_0} + o(h^2), \tag{3.6}$$

où

$$\mu_0 = \int K(u)du, \qquad \mu_2 = \int u^2K(u)du$$

et où le terme $f_2'(x)$ représente la dérivée première de $f_2(x)$ par rapport à x, c'est-à-dire $f_2'(x) = df_2(x)/dx$. En examinant d'un peu plus près les deux formules ci-dessus, on remarque que, quelque soit le choix du noyau, la valeur de μ_0 est toujours de 1. De plus, lorsque le choix du noyau est fixé à un noyau gaussien, on se retrouve avec l'égalité $\mu_2 = 1$. Cette dernière égalité découle du fait que le noyau gaussien est de moyenne 0 et de variance 1. D'autre part, soit

$$\sigma^2 = \frac{b(x)v_2}{\mu_0^2 f_2(x)[\phi''(0|x)]^2},$$

où

$$v_2 = \int K^2(u)du < \infty, \qquad b(x) = E\{Y^2[1 - G(Y)]^{-1}|X = x\}$$

et

$$\phi(t|x) = E[\rho_\alpha(T - \beta_0(x) + t)|X = x],$$

où l'espérance E est calculée par rapport à $F(y|x)$. Une formule pour la variance de l'estimateur des percentiles conditionnels par la méthode utilisant la pondération proposée par Stute s'écrit comme suit :

$$Var[\hat{q}_{n,\alpha}(x) \mid x_1, ...x_n] = \frac{\sigma^2}{nh}. \tag{3.7}$$

Pour sa part, l'EQM asymptotique est obtenue de la manière qui suit :

$$EQMA = h^4[\beta_1(x)f_2'(x)\mu_0^{(-1)}\mu_2]^2 + \frac{1}{nh}\frac{b(x)v_2}{\mu_0^2 f_2(x)[\phi''(0|x)]^2}.$$

Après avoir dérivé cette équation par rapport à h, la fenêtre de lissage optimale est trouvée :

$$h_{opt} = \left\{ \frac{1}{4} \frac{b(x)v_2}{f_2(x)[\phi''(0|x)]^2[\beta_1(x)f_2'(x)\mu_2]^2} \right\} n^{-1/5}.$$

Comme pour toutes les fenêtres de lissage optimales vues jusqu'à maintenant dans ce mémoire, on remarque que cette fenêtre de lissage est inversement proportionnelle au facteur $n^{1/5}$, ce qui concorde une autre fois parfaitement avec le point 3 de la section 1.2.1. En effet, ce point spécifiait que la fenêtre de lissage optimale est généralement inversement proportionnelle à une certaine puissance de n, c'est-à-dire $1/5$ dans le cas actuel.

Selon Gannoun *et al.* (2005), cette méthode présente de nombreux avantages. Entre autres, elle amènerait une réduction du biais de l'estimateur $\hat{q}_{n,\alpha}(x)$ et, de surcroît, ce biais ne varierait pas lorsque les estimateurs sont évalués à des points d'estimation x qui sont localisés aux extrémités. Mais la validité de cette affirmation sera testée au chapitre 4, à l'aide de simulations.

3.2 La méthode de Bowman et Wright

Une troisième méthode utilisant une approche différente des deux méthodes vues précédemment est la méthode de Bowman et Wright (2000). Celle-ci sera d'ailleurs comparée aux deux autres méthodes par voie de simulation au chapitre 4. Tout d'abord, il faut savoir que l'estimateur de la fonction de survie à la base de cette méthode est encore une fois le Kaplan-Meier généralisé, c'est-à-dire qu'il s'écrit comme suit :

$$\hat{S}_{KMG}(Y_{(i)} \mid x) = \prod_{l=1}^{i} \left[1 - \frac{K_h(x - X_{(l)})}{\sum_{k=l}^{n} K_h(x - X_{(k)})} \right]^{\delta_{(l)}}.$$

Or, contrairement aux deux méthodes précédentes, il n'est nullement nécessaire de calculer des sauts pour obtenir l'estimateur par la méthode de Bowman et Wright (2000). En effet, cet estimateur est trouvé en résolvant le problème de minimisation qui suit :

$$\min_{\beta_0,\ \beta_1} \sum_{i=1}^{n} \delta_i \{Y_{(i)} - \beta_0 - \beta_1(q_i^{(x)} - q)\}^2 K_g(q_i^{(x)} - q), \tag{3.8}$$

où $q_i^{(x)} = \hat{F}_{KMG}(Y_{(i)} \mid x)$ et q est l'ordre du percentile pour lequel on souhaite obtenir une estimation, par exemple lorsqu'on désire estimer le 25^e percentile, on prend $q = 0.25$.

De plus, g est le paramètre représentant une fenêtre de lissage, possiblement différente de celle de l'estimateur du Kaplan-Meier généralisé, qui est quant à elle dénotée h.

Le principal avantage de cette méthode est qu'il existe une solution explicite au problème de minimisation (3.8) ; le calcul par itérations n'est donc pas nécessaire dans ce cas-ci. En effet, un calcul explicite pour ce problème de minimisation (voir annexe A.2) conduit à la solution suivante :

Soit

$$B_i(q) = \delta_i K_g(q_i^{(x)} - q),$$

alors

$$\hat{\beta}_0 = \frac{\sum_{i=1}^n B_i(q) Y_i}{\sum_{i=1}^n B_i(q)} - \hat{\beta}_1 \frac{\sum_{i=1}^n B_i(q)(q_i^{(x)} - q)}{\sum_{i=1}^n B_i(q)} \tag{3.9}$$

et

$$\hat{\beta}_1 = \frac{\sum_{i=1}^n B_i(q) Y_i (q_i^{(x)} - q) - \frac{\sum_{i=1}^n B_i(q) Y_i}{\sum_{i=1}^n B_i(q)} \sum_{i=1}^n B_i(q)(q_i^{(x)} - q)}{\sum_{i=1}^n B_i(q)[(q_i^{(x)} - q)]^2 - \frac{[\sum_{i=1}^n B_i(q)(q_i^{(x)} - q)]^2}{\sum_{i=1}^n B_i(q)}}. \tag{3.10}$$

Il va de soit que cette méthode ne comporte pas que des avantages, mais également certains inconvénients. Principalement, contrairement aux deux méthodes précédentes où un seul paramètre de lissage, h, était présent dans les problèmes de minimisation (3.4) et (3.5), le problème de minimisation pour la méthode de Bowman et Wright (2000) contient deux fenêtres de lissage, h et g. Ceci rend l'estimation un peu plus compliquée puisqu'il faut maintenant trouver des valeurs raisonnables pour non pas une, mais deux fenêtres de lissage. De plus, les propriétés de l'estimateur pour cette troisième méthode ne sont pas connues jusqu'à ce jour puisque Bowman et Wright (2000) n'en dérivent que quelques propriétés en l'absence de censure.

3.3 Exemple

Afin de s'assurer que l'information donnée antérieurement dans ce chapitre a bien été saisie, les deux méthodes qui ont été présentées à la section 3.1, plus précisément, la méthode du KMG et la méthode des poids proposés par Stute, seront appliquées à un jeu de données. Toutefois, la méthode de Bowman et Wright (2000) ne sera appliquée à aucun exemple. En effet, cette dernière méthode nécessite de trouver deux fenêtres

de lissage, contrairement aux deux autres méthodes pour lesquelles une seule fenêtre de lissage devait être obtenue, et, de plus, aucun calcul de saut ne doit être effectué pour cette méthode, ce qui la rend évidemment moins comparable aux deux autres. Bien entendu, les données qui servent d'exemple dans ce chapitre ne peuvent être les mêmes qu'aux deux chapitres précédents étant donné qu'une variable de censure doit maintenant être présente dans le jeu de données qui sera utilisé.

L'ensemble de données qui a donc finalement été choisi afin de servir d'exemple pour ce chapitre est un jeu de données qui provient du livre de Klein et Moeschberger (2003, section 1.7) et qui présente des données observées sur 863 patients ayant subi une greffe de rein. L'intérêt est en fait porté sur le temps de décès de ces derniers à partir du moment où ils ont subi la greffe. La répartition du type de personnes faisant partie de cette étude, selon le sexe et la race, est la suivante : 432 hommes blancs, 92 hommes noirs, 280 femmes blanches et 59 femmes noires. Dans cette section, une analyse sera effectuée pour les hommes blancs et une autre le sera pour les femmes blanches. Les hommes et les femmes noirs ne feront donc pas partie de la présente étude étant donné qu'ils ne sont pas assez nombreux pour estimer des percentiles de régression de façon non paramétrique. Des régressions non paramétriques des percentiles avec variable réponse censurée à droite du temps de décès suite à une greffe de rein des patients en fonction de leur âge au moment de cette greffe seront donc effectuées. Pour être plus précis, quatre régressions seront produites : une pour les hommes blancs et une autre pour les femmes blanches pour ce qui est de la méthode du KMG et les deux mêmes pour la méthode des poids proposés par Stute. Il faut également souligner le fait que les analyses seront produites uniquement à partir des données pour lesquelles l'âge de l'individu au moment de la greffe se situe entre 30 et 60 ans, puisque c'est dans cette classe d'âge que la quantité de données est la plus abondante. En effet, pour obtenir de bons résultats en régression non paramétrique, il est important qu'il y ait un grand nombre de données. Tout comme aux chapitres antérieurs, les programmes en langage C et en langage R ayant été utiles à cette analyse sont présentés à l'annexe C.

Pour débuter l'analyse, deux fichiers, un pour chacun des deux groupes d'intérêt, ont d'abord été créés. Ensuite, tel qu'il a été mentionné à maintes reprises dans les sections précédentes, une fenêtre de lissage optimale doit être évaluée avant que l'on soit en mesure de procéder aux analyses. Il y a donc quatre fenêtres de lissage qui ont été obtenues, une pour chacune des régressions. La méthode du double noyau, qui a été utilisée aux deux chapitres précédents, a été conservée dans cette section afin de choisir ces fenêtres de lissage. Les raisons expliquant le choix de cette méthode sont les mêmes que celles qui ont déjà été mentionnées aux chapitres 1 et 2, c'est-à-dire que la fenêtre de lissage optimale calculée dans ce chapitre pour la méthode des poids proposés par Stute n'est pas utilisée dans cet exemple, car elle implique des quantités inconnues qui

doivent être estimées, comme par exemple le calcul de $f_2'(x)$. Ainsi, au lieu de devoir utiliser des méthodes adaptatives, nous préférons procéder par la méthode du double noyau. Donc, dans le cas de la méthode de Kaplan-Meier généralisé, les estimateurs ayant été calculés sont, d'une part, l'estimateur par la méthode localement linéaire, qui est trouvé en solutionnant le problème de minimisation (3.4), soit

$$\arg \min_{\beta_0,\ \beta_1} \sum_{i=1}^{n} \rho_\alpha\{Y_{(i)} - \beta_0 - \beta_1(X_{(i)} - x)\}\ \Delta_{KMG}(Y_{(i)}|x)$$

et en posant $\hat{q}^1_{n,\alpha}(x) = \hat{\beta}_0$ et, d'autre part, l'estimateur par la méthode localement quadratique, qui est obtenu en minimisant

$$\arg \min_{\beta_0,\ \beta_1} \sum_{i=1}^{n} \rho_\alpha\{Y_{(i)} - \beta_0 - \beta_1(X_{(i)} - x) - \beta_2(X_{(i)} - x)^2\}\ \Delta_{KMG}(Y_{(i)}|x)$$

et en posant $\hat{q}^2_{n,\alpha}(x) = \hat{\beta}_0$. Dans le cas de la méthode par les poids proposés par Stute, l'estimateur par la méthode localement linéaire est obtenu en trouvant la solution au problème de minimisation (3.5), c'est-à-dire

$$\arg \min_{\beta_0,\ \beta_1} \sum_{i=1}^{n} \rho_\alpha\{Y_{(i)} - \beta_0 - \beta_1(X_{(i)} - x)\}\ \Delta_{Stute}(Y_{(i)}|x)$$

et en posant par la suite $\hat{q}^1_{n,\alpha}(x) = \hat{\beta}_0$, alors que l'estimateur par la méthode localement quadratique est trouvé en résolvant

$$\arg \min_{\beta_0,\ \beta_1} \sum_{i=1}^{n} \rho_\alpha\{Y_{(i)} - \beta_0 - \beta_1(X_{(i)} - x) - \beta_2(X_{(i)} - x)^2\}\ \Delta_{Stute}(Y_{(i)}|x)$$

et en posant $\hat{q}^2_{n,\alpha}(x) = \hat{\beta}_0$.

Pour chacune des deux méthodes, la minimisation de l'expression ci-dessous doit par la suite être obtenue

$$\sum_{x \in \mathcal{G}} [\hat{q}^1_{n,\alpha}(x) - \hat{q}^2_{n,\alpha}(x)]^2.$$

Ainsi, pour chacune des trois valeurs de α, plus précisément 0.25, 0.50 et 0.75, la solution de cette dernière équation est trouvée pour plusieurs valeurs de h. Finalement, pour un α donné, la valeur de h qui a conduit à l'obtention de la quantité la plus minime est celle qui sera conservée. Les valeurs de h qui ont finalement été retenues sont présentées au tableau 3.1 ci-dessous.

Ces valeurs étant déterminées, il devient alors possible d'obtenir les courbes des percentiles des temps de décès suite à une greffe du rein en fonction de l'âge pour les

Methode	**Domaine**	α=0.25	α=0.5	α=0.75
KMG	**femmes**	24	18	2
	hommes	4	13	2
Stute	**femmes**	22	13	2
	hommes	3	29	2

TAB. 3.1 – *Meilleures fenêtres de lissage obtenues pour chacune des deux méthodes pour les femmes blanches et pour les hommes blancs.*

femmes blanches et les hommes blancs. Pour y arriver, les estimateurs trouvés par la méthode localement linéaire sont simplement calculés en plusieurs points de la grille, pour chacune des trois valeurs de α et des fenêtres de lissage h leur correspondant, et sont par la suite reliés. Ces étapes sont répétées pour les deux types de personnes, les hommes et les femmes, et, pour chacun d'entre eux, pour les méthodes du KMG et des poids pondérés par Stute. Les courbes de régression des percentiles ainsi obtenues sont montrées à la figure 3.1, pour les estimations obtenues par la méthode de Kaplan-Meier généralisé, et à la figure 3.2, pour celles trouvées par la méthode des poids proposés par Stute. Avant de s'intéresser à ces graphiques, il serait important de mentionner le fait que, sur ces derniers, les pics ayant la forme de stalagtites très longs et étroits ne sont pas causés par des défauts des estimateurs, mais plutôt par des instabilités numériques dans les optimisations. Pour être plus précis, la fonction `optim`() du langage R a été incapable de faire converger l'estimateur en ces points.

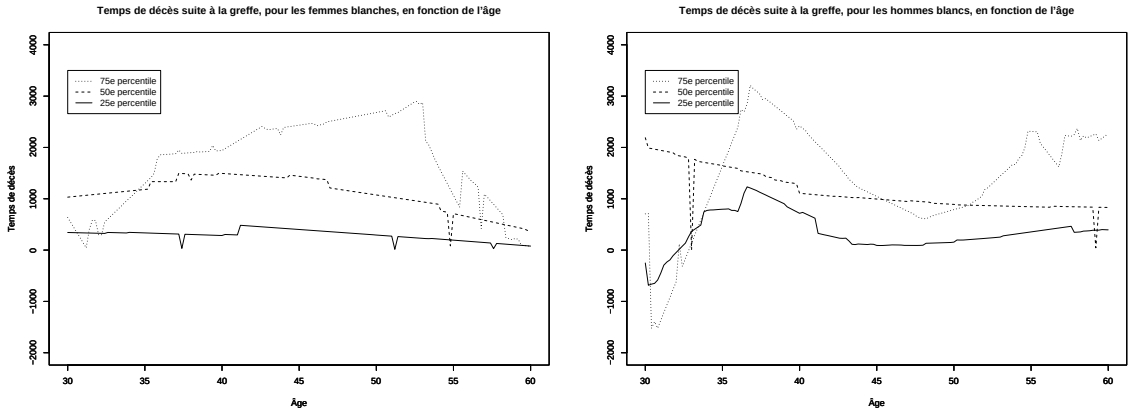


FIG. 3.1 – *Régression non paramétrique des* 25^e, 50^e *et* 75^e *percentiles par la méthode de Kaplan-Meier Généralisé produite à partir des paramètres de lissage du tableau 3.1 pour le temps de décès suite à une greffe de rein en fonction de l'âge pour (a) les femmes blanches (b) les hommes blancs.*

La comparaison de ces graphiques, pour un groupe d'individus déterminé, permet

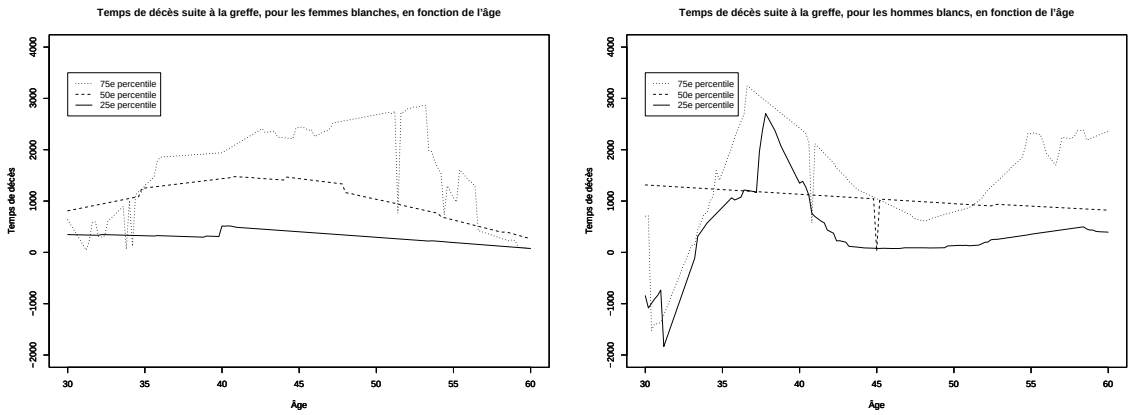


FIG. 3.2 – *Régression non paramétrique des* 25^e, 50^e *et* 75^e *percentiles par la méthode des poids proposés par Stute produite à partir des paramètres de lissage du tableau 3.1 pour le temps de décès suite à une greffe de rein en fonction de l'âge pour (a) les femmes blanches (b) les hommes blancs.*

de constater que les courbes de régression obtenues par les deux différentes méthodes ont une forme plutôt semblable, c'est-à-dire qu'elles semblent être relativement linéaires et légèrement décroissantes.

Par contre, au regard de ces quatre graphiques, on s'aperçoit que les fenêtres de lissage choisies ne semblent pas être très aptes à bien décrire les données observées, en particulier dans le cas des variables mesurées sur les hommes. En effet, chez les hommes, les courbes de lissage présentent de grandes fluctuations, alors que dans le cas des femmes, il semble y avoir moins de problèmes de ce genre, à l'exception de la courbe du 75^e percentile. Pour cette raison, quatre autres graphiques, pour lesquels la fenêtre de lissage h semble être plus appropriée, ont été obtenus. Ces quatre nouveaux graphiques ont en fait été trouvés en fixant diverses valeurs pour le paramètre de lissage h et en observant, pour chacune d'entre elles, le graphique ainsi trouvé. Les fenêtres de lissage produisant les graphiques qui semblent le mieux décrire la distribution des données sont donc finalement conservées et ce sont elles qui ont été prises en considération dans la création des graphiques se trouvant aux figures 3.3 et 3.4 ci-dessous. Ces données sont pour leur part exprimées au tableau 3.2.

Ces quatre nouveaux graphiques semblent en effet mieux décrire la relation existant entre le temps de décès suite à une greffe de rein des patients en fonction de leur âge au moment de cette greffe, étant donné qu'il y a beaucoup moins de grandes fluctuations sur ces graphiques que sur les précédents. Chez les hommes, ce temps de décès semble être légèrement décroissant en fonction de l'âge, alors que pour les femmes, le temps de décès semble augmenter jusqu'à l'âge d'environ quarante-deux ans et ensuite diminuer à partir

Methode	**Domaine**	α=0.25	α=0.5	α=0.75
KMG	**femmes**	24	18	20
	hommes	12	13	15
Stute	**femmes**	18	13	18
	hommes	20	29	20

TAB. 3.2 – *Meilleures fenêtres de lissage obtenues en visualisant les graphiques, et ce, pour chacune des deux méthodes pour les femmes blanches et pour les hommes blancs.*

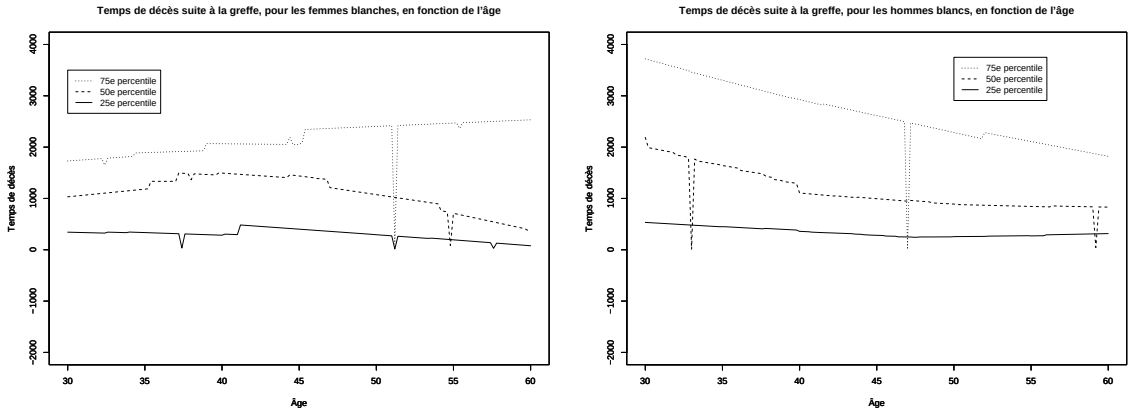


FIG. 3.3 – *Régression non paramétrique des* 25^e, 50^e *et* 75^e *percentiles par la méthode de Kaplan-Meier Généralisé produite à partir des paramètres de lissage du tableau 3.2 pour le temps de décès suite à une greffe de rein en fonction de l'âge pour (a) les femmes blanches (b) les hommes blancs.*

de cet âge. Ces remarques peuvent d'ailleurs être appuyées par Klein et Moeschberger (2003, p. 274), puisqu'une étude de ce jeu de données y a également été effectuée. Dans cette étude, la covariable continue "âge" a été traitée comme une variable binaire et, pour ce faire, un point de coupure de cette variable a été déterminé. Il a donc ensuite été possible de conclure s'il y a une différence significative entre les temps de décès pour les gens plus âgés que ce point de coupure et ceux qui le sont moins. À cette fin, un risque relatif a pu être estimé afin de voir le risque relatif de décès des personnes plus âgées que ce point de coupure par rapport aux plus jeunes. Ainsi, un risque relatif estimé de 2.6 et un p-value < 0.001 ont été obtenus pour les hommes blancs, alors que pour les femmes blanches, un risque relatif de 4.4 et un p-value de 0.001 ont été trouvés. Ces résultats suggèrent donc que, autant pour les hommes blancs que pour les femmes blanches, il y a un risque de décès plus élevé pour les gens plus âgés que pour les plus jeunes, tout comme les figures précédentes le montraient.

