Université de Montréal

Estimateur des moindres carrés tronqués adaptatif par rééchantillonnage

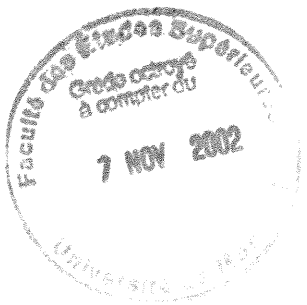
par

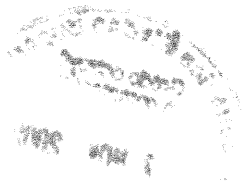
Jean-François Boudreau

Département de mathématiques et de statistique
Faculté des arts et des sciences

Mémoire présenté à la Faculté des études supérieures
en vue de l'obtention du grade de
Maître ès sciences (M.Sc.)
en statistique

août 2002





Université de Montréal

Faculté des études supérieures

Ce mémoire intitulé

Estimateur des moindres carrés tronqués adaptatif par rééchantillonnage

présenté par

Jean-François Boudreau

a été évalué par un jury composé des personnes suivantes :

Roch Roy

(président-rapporteur)

Christian Léger

(directeur de recherche)

François Perron

(membre du jury)

Mémoire accepté le:

_____

SOMMAIRE

Afin d'éviter l'influence de données aberrantes dans les contextes de régression linéaire, des estimateurs robustes tels que les moindres carrés tronqués (avec paramètre de troncature atteignant la robustesse maximale au niveau du point de rupture) sont habituellement utilisés. Une perte appréciable de performance est observée lorsqu'un jeu de données ne contenant que peu ou aucune donnée aberrante est analysé à l'aide de ces estimateurs.

Nous développerons dans ce mémoire un estimateur adaptatif, soit l'estimateur par moindres carrés tronqués de paramètre de troncature correspondant au minimum de l'estimation par rééchantillonnage de l'erreur quadratique moyenne de l'espérance au centre des données. Cet estimateur permettra entre autres de diminuer la variance de l'estimateur de β tout en conservant un biais raisonnable.

Nous nous intéresserons à deux estimateurs adaptatifs qui se différencieront par la technique de rééchantillonnage utilisée pour estimer le critère de performance. Une étude comparative des deux techniques sur les mêmes jeux de données nous portera à préférer pour sa simplicité le rééchantillonnage des paires.

Les estimateurs adaptatifs utiliseront l'estimateur par moindres carrés tronqués à robustesse maximale lors de l'estimation par rééchantillonnage du critère, ce qui ne l'empêchera pas de se démarquer de ce dernier, en particulier en ce qui concerne l'efficacité par rapport aux moindres carrés tronqués de paramètre optimal. De plus, la polyefficacité de l'estimateur adaptatif basée sur les 15 modèles à l'étude sera supérieure à celle de tous les estimateurs par moindres carrés tronqués à proportion de troncature fixe. L'estimateur des moindres carrés tronqués adaptatif par rééchantillonnage développé dans ce mémoire est donc supérieur, au sens de l'erreur quadratique moyenne de l'espérance à la moyenne, à tous les

estimateurs par moindres carrés tronqués dans une grande variété de situations de régression linéaire avec données aberrantes.

Mots clés : bootstrap, moindres carrés tronqués, régression robuste, données aberrantes, estimation adaptative.

SUMMARY

To avoid the influence of outliers in linear regression settings, robust estimators such as the least trimmed squares (with truncation parameter leading to the maximum breakdown) are often used. A great loss of performance is observed when a dataset containing few or no outlier is analysed with these estimators.

We are going to develop in this thesis an adaptive estimator. This estimator will be the least trimmed squares estimator with the truncation parameter corresponding to the minimum bootstrap estimation of the mean squared error of the expectation at the data's center. It will allow a reduction of the estimator's variance compared to the maximum breakdown estimator, while preserving a reasonably low bias.

We are going to focus on two adaptive estimators that differenciate themselves by the resampling technique used to estimate the performance criterion. A comparative study of the two techniques on the same datasets will lead us to prefer pair resampling for its simplicity.

Even though the adaptive estimator is going to use the maximum breakdown least trimmed squares estimator in the estimation of the criterion, the two estimators behave quite differently especially from an efficiency point of view. Moreover, the polyefficiency of the adaptive estimator based on the 15 models studied will be superior to any least trimmed squares estimator's polyefficiency. The resampling adaptive least trimmed squares estimator developped in this thesis is therefore superior, according to the mean squared error of expected value at the mean, to all the least trimmed squares estimators in a great variety of linear regression situations with outliers.