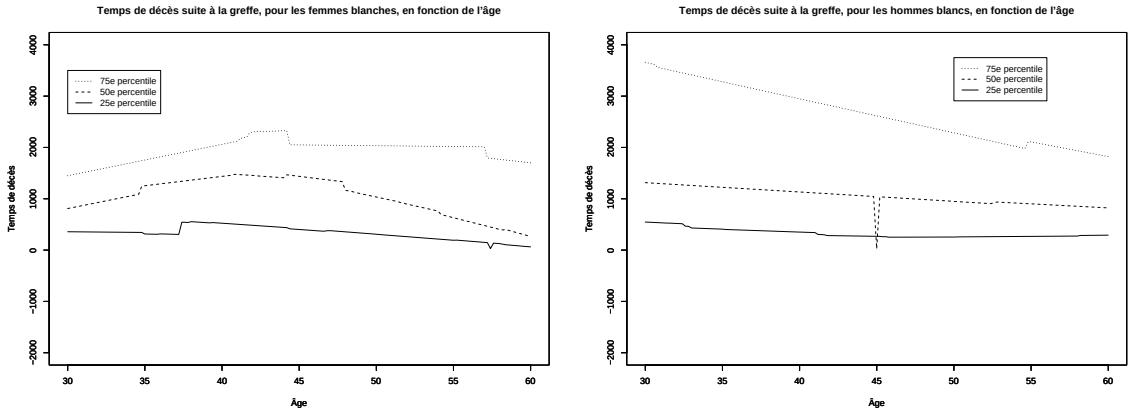


FIG. 3.4 – *Régression non paramétrique des* 25^e, 50^e *et* 75^e *percentiles par la méthode des poids proposés par Stute produite à partir des paramètres de lissage du tableau 3.2 pour le temps de décès suite à une greffe de rein en fonction de l'âge pour (a) les femmes blanches (b) les hommes blancs.*

Chapitre 4

Les simulations

Au chapitre 3, trois méthodes de régression non paramétrique des percentiles avec variable réponse censurée à droite ont été présentées : celle employant l'estimateur de Kaplan-Meier généralisé, celle employant une optimisation pondérée par les poids Stute et celle proposée par Bowman et Wright (2000). Or, seules la description de chacune de ces méthodes et quelques formules de mesures de la qualité de l'estimateur trouvées par la méthode pondérée par les poids Stute y ont été présentés. Dans le présent chapitre, ces trois méthodes d'estimation des percentiles seront comparées à l'aide de résultats ayant été obtenus par voie de simulations et, si cela s'avère possible, on tentera également de déterminer la méthode qui conduit aux meilleurs résultats.

La section 4.1 porte exclusivement sur le modèle qui a été utilisé afin de simuler les données ainsi que les paramètres lui étant associés. Les résultats obtenus pour les trois méthodes à partir des simulations seront par ailleurs présentés à la section 4.2. Les différentes fenêtres de lissage trouvées et ayant servi aux simulations (section 4.2.1), les résultats des simulations (section 4.2.2) ainsi qu'une analyse de ces résultats (section 4.2.3) sont les sujets qui y seront traités.

4.1 Le modèle et ses paramètres

Avant d'être en mesure de discuter de la performance de chacune des trois méthodes d'estimation des percentiles nommées ci-dessus, il est tout d'abord indispensable de procéder à la simulation d'un échantillon de valeurs. Sous cette optique, le choix du modèle de simulation s'est arrêté sur le modèle donné par Hall *et al.* (1999). Le choix de

ce modèle se justifie par le fait que ce dernier a déjà été utilisé dans une étude similaire à celle présentée dans ce mémoire, mais surtout parce qu'il est composé de pics. En effet, la présence de ces pics dans le modèle permet de vérifier si les trois méthodes comparées ont la capacité de tenir compte de ces pics pour effectuer les estimations ou si, au contraire, ces derniers seront tout simplement aplatis. Les paramètres composant ce modèle sont décrits dans les lignes qui suivent. Dans un premier temps, il faut savoir que la variable exogène X_i suit une loi normale,

$$X_i \sim \text{Normale}(0, \sigma^2),$$

alors que la variable endogène Y_i suit une loi log normale,

$$Y_i = \exp\{2\sin(\pi x) + \epsilon_i\},$$

où

$$\epsilon_i \sim \text{Normale}(0, \sigma^2).$$

Par ailleurs, la variable des temps de censure C_i suit une loi exponentielle :

$$C_i \sim \text{exponentielle } (\lambda),$$

où λ représente le taux de panne des temps de censure. Il devient donc clair que la moyenne des temps de censure est quant à elle dénotée $1/\lambda$. Ainsi, la variable des temps observés T_i est

$$T_i = \min(Y_i, C_i)$$

et la variable indicatrice des temps de censure δ_i est

$$\delta_i = I(Y_i \leq C_i) = I(Y_i = T_i).$$

La forme de l'échantillon final utilisé pour comparer les trois méthodes d'estimation des percentiles est alors celle qui suit :

$$(T_i, \delta_i, X_i), \qquad i = 1, 2, ..., n.$$

Afin de comparer les trois méthodes décrites au chapitre 3, plusieurs échantillons de cette forme ont été produits par 1000 simulations pour différentes valeurs de la taille échantillonnale ($n = 200$ et $n = 500$), pour différents taux de panne des temps de censure ($\lambda = 0.1$ et $\lambda = 0.6$) et pour différents seuils ($\alpha = 0.25, 0.5$ et 0.75). Par ailleurs, la même variance, $\sigma^2 = 0.5$, a été conservée tout au long de cette étude. Les données ont toutes été simulées à partir du logiciel R, alors que les trois méthodes ont été programmées dans deux logiciels : en C et en R. Pour être plus clair, le code

C a été invoqué par un programme R. Il est également indispensable de spécifier que les estimations ont été calculées en seulement certains points, ceux faisant partie de la grille $\mathcal{G} = \{-1.0, -0.9, \ldots, 0.9, 1.0\}$. Enfin, pour chacune des douze combinaisons possibles de n, λ et α, on a tenté de déterminer la méthode qui semble produire les meilleures estimations en calculant, pour chacune d'entre elles, le biais, la variance, l'erreur quadratique moyenne (EQM),

$$EQM(x) \quad = \quad \sum_{b=1}^{1000} [\hat{g}_{\alpha}^{(b)}(x) - g_{\alpha}(x)]^2/1000,$$

ainsi que l'erreur quadratique moyenne intégrée ($EQMI$),

$$EQMI \quad = \quad \sum_{x \in \mathcal{G}} EQM(x),$$

où $\mathcal{G} = \{-1.0, -0.9, \ldots, 0.9, 1.0\}$ représente la grille d'évaluation en différents points de l'estimateur.

Il pourrait être très intéressant de voir l'allure de certaines courbes de percentiles théoriques pour ce modèle. Ainsi, les 25^e, 50^e et 75^e percentiles théoriques pour les valeurs de X comprises dans l'intervalle [-1,1] sont illustrés à la figure 4.1. L'équation

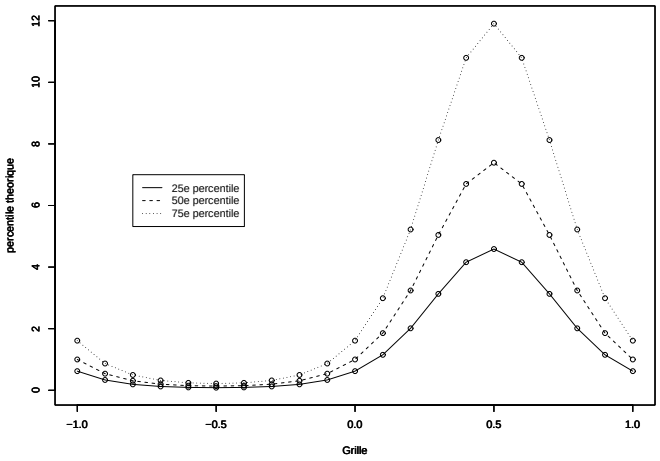


FIG. 4.1 – *Graphique des 25^e, 50^e et 75^e percentiles théoriques en fonction de la grille.*

permettant d'obtenir ces percentiles théoriques est donnée par

$$q_{\alpha}(x) = e^{2\sin(\pi x) + \sigma Z_{\alpha}},$$

où Z_{α} correspond à la valeur que prend une variable aléatoire suivant une loi normale standard lorsque l'aire sous la courbe de la fonction de distribution de cette variable,

à gauche de cette valeur, est de α. Par exemple, pour un seuil $\alpha = 0.25$, Z_α prend approximativement la valeur -0.675. On remarque sur cette figure que la courbe des percentiles théoriques est en forme de pic au point 0.5 de la grille des valeurs de X, ce qui est, comme il a déjà été mentionné dans cette section, la raison principale du choix de ce modèle pour la simulation des données.

4.2 Les résultats

Dans cette section, les résultats obtenus pour les trois méthodes de régression des percentiles avec variable réponse censurée à droite vues au chapitre 3 seront divulgués. Ainsi, la première sous-section traite des fenêtres de lissage permettant d'optimiser les méthodes qui ont été choisies. Par la suite, les résultats seront présentés sous forme de graphique pour différentes valeurs des paramètres discutés à la section 4.1, alors qu'à la sous-section 4.2.3, un jugement d'ensemble sera porté sur ces résultats.

4.2.1 Choix des fenêtres

Avant d'entreprendre l'estimation des percentiles conditionnels, il faut tout d'abord tenter de déterminer la valeur de la fenêtre de lissage h qui mènera à la minimisation de l'$EQMI$. À cette fin, une petite étude permettant d'évaluer cette valeur h a été produite pour chacune des trois méthodes. Pour y arriver, 500 simulations du modèle présenté au début de ce chapitre ont été effectuées, pour différentes valeurs du paramètre h, et, pour chacune de ces valeurs, la différence entre l'estimateur trouvé et la valeur réelle était calculée. La moyenne de la somme en tout point de la grille de cette différence élevée au carré,

$$\frac{1}{500}\sum_{b=1}^{500}\left[\sum_{x\in\mathcal{G}}(\hat{q}_\alpha^b(x) - q_\alpha(x))^2\right], \tag{4.1}$$

était ensuite notée. Ainsi, on cherchait la valeur de la fenêtre qui permettait d'obtenir la moyenne la plus faible. On remarque que pour trouver la fenêtre de lissage optimale, le calcul est effectué à l'aide de 500 simulations, contrairement au calcul des résultats finaux, pour lequel 1000 simulations sont effectuées. En fait, cela permet d'économiser une quantité accrue de temps, puisque toutes ces simulations sont extrêmement longues à faire rouler en R. De plus, on se permet une telle souplesse étant donné que la recherche d'une fenêtre de lissage optimale h n'est en fait qu'une simple analyse exploratoire. Ce point étant mis au clair, il faut par la suite comprendre que l'on obtient, pour

chacune des méthodes du KMG et des poids proposés par Stute, douze fenêtres de lissage optimales. En effet, une fenêtre de lissage optimale est obtenue pour chacun des croisements des trois valeurs possibles pour le seuil ($\alpha = 0.25, 0.5$ et 0.75), des deux valeurs de la taille échantillonnale ($n = 200$ et $n = 500$) et des deux valeurs de la moyenne des temps de censure ($1/\lambda = 10$ et $1/\lambda = 5/3$). Dans le cas de la méthode de Bowman et Wright (2000), on comprend que 24 fenêtres de lissage sont obtenues, car deux fenêtres de lissage (h et g) sont nécessaires dans chacun des douze cas. Les valeurs de ces deux fenêtres de lissage ont été les plus ardues à obtenir. En fait, il fallait procéder de la façon suivante : pour un paramètre h, on fait varier la valeur de g et on note alors la valeur g qui permet d'obtenir la plus petite moyenne décrite plus haut dans ce paragraphe. Mais cela devait évidemment être répété pour plusieurs valeurs de h. De toutes ces situations, les valeurs h et g conservées était celles qui avaient produit, encore une fois, la moyenne la plus minime qui soit, parmi toutes les tentatives effectuées. Les valeurs retenues pour les fenêtres de lissage optimales pour ces trois méthodes sont donc finalement présentées aux tableaux 4.1 et 4.2 qui suivent.

TAB. 4.1 – *Meilleures fenêtres de lissage obtenues pour chacune des 3 méthodes avec un taux de panne des temps de censure de λ =0.1.*

		α					
		0.25		0.5		0.75	
Methode	**Fenetre**	**n**=200	**n**=500	**n**=200	**n**=500	**n**=200	**n**=500
KMG	**h**	0.13	0.1	0.12	0.11	0.13	0.11
Stute	**h**	0.13	0.11	0.14	0.11	0.18	0.15
BW	**h**	0.15	0.15	0.13	0.1	0.1	0.1
	g	0.07	0.07	0.05	0.04	0.05	0.04

TAB. 4.2 – *Meilleures fenêtres de lissage obtenues pour chacune des 3 méthodes avec un taux de panne des temps de censure de λ =0.6.*

		α					
		0.25		0.5		0.75	
Methode	**Fenetre**	**n**=200	**n**=500	**n**=200	**n**=500	**n**=200	**n**=500
KMG	**h**	0.16	0.14	0.18	0.15	0.22	0.19
Stute	**h**	0.18	0.15	0.21	0.13	0.28	0.21
BW	**h**	0.18	0.15	0.16	0.13	0.15	0.12
	g	0.05	0.06	0.06	0.07	0.08	0.08

La comparaison de ces fenêtres de lissage porte à réaliser que lorsque les autres

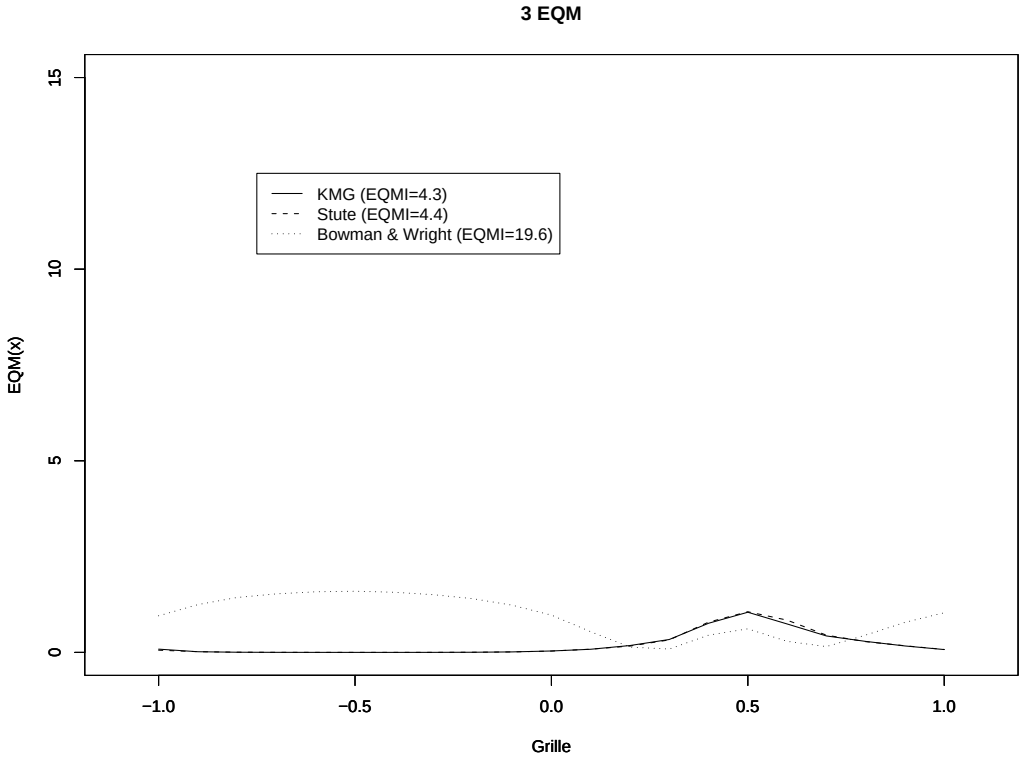
paramètres sont fixés, la fenêtre de lissage optimale obtenue par la méthode des poids proposés par Stute est supérieure ou égale à celle obtenue par la méthode du KMG, et ce, dans tous les cas, à l'exception de celui où $\lambda = 0.6, \alpha = 0.5$ et $n = 500$. De plus, l'examen des valeurs trouvées par la méthode de Bowman et Wright (2000) permet de constater que la deuxième fenêtre de lissage (g) est toujours inférieure à la fenêtre h. En effet, le paramètre g se situe toujours entre 0.04 et 0.08, alors que la valeur de h est toujours comprise entre 0.1 et 0.15 dans le cas où le taux de panne des temps de censure, λ, est de 0.1 et entre 0.12 et 0.18 lorsque ce taux de panne des temps de censure passe à 0.6.

4.2.2 Résultats des simulations

Puisque toutes les valeurs que prendront chacun des paramètres du modèle sont connues, les résultats que chacune des trois méthodes ont produits peuvent maintenant être examinés et comparés. Ainsi, les graphiques qui suivent montrent l'allure des courbes de l'erreur quadratique moyenne obtenues lors de l'estimation des percentiles du modèle développé à la section 4.1, et ce, dans les douze cas qui croisent les différentes valeurs possibles des paramètres.

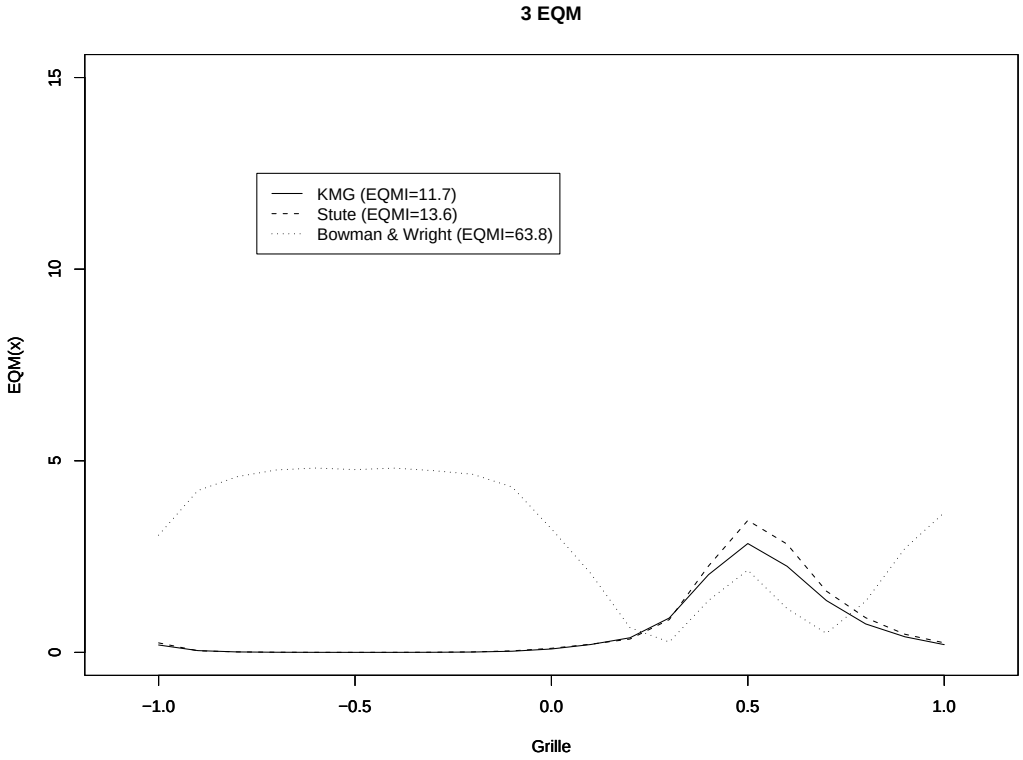
Ces graphiques sont donc illustrés ci-dessous et, pour chacun d'eux, une courte description du graphique comparant les trois différentes courbes d'EQM suit. Une discussion à propos de l'ensemble de ces graphiques sera par la suite présentée à la section 4.2.3. Les graphiques présentant aussi les courbes du biais et de la variance obtenues se retrouvent pour leur part à l'annexe B.

FIG. 4.2 – *Graphique présentant les EQM obtenues par chacune des trois méthodes pour n=200, $\alpha = 0.25$ et λ=0.1.*



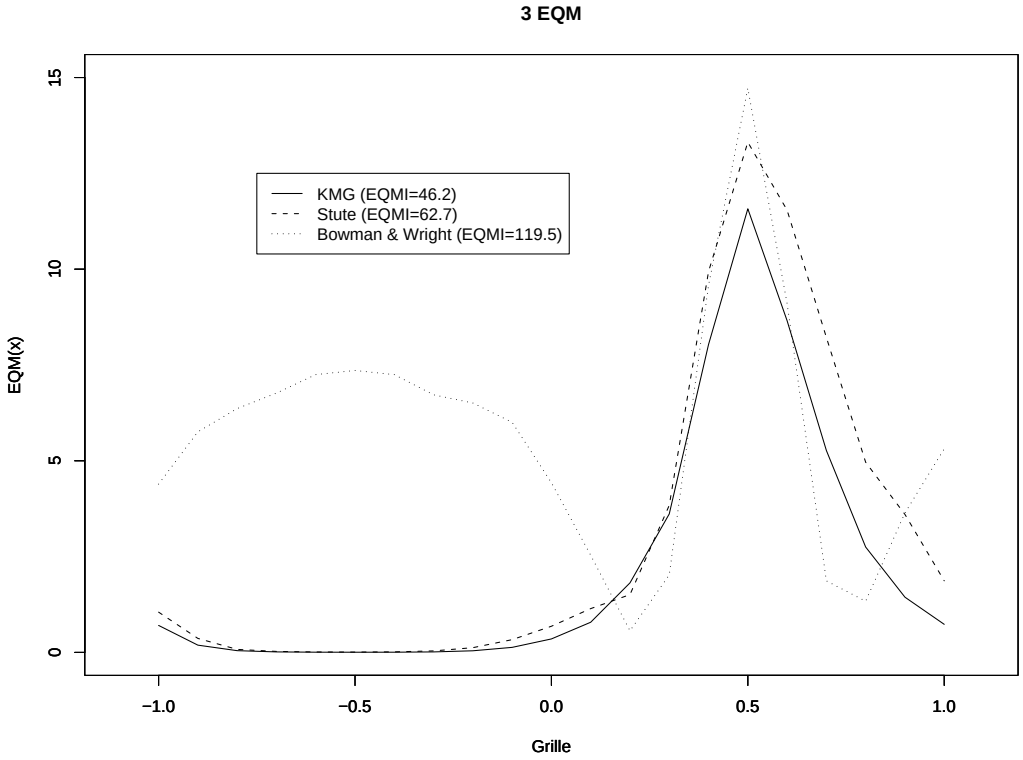
Sur le graphique ci-dessus, qui présente les courbes des EQM obtenues par les trois différentes méthodes, on voit que la méthode de Bowman et Wright produit toujours une EQM plus élevée que les méthodes de KMG et Stute aux valeurs les plus petites et les plus élevées de la grille. De plus, ce graphique montre que l'$EQMI$ la plus faible est obtenue par la méthode de KMG, alors que la plus grande est trouvée par Bowman et Wright .

FIG. 4.3 – *Graphique présentant les EQM obtenues par chacune des trois méthodes pour n=200, $\alpha = 0.5$ et λ=0.1.*



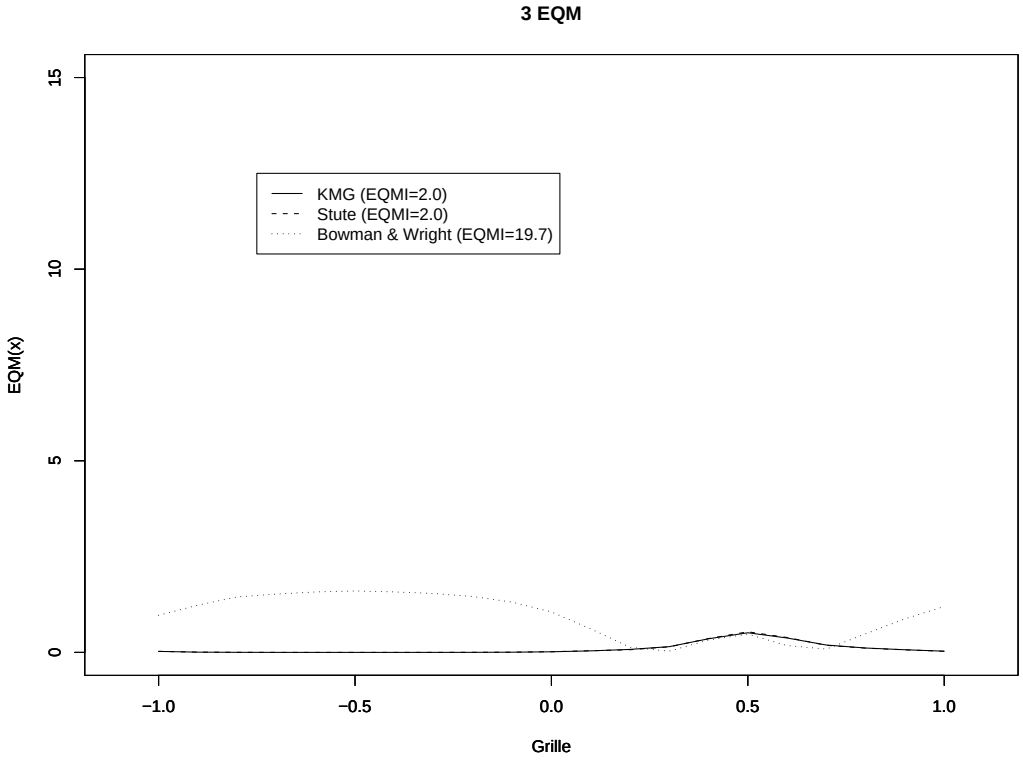
Le graphique ci-haut, qui compare les courbes des EQM acquises par les trois différentes méthodes lorsque $n = 200$, $\alpha = 0.5$ et $\lambda = 0.1$, est très similaire au précédent. En effet, on y voit que la méthode de Bowman et Wright produit des EQM plus élevées que celles trouvées par les autres méthodes aux points qui sont éloignés de la valeur 0.5, alors qu'elles sont plus faibles à proximité de ce point. Or, on remarque que l'écart entre les EQM obtenues par Bowman et Wright et celles des deux autres estimateurs est amplifié sur ce graphique.

FIG. 4.4 – *Graphique présentant les EQM obtenues par chacune des trois méthodes pour n=200, $\alpha = 0.75$ et λ=0.1.*



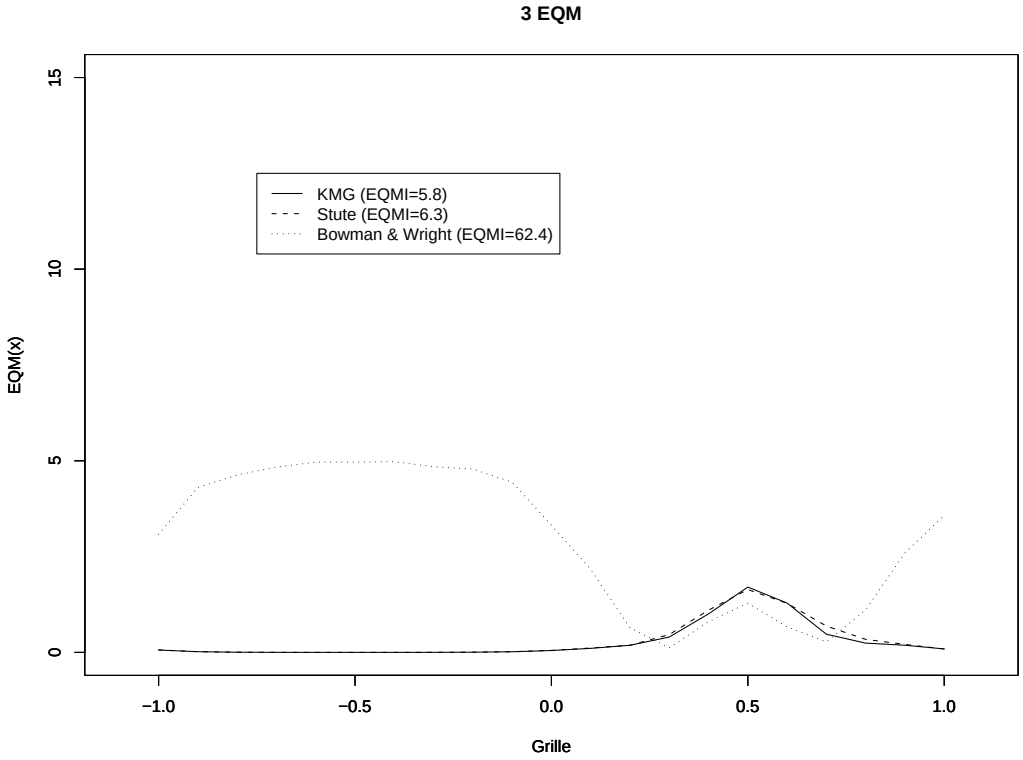
Dans le cas où $n = 200$, $\alpha = 0.75$ et $\lambda = 0.1$, les EQM les plus élevées sont celles qui sont obtenues par la méthode de Bowman et Wright, et ce, sur presque toutes les valeurs de la grille. De plus, en comparant les valeurs des EQM calculées près du point d'estimation $x = 0.5$ de ce graphique à celles évaluées aux deux graphiques précédents, on voit que celles illustrées sur le graphique ci-dessus sont supérieures.

FIG. 4.5 – *Graphique présentant les EQM obtenues par chacune des trois méthodes pour n=500, $\alpha = 0.25$ et λ=0.1.*



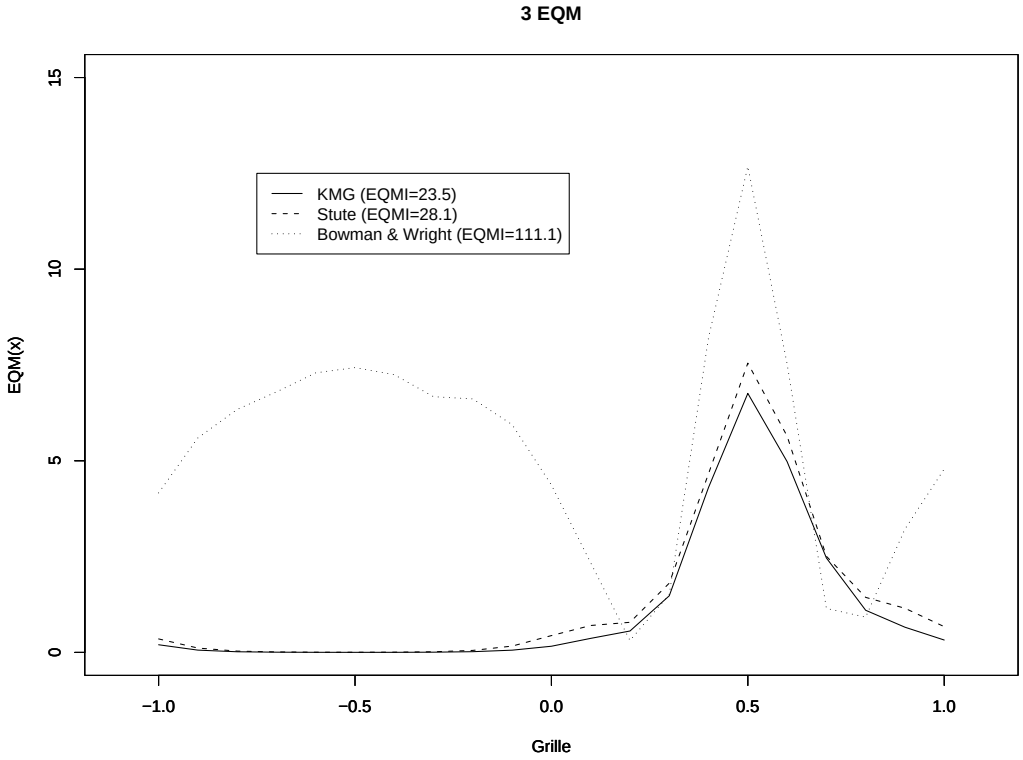
Comme tous les graphiques vus jusqu'à présent dans cette section, ce dernier permet de constater que Bowman et Wright produit des EQM plus fortes aux points supérieurs et inférieurs de la grille $\mathcal{G}$ que celles trouvées à partir des méthodes de KMG et de Stute. En revanche, contrairement aux graphiques antérieurs, les trois courbes des EQM illustrées ci-dessus sont pratiquement les mêmes aux points situés près de $x = 0.5$.

FIG. 4.6 – *Graphique présentant les EQM obtenues par chacune des trois méthodes pour n=500, $\alpha = 0.5$ et λ=0.1.*



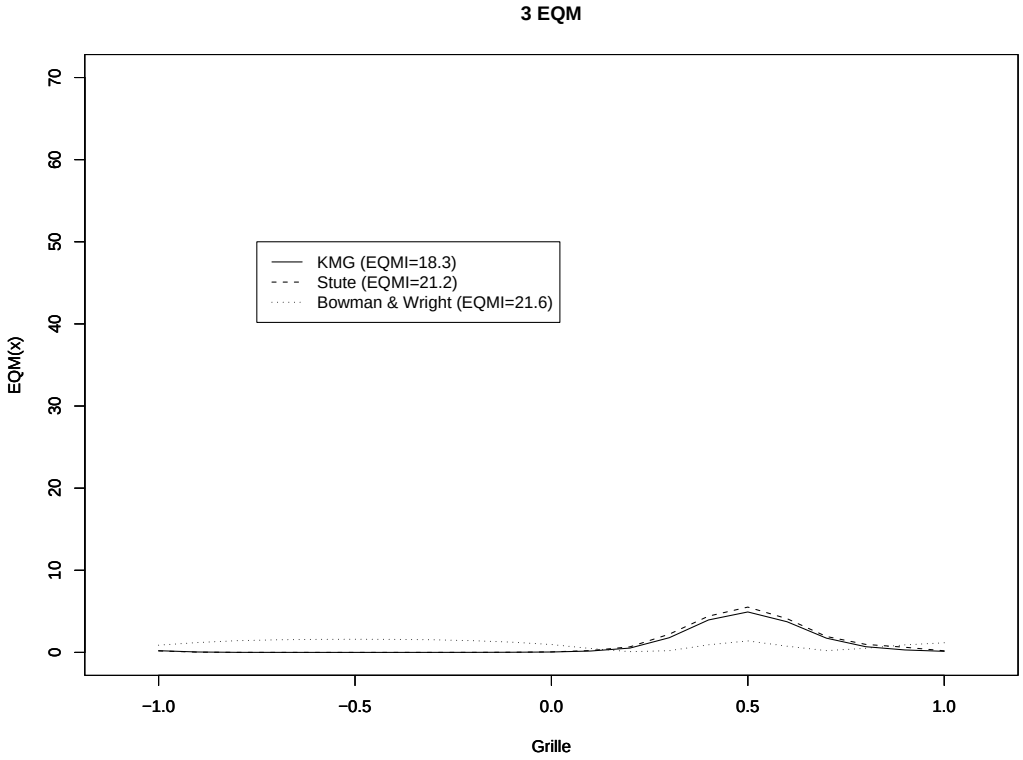
Le graphique 4.6 montre, comme tous les graphiques précédents, que les EQM trouvées par la méthode de Bowman et Wright sont plus importantes que celles obtenues par les méthodes de KMG et de Stute aux points éloignés du point d'estimation 0.5. Par contre, près de cette demie-unité, Bowman et Wright donne des EQM plus faibles. Les deux méthodes de Stute et KMG produisent environ les mêmes EQM sur toutes les valeurs de la grille. On remarque aussi que l'$EQMI$ trouvée par Bowman et Wright est plus grande que celle obtenue par la méthode de Stute, qui est pour sa part plus élevée que celle acquise par la méthode de KMG.

FIG. 4.7 – *Graphique présentant les EQM obtenues par chacune des trois méthodes pour n=500, $\alpha = 0.75$ et λ=0.1.*



La figure ci-dessus, qui illustre les résultats obtenus avec une moyenne des temps de censure de $1/\lambda = 10$, est très semblable à la figure 4.4, pour laquelle la taille échantillonnale n est le seul paramètre qui diffère. En effet, l'EQM est plus grande pour Bowman et Wright que pour les deux autres méthodes, et ce, sur presque tous les points d'estimation.

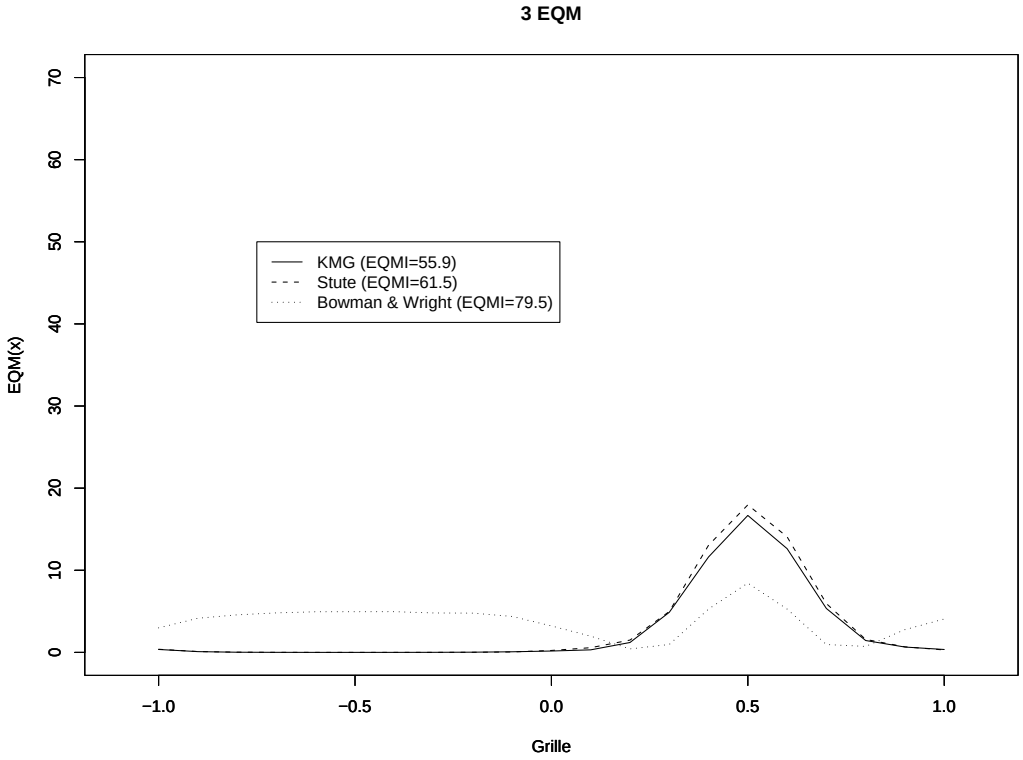
FIG. 4.8 – *Graphique présentant les EQM obtenues par chacune des trois méthodes pour n=200, $\alpha = 0.25$ et λ=0.6.*



À partir de maintenant, c'est-à-dire pour la figure ci-dessus jusqu'à la figure 4.13, les graphiques et les analyses seront effectués à l'aide d'un taux de panne des temps de censure $\lambda = 0.6$ au lieu de $\lambda = 0.1$.

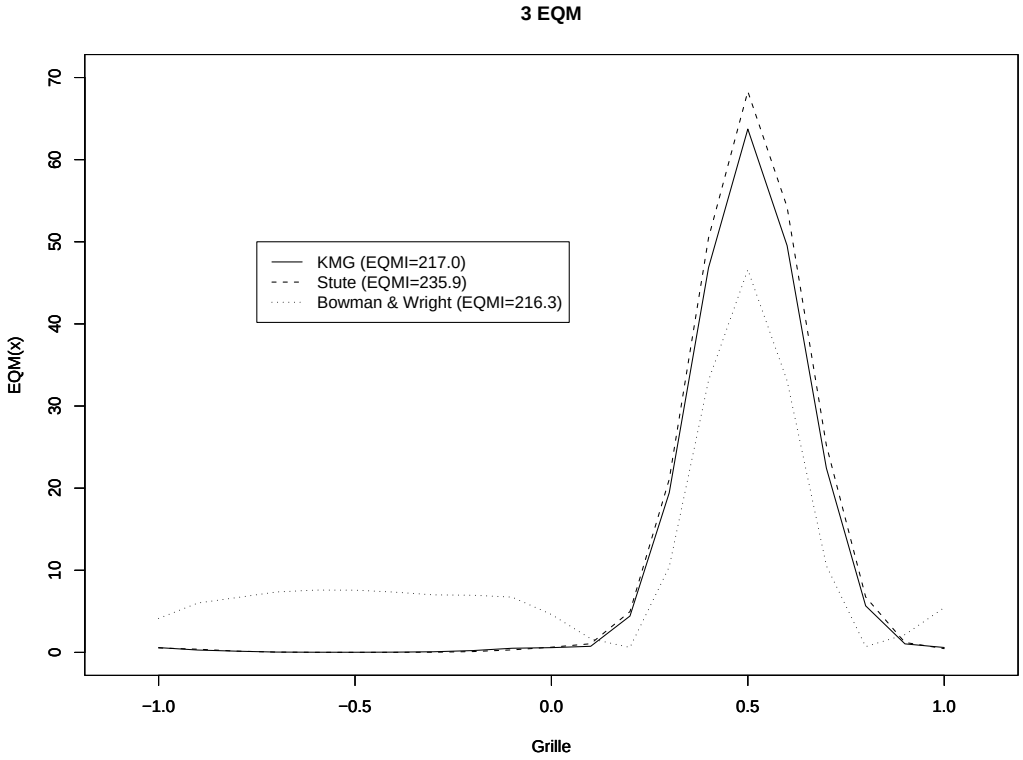
L'examen de la figure 4.8 permet de constater que le résultat observé sur tous les graphiques passés est également valide dans le cas présent. En fait, aux points inférieurs et supérieurs de la grille $\mathcal{G}$, les EQM acquises par les méthodes de KMG et Stute sont plus petites que celles trouvées par la méthode de Bowman et Wright. Or, l'écart entre celles-ci est moins éminent que dans les cas traités précédemment. De plus, l'EQM par la méthode de Bowman et Wright est moins élevée que pour les autres méthodes lorsqu'elle est évaluée aux points à proximité de 0.5.

FIG. 4.9 – *Graphique présentant les EQM obtenues par chacune des trois méthodes pour n=200, $\alpha = 0.5$ et λ=0.6.*



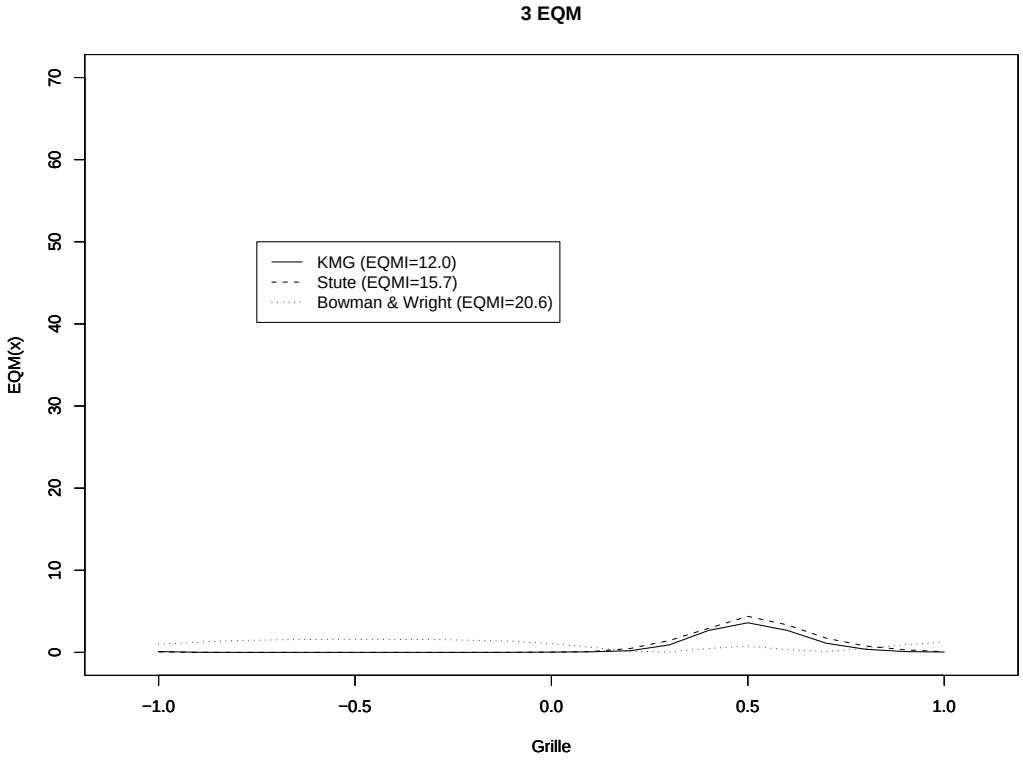
Cette figure est réellement similaire à celle analysée à la figure 4.8. En effet, les méthodes de KMG et de Stute produisent des EQM plus minimes que pour la méthode de Bowman et Wright aux extrémités de la grille $\mathcal{G}$, alors que la situation inverse survient aux points d'estimations qui se situent autour de la valeur 0.5. Par contre, sur tous les points de la grille, la figure ci-dessus admet des valeurs d'EQM plus élevées, ou du moins égales, à celles du graphique précédent. Conséquemment, l'$EQMI$ est plus élevée qu'au graphique 4.8 pour chacune des trois méthodes.

FIG. 4.10 – *Graphique présentant les EQM obtenues par chacune des trois méthodes pour n=200, $\alpha = 0.75$ et λ=0.6.*



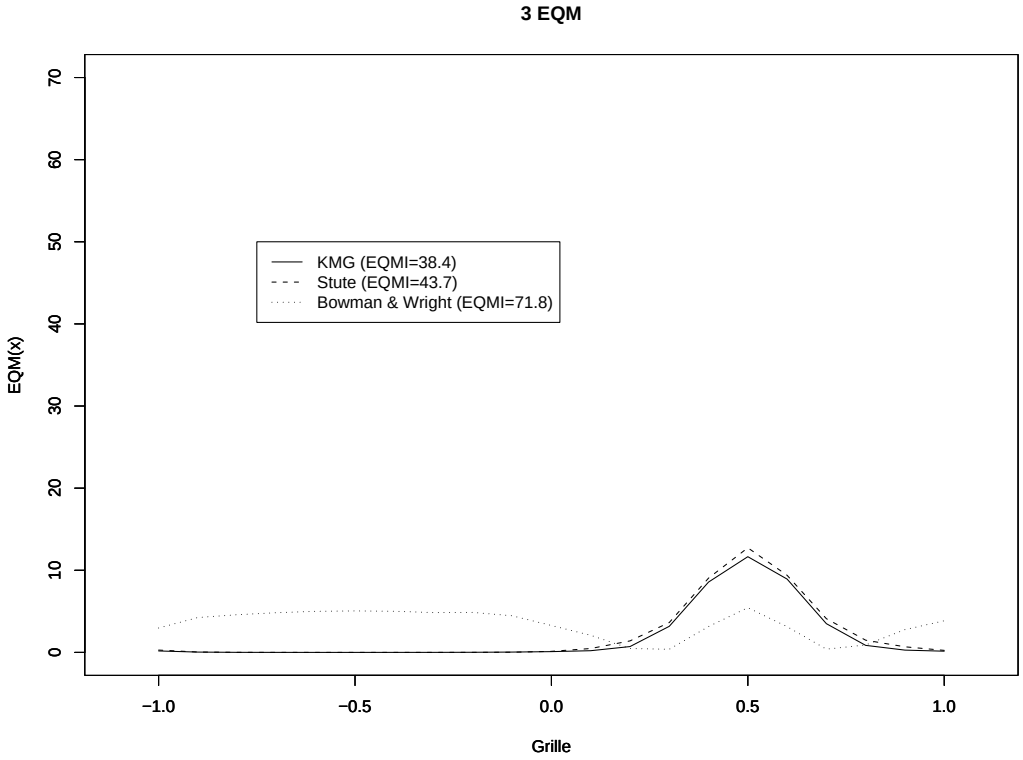
Le cas présent peut être traité comme une suite logique aux deux graphiques précédents 4.8 et 4.9. On voit effectivement qu'aux points inférieurs et supérieurs de la grille $\mathcal{G}$, les EQM acquises par les méthodes de KMG et Stute sont plus petites que celles dérivées de la méthode de Bowman et Wright. De plus, l'EQM par la méthode de Bowman et Wright est moins élevée que pour les autres méthodes lorsqu'elle est évaluée aux points à proximité de 0.5. Par ailleurs, les EQM observées sur ce graphique sont toujours plus grandes ou égales à celles des deux autres graphiques. Les trois $EQMI$ actuelles sont donc subséquemment plus élevées que celles étudiées aux deux graphiques antérieurs.