Key Words : bootstrap, least trimmed squares, robust regression, outliers, adaptive estimation.

REMERCIEMENTS

Un mémoire de maîtrise constitue un projet de longue haleine. Je veux donc en premier lieu remercier parents et amis qui m'ont supporté pendant ces deux dernières années. Vous en connaissez malgré vous un peu plus sur la régression robuste et le rééchantillonnage, et je vous en remercie.

Au cours de ma recherche, j'ai eu l'occasion de valider la réputation de Christian Léger parmi les étudiants, soit qu'on a davantage de questions non résolues en sortant de son bureau qu'en y entrant. Merci de m'avoir poussé à me surpasser.

Je profite de l'occasion pour remercier les professeurs, en particulier mon père, Christian Léger, et Jean-François Angers pour leurs judicieux conseils qui m'ont aidé à ressortir grandi de ma charge de cours.

Et finalement, merci au Conseil de recherches en sciences naturelles et en génie du Canada pour son support financier tout au long de ma maîtrise.

Table des matières

Table des figures

Liste des tableaux

INTRODUCTION

Lorsque le statisticien suspecte la présence de données aberrantes parmi les données à sa disposition, des estimateurs robustes du paramètre β du modèle de régression linéaire sont alors préférés à l'estimation par moindres carrés traditionnelle. L'estimateur par moindres carrés tronqués (MCT) avec paramètre de troncature h, et en particulier les MCT avec paramètre h atteignant la robustesse maximale au niveau du point de rupture, représentent une telle alternative.

Nous nous convaincrons par simulation que le paramètre h optimal (au sens du biais et de la variance, par exemple) n'est pas nécessairement celui menant à l'estimateur le plus robuste, ce dernier étant très variable. Nous nous proposons donc dans ce travail d'estimer par rééchantillonnage le paramètre h correspondant à l'estimateur par moindres carrés tronqués le plus performant au sens du biais et de la variance. En fait, nous adopterons un critère représentant l'erreur quadratique moyenne de l'espérance au centre des données. Nous étudierons deux méthodes de rééchantillonnage pour estimer le modèle de probabilité ayant généré les données, celles-ci correspondant aux cas où nous considérons le nombre de données aberrantes fixe ou aléatoire.

Les estimateurs adaptatifs ainsi définis représentent une alternative aux estimateurs à robustesse maximale ; ils ont entre autres l'avantage d'inclure un nombre de données h variant d'un jeu de données à l'autre. On peut ainsi souhaiter que le h choisi pour un jeu de données contenant n_a données aberrantes soit plus petit ou égal à $n-n_a$, évitant ainsi l'influence des données aberrantes qui aurait causé la rupture de l'estimateur par moindres carrés.

Les résultats des simulations basées sur 15 modèles différents montreront que l'estimateur adaptatif retenu rencontre bien les attentes formulées à son endroit,

en particulier concernant son efficacité par rapport à l'estimateur par moindres carrés tronqués de paramètre optimal.

Le premier chapitre sera consacré à l'étude du modèle de régression linéaire, aux estimateurs par moindres carrés tronqués, ainsi qu'à leurs propriétés de robustesse.

Au deuxième chapitre seront introduits les estimateurs par rééchantillonnage des caractéristiques de la distribution des estimateurs de β dans les contextes de régression, i.e. biais, variance, erreur quadratique moyenne, ainsi que les estimateurs adaptatifs définis par rééchantillonnage.

Les différents modèles étudiés, en particulier la distribution du nombre de données aberrantes qui peut être fixe ou aléatoire, ainsi que notre estimateur adaptatif seront discutés au troisième chapitre. Nous aborderons entre autres les sujets du choix du critère à optimiser et de l'estimateur $t(\hat{F})$ de β nécessaire aux estimations par rééchantillonnage du biais et de l'erreur quadratique moyenne. Notre choix pour $t(\hat{F})$ se portera sur l'estimateur par moindres carrés tronqués de robustesse maximale au niveau du point de rupture. Nous argumenterons enfin sur la légère perte de robustesse de l'estimateur adaptatif par rapport aux moindres carrés tronqués de robustesse maximale, habituellement utilisés dans les contextes de régression linéaire avec données aberrantes.

Le comportement, et en particulier l'efficacité de l'estimateur adaptatif par rapport à l'estimateur par moindres carrés tronqués à robustesse maximale, sera étudié à l'aide de simulations au chapitre quatre. Nous comparerons aussi les techniques de rééchantillonnage et en viendrons à la conclusion que le rééchantillonnage des paires, notamment grâce à sa simplicité, peut être utilisé avec succès dans tous les cas à l'étude.