FIG. 4.11 – *Graphique présentant les EQM obtenues par chacune des trois méthodes pour n=500, $\alpha = 0.25$ et λ=0.6.*



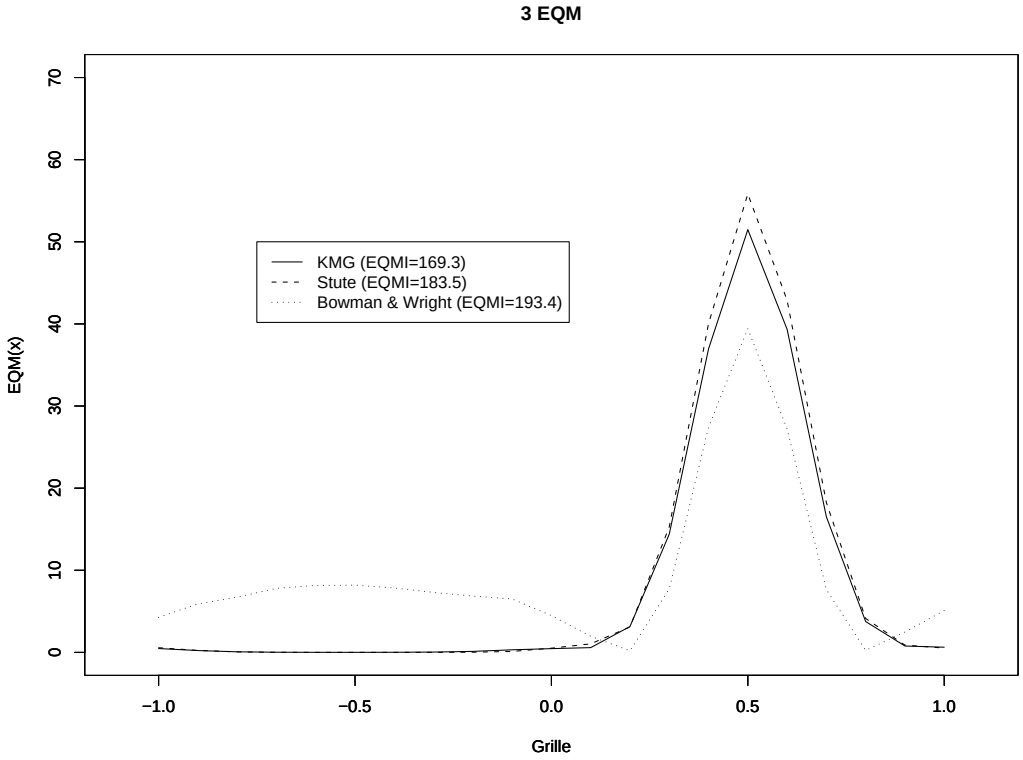
Cette figure présente une fois de plus les mêmes conclusions tirées des graphiques pour la moyenne des temps de censure $1/\lambda = 5/3$ antérieurs. En effet, la méthode de Bowman et Wright produit des EQM plus élevées que celles obtenues par les méthodes de KMG et de Stute aux extrémités de la grille, alors que la situation inverse survient lorsque l'EQM est mesurée aux points d'estimations près de la valeur 0.5. Par ailleurs, à la lumière de ce graphique, on voit encore une fois que l'$EQMI$ associée à la méthode de Bowman et Wright est plus proéminente que celle trouvée par les deux autres méthodes ; celle obtenue par la méthode du KMG étant inférieure.

FIG. 4.12 – *Graphique présentant les EQM obtenues par chacune des trois méthodes pour n=500, $\alpha = 0.5$ et λ=0.6.*



Les informations qui se dégagent de cette image ressemblent en plusieurs points aux trois premières illustrées pour une valeur de λ=0.6. Cela se vérifie par le fait que les EQM pour Bowman et Wright sont plus grandes que pour les autres méthodes aux petites et aux grandes valeurs de la grille, mais elles sont plus petites lorsqu'elles sont évaluées près de 0.5.

FIG. 4.13 – *Graphique présentant les EQM obtenues par chacune des trois méthodes pour n=500, $\alpha = 0.75$ et λ=0.6.*



Les résultats qui ont été obtenus antérieurement et qui stipulent que la méthode de Bowman et Wright produit des EQM plus élevées que celles obtenues par les méthodes de KMG et de Stute aux extrémités de la grille, alors que le contraire survient lorsque l'EQM est calculée autour de 0.5 s'appliquent également à ce graphique. Par ailleurs, comme les figures antérieures l'ont déjà démontré, l'$EQMI$ associée à la méthode de Bowman et Wright est plus grande que les $EQMI$ calculée par les deux autres méthodes. Sa valeur de 193.4 est en effet plus élevée que celles reliées aux méthodes de KMG (169.3) et des poids proposés par Stute (183.5).

4.2.3 Discussion

Une des premières constatations qui se dégagent de ces multiples graphiques est que, pour une méthode donnée et lorsque tous les autres paramètres sont fixés, plus la valeur de α augmente, plus l'EQM devient élevée. Cette situation apparaît pour chacune des trois méthodes d'estimation et elle se justifie par le fait que plus une valeur estimée est grande, plus le biais et la variance qui lui sont associés sont élevés si toutes les valeurs sont estimées de la même façon. Évidemment, on sait que le 75^e percentile est plus élevé que le 50^e, qui est à son tour plus grand que le 25^e percentile. La figure 4.1, qui présente les percentiles théoriques pour le modèle ayant servi de modèle pour les simulations, permet d'ailleurs aisément de visualiser cette réalité. Il est donc naturel de prétendre que le biais associé au 75^e percentile est plus élevé que celui associé au 50^e et au 25^e percentile. Par ailleurs, une autre raison pour laquelle le biais peut être accentué lorsque la valeur de α augmente est que l'on dispose de moins de données pour estimer le 75^e percentile, car moins de valeurs sont "vivantes et non censurées" à ce niveau de la variable Y.

Dans un deuxième temps, l'analyse de ces douze graphiques permet d'avoir une idée de l'effet de la taille échantillonnale n et du taux de panne des temps de censure λ sur les EQM, et ce, pour une méthode donnée et lorsque tous les autres paramètres sont fixes. Ces graphiques permettent en effet de voir, qu'en général, plus la taille échantillonnale n augmente, plus les EQM diminuent et, par ailleurs, que plus le taux de panne des temps de censure est large, plus les EQM augmentent.

De plus, on remarque qu'aux valeurs les plus élevées ainsi qu'aux plus faibles de la grille, l'EQM acquise par la méthode de Bowman et Wright (2000) est toujours plus proéminente que celles trouvées à l'aide des deux autres méthodes. En effet, bien que cette différence soit parfois minime, mais aussi parfois énorme, elle se présente toutefois dans chacune des douze situations analysées.

D'autre part, ces graphiques montrent que, dans onze cas sur douze, l'$EQMI$ trouvée par la méthode du KMG est inférieure ou égale à celle obtenue par la méthode des poids proposés par Stute, qui à son tour est inférieure ou égale à l'$EQMI$ que donne la méthode de Bowman et Wright (2000). Le cas pour lequel cette situation ne se produit pas est celui pour lequel une moyenne des temps de censure $1/\lambda = 5/3$, une taille échantillonnale $n = 200$ et un seuil $\alpha = 0.75$ sont fixés. En effet, dans cette situation, on trouve une $EQMI$ de 217.0 pour la méthode du KMG, une $EQMI$ de 235.9 pour la méthodes des poids proposés par Stute et une $EQMI$ de 216.3 pour la méthode de Bowman et Wright (2000).

Par ailleurs, les $EQMI$ varient également beaucoup en fonction de la méthode utilisée et des trois paramètres qui sont changés. Le tableau 4.3 évoque justement les différences qui existent entre les trois méthodes en montrant les $EQMI$ trouvées à partir de la combinaison des différents paramètres. À la vue de ce tableau, on s'aperçoit

TAB. 4.3 – *EQMI obtenues pour chacune des 3 méthodes et pour chacun des paramètres (erreur standard).*

		α					
		0.25		0.5		0.75	
Methode	λ	**n**=200	**n**=500	**n**=200	**n**=500	**n**=200	**n**=500
KMG	**0.1**	4.3 (0.06)	2.0 (0.04)	11.7 (0.09)	5.8 (0.06)	46.2 (0.19)	23.5 (0.13)
	0.6	18.3 (0.10)	12.0 (0.08)	55.9 (0.12)	38.4 (0.11)	217.0 (0.13)	169.3 (0.12)
Stute	**0.1**	4.4 (0.06)	2.0 (0.04)	13.6 (0.10)	6.3 (0.07)	62.7 (0.21)	28.1 (0.13)
	0.6	21.2 (0.12)	15.7 (0.11)	61.5 (0.13)	43.7 (0.12)	235.9 (0.14)	183.5 (0.13)
BW	**0.1**	19.6 (0.03)	19.7 (0.02)	63.8 (0.06)	62.4 (0.04)	119.5 (0.08)	111.1 (0.06)
	0.6	21.6 (0.04)	20.6 (0.02)	79.5 (0.05)	71.8 (0.04)	216.3 (0.07)	193.4 (0.06)

que les $EQMI$ minimales, qui se chiffrent à 2, sont acquises par les méthodes de KMG et de Stute, et ce, lorsque $\lambda = 0.1$, $n = 500$ et $\alpha = 0.25$. On remarque également que l'$EQMI$ maximale vaut 235.9 et elle est atteinte avec la méthode des poids proposés par Stute. Pour être plus précis, cette situation apparaît dans le cas qui a été mentionné au paragraphe précédent, c'est-à-dire lorsque $\lambda = 0.6$, $n = 200$ et $\alpha = 0.75$. Par ailleurs, ce tableau permet aussi de renforcer les remarques formulées au début de cette section. En effet, on voit sur ce dernier que, pour une méthode donnée et lorsque α et λ sont fixés, l'$EQMI$ diminue lorsque la taille échantillonnale n augmente. La situation contraire se produit lorsque le seuil α ou le taux de panne des temps de censure λ sont augmentés. Par exemple, pour une méthode donnée et si les autres paramètres sont fixés, l'$EQMI$ augmente lorsque α s'élargit.

L'examen de tous ces graphiques permet également de constater que les EQM sont élevées au point $x = 0.5$. À la lumière des graphiques fournis à l'annexe B, on s'aperçoit que cette hausse de l'EQM près du pic est causée par un important biais négatif à ce point, c'est-à-dire qu'elle est due à une sous-estimation du pic. Puisque le biais est élevé

au carré dans le calcul de l'EQM, il est logique que l'EQM soit élevée à ce point.

Finalement, tel qu'il a été précisé à maintes reprises au cours de la présentation des différents graphiques, tous les cas pour lesquels $\lambda = 0.6$ ont un point en commun. En effet, dans chacune de ces six situations, la méthode de Bowman et Wright produit des EQM moindres que celles obtenues par les méthodes de KMG et de Stute aux points d'estimations près de la valeur 0.5, alors que la situation inverse survient lorsque l'EQM est mesurée aux extrémités de la grille. Dans le cas où $\lambda = 0.1$, on assiste à la même chose dans quelques cas, mais il arrive également que l'EQM trouvé par la méthode de Bowman et Wright soit plus élevé que celui obtenu par les deux autres méthodes aux points d'estimations près de la valeur 0.5.

Donc, en général, on peut dire que les deux méthodes de KMG et des poids proposés par Stute sont semblables sur tous les points de la grille, mais que la méthode de KMG semble tout de même être légèrement meilleure, puisque son EQM est généralement un tout petit peu moins élevé. On peut même aller jusqu'à dire que la méthode de KMG est meilleure que les deux autres méthodes dans à peu près tous les cas illustrés précédemment. En effet, la méthode de Bowman et Wright est particulièrement moins bonne aux points d'estimation situés près des bornes. Or, dans le cas où le taux de panne des temps de censure est de $\lambda = 0.6$ et que les estimés sont évalués près du point $x = 0.5$, il serait de mise de choisir la méthode de Bowman et Wright, puisqu'elle y produit des EQM moins élevés que les autres.

Chapitre 5

Conclusion

Dans ce mémoire, nous avons étudié diverses méthodes de régression non paramétrique. Au premier chapitre, la régression non paramétrique dans sa forme la plus simple a été abordée. Le lien existant entre la régression et la minimisation d'une espérance conditionnelle a tout d'abord été démontré. Par la suite, la régression non paramétrique par la méthode du noyau a été présentée. Dans un premier temps, les étapes permettant d'arriver à un estimateur par la méthode du noyau ont été décrites. On a vu qu'il faut d'abord choisir un point d'estimation, une fonction de noyau et une fenêtre de lissage pour faire les calculs. Par la suite, le poids associé à chacune des données peut être calculé afin d'obtenir l'estimateur du noyau de Nadaraya-Watson. La courbe de régression de l'estimateur en fonction de x est finalement obtenue en joignant chacun des estimateurs. Les propriétés de cette méthode, telles le biais, la variance et l'EQM, ainsi que les problèmes lui étant reliés, comme par exemple le problème d'aplatissement des pics et des vallées, ont ensuite été explorés aux sections 1.2.2 et 1.2.3. Après avoir vu en détails cette méthode, une deuxième méthode, la méthode par polynômes locaux, a été expliquée. Une fois que l'estimateur obtenu à l'aide de cette méthode a été abordé, les propriétés lui étant rattachées ont été discutées. Afin de clore le premier chapitre et de mieux comprendre toute la théorie qui y a été donnée, un exemple pratique exécuté à partir de données provenant de Johnson (1995) a permis de faire une régression non paramétrique par la méthode localement linéaire du pourcentage de graisse des individus en fonction de leur poids.

Un peu plus loin dans ce mémoire, la régression non paramétrique des percentiles a fait l'objet du chapitre 2. Afin de permettre une meilleure compréhension de cette méthode, la définition formelle d'un percentile conditionnel a été donnée à la section 2.1. Ce point étant éclairci, la méthode du noyau et celles des polynômes locaux pour la régression non paramétrique des percentiles ont été présentées, respectivement aux sec-

tions 2.2 et 2.3. Chacune de ces sections était évidemment accompagnée des propriétés pour chacun des deux estimateurs. On s'est conséquemment aperçu que les méthodes étudiées dans ce chapitre sont très similaires à celles du chapitre 1, la différence résidant dans l'utilisation de la fonction $\rho_\alpha(z)$, au lieu de z à la puissance 2, dans le problème de minimisation (2.2). Tout comme au premier chapitre, le chapitre 2 a été clos par un exemple. Cet exemple a en fait été appliqué aux mêmes données qu'au chapitre 1 et il a permis de faire, dans ce cas-ci, une régression non paramétrique des percentiles par la méthode localement linéaire du pourcentage de graisse des individus étudiés en fonction de leur poids.

Pour sa part, le chapitre 3 présentait la régression non paramétrique des percentiles avec variable réponse censurée à droite. Tout d'abord, deux méthodes permettant d'effectuer ce type de régression étaient expliquées à la section 3.1, plus précisément, la méthode du Kaplan-Meier généralisé (KMG) et la méthode des poids proposés par Stute. Ces deux méthodes sont toutes deux basées sur la méthode localement linéaire et les poids nécessaires à l'obtention de l'estimateur ne dépendent dorénavant plus uniquement de la i^e observation, mais plutôt de l'échantillon en entier, étant donné que les poids des Y_i censurés doivent être redistribués aux poids des données non censurées. Pour être plus précis, les poids utilisés pour la méthode du KMG correspondent aux sauts que fait la fonction de survie du KMG aux points Y_i sachant $\{X_i = x\}$, alors que pour ce qui est de la méthode des poids proposés par Stute, les poids dépendent des sauts que fait la fonction de survie du Kaplan-Meier ordinaire aux points Y_i. Seules certaines propriétés concernant la méthode des poids proposés par Stute, tirées de l'article de Gannoun *et al.* (2005), ont été mentionnées dans cette section. Une troisième méthode a finalement été abordée à la section 3.2, soit la méthode de Bowman et Wright (2000). Cette méthode est en fait totalement différente des deux méthodes précédentes en ce qui a trait au fait qu'aucun calcul de sauts n'est nécessaire. Le principal avantage de cette méthode est qu'il existe une solution explicite au problème de minimisation (3.8). Or, cette dernière méthode est plus complexe, puisque deux fenêtres de lissage doivent être déterminées, au lieu d'une seule. Enfin, une application de ces méthodes à un jeu de données (Klein et Moeschberger, 2003) a également été présentée dans ce chapitre. Les données ayant été utilisées à cette fin diffèrent évidemment de celles employées aux chapitres antérieurs, puisque la présence de variables de censure est obligatoire. Seules les méthodes du KMG et des poids proposés par Stute ont servi à la régression non paramétrique des percentiles avec variable réponse censurée à droite des temps de décès d'individus à partir du moment où ils ont subi une greffe de rein en fonction de leur âge.

Finalement, le chapitre 4 s'intéressait aux simulations produites à l'aide des trois méthodes mentionnées au chapitre 3. En premier lieu, la description du modèle ayant

servi à ces simulations ainsi que les paramètres de ce dernier ont été abordés à la section 4.1. Enfin, le résultat des simulations a été présenté à la section 4.2. Les fenêtres de lissage produisant l'$EQMI$ la plus minime et ayant été obtenues par 500 simulations ont d'abord été données pour tous les différents cas possibles des seuils (α), des tailles échantillonnales (n) et des moyennes des temps de censure ($1/\lambda$). L'obtention de ces fenêtres de lissage a subséquemment permis de tracer des courbes de biais, de variance et d'EQM pour chacune des trois méthodes, et ce, pour tous les cas abordés ci-dessus. Les résultats globaux découlant de ces graphiques ont finalement été discutés à la section 4.2.3. Une des premières leçons générales qui en ressort est que les méthodes de KMG et des poids proposés par Stute donnent des résultats très similaires. Par ailleurs, la méthode de KMG semble être meilleure que les deux autres méthodes pour la très grande majorité des douze cas présentés à la section 4.2.2. Les cas pour lesquels cette méthode ne semble pas être optimale surviennent pour leur part lorsque la moyenne des temps de censure est de $1/\lambda = 5/3$ et que les estimés sont évalués aux extrémités des données.

Dans des travaux futurs, il serait intéressant d'étudier les méthodes de choix des fenêtres pour les méthodes du chapitre 3, puisque la méthode qui a été employée dans ce mémoire ne permet pas toujours de déduire avec certitude la meilleure fenêtre. En effet, la courbe formée par la liaison des différentes valeurs obtenues par l'équation (4.1), ordonnées par ordre croissant de h, n'a pas la forme parfaite d'une parabole dans tous les cas, mais elle présente parfois de petites oscillations près de son minimum. Il pourrait donc être utile d'évaluer si une autre méthode fournirait de meilleurs résultats. D'un autre côté, puisque cela n'a pas été fait dans ce mémoire, il pourrait aussi être intéressant d'étudier les propriétés théoriques des estimateurs obtenus par la méthode des poids proposés par Stute en les comparant aux résultats obtenus par simulations. Cette même étude pourrait également être effectuée pour les estimateurs trouvés par la méthode de KMG et de Bowman et Wright (2000). Il deviendrait alors possible de comparer les propriétés théoriques de ces trois méthodes d'estimation. Par ailleurs, il est évident que l'étude présentée dans ce mémoire serait plus complète si les simulations effectuées au chapitre 4 étaient produites pour plusieurs autres valeurs des paramètres du modèle. En effet, diverses autres valeurs pour la variance de la variable exogène X_i, c'est-à-dire σ^2, pourraient être utilisées pour faire d'autres simulations. Changer les valeurs du paramètre des moyennes des temps de censure $1/\lambda$ permettrait également peut-être de déceler un comportement des méthodes qui n'aurait malheureusement pas pu être trouvé par les simulations ayant déjà été produites dans ce mémoire.

Bibliographie

Beran, R. (1981). Nonparametric regression with randomly censored survival data. Rapport technique, University of California, Berkeley.

Bowman, A. W. et Wright, E. M. (2000). Graphical exploration of covariate effects on survival data through nonparametric quantile curves. *Biometrics*, 56:563–570.

Casella, G. et Berger, R. L. (2001). *Statistical Inference*. Duxbury Thomson Learning, Pacific Grove.

Duong, T. (2001). An introduction to kernel density estimation. Séminaire : Weatherburn Lecture series, Department of Mathematics and Statistics, University of Western Australia, Australia. Disponibles en ligne : http://web.maths.unsw.edu.au/~tduong/seminars/intro2kde/ (Page consultée le 1 février 2007).

Fan, J., Hu, T.-C. et Truong, Y. K. (1994). Robust non-parametric function estimation. *Scandinavian Journal of Statistics*, 21:433–446.

Fox, J. (2002). Nonparametric regression. Appendix to An R and S-PLUS Companion to Applied Regression. Disponibles en ligne : http ://socserv.mcmaster.ca/jfox/Books/Companion/appendix-nonparametric-regression.pdf (Page consultée le 1 février 2007).

Fox, J. (2004). Nonparametric regression. Ébauche d'un article pour l'Encyclopedia of Behavioral Statistics. Disponibles en ligne : http://socserv.mcmaster.ca/jfox/Nonparametric-regression.pdf (Page consultée le 1 février 2007).

Gannoun, A., Saracco, J., Yuan, A. et Bonney, G. E. (2005). Non-parametric quantile regression with censored data. *Scandinavian Journal of Statistics*, 32:527–550.

Hall, P., Wolff, R. et Yao, Q. (1999). Methods for estimating a conditional distribution function. *Journal of the american statistical association*, 94:154–163.

Härdle, W. (1990). *Smoothing Techniques With Implementation in S*. Springer Series in Statistics. Springer-Verlag, New York.

Johnson, R. (1995). Lists estimates of the percentage of body fat determined by underwater weighing and various body circumference measurements for 252 men. Disponibles en ligne : http://lib.stat.cmu.edu/datasets/bodyfat (Page consultée le 1 février 2007).

Jones, M. C. et Hall, P. (1990). Mean squared error properties of kernel estimates of regression quantiles. *Statistics & Probability Letters*, 10(4):283–289.

Klein, J. P. et Moeschberger, M. L. (2003). *Survival Analysis : techniques for censored and truncated data*. Springer-Verlag, New York.

Koenker, R. (2005). *Quantile Regression*. Cambridge University Press, Cambridge.

Leconte, E., Casanova, S. P. et Agnan, C. T. (2002). Smooth conditional distribution function and quantiles under random censorship. *Lifetime data analysis*, 8:229–246.

Nelder, J. A. et Mead, R. (1965). A simplex algorithm for function minimization. *Computer Journal*, 7:308–313.

Schimek, M. G. (2000). *Smoothing and regression : approaches, computation, and application*. John Wiley & Sons, Inc., New York.

Simonoff, J. S. (1996). *Smoothing Methods in Statistics*. Springer Series in Statistics. Springer-Verlag, New York.

Stute, W. (1993). Consistent estimation under random censorship when covariables are present. *Journal of Multivariate Analysis*, 45:89–103.

Annexe A

Définitions et démonstrations

A.1 Définition des notations $O(h_n)$, $o(h_n)$, $O_p(H_n)$ et $o_p(H_n)$

La définition des notations standards pour la convergence des séries se lit de la façon qui suit :

(1) Pour les variables non aléatoires :

Soit x_n et h_n deux séries de nombres réels. Alors, lorsque $n \to \infty$,

(a) $x_n = O(h_n) \Leftrightarrow \limsup_{n\to\infty} |x_n/h_n| < \infty$,

(b) $x_n = o(h_n) \Leftrightarrow \lim_{n\to\infty} |x_n/h_n| = 0$.

(2) Pour les variables aléatoires :

Soit X_n et H_n deux séries de nombres réels. Alors, lorsque $n \to \infty$,

(a) $X_n = O_p(H_n) \Leftrightarrow \forall\ \epsilon > 0$, $\exists\ \delta$ et N tels que $P(|X_n/H_n| > \delta) < \epsilon$, $\forall\ n > N$,

(b) $X_n = o_p(H_n) \Leftrightarrow \forall\ \epsilon > 0$, $\lim_{n\to\infty} P(|X_n/H_n| > \epsilon) = 0$.

A.2 Démonstration des équations (3.9) et (3.10)

Soit $A = \sum_{i=1}^{n} \delta_i K_g(q_i^{(x)} - q)\Big\{Y_i - \big[\beta_0 + \beta_1(q_i^{(x)} - q)\big]\Big\}^2$. Pour obtenir la formule pour $\hat{\beta}_0$, il faut d'abord dériver A par rapport à β_0 :

$$\begin{aligned}
\frac{d}{d\beta_0}A &= -2\sum_{i=1}^{n} \delta_i K_g(q_i^{(x)} - q)\Big\{Y_i - \big[\beta_0 + \beta_1(q_i^{(x)} - q)\big]\Big\} \\
&= -2\sum_{i=1}^{n} \delta_i K_g(q_i^{(x)} - q)Y_i + 2\sum_{i=1}^{n} \delta_i K_g(q_i^{(x)} - q)\big[\beta_0 - \beta_1(q_i^{(x)} - q)\big] \\
&= 0.
\end{aligned}$$

Ainsi, en changeant de côté de l'équation le terme ayant β_0 en facteur, on se retrouve avec l'équation qui suit :

$$\beta_0 \sum_{i=1}^{n} \delta_i K_g(q_i^{(x)} - q) = \sum_{i=1}^{n} \delta_i K_g(q_i^{(x)} - q)Y_i - \beta_1 \sum_{i=1}^{n} \delta_i K_g(q_i^{(x)} - q)(q_i^{(x)} - q).$$

Il devient alors évident que

$$\hat{\beta}_0 = \frac{\sum_{i=1}^{n} \delta_i K_g(q_i^{(x)} - q)Y_i - \beta_1 \sum_{i=1}^{n} \delta_i K_g(q_i^{(x)} - q)(q_i^{(x)} - q)}{\sum_{i=1}^{n} \delta_i K_g(q_i^{(x)} - q)}.$$

En décomposant en deux termes le côté droit de l'équation, on trouve

$$\hat{\beta}_0 = \frac{\sum_{i=1}^{n} \delta_i K_g(q_i^{(x)} - q)Y_i}{\sum_{i=1}^{n} \delta_i K_g(q_i^{(x)} - q)} - \frac{\beta_1 \sum_{i=1}^{n} \delta_i K_g(q_i^{(x)} - q)(q_i^{(x)} - q)}{\sum_{i=1}^{n} \delta_i K_g(q_i^{(x)} - q)}.$$

Si on pose $B_i(q) = \delta_i K_g(q_i^{(x)} - q)$, on obtient finalement le résultat

$$\hat{\beta}_0 = \frac{\sum_{i=1}^{n} B_i(q)Y_i}{\sum_{i=1}^{n} B_i(q)} - \hat{\beta}_1 \frac{\sum_{i=1}^{n} B_i(q)(q_i^{(x)} - q)}{\sum_{i=1}^{n} B_i(q)}. \tag{A.1}$$

Pour ce qui est de la formule pour $\hat{\beta}_1$, une plus grande quantité de calculs est nécessaire. Pour cette raison, le terme $B_i(q) = \delta_i K_g(q_i^{(x)} - q)$ sera utilisé tout au long de cette démonstration afin d'alléger le contenu. Dans un premier temps, il faut décomposer le

terme A, défini plus haut, comme suit :

$$\begin{aligned}
A &= \sum_{i=1}^{n} \delta_i K_g(q_i^{(x)} - q)\Big\{Y_i - \big[\beta_0 + \beta_1(q_i^{(x)} - q)\big]\Big\}^2 \\
&= \sum_{i=1}^{n} B_i(q)\Big\{Y_i - \big[\beta_0 + \beta_1(q_i^{(x)} - q)\big]\Big\}^2 \\
&= \sum_{i=1}^{n} B_i(q)\big[Y_i^2 - 2Y_i\beta_0 - 2Y_i\beta_1(q_i^{(x)} - q) + \beta_0^2 + 2\beta_0\beta_1(q_i^{(x)} - q) + \beta_1^2(q_i^{(x)} - q)^2\big] \\
&= \sum_{i=1}^{n} B_i(q)Y_i^2 - 2\beta_0\sum_{i=1}^{n} B_i(q)Y_i - 2\beta_1\sum_{i=1}^{n} B_i(q)Y_i(q_i^{(x)} - q) + \beta_0^2\sum_{i=1}^{n} B_i(q) \\
&\quad +2\beta_0\beta_1\sum_{i=1}^{n} B_i(q)(q_i^{(x)} - q) + \beta_1^2\sum_{i=1}^{n} B_i(q)(q_i^{(x)} - q)^2.
\end{aligned}$$

Pour obtenir l'estimateur $\hat{\beta}_1$, il faut dériver ce A par rapport à β_1, de la manière qui suit :

$$\begin{aligned}
\frac{d}{d\beta_1}A &= -2\sum_{i=1}^{n} B_i(q)Y_i(q_i^{(x)} - q) + 2\beta_0\sum_{i=1}^{n} B_i(q)(q_i^{(x)} - q) + 2\beta_1\sum_{i=1}^{n} B_i(q)(q_i^{(x)} - q)^2 \\
&= 0.
\end{aligned}$$

En substituant (A.1) dans l'équation ci-dessus et en réorganisant celle-ci, la nouvelle équation ci-dessous est obtenue :

$$\beta_1\sum_{i=1}^{n} B_i(q)(q_i^{(x)} - q)^2 + \left[\frac{\sum_{i=1}^{n} B_i(q)Y_i}{\sum_{i=1}^{n} B_i(q)} - \hat{\beta}_1\frac{\sum_{i=1}^{n} B_i(q)(q_i^{(x)} - q)}{\sum_{i=1}^{n} B_i(q)}\right]\sum_{i=1}^{n} B_i(q)(q_i^{(x)} - q) = \sum_{i=1}^{n} B_i(q)Y_i(q_i^{(x)} - q)$$

En réordonnant les termes, on trouve

$$\beta_1\left[\sum_{i=1}^{n} B_i(q)(q_i^{(x)} - q)^2 - \frac{\sum_{i=1}^{n} B_i(q)(q_i^{(x)} - q)}{\sum_{i=1}^{n} B_i(q)}\sum_{i=1}^{n} B_i(q)(q_i^{(x)} - q)\right] = \sum_{i=1}^{n} B_i(q)Y_i(q_i^{(x)} - q) - \frac{\sum_{i=1}^{n} B_i(q)Y_i}{\sum_{i=1}^{n} B_i(q)}\sum_{i=1}^{n} B_i(q)(q_i^{(x)} - q)$$

Enfin, en isolant $\hat{\beta}_1$, le résultat final est trouvé

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n} B_i(q)Y_i(q_i^{(x)} - q) - \frac{\sum_{i=1}^{n} B_i(q)Y_i}{\sum_{i=1}^{n} B_i(q)}\sum_{i=1}^{n} B_i(q)(q_i^{(x)} - q)}{\sum_{i=1}^{n} B_i(q)[(q_i^{(x)} - q)]^2 - \frac{[\sum_{i=1}^{n} B_i(q)(q_i^{(x)} - q)]^2}{\sum_{i=1}^{n} B_i(q)}}. \tag{A.2}$$

Annexe B

Résultats des simulations du chapitre 4

Cette section illustre les résultats trouvés à partir des simulations effectuées au chapitre 4. Les graphiques qui suivent montrent donc l'allure des courbes du biais, de la variance et de l'erreur quadratique moyenne obtenus lors de l'estimation des percentiles du modèle développé à la section 4.1, et ce, dans les douze cas qui croisent les différentes valeurs possibles des paramètres. Par ce fait même, il y a en réalité un total de douze blocs constitués de trois graphiques. En effet, chacun de ces blocs est composé de trois graphiques présentant chacun les courbes des trois mesures permettant d'évaluer la qualité d'un estimateur obtenues par une des trois méthodes. En fait, le graphique situé dans la partie supérieure de chacun des blocs montre les 3 courbes du biais, de la variance et de l'EQM trouvés par la méthode du KMG, le graphique qui se trouve dans la partie centrale de ces blocs présente ces trois mêmes courbes, mais dans le cas où la méthode de Bowman et Wright (2000) est utilisée, alors que celui qui se trouve complètement au bas de ces blocs illustre pour sa part ces trois courbes, mais dans le cas où la méthode des poids proposés par Stute est choisie.

FIG. B.1 – *Graphiques présentant les résultats trouvés pour chacune des trois méthodes pour n=200, $\alpha = 0.25$ et λ=0.1.*

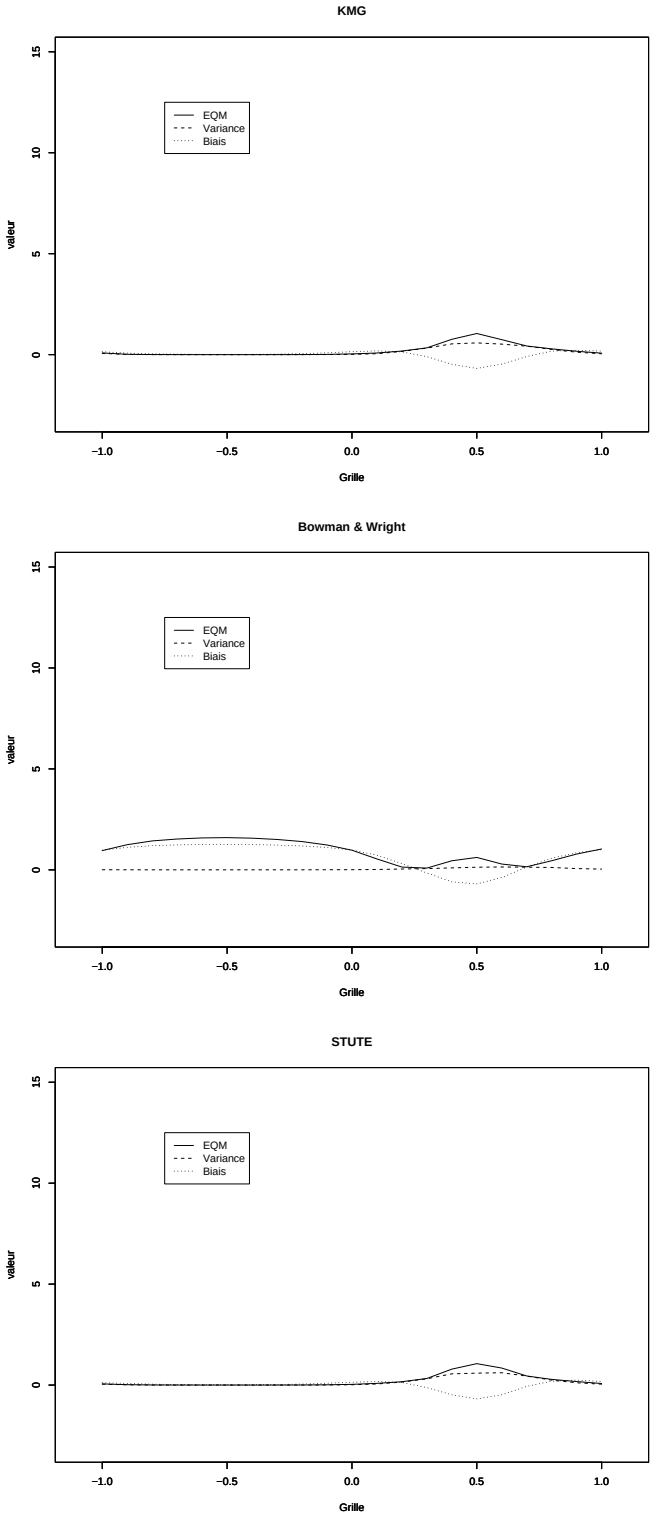


FIG. B.2 – *Graphiques présentant les résultats trouvés pour chacune des trois méthodes pour n=200, $\alpha = 0.5$ et λ=0.1.*

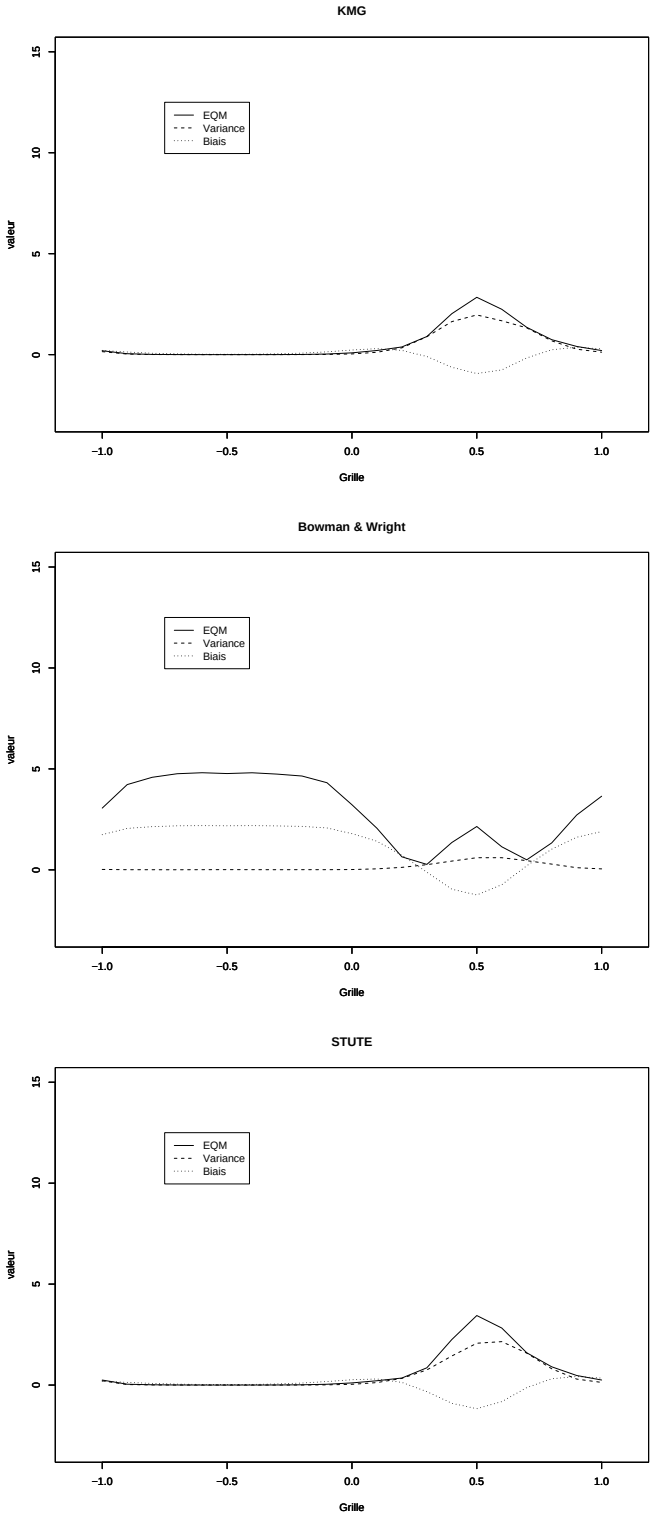


FIG. B.3 – *Graphiques présentant les résultats trouvés pour chacune des trois méthodes pour n=200, $\alpha = 0.75$ et λ=0.1.*

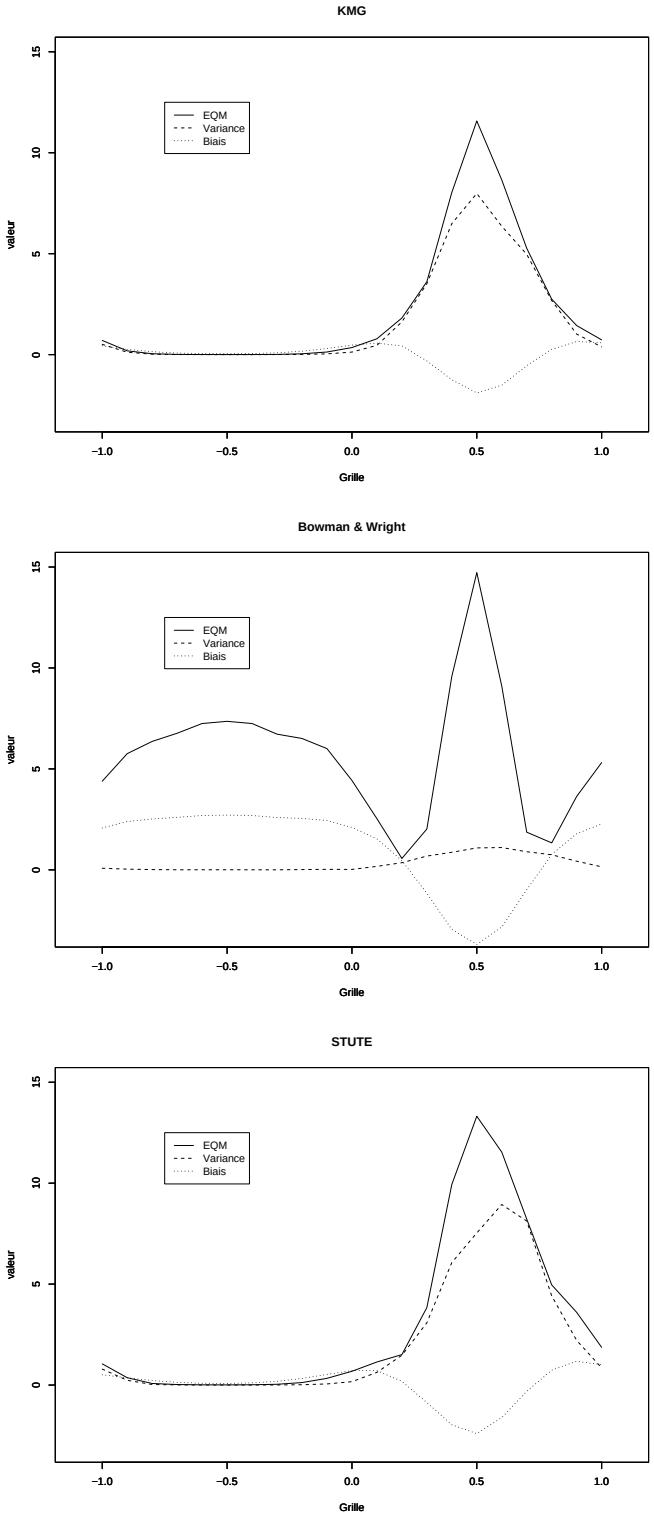


FIG. B.4 – *Graphiques présentant les résultats trouvés pour chacune des trois méthodes pour n=500, $\alpha = 0.25$ et λ=0.1.*

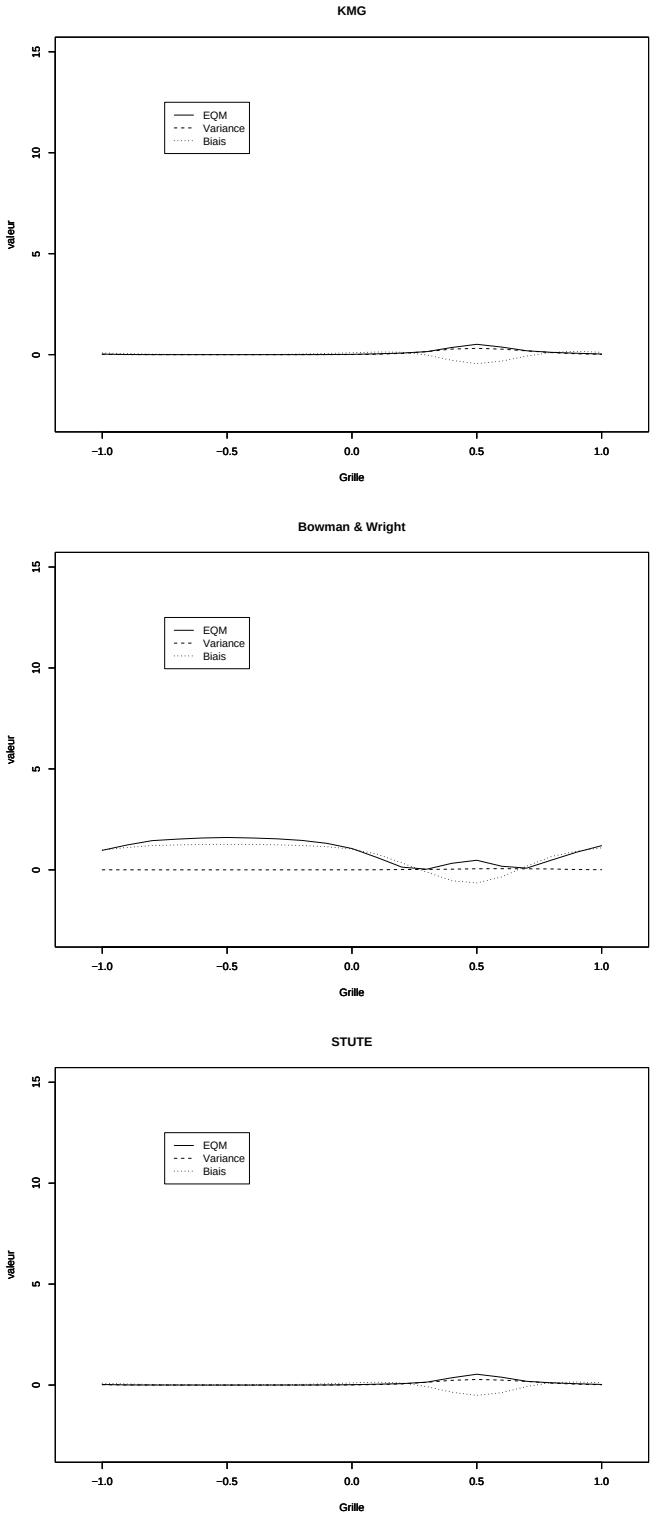


FIG. B.5 – *Graphiques présentant les résultats trouvés pour chacune des trois méthodes pour n=500, $\alpha = 0.5$ et λ=0.1.*

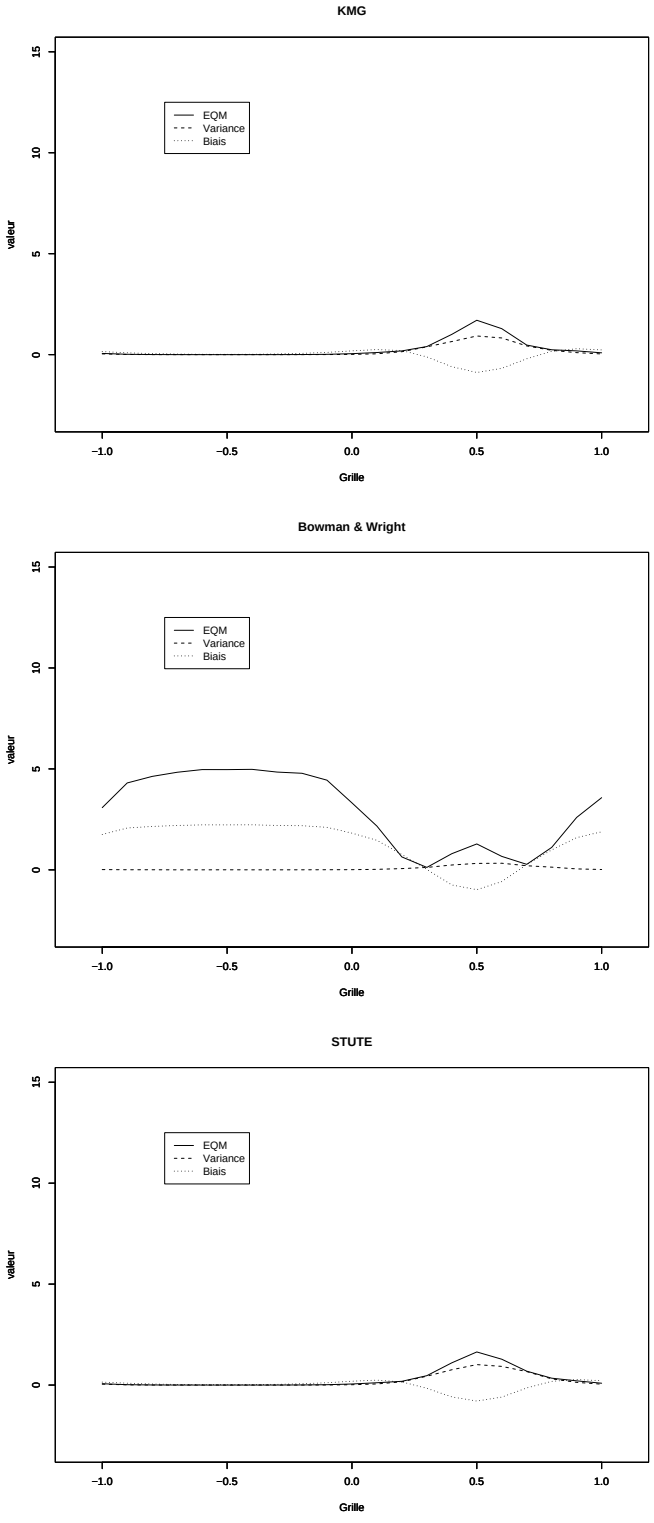


FIG. B.6 – *Graphiques présentant les résultats trouvés pour chacune des trois méthodes pour n=500, $\alpha = 0.75$ et λ=0.1.*