Chapitre 1

RÉGRESSION LINÉAIRE À L'AIDE DES MOINDRES CARRÉS TRONQUÉS

1.1. INTRODUCTION

Nous nous intéresserons au modèle de régression linéaire suivant :

$$\mathbb{E}[\mathbf{y}|X] = X\beta, \tag{1.1}$$

où $\mathbf{y}_{(n\times 1)}$ est la variable expliquée, $X_{(n\times p)} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n)' = (\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^p)$ contenant la (les) variable(s) explicative(s) est une matrice de plein rang et $\beta_{(p\times 1)}$ est le vecteur des paramètres.

L'estimateur par moindres carrés (MC) de β est défini comme suit :

$$\begin{aligned}\hat{\beta}_{MC} &= \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^{n} (y_i - \mathbf{x}_i'\beta)^2 \\ &= \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^{n} r_i^2(\beta) \\ &= (X'X)^{-1}X'\mathbf{y}, \qquad (1.2)\end{aligned}$$

$$\text{où } r_i(\beta) = y_i - \mathbf{x}_i'\beta. \tag{1.3}$$

Cet estimateur, qui date des environs de 1800, est très répandu entre autres parce qu'il peut être calculé explicitement à l'aide de la formule (1.2). La distribution normale ou Gaussienne a été introduite par la suite comme distribution d'erreurs pour laquelle l'estimateur MC est optimal (voir Huber, 1972, p.1042 et Le Cam, 1986, p.79). Mais cet estimateur performe très mal lorsqu'une ou

plusieurs données ne suivent pas la relation linéaire $\mathbb{E}[\mathbf{y}|\mathbf{x}] = \mathbf{x}'\beta$. On dit que l'estimateur par moindres carrés est non robuste aux valeurs aberrantes.

Pour éviter que quelques points puissent ruiner complètement l'estimation du vecteur de paramètres β, mais aussi pour satisfaire d'autres critères, plusieurs autres estimateurs ont été proposés depuis. Pensons aux moindres valeurs absolues (Edgeworth, 1887) :

$$\hat{\beta}_{MVA} = \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^{n} |r_i(\beta)|, \tag{1.4}$$

où $r_i(\beta)$ est défini à l'équation (1.3). Il s'agit de la régression L_1, alors que MC est aussi appelé régression L_2. Cet estimateur diminue l'influence des points éloignés qui ne sont plus élevés au carré ici. Il est robuste à une valeur aberrante en y, mais non robuste à une valeur aberrante en $\mathbf{x}$.

Nous pouvons expliquer ce comportement à l'aide de la fonction d'influence, qui nous donne le changement dans un estimateur qui peut être causé par une quantité infinitésimale de contamination des données (Hampel et al., 1985). En particulier, si la fonction d'influence n'est pas bornée, un point peut avoir une très grande influence sur l'estimateur. Il est aussi souhaitable que la fonction d'influence soit continue, sinon l'estimateur sera instable autour des points de discontinuité.

L'estimateur (1.4) possède une fonction d'influence continue, mais non bornée. L'étude de la fonction nous montre qu'une valeur éloignée en $\mathbf{x}$ aura une influence non bornée sur l'estimateur $\hat{\beta}_{MVA}$ (Hampel et al., 1985, p.313).

Plus récemment, une famille d'estimateurs (qui inclut les deux précédents) a été proposée par Huber (1973). Les M-estimateurs sont définis comme suit :

$$\hat{\beta}_{M} = \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^{n} \rho(r_i(\beta)), \tag{1.5}$$

où $r_i(\beta)$ est défini à l'équation (1.3), et $\rho(u)$ est une fonction symétrique avec un seul minimum en $u = 0$. Ici encore, on peut diminuer l'influence des points éloignés en choisissant par exemple une fonction ρ bornée ; mais cet estimateur n'est toujours pas robuste à une valeur aberrante en $\mathbf{x}$. Sa fonction d'influence n'est pas bornée (voir Hampel et al., 1985, p.313).

De nouveaux estimateurs ont réussi où les précédents ont échoué : ils peuvent résister à une, et même plusieurs valeurs aberrantes. Dans ce qui suit, nous introduirons formellement une autre mesure de robustesse, et nous discuterons des différentes propriétés d'un estimateur de β très robuste : les moindres carrés tronqués (MCT).