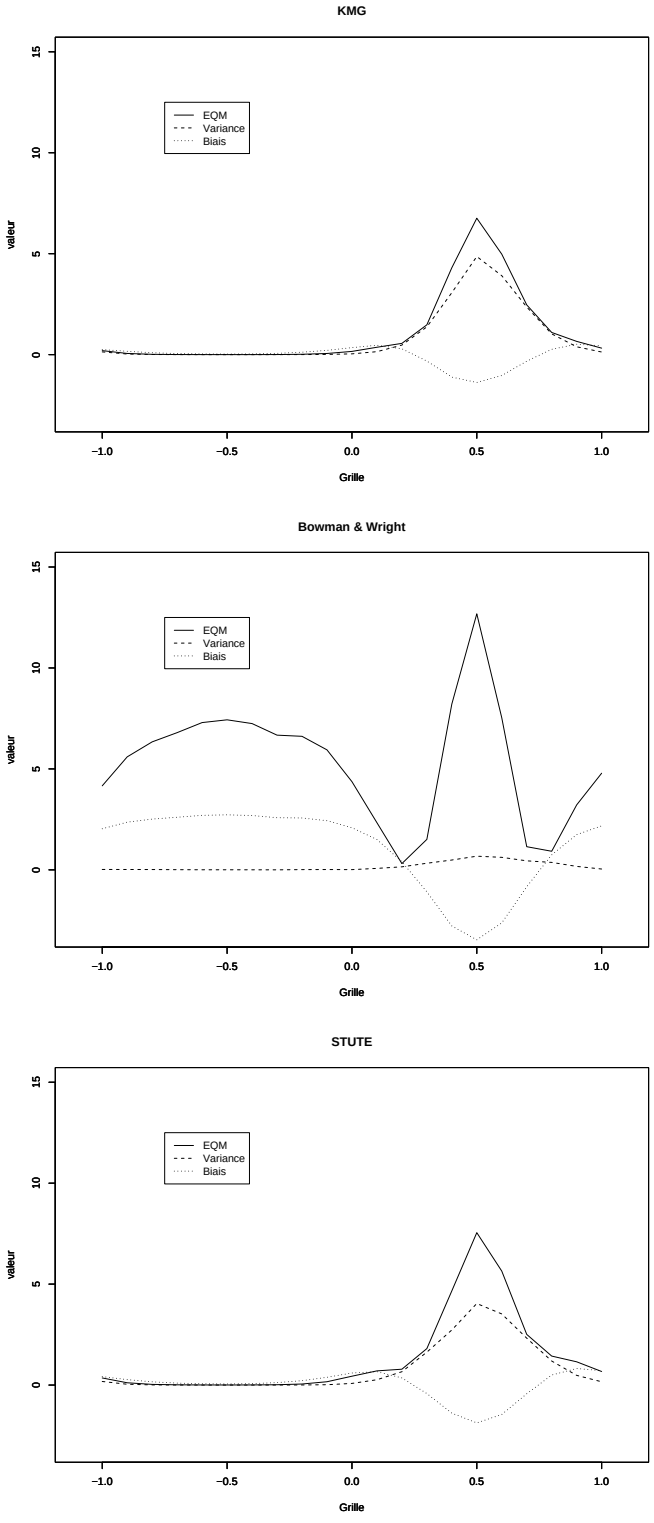


FIG. B.7 – *Graphiques présentant les résultats trouvés pour chacune des trois méthodes pour n=200, $\alpha = 0.25$ et λ=0.6.*

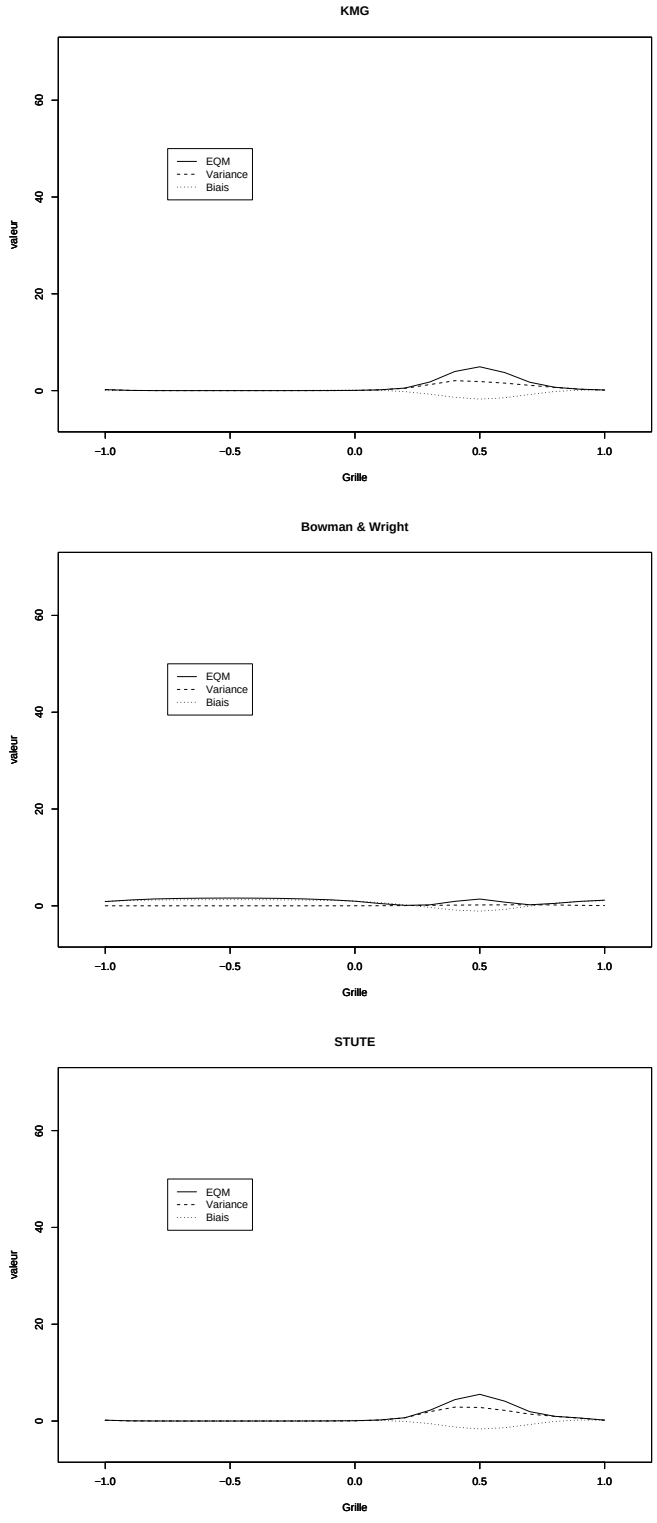


FIG. B.8 – *Graphiques présentant les résultats trouvés pour chacune des trois méthodes pour n=200, $\alpha = 0.5$ et λ=0.6.*

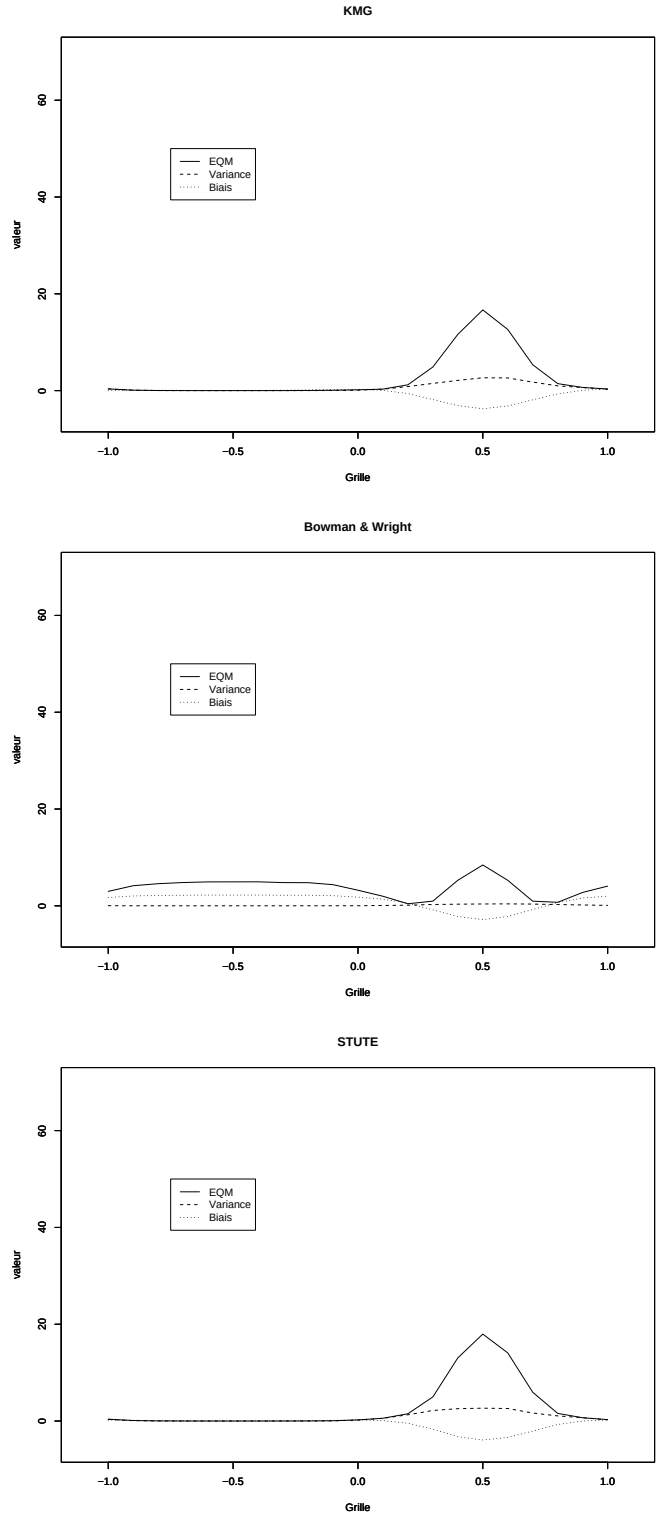


FIG. B.9 – *Graphiques présentant les résultats trouvés pour chacune des trois méthodes pour n=200, $\alpha = 0.75$ et λ=0.6.*

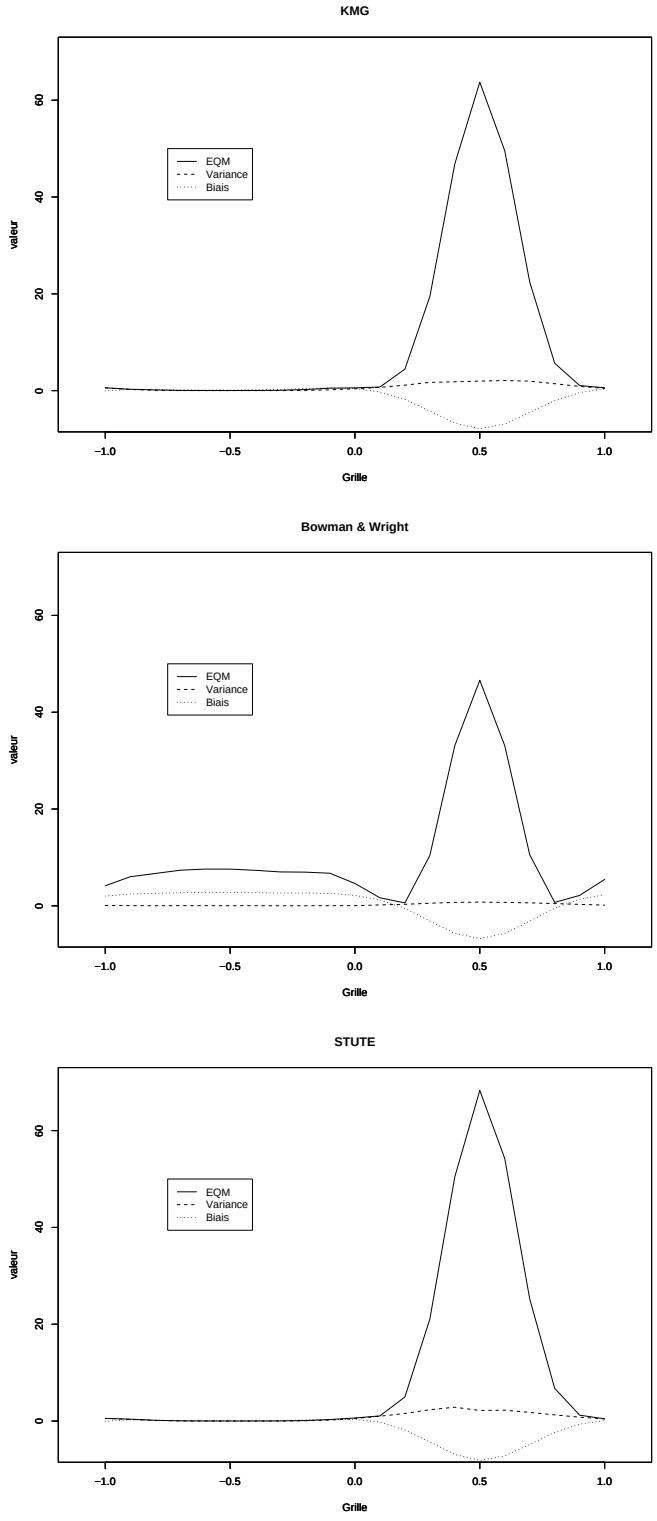


FIG. B.10 – *Graphiques présentant les résultats trouvés pour chacune des trois méthodes pour n=500, $\alpha = 0.25$ et λ=0.6.*

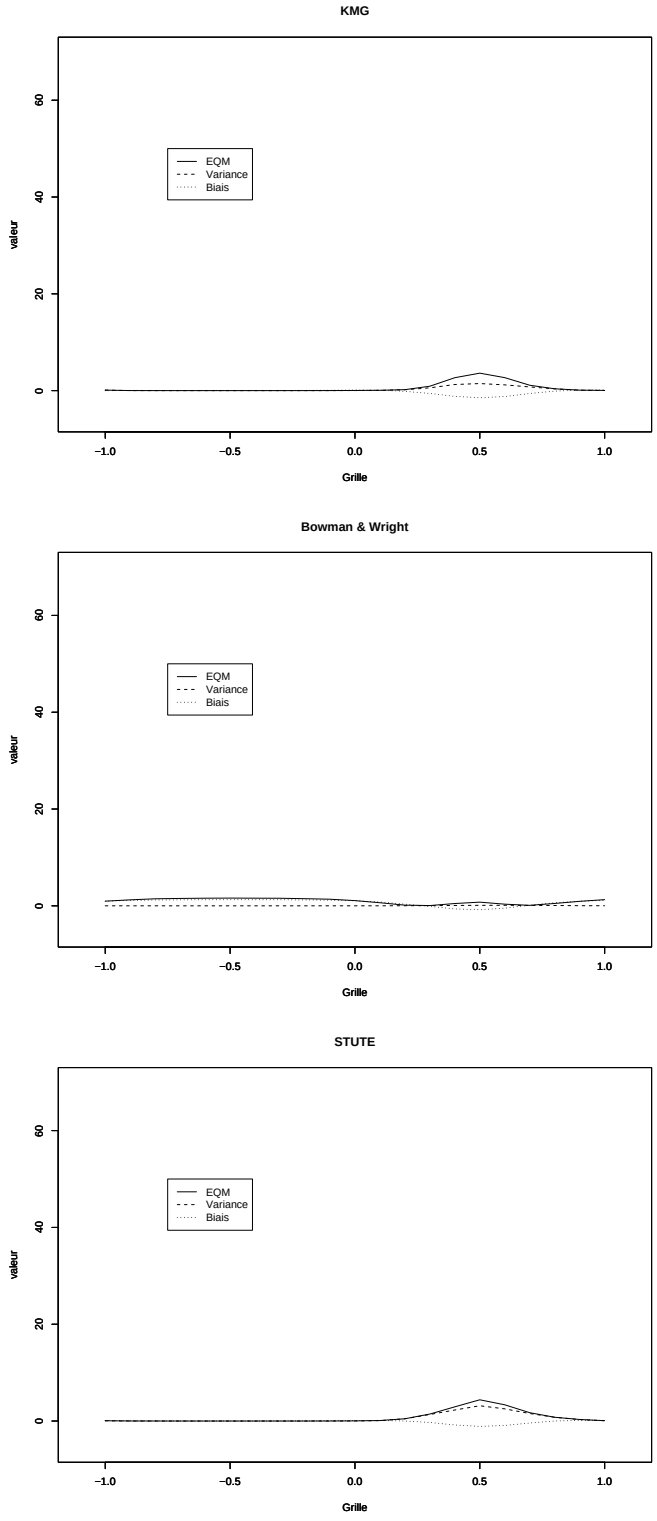


FIG. B.11 – *Graphiques présentant les résultats trouvés pour chacune des trois méthodes pour n=500, $\alpha = 0.5$ et λ=0.6.*

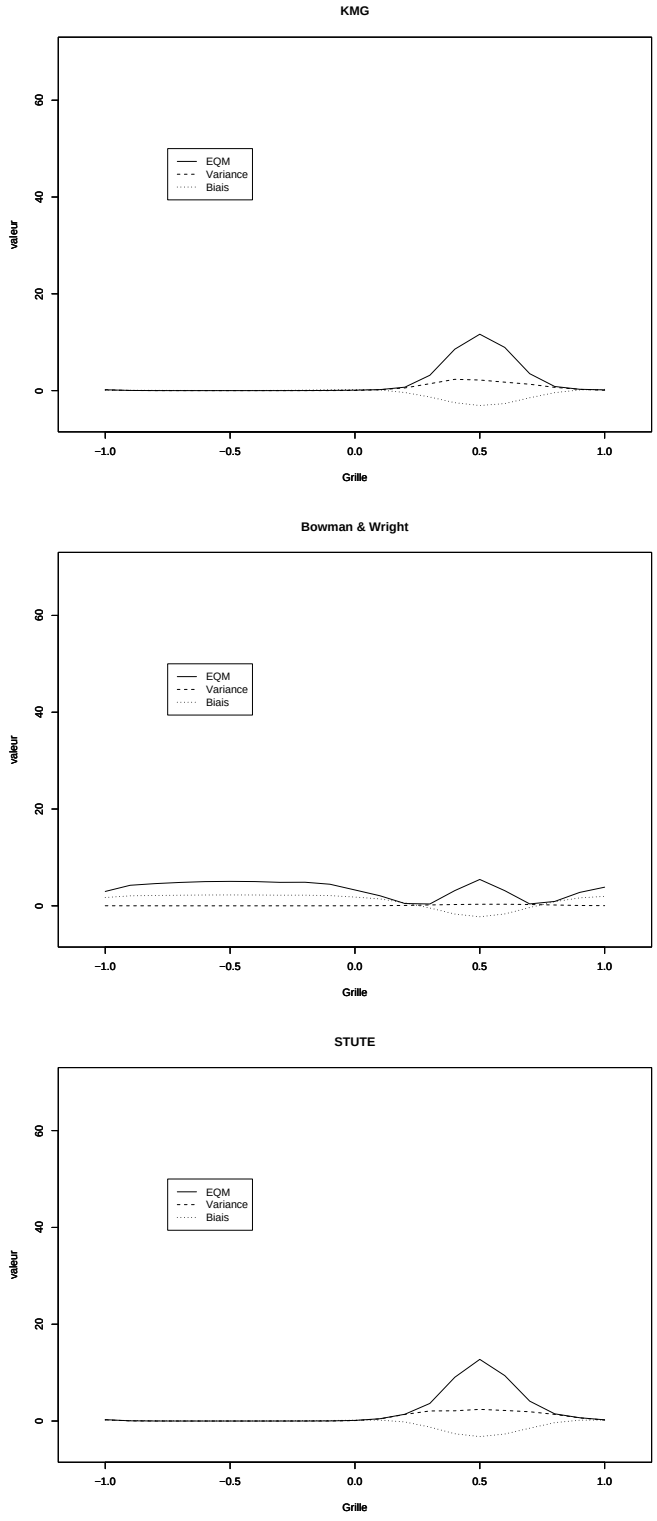
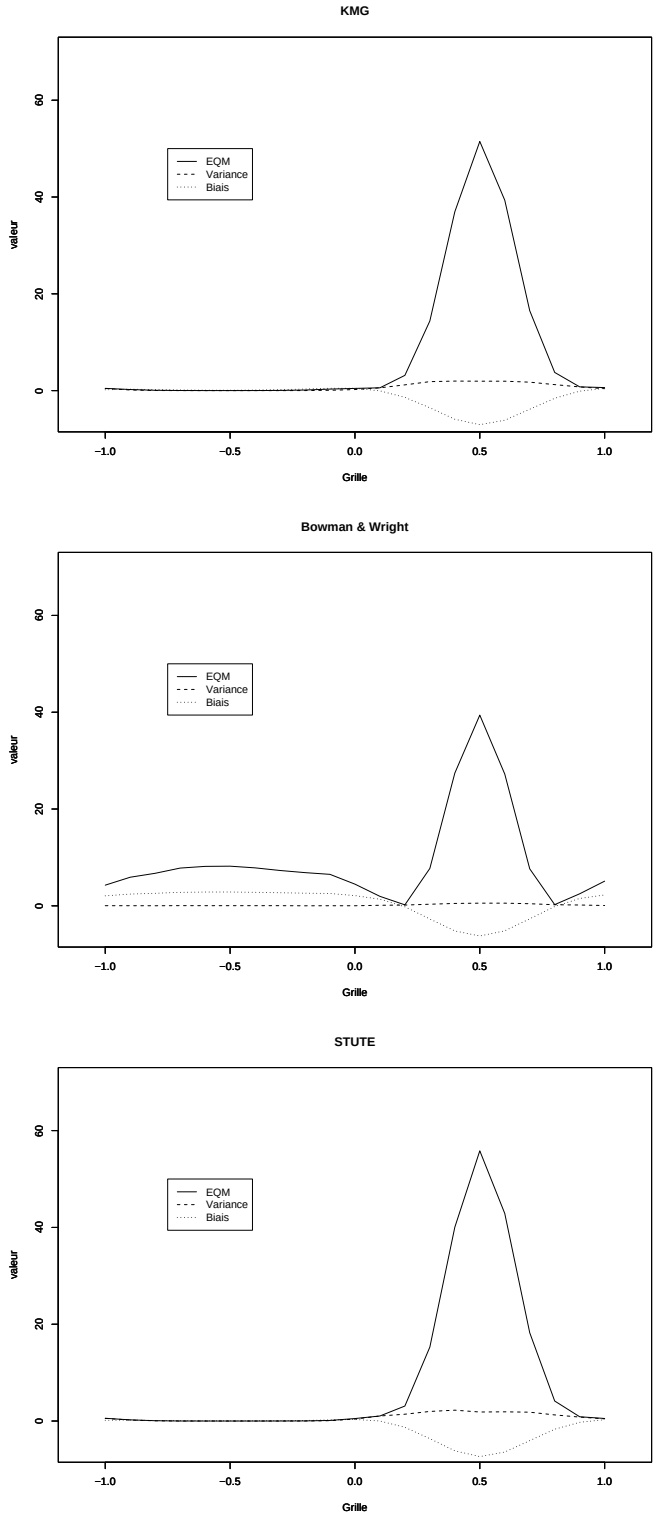


FIG. B.12 – *Graphiques présentant les résultats trouvés pour chacune des trois méthodes pour n=500, $\alpha = 0.75$ et λ=0.6.*



Annexe C

Programmes en langages R et C

C.1 Programmes en langage C

Voici tout d'abord le code C qui a été incorporé en langage R dans les analyses qui ont été effectuées dans ce mémoire. Afin de permettre cette incorporation, les trois fonctions ci-dessous doivent être sauvées dans un même fichier .c, qui sera par la suite compilé et chargé en R.

```
#include <R.h>
#include <Rmath.h>
#include <R_ext/Applic.h>

/* Poidsch12 calcul les poids Wi(x) à chaque point de la grille */

void Poidsch12 (double *echx,int *nech,double *hh,int *ngrid,double *grid,
                                                              double *repo)
{
  int i, j, k;
  double sum;
/* Calcul des Kh sur somme des Kh pour tous les points */
  for(j=0;j<*ngrid;j++){
    sum=0;
    for(i=0;i<*nech;i++){
      sum+=dnorm((echx[i]-grid[j])/ *hh,0,1,0);
      }
    for (k=0;k<*nech;k++){
```

```
      repo[k+(*nech*(j))]=dnorm((echx[k]-grid[j])/ *hh,0,1,0)/sum;
      }
    }
}

/* SautsGKM calcul la matrice des poids aux sauts dans le KMG */

void SautsGKM (double *echx,double *echy,double *echdelta,int *nech,
                    double *hh,int *ngrid,double *grid,double *repo)
{
  int i, j, k, l;
  double sum, prod;
  double KsurK[(*nech)*(*ngrid)];

/* Calcul des Kh sur somme des Kh pour tous les points */
  for(j=0; j<*ngrid; j++){
    for(i=0; i<*nech; i++){
      sum=0;
      for(k=i;k<*nech;k++){
        sum+=dnorm((echx[k]-grid[j])/ *hh,0,1,0);
        }
      KsurK[(*nech)*j+i]=dnorm((echx[i]-grid[j])/ *hh,0,1,0)/sum;
      }
    }

/* Calcul des sauts en Y_1 a chaque point de la grille */
  for(j=0; j<*ngrid; j++){
    repo[(*nech)*j]=KsurK[(*nech)*j] * echdelta[0];
    }

/* Calcul des sauts en Y_2, ..., Y_n a chaque point de la grille */
  for(j=0; j<*ngrid; j++){
    for(i=1;i<*nech; i++){
      prod=1;
      for(l=0;l<i;l++){
        prod=prod*pow(1-KsurK[(*nech)*j+l],echdelta[l]);
        }
      repo[((*nech)*j)+i]=echdelta[i]*KsurK[(*nech)*j+i]*prod;
      }
    }
}

/* SautsStute calcul la matrice des poids aux sauts donnés par Stute */
```

```
void SautsStute (double *echx1, double *echx,double *echy,
double *echdelta,int *nech,double *hh,int *ngrid,double *grid,
double *repo)
{
  int i, j, k, l;
  double sum, prod;
  double Stute[(*nech)*(*ngrid)], KsurK[(*nech)*(*ngrid)];

/* Calcul des Kh sur somme des Kh pour tous les points */
  for(j=0; j<*ngrid; j++){
    for(i=0; i<*nech; i++){
      sum=0;
      for(k=i;k<*nech;k++){
        sum+=dnorm((echx[k]-grid[j])/ *hh,0,1,0);
        }
      KsurK[(*nech)*j+i]=dnorm((echx[i]-grid[j])/ *hh,0,1,0)/sum;
      Stute[(*nech)*j+i]=dnorm((echx1[i]-grid[j])/ *hh,0,1,0);
        }
    }

/* Calcul des sauts en Y_1 a chaque point de la grille */
  for(j=0; j<*ngrid; j++){
    repo[(*nech)*j]=KsurK[(*nech)*j] * echdelta[0]*Stute[(*nech)*j];
    }

/* Calcul des sauts en Y_2, ..., Y_n a chaque point de la grille */
  for(j=0; j<*ngrid; j++){
    for(i=1;i<*nech; i++){
      prod=1;
      for(l=0;l<i;l++){
        prod=prod*pow(1-KsurK[(*nech)*j+l],echdelta[l]);
        }
      repo[((*nech)*j)+i]=echdelta[i]*KsurK[(*nech)*j+i]*prod
                                          *Stute[(*nech)*j+i];
      }
    }
}
```

C.2 Programmes en langage R

Voici maintenant le code R ayant permis d'effectuer les exemples présentés à la fin de chacun des trois premiers chapitres ainsi que les simulations que l'on voit au chapitre 4. Or, il faut tout d'abord savoir que la ligne de code qui suit doit être transcrite dans le logiciel R avant de faire rouler n'importe lequel des programmes en code R de cette section.

```
# Permet le chargement du code C en R
dyn.load("/vroy/R/Calcul_poids.so")
```

C.2.1 Programme qui permet d'effectuer les exemples des chapitres 1 et 2

```
# Cette fonction effectue le calcul des poids, à partir du code C

calculpoids<- function(ech,Grille,hh)
{
  y<-ech[,1]
  x<-ech[,2]
  nech<-length(ech[,1])  #nb de donnees dans ech
  ngrid<-length(Grille)  #nb de points sur grille
  a<-rep(1,nech*ngrid)
  poids.c <-.C("Poidsch12",echx=as.double(x),as.integer(nech),as.double(hh),
               as.integer(ngrid),grille=as.double(Grille),repo=as.double(a))
  output<-poids.c$repo
  return(output)
}

# Fonction à minimiser pour le chapitre 1 (localement linéaire)

A.minimiser<-function(ab,j,Ech,Bigmatrice,gril)
{
  z<-Ech[,1]-ab[1]-ab[2]*(Ech[,2]-gril[j])
  return(sum(z^2*Bigmatrice[,j]))
}

# Minimisation de la fonction A.minimiser : obtention de l'estimateur
```

```
Estimateur.j<-function(jj,Echa,Bigmat,grill,deb=1,fin=5)
{
  ab<-optim(c(deb,fin),A.minimiser,j=jj,Ech=Echa,Bigmatrice=Bigmat,
                                                    gril=grill)$par
  return(ab[1])
}

# Fonction à minimiser pour le chapitre 2 (localement linéaire)

A.minimiser.ch2<-function(ab,j,Ech,Bigmatrice,gril,alpha=0.25)
{
  z<-Ech[,1]-ab[1]-ab[2]*(Ech[,2]-gril[j])
  return(sum((abs(z) + (2*alpha-1)*z)*Bigmatrice[,j]))
}

# Minimisation de la fonction A.minimiser.ch2 : obtention de l'estimateur

Estimateur.j.ch2<-function(jj,Echa,Bigmat,grill,deb=1,fin=5,Alpha=0.25)
{
  ab<-optim(c(deb,fin),A.minimiser.ch2,j=jj,Ech=Echa,Bigmatrice=Bigmat,
                                        gril=grill,alpha=Alpha)$par
  return(ab[1])
}

# Fonction à minimiser pour le chapitre 1 (localement quadratique)

A.minimiser.quad<-function(abquad,j,Ech,Bigmatrice,gril)
{
  z<-Ech[,1]-abquad[1]-abquad[2]*(Ech[,2]-gril[j])-abquad[3]*
                                       (Ech[,2]-gril[j])^2
  return(sum(z^2*Bigmatrice[,j]))
}

# Minimisation de la fonction A.minimiser.quad : obtention de l'estimateur

Estimateur.j.quad<-function(jj,Echa,Bigmat,grill,deb=1,fin=5,fin2)
{
  ab<-optim(c(deb,fin,fin2),A.minimiser.quad,j=jj,Ech=Echa,
                          Bigmatrice=Bigmat,gril=grill)$par
  return(ab[1])
}

# Fonction à minimiser pour le chapitre 2 (localement quadratique)
```

```
A.minimiser.quad.ch2<-function(abquad,j,Ech,Bigmatrice,gril,alpha=0.25)
{
  z<-Ech[,1]-abquad[1]-abquad[2]*(Ech[,2]-gril[j])-abquad[3]*
                                          (Ech[,2]-gril[j])^2
  return(sum((abs(z) + (2*alpha-1)*z)*Bigmatrice[,j]))
}

# Minimisation de la fonction A.minimiser.quad.ch2 : obtention de
# l'estimateur

Estimateur.j.quad.ch2<-function(jj,Echa,Bigmat,grill,deb=1,fin=5,fin2,
                                                          Alpha=0.25)
{
  ab<-optim(c(deb,fin,fin2),A.minimiser.quad.ch2,j=jj,Ech=Echa,
                  Bigmatrice=Bigmat,gril=grill,alpha=Alpha)$par
  return(ab[1])
}

# Permet d'obtenir la meilleure fenêtre de lissage pour le chapitre 1
# à l'aide du calcul de la différence entre les estimateurs localement
# linéaires et localement quadratiques.

Fenetre.ch1<-function(grille.x=seq(118,300,by=2),fenetre=10,depart.a=1,
                                              depart.b=1,depart.c=1)
{
  ech<-read.table("/vroy/R/bodyfat_donnees.txt")
  y<-ech$V2
  x<-ech$V4
  echant<-cbind(y,x)
  Echant<-echant[order(echant[,1],echant[,2]),]
  BigVecteur<-calculpoids(Echant,grille.x,fenetre)
  BigMatriceGKM<-matrix(BigVecteur,nrow=length(Echant[,1]),byrow=F)
  ngrid<-length(grille.x)
  Estimateurs<-unlist(lapply((1:ngrid),Estimateur.j,Echa=Echant,
     Bigmat=BigMatriceGKM,grill=grille.x,deb=depart.a,fin=depart.b))
  Estimateurs.quad<-unlist(lapply((1:ngrid),Estimateur.j.quad,
              Echa=Echant,Bigmat=BigMatriceGKM,grill=grille.x,
                     deb=depart.a,fin=depart.b,fin2=depart.c))
  Output2<-Estimateurs-Estimateurs.quad
  Output2<-sum(Output2)^2
  print(Output2)
}
```

```
# Permet d'obtenir la meilleure fenêtre de lissage pour le chapitre 2
# à l'aide du calcul de la différence entre les estimateurs localement
# linéaires et localement quadratiques.

Fenetre.ch2<-function(grille.x=seq(118,300,by=2),fenetre=10,depart.a=1,
                                        depart.b=1,depart.c=1,ALPHA=0.25)
{
  ech<-read.table("/vroy/R/bodyfat_donnees.txt")
  y<-ech$V2
  x<-ech$V4
  echant<-cbind(y,x)
  Echant<-echant[order(echant[,1],echant[,2]),]
  BigVecteur<-calculpoids(Echant,grille.x,fenetre)
  BigMatriceGKM<-matrix(BigVecteur,nrow=length(Echant[,1]),byrow=F)
    ngrid<-length(grille.x)
  Estimateurs<-unlist(lapply((1:ngrid),Estimateur.j.ch2,Echa=Echant,
                                Bigmat=BigMatriceGKM,grill=grille.x,
                              deb=depart.a,fin=depart.b,Alpha=ALPHA))
  Estimateurs.quad<-unlist(lapply((1:ngrid),Estimateur.j.quad.ch2,
                  Echa=Echant,Bigmat=BigMatriceGKM,grill=grille.x,
             deb=depart.a,fin=depart.b,fin2=depart.c,Alpha=ALPHA))
  Output2<-Estimateurs-Estimateurs.quad
  Output2<-sum(Output2)^2
  print(Output2)
}

# Fonction produisant les estimateurs en chaque point de la grille pour le
# chapitre 1 qui ont ultérieurement servis à la création des graphiques

Estim.ch1<-function(grille.x=seq(118,300,by=2),fenetre=10,depart.a=1,
                                                          depart.b=5)
{
  ech<-read.table("/vroy/R/bodyfat_donnees.txt")
  y<-ech$V2
  x<-ech$V4
  echant<-cbind(y,x)
  Echant<-echant[order(echant[,1],echant[,2]),]
  BigVecteur<-calculpoids(Echant,grille.x,fenetre)
  BigMatriceGKM<-matrix(BigVecteur,nrow=length(Echant[,1]),byrow=F)
  ngrid<-length(grille.x)
  Estimateurs0<-unlist(lapply((1:ngrid),Estimateur.j,Echa=Echant,
   Bigmat=BigMatriceGKM,grill=grille.x,deb=depart.a,fin=depart.b))
```

```
  print(Estimateurs0)
}

# Fonction produisant les estimateurs en chaque point de la grille pour le
# chapitre 2 qui ont ultérieurement servis à la création des graphiques

Estim.ch2<-function(grille.x=seq(118,270,by=2),fenetre1=10,fenetre2=10,
                                        fenetre3=10,depart.a=1,depart.b=5)
{
  ech<-read.table("/vroy/R/bodyfat_donnees.txt")
  y<-ech$V2
  x<-ech$V4
  echant<-cbind(y,x)
  Echant<-echant[order(echant[,1],echant[,2]),]
  ngrid<-length(grille.x)
  ALPHA=c(0.25,0.5,0.75)

  BigVecteur<-calculpoids(Echant,grille.x,fenetre1)
  BigMatriceGKM<-matrix(BigVecteur,nrow=length(Echant[,1]),byrow=F)
  Estimateurs1<-unlist(lapply((1:ngrid),Estimateur.j.ch2,Echa=Echant,
                                  Bigmat=BigMatriceGKM,grill=grille.x,
                              deb=depart.a,fin=depart.b,Alpha=ALPHA[1]))
  print(Estimateurs1)
  BigVecteur<-calculpoids(Echant,grille.x,fenetre2)
  BigMatriceGKM<-matrix(BigVecteur,nrow=length(Echant[,1]),byrow=F)
  Estimateurs2<-unlist(lapply((1:ngrid),Estimateur.j.ch2,Echa=Echant,
                                  Bigmat=BigMatriceGKM,grill=grille.x,
                              deb=depart.a,fin=depart.b,Alpha=ALPHA[2]))
  print(Estimateurs2)
  BigVecteur<-calculpoids(Echant,grille.x,fenetre3)
  BigMatriceGKM<-matrix(BigVecteur,nrow=length(Echant[,1]),byrow=F)
  Estimateurs3<-unlist(lapply((1:ngrid),Estimateur.j.ch2,Echa=Echant,
                                  Bigmat=BigMatriceGKM,grill=grille.x,
                              deb=depart.a,fin=depart.b,Alpha=ALPHA[3]))
  print(Estimateurs3)
}
```

C.2.2 Programme permettant d'effectuer les exemples du chapitre 3

Le programme qui a servi à la méthode de KMG sera tout d'abord présenté et celui pour la méthode des poids pondérés par Stute suivra ensuite.

```
####################
#  Méthode de KMG  #
####################

# Calcul des sauts avec la methode de KMG, à partir du code C

sautsGKM.v2 <- function(ech,Grille,hh)
{
  y<-ech[,1]
  delta<-ech[,2]
  x<-ech[,3]
  nech<-length(ech[,1])  #nb de donnees dans ech
  ngrid<-length(Grille) #nb de points sur grille
  a<-rep(1,nech*ngrid)
  sautsGKM.c <-.C("SautsGKM",echx=as.double(x),echy=as.double(y),
        echdelta=as.double(delta),as.integer(nech),as.double(hh),
    as.integer(ngrid),grille=as.double(Grille),repo=as.double(a))
  output<-sautsGKM.c$repo
  return(output)
}

# Fonction à minimiser pour la méthode localement linéaire

A.minimiser<-function(ab,j,Ech,Bigmatrice,gril,alpha=0.25)
{
  z<-Ech[,1]-ab[1]-ab[2]*(Ech[,3]-gril[j])
  return(sum((abs(z) + (2*alpha-1)*z)*Bigmatrice[,j]))
}

# Minimisation de la fonction A.minimiser : obtention de l'estimateur

Estimateur.j<-function(jj,Echa,Bigmat,grill,deb=1,fin=5,Alpha=0.25)
{
  ab<-optim(c(deb,fin),A.minimiser,j=jj,Ech=Echa,Bigmatrice=Bigmat,
                                        gril=grill,alpha=Alpha)$par
```

```
  return(ab[1])
}

# Fonction à minimiser pour la méthode localement quadratique

A.minimiser.quad<-function(ab,j,Ech,Bigmatrice,gril,alpha=0.25)
{
  z<-Ech[,1]-ab[1]-ab[2]*(Ech[,3]-gril[j])-ab[3]*(Ech[,3]-gril[j])^2
  return(sum((abs(z) + (2*alpha-1)*z)*Bigmatrice[,j]))
}

# Minimisation de la fonction A.minimiser.quad : obtention de l'estimateur

Estimateur.j.quad<-function(jj,Echa,Bigmat,grill,deb=1,fin=5,fin2=5,
                                                        Alpha=0.25)
{
  ab<-optim(c(deb,fin,fin2),A.minimiser.quad,j=jj,Ech=Echa,Bigmatrice=Bigmat,
                                                  gril=grill,alpha=Alpha)$par
  return(ab[1])
}

# Permet d'obtenir la meilleure fenêtre de lissage, pour les femmes,
# à l'aide du calcul de la différence entre les estimateurs locale-
# ment linéaires et localement quadratiques.

Fenetre.kmg.fembl<-function(grille.x=seq(1,71,by=2),fenetre=1,depart.a=1,
                                    depart.b=5,depart.c=5,ALPHA=0.25)
{
  echfembl<-read.table("/vroy/R/femmesblanches.txt")
  y<-echfembl$V2
  delta<-echfembl$V3
  x<-echfembl$V6
  echant<-cbind(y,delta,x)
  Echant<-echant[order(echant[,1],echant[,3]),]
  ngrid<-length(grille.x)
  BigVecteur<-sautsGKM.v2(Echant,grille.x,fenetre)
  BigMatriceGKM<-matrix(BigVecteur,nrow=length(Echant[,1]),byrow=F)
  Estimateurs<-unlist(lapply((1:ngrid),Estimateur.j,Echa=Echant,Bigmat=
   BigMatriceGKM,grill=grille.x,deb=depart.a,fin=depart.b,Alpha=ALPHA))
  Estimateurs.quad<-unlist(lapply((1:ngrid),Estimateur.j.quad,Echa=Echant,
            Bigmat=BigMatriceGKM,grill=grille.x,deb=depart.a,fin=depart.b,
                                              fin2=depart.c,Alpha=ALPHA))
  Output2<-Estimateurs-Estimateurs.quad
```

```
  Output2<-sum(Output2)^2
  print(Output2)
}
# Permet d'obtenir la meilleure fenêtre de lissage, pour les hommes,
# à l'aide du calcul de la différence entre les estimateurs locale-
# ment linéaires et localement quadratiques.

Fenetre.kmg.hombl<-function(grille.x=seq(1,75,by=2),fenetre=1,depart.a=1,
                                      depart.b=5,depart.c=5,ALPHA=0.25)
{
  echhombl<-read.table("/vroy/R/hommesblanc.txt")
  y<-echhombl$V2
  delta<-echhombl$V3
  x<-echhombl$V6
  echant<-cbind(y,delta,x)
  Echant<-echant[order(echant[,1],echant[,3]),]
  ngrid<-length(grille.x)
  BigVecteur<-sautsGKM.v2(Echant,grille.x,fenetre)
  BigMatriceGKM<-matrix(BigVecteur,nrow=length(Echant[,1]),byrow=F)
  Estimateurs<-unlist(lapply((1:ngrid),Estimateur.j,Echa=Echant,Bigmat=
    BigMatriceGKM,grill=grille.x,deb=depart.a,fin=depart.b,Alpha=ALPHA))
  Estimateurs.quad<-unlist(lapply((1:ngrid),Estimateur.j.quad,Echa=Echant,
            Bigmat=BigMatriceGKM,grill=grille.x,deb=depart.a,fin=depart.b,
                                              fin2=depart.c,Alpha=ALPHA))
  Output2<-Estimateurs-Estimateurs.quad
  Output2<-sum(Output2)^2
  print(Output2)
}

# Fonction produisant, pour les femmes, les estimateurs en chaque point
# de la grille qui ont ultérieurement servis à la création des graphiques

Estim.kmg.fembl<-function(grille.x=seq(1,71,by=2),fenetre1=1,
        fenetre2=1,fenetre3=1,depart.a=1,depart.b=5,depart.c=5)
{
  echfembl<-read.table("/vroy/R/femmesblanches.txt")
  y<-echfembl$V2
  delta<-echfembl$V3
  x<-echfembl$V6
  echant<-cbind(y,delta,x)
  Echant<-echant[order(echant[,1],echant[,3]),]
  ngrid<-length(grille.x)
  ALPHA=c(0.25,0.5,0.75)
```

```
  BigVecteur<-sautsGKM.v2(Echant,grille.x,fenetre1)
  BigMatriceGKM<-matrix(BigVecteur,nrow=length(Echant[,1]),byrow=F)
  Estimateurs1<-unlist(lapply((1:ngrid),Estimateur.j,Echa=Echant,
   Bigmat=BigMatriceGKM,grill=grille.x,deb=depart.a,fin=depart.b,
                                                Alpha=ALPHA[1]))
  print(Estimateurs1)

  BigVecteur<-sautsGKM.v2(Echant,grille.x,fenetre2)
  BigMatriceGKM<-matrix(BigVecteur,nrow=length(Echant[,1]),byrow=F)
  Estimateurs2<-unlist(lapply((1:ngrid),Estimateur.j,Echa=Echant,
   Bigmat=BigMatriceGKM,grill=grille.x,deb=depart.a,fin=depart.b,
                                                Alpha=ALPHA[2]))
  print(Estimateurs2)

  BigVecteur<-sautsGKM.v2(Echant,grille.x,fenetre3)
  BigMatriceGKM<-matrix(BigVecteur,nrow=length(Echant[,1]),byrow=F)
  Estimateurs3<-unlist(lapply((1:ngrid),Estimateur.j,Echa=Echant,
   Bigmat=BigMatriceGKM,grill=grille.x,deb=depart.a,fin=depart.b,
                                                Alpha=ALPHA[3]))
  print(Estimateurs3)
}

# Fonction produisant, pour les hommes, les estimateurs en chaque point
# de la grille qui ont ultérieurement servis à la création des graphiques

Estim.kmg.hombl<-function(grille.x=seq(1,75,by=2),fenetre1=1,
        fenetre2=1,fenetre3=1,depart.a=1,depart.b=5,depart.c=5)
{
  echhombl<-read.table("/vroy/R/hommesblancs.txt")
  y<-echhombl$V2
  delta<-echhombl$V3
  x<-echhombl$V6
  echant<-cbind(y,delta,x)
  Echant<-echant[order(echant[,1],echant[,3]),]
  ngrid<-length(grille.x)
  ALPHA=c(0.25,0.5,0.75)

  BigVecteur<-sautsGKM.v2(Echant,grille.x,fenetre1)
  BigMatriceGKM<-matrix(BigVecteur,nrow=length(Echant[,1]),byrow=F)
  Estimateurs1<-unlist(lapply((1:ngrid),Estimateur.j,Echa=Echant,
   Bigmat=BigMatriceGKM,grill=grille.x,deb=depart.a,fin=depart.b,
                                                Alpha=ALPHA[1]))
```

```
  print(Estimateurs1)

  BigVecteur<-sautsGKM.v2(Echant,grille.x,fenetre2)
  BigMatriceGKM<-matrix(BigVecteur,nrow=length(Echant[,1]),byrow=F)
  Estimateurs2<-unlist(lapply((1:ngrid),Estimateur.j,Echa=Echant,
   Bigmat=BigMatriceGKM,grill=grille.x,deb=depart.a,fin=depart.b,
                                                    Alpha=ALPHA[2]))
  print(Estimateurs2)

  BigVecteur<-sautsGKM.v2(Echant,grille.x,fenetre3)
  BigMatriceGKM<-matrix(BigVecteur,nrow=length(Echant[,1]),byrow=F)
  Estimateurs3<-unlist(lapply((1:ngrid),Estimateur.j,Echa=Echant,
   Bigmat=BigMatriceGKM,grill=grille.x,deb=depart.a,fin=depart.b,
                                                    Alpha=ALPHA[3]))
  print(Estimateurs3)
}

########################################
#  Méthode des poids proposés par STUTE  #
########################################

# Calcul des sauts avec la methode des poids pondérés par Stute, à partir
# du code C

sautsStute.v2 <- function(ech1,ech,Grille,hh)
{
  y<-ech[,1]
  delta<-ech[,2]
  x<-ech[,3]
  x1<-ech1[,3]
  nech<-length(ech[,1])  #nb de donnees dans ech
  ngrid<-length(Grille) #nb de points sur grille
  a<-rep(1,nech*ngrid)
  sautsStute.c <-.C("SautsStute",echx1=as.double(x1),echx=as.double(x),
          echy=as.double(y),echdelta=as.double(delta),as.integer(nech),
              as.double(hh),as.integer(ngrid),grille=as.double(Grille),
                                                    repo=as.double(a))
  output<-sautsStute.c$repo
  return(output)
}

# Fonction à minimiser pour la méthode localement linéaire
```

```
A.minimiser.2<-function(ab,j,Ech,Bigmatrice,gril,alpha=0.25)
{
  z<-Ech[,1]-ab[1]-ab[2]*(Ech[,3]-gril[j])
  return(sum((abs(z) + (2*alpha-1)*z)*Bigmatrice[,j]))
}

# Minimisation de la fonction A.minimiser.2 : obtention de l'estimateur

Estimateur.j.2<-function(jj,Echa,Bigmat,grill,deb=1,fin=5,Alpha=0.25)
{
  ab<-optim(c(deb,fin),A.minimiser.2,j=jj,Ech=Echa,Bigmatrice=Bigmat,
                                          gril=grill,alpha=Alpha)$par
  return(ab[1])
}

# Fonction à minimiser pour la méthode localement quadratique

A.minimiser.quad.2<-function(ab,j,Ech,Bigmatrice,gril,alpha=0.25)
{
  z<-Ech[,1]-ab[1]-ab[2]*(Ech[,3]-gril[j])-ab[3]*(Ech[,3]-gril[j])^2
  return(sum((abs(z) + (2*alpha-1)*z)*Bigmatrice[,j]))
}

# Minimisation de la fonction A.minimiser.quad.2 : obtention de l'estimateur

Estimateur.j.quad.2<-function(jj,Echa,Bigmat,grill,deb=1,fin=5,fin2=5,
                                                              Alpha=0.25)
{
  ab<-optim(c(deb,fin,fin2),A.minimiser.quad.2,j=jj,Ech=Echa,Bigmatrice=
                                      Bigmat,gril=grill,alpha=Alpha)$par
  return(ab[1])
}

# Permet d'obtenir la meilleure fenêtre de lissage, pour les femmes,
# à l'aide du calcul de la différence entre les estimateurs locale-
# ment linéaires et localement quadratiques.

Fenetre.stute.fembl<-function(grille.x=seq(1,71,by=2),fenetre=1,depart.a=1,
                                        depart.b=5,depart.c=5,ALPHA=0.25)
{
  echfembl<-read.table("/vroy/R/femmesblanches.txt")
  y<-echfembl$V2
  delta<-echfembl$V3
```

```
  x<-echfembl$V6
  echant<-cbind(y,delta,x)
  Echant<-echant[order(echant[,1],echant[,3]),]
  echx.2<-rep(x[25], length(Echant[,3]))
  Echant.2<-cbind(Echant[,1],Echant[,2],echx.2)
  BigVecteur<-sautsStute.v2(Echant,Echant.2,grille.x,fenetre)
  BigMatriceGKM<-matrix(BigVecteur,nrow=length(Echant[,1]),byrow=F)
  ngrid<-length(grille.x)
  Estimateurs<-unlist(lapply((1:ngrid),Estimateur.j.2,Echa=Echant,
    Bigmat=BigMatriceGKM,grill=grille.x,deb=depart.a,fin=depart.b,
                                                    Alpha=ALPHA))
  Estimateurs.quad<-unlist(lapply((1:ngrid),Estimateur.j.quad.2,
   Echa=Echant,Bigmat=BigMatriceGKM,grill=grille.x,deb=depart.a,
                        fin=depart.b,fin2=depart.c,Alpha=ALPHA))
  Output2<-Estimateurs-Estimateurs.quad
  Output2<-sum(Output2)^2
  print(Output2)
}

# Permet d'obtenir la meilleure fenêtre de lissage, pour les hommes,
# à l'aide du calcul de la différence entre les estimateurs locale-
# ment linéaires et localement quadratiques.

Fenetre.stute.hombl<-function(grille.x=seq(1,75,by=2),fenetre=1,depart.a=1,
                                        depart.b=5,depart.c=5,ALPHA=0.25)
{
  echhombl<-read.table("/vroy/R/hommesblanc.txt")
  y<-echhombl$V2
  delta<-echhombl$V3
  x<-echhombl$V6
  echant<-cbind(y,delta,x)
  Echant<-echant[order(echant[,1],echant[,3]),]
  echx.2<-rep(x[25], length(Echant[,3]))
  Echant.2<-cbind(Echant[,1],Echant[,2],echx.2)
  BigVecteur<-sautsStute.v2(Echant,Echant.2,grille.x,fenetre)
  BigMatriceGKM<-matrix(BigVecteur,nrow=length(Echant[,1]),byrow=F)
  ngrid<-length(grille.x)
  Estimateurs<-unlist(lapply((1:ngrid),Estimateur.j.2,Echa=Echant,
    Bigmat=BigMatriceGKM,grill=grille.x,deb=depart.a,fin=depart.b,
                                                    Alpha=ALPHA))
  Estimateurs.quad<-unlist(lapply((1:ngrid),Estimateur.j.quad.2,
   Echa=Echant,Bigmat=BigMatriceGKM,grill=grille.x,deb=depart.a,
                        fin=depart.b,fin2=depart.c,Alpha=ALPHA))
```

```
  Output2<-Estimateurs-Estimateurs.quad
  Output2<-sum(Output2)^2
  print(Output2)
}

# Fonction produisant, pour les femmes, les estimateurs en chaque point
# de la grille qui ont ultérieurement servis à la création des graphiques

Estim.stute.fembl<-function(grille.x=seq(1,71,by=2),fenetre1=1,
          fenetre2=1,fenetre3=1,depart.a=1,depart.b=5,depart.c=5)
{
  echfembl<-read.table("/vroy/R/femmesblanches.txt")
  y<-echfembl$V2
  delta<-echfembl$V3
  x<-echfembl$V6
  echant<-cbind(y,delta,x)
  Echant<-echant[order(echant[,1],echant[,3]),]
  echx.2<-rep(x[25], length(Echant[,3]))
  Echant.2<-cbind(Echant[,1],Echant[,2],echx.2)
  ALPHA=c(0.25,0.5,0.75)
  ngrid<-length(grille.x)

  BigVecteur<-sautsStute.v2(Echant,Echant.2,grille.x,fenetre1)
  BigMatriceGKM<-matrix(BigVecteur,nrow=length(Echant[,1]),byrow=F)
  Estimateurs1<-unlist(lapply((1:ngrid),Estimateur.j.2,Echa=Echant,
     Bigmat=BigMatriceGKM,grill=grille.x,deb=depart.a,fin=depart.b,
                                                  Alpha=ALPHA[1]))
  print(Estimateurs1)

  BigVecteur<-sautsStute.v2(Echant,Echant.2,grille.x,fenetre2)
  BigMatriceGKM<-matrix(BigVecteur,nrow=length(Echant[,1]),byrow=F)
  Estimateurs2<-unlist(lapply((1:ngrid),Estimateur.j.2,Echa=Echant,
     Bigmat=BigMatriceGKM,grill=grille.x,deb=depart.a,fin=depart.b,
                                                  Alpha=ALPHA[2]))
  print(Estimateurs2)

  BigVecteur<-sautsStute.v2(Echant,Echant.2,grille.x,fenetre3)
  BigMatriceGKM<-matrix(BigVecteur,nrow=length(Echant[,1]),byrow=F)
  Estimateurs3<-unlist(lapply((1:ngrid),Estimateur.j.2,Echa=Echant,
     Bigmat=BigMatriceGKM,grill=grille.x,deb=depart.a,fin=depart.b,
                                                  Alpha=ALPHA[3]))
  print(Estimateurs3)
}
```

```
# Fonction produisant, pour les hommes, les estimateurs en chaque point
# de la grille qui ont ultérieurement servis à la création des graphiques

Estim.stute.hombl<-function(grille.x=seq(1,75,by=2),fenetre1=1,
          fenetre2=1,fenetre3=1,depart.a=1,depart.b=5,depart.c=5)
{
  echhombl<-read.table("/vroy/R/hommesblanc.txt")
  y<-echhombl$V2
  delta<-echhombl$V3
  x<-echhombl$V6
  echant<-cbind(y,delta,x)
  Echant<-echant[order(echant[,1],echant[,3]),]
  echx.2<-rep(x[25], length(Echant[,3]))
  Echant.2<-cbind(Echant[,1],Echant[,2],echx.2)
  ALPHA=c(0.25,0.5,0.75)
  ngrid<-length(grille.x)

  BigVecteur<-sautsStute.v2(Echant,Echant.2,grille.x,fenetre1)
  BigMatriceGKM<-matrix(BigVecteur,nrow=length(Echant[,1]),byrow=F)
  Estimateurs1<-unlist(lapply((1:ngrid),Estimateur.j.2,Echa=Echant,
     Bigmat=BigMatriceGKM,grill=grille.x,deb=depart.a,fin=depart.b,
                                                    Alpha=ALPHA[1]))
  print(Estimateurs1)

  BigVecteur<-sautsStute.v2(Echant,Echant.2,grille.x,fenetre2)
  BigMatriceGKM<-matrix(BigVecteur,nrow=length(Echant[,1]),byrow=F)
  Estimateurs2<-unlist(lapply((1:ngrid),Estimateur.j.2,Echa=Echant,
     Bigmat=BigMatriceGKM,grill=grille.x,deb=depart.a,fin=depart.b,
                                                    Alpha=ALPHA[2]))
  print(Estimateurs2)

  BigVecteur<-sautsStute.v2(Echant,Echant.2,grille.x,fenetre3)
  BigMatriceGKM<-matrix(BigVecteur,nrow=length(Echant[,1]),byrow=F)
  Estimateurs3<-unlist(lapply((1:ngrid),Estimateur.j.2,Echa=Echant,
     Bigmat=BigMatriceGKM,grill=grille.x,deb=depart.a,fin=depart.b,
                                                    Alpha=ALPHA[3]))
  print(Estimateurs3)
}
```

C.2.3 Programmes ayant servis à obtenir le biais, la variance et l'EQM au chapitre 4

Il est important de mentionner qu'avant d'utiliser un des trois programmes ci-dessous, il faut d'abord faire rouler la fonction Simul.echant ci-dessous.

```
#
# On commence par simuler des donnees,
# un echantillon de n observations
#

Simul.echant <-function(n,sigma2=2, rate2=0.6)
{
  epsilon <- rnorm(n,0,sqrt(sigma2))
  z <- rnorm(n,0,sqrt(sigma2))
  x <- exp(2*sin(pi*z) + epsilon)
  cens <- rexp(n, rate=rate2)
  delt <- 1*(x<=cens)
  x <- delt * x + (1-delt) * cens
  echa<-cbind(x,delt,z)
  echa<-echa[order(echa[,1],echa[,3]),]
}

####################
#  Methode de KMG  #
####################

#
#  On cree tout d'abord une matrice qui contient tous les sauts,
#  ou chaque colonne represente un point de la Grille
#  et chaque rangee represente une valeur de Y.
#

sautsGKM.v2 <- function(ech,Grille,hh)
{
  y<-ech[,1]
  delta<-ech[,2]
  x<-ech[,3]
  nech<-length(ech[,1])  #nb de donnees dans ech
  ngrid<-length(Grille) #nb de points sur grille
  a<-rep(1,nech*ngrid)
  sautsGKM.c <-.C("SautsGKM",echx=as.double(x),echy=as.double(y),echdelta=
```

```
        as.double(delta),as.integer(nech),as.double(hh),as.integer(ngrid),
                                grille=as.double(Grille),repo=as.double(a))
  output<-sautsGKM.c$repo
  return(output)
}

#
# Ensuite, on peut passer a l'estimation. Une premiere fonction définit
# la fonction objectif et une seconde fonction calcule l'estimateur au
# j-eme point de la grille.
#

A.minimiser<-function(ab,j,Ech,Bigmatrice,gril,alpha=0.25)
{
  z<-Ech[,1]-ab[1]-ab[2]*(Ech[,3]-gril[j])
  return(sum((abs(z) + (2*alpha-1)*z)*Bigmatrice[,j]))
}

Estimateur.j<-function(jj,Echa,Bigmat,grill,deb=1,fin=5,Alpha=0.25)
{
  ab<-optim(c(deb,fin),A.minimiser,,j=jj,Ech=Echa,Bigmatrice=Bigmat,
                                          gril=grill,alpha=Alpha)$par
  return(ab[1])
}

# On est pret a lancer la simulation!

# On trouve tout d'abord la meilleure fenêtre de lissage...

Simuler.fenetre<-function(N,taille,grille.x=seq(-1,1,by=0.2),fenetre=1,
                     sigma=2,rate3=0.6,depart.a=1,depart.b=5,ALPHA=0.25)
{
  Output2<-0
  for (i in (1:N)) {
    # taille: taille d'echantillon
    # grille.x: points ou l'on veut estimer les quantiles
    # fenetre: valeur de h, la fenetre de lissage
    # sigma: variance des epsilons dans la simulation
    # depart.a: valeur initiale de a dans les iterations
    # depart.b: valeur initiale de b dans les iterations
    # ALPHA: quantile a estimer
    Echant<-Simul.echant(n=taille,sigma2=sigma,rate2=rate3)
    BigVecteur<-sautsGKM.v2(Echant,grille.x,fenetre)
```

```
    BigMatriceGKM<-matrix(BigVecteur,nrow=length(Echant[,1]),byrow=F)
    ngrid<-length(grille.x)
    Estimateurs<-unlist(lapply((1:ngrid),Estimateur.j,Echa=Echant,
                           Bigmat=BigMatriceGKM,grill=grille.x,
                         deb=depart.a,fin=depart.b,Alpha=ALPHA))
    Vraies.valeurs<-exp(2*sin(pi*grille.x)+sqrt(sigma)*qnorm(ALPHA))
    u1 <- sum((Estimateurs-Vraies.valeurs)^2)
    Output2<-Output2 + u1
    }
  Output2<-Output2/N
  print(Output2)
}

# et ensuite, on peut passer au calcul des estimations.

Simuler.estim<-function(N,taille,grille.x=seq(-1,1,by=0.2),fenetre=1,sigma=2,
                           rate3=0.6,depart.a=1,depart.b=5,ALPHA=0.25)
{
  Output2<-0

  ngrid<-length(grille.x)
  u<-matrix(rep(1, N*ngrid),N,ngrid)
  for (i in (1:N)){
    # taille: taille d'echantillon
    # grille.x: points ou l'on veut estimer les quantiles
    # fenetre: valeur de h, la fenetre de lissage
    # sigma: variance des epsilons dans la simulation
    # depart.a: valeur initiale de a dans les iterations
    # depart.b: valeur initiale de b dans les iterations
    # ALPHA: quantile a estimer
    Echant<-Simul.echant(n=taille,sigma2=sigma,rate2=rate3)
    BigVecteur<-sautsGKM.v2(Echant,grille.x,fenetre)
    BigMatriceGKM<-matrix(BigVecteur,nrow=length(Echant[,1]),byrow=F)
    Estimateurs<-unlist(lapply((1:ngrid),Estimateur.j,Echa=Echant,
                               Bigmat=BigMatriceGKM,grill=grille.x,
                             deb=depart.a,fin=depart.b,Alpha=ALPHA))
    Vraies.valeurs<-exp(2*sin(pi*grille.x) + sqrt(sigma)*qnorm(ALPHA))
    u[i,]  <- Estimateurs-Vraies.valeurs
    }
  biais.kmg<-colMeans(u)
  variance.kmg<-diag(var(u))
  EQM.kmg<-biais.kmg^2+variance.kmg
  repo<-rbind(biais.kmg,variance.kmg,EQM.kmg)
```

```
  biais.integer.kmg<-sum(biais.kmg)
  var.integer.kmg<-sum(variance.kmg)
  EQM.integer.kmg<-sum(EQM.kmg)
  repo2<-c(biais.integer.kmg,var.integer.kmg,EQM.integer.kmg)
  print(repo)
  print(repo2)
}

########################################
#  Méthode des poids proposés par STUTE  #
########################################

#
#  On cree tout d'abord une matrice qui contient tous les sauts,
#  ou chaque colonne represente un point de la Grille
#  et chaque rangee represente une valeur de Y.
#

sautsStute.v2 <- function(ech1,ech,Grille,hh)
{
  y<-ech[,1]
  delta<-ech[,2]
  x<-ech[,3]
  x1<-ech1[,3]
  nech<-length(ech[,1])  #nb de donnees dans ech
  ngrid<-length(Grille) #nb de points sur grille
  a<-rep(1,nech*ngrid)
  sautsStute.c <-.C("SautsStute",echx1=as.double(x1),echx=as.double(x), echy=
     as.double(y),echdelta=as.double(delta),as.integer(nech),as.double(hh),
              as.integer(ngrid),grille=as.double(Grille),repo=as.double(a))
  output<-sautsStute.c$repo
  return(output)
}

#
# Ensuite, on peut passer a l'estimation. Une premiere fonction définit
# la fonction objectif et une seconde fonction calcule l'estimateur au
# j-eme point de la grille.
#

A.minimiser.2<-function(ab,j,Ech,Bigmatrice,gril,Hh,alpha=0.25)
{
  z<-Ech[,1]-ab[1]-ab[2]*(Ech[,3]-gril[j])
```

```
  return(sum((abs(z) + (2*alpha-1)*z)*Bigmatrice[,j]))
}

Estimateur.j.2<-function(jj,Echa,Bigmat,grill,hhH,deb=1,fin=5,Alpha=0.25)
{
  ab<-optim(c(deb,fin),A.minimiser.2,,j=jj,Ech=Echa,Bigmatrice=Bigmat,
                                    gril=grill,Hh=hhH,alpha=Alpha)$par
  return(ab[1])
}

# On est pret a lancer la simulation!

# On trouve tout d'abord la meilleure fenêtre de lissage...

Simuler.fenetre.stute<-function(N,taille,grille.x=seq(-1,1,by=0.2),
      fenetre=1,sigma=2,rate3=0.6,depart.a=1,depart.b=5,ALPHA=0.25)
{
  Output2<-0
  for (i in (1:N)){
    Echant<-Simul.echant(n=taille,sigma2=sigma,rate2=rate3)
    echx.2<-rep(Echant[1,3], length(Echant[,3]))
    Echant.2<-cbind(Echant[,1],Echant[,2],echx.2)
    BigVecteur<-sautsStute.v2(Echant,Echant.2,grille.x,fenetre)
    BigMatriceGKM<-matrix(BigVecteur,nrow=length(Echant[,1]),byrow=F)
    ngrid<-length(grille.x)
    Estimateurs<-unlist(lapply((1:ngrid),Estimateur.j.2,Echa=Echant,
                          Bigmat=BigMatriceGKM,grill=grille.x,
    hh=fenetre,deb=depart.a,fin=depart.b,Alpha=ALPHA))
    Vraies.valeurs<-exp(2*sin(pi*grille.x)+sqrt(sigma)*qnorm(ALPHA))
    u1 <- sum((Estimateurs-Vraies.valeurs)^2)

    Output2<-Output2 + u1
    }
  Output2<-Output2/N
  print(Output2)
}

# et ensuite, on peut passer au calcul des estimations.

Simuler.estim.stute<-function(N,taille,grille.x=seq(-1,1,by=0.2),fenetre=1,
                  sigma=2,rate3=0.6,depart.a=1,depart.b=5,ALPHA=0.25)
{
  Output2<-0
```

```
  ngrid<-length(grille.x)
  u<-matrix(rep(1, N*ngrid),N,ngrid)
  for (i in (1:N)){
    Echant<-Simul.echant(n=taille,sigma2=sigma,rate2=rate3)
    echx.2<-rep(Echant[1,3], length(Echant[,3]))
    Echant.2<-cbind(Echant[,1],Echant[,2],echx.2)
    BigVecteur<-sautsStute.v2(Echant,Echant.2,grille.x,fenetre)
    BigMatriceGKM<-matrix(BigVecteur,nrow=length(Echant[,1]),byrow=F)
    Estimateurs<-unlist(lapply((1:ngrid),Estimateur.j.2,Echa=Echant,
        Bigmat=BigMatriceGKM,grill=grille.x,hh=fenetre,deb=depart.a,
                                          fin=depart.b,Alpha=ALPHA))
    Vraies.valeurs<-exp(2*sin(pi*grille.x) + sqrt(sigma)*qnorm(ALPHA))
    u[i,] <-Estimateurs-Vraies.valeurs
    }
  biais.stute<-colMeans(u)
  variance.stute<-diag(var(u))
  EQM.stute<-biais.stute^2+variance.stute
  repo<-rbind(biais.stute,variance.stute,EQM.stute)
  biais.integer.stute<-sum(biais.stute)
  var.integer.stute<-sum(variance.stute)
  EQM.integer.stute<-sum(EQM.stute)
  repo2<-c(biais.integer.stute,var.integer.stute,EQM.integer.stute)
  print(repo)
  print(repo2)
}

##################################
#  Methode de Bowman et Wright  #
##################################

# Les sauts utilisés dans cette méthode sont les sauts du KMG!

#
# On peut donc passer a l'estimation. Une premiere fonction définit
# la fonction objectif et une seconde fonction calcule l'estimateur
# au j-eme point de la grille.
#

A.minimiser.bw<-function(ab,j,Ech,Bigmatrice,hh2,gril,alpha=0.25)
{
  z<-Ech[,1]-ab[1]-ab[2]*(Bigmatrice[j]-alpha)
  return(sum(Ech[,2]*z^2*dnorm((Bigmatrice[,j]-alpha)/hh2)))
}
```

```
Estimateur.j.bw<-function(jj,Echa,Bigmat,grill,deb=1,Hh2,fin=5,Alpha=0.25)
{
  ab<-optim(c(deb,fin),A.minimiser.bw,,j=jj,Ech=Echa,Bigmatrice=Bigmat,
                                    hh2=Hh2,gril=grill,alpha=Alpha)$par
  return(ab[1])
}

# On est pret a lancer la simulation!

# On trouve tout d'abord la meilleure fenêtre de lissage...

Simuler.fenetre.bw<-function(N,taille,grille.x=seq(-1,1,by=0.2),fenetre=1,
                  hhh2,sigma=2,rate3=0.6,depart.a=1,depart.b=5,ALPHA=0.25)
{
  Output2<-0
  ngrid<-length(grille.x)
  u<-matrix(rep(1, N*ngrid),N,ngrid)
  for (i in (1:N)) {
    Echant<-Simul.echant(n=taille,sigma2=sigma,rate2=rate3)
    BigVecteur<-sautsGKM.v2(Echant,grille.x,fenetre)
    BigMatriceGKM2<-matrix(rep(1,taille*ngrid),taille,ngrid)
    BigMatriceGKM<-matrix(BigVecteur,nrow=length(Echant[,1]),byrow=F)
    for (j in (1:ngrid)){
      BigMatriceGKM2[,j]<-cumsum(BigMatriceGKM[,j])
      }
    Estimateurs<-unlist(lapply((1:ngrid),Estimateur.j.bw,Echa=Echant,
                      Bigmat=BigMatriceGKM2,Hh2=hhh2,grill=grille.x,
                            deb=depart.a,fin=depart.b,Alpha=ALPHA))
    Vraies.valeurs<-exp(2*sin(pi*grille.x)+sqrt(sigma)*qnorm(ALPHA))
    u1 <- sum((Estimateurs-Vraies.valeurs)^2)

    Output2<-Output2 + u1
    }
  Output2<-Output2/N
  print(Output2)
}

# et ensuite, on peut passer au calcul des estimations.

Simuler.estim.bw<-function(N,taille,grille.x=seq(-1,1,by=0.2),fenetre=1,hhh2,
                    sigma=2,rate3=0.6,depart.a=1,depart.b=5,ALPHA=0.25)
{
```

```
  Output2<-0
  ngrid<-length(grille.x)
  u<-matrix(rep(1, N*ngrid),N,ngrid)
  for (i in (1:N)){
    Echant<-Simul.echant(n=taille,sigma2=sigma,rate2=rate3)
    BigVecteur<-sautsGKM.v2(Echant,grille.x,fenetre)
    BigMatriceGKM2<-matrix(rep(1,taille*ngrid),taille,ngrid)
    BigMatriceGKM<-matrix(BigVecteur,nrow=length(Echant[,1]),byrow=F)
    for (j in (1:ngrid)){
      BigMatriceGKM2[,j]<-cumsum(BigMatriceGKM[,j])
      }
    Estimateurs<-unlist(lapply((1:ngrid),Estimateur.j.bw,Echa=Echant,
                       Bigmat=BigMatriceGKM2,Hh2=hhh2,grill=grille.x,
                              deb=depart.a,fin=depart.b,Alpha=ALPHA))
    Vraies.valeurs<-exp(2*sin(pi*grille.x) + sqrt(sigma) * qnorm(ALPHA))
    u[i,]  <- Estimateurs-Vraies.valeurs
    }
  biais.bw<-colMeans(u)
  variance.bw<-diag(var(u))
  EQM.bw<-biais.bw^2+variance.bw
  repo<-rbind(biais.bw,variance.bw,EQM.bw)
  biais.integer.bw<-sum(biais.bw)
  var.integer.bw<-sum(variance.bw)
  EQM.integer.bw<-sum(EQM.bw)
  repo2<-c(biais.integer.bw,var.integer.bw,EQM.integer.bw)
  print(repo)
  print(repo2)
}
```