1.2. Moindres carrés tronqués (MCT)

1.2.1. Définition et robustesse des MCT

L'estimateur par moindres carrés tronqués introduit par Rousseeuw (1984) est défini comme suit :

$$\hat{\beta}_h = \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^{h} (r^2(\beta))_{(i)}, \tag{1.6}$$

où $r^2(\beta) = (y - \mathbf{x}'\beta)^2$ et $(r^2(\beta))_{(1)} \leq (r^2(\beta))_{(2)} \leq \ldots \leq (r^2(\beta))_{(n)}$ sont les statistiques d'ordre des résidus carrés. Il s'agit donc de minimiser la somme des h plus petits résidus carrés.

Afin de comparer entre eux différents estimateurs en ce qui a trait à leur robustesse, nous définissons le concept de point de rupture ("breakdown point"). Nous utiliserons ici une version échantillonnale introduite par Donoho & Huber (1983). Soit un échantillon Z de n points et T un estimateur du vecteur des paramètres β, i.e. $T(Z) = \hat{\beta}$. Considérons tous les échantillons possibles Z' obtenus à partir de Z en remplaçant m des points originaux par des valeurs arbitraires. Finalement, notons $biais(m; T, Z)$ le biais maximal qui peut être causé à l'estimateur $T(Z)$ par contamination :

$$biais(m; T, Z) = \sup_{Z'} ||T(Z') - T(Z)||.$$

La notion de biais utilisée ici est donc différente de celle habituellement utilisée en probabilité et en statistique, soit la différence entre l'espérance de l'estimateur et la quantitée estimée. Notez que si $biais(m; T, Z)$ est infini, m valeurs aberrantes

peuvent avoir un effet arbitrairement grand sur T ; c'est donc dire que l'estimateur a rompu. Nous définissons donc le point de rupture (pour échantillon) de l'estimateur T pour l'échantillon Z.

Définition 1.1 (point de rupture). *Soit l'estimateur T pour l'échantillon Z de taille n. Le point de rupture (pour échantillon) de T pour Z est*

$$\epsilon_n(T, Z) = min\left\{\frac{m}{n} \ : \textit{tel que biais}(m; T, Z) \textit{ est infini}\right\}.$$

Le point de rupture de l'estimateur T est donc la plus petite fraction de contamination des données Z qui peut le modifier de façon arbitraire.

Les régressions L_2 (1.2) et L_1 (1.4) ainsi que les M-estimateurs (1.5) possèdent un point de rupture de $1/n$, puisqu'une seule valeur aberrante en $\mathbf{x}$ peut modifier les estimateurs de façon arbitraire. Nous avons déjà noté que ceci est dû aux fonctions d'influence qui ne sont pas bornées.

Puisqu'une fonction d'influence non bornée signifie qu'un point peut avoir une très grande influence sur l'estimateur, les estimateurs T dont le point de rupture $\epsilon_n(T, Z)$ est supérieur à $1/n$ ont des fonctions d'influence bornées. Mais il est important de noter ici qu'une fraction de contamination inférieure à $\epsilon_n(T, Z)$ n'implique pas que $T(Z)$ ne sera pas du tout affecté par des données aberrantes.

La proposition suivante montre que le point de rupture en régression possède une limite supérieure.

Proposition 1.1 (Rousseeuw & Leroy (1987), p.125). *Tout estimateur T équivariant en régression*
(c'est-à-dire $T(\{(\mathbf{x}_i, y_i + \mathbf{x}_i\mathbf{v}) ; i = 1, 2, \ldots, n\}) = T(\{(\mathbf{x}_i, y_i) ; i = 1, 2, \ldots, n\}) + \mathbf{v}$ pour un vecteur colonne $\mathbf{v}$) satisfait

$$\epsilon_n(T, Z) \leq \frac{\left[\frac{n-p}{2}\right] + 1}{n} \textit{ pour tout échantillon } Z,$$

où $[\ ]$ est la valeur entière, n la taille échantillonnale, et p le nombre de paramètres à estimer.

Cette limite d'environ 50% ne pourrait en toute logique être dépassée. Sinon comment pourrions-nous discerner les données appartenant au modèle des valeurs aberrantes sans les inverser ?

Proposition 1.2 (Rousseeuw & Leroy (1987), p.132). *Le point de rupture des moindres carrés tronqués (1.6) avec* $h = \left[\frac{n}{2}\right] + \left[\frac{p+1}{2}\right]$ *est égal à* $\left(\left[\frac{n-p}{2}\right] + 1\right)/n$, *où* $[\ ]$ *est la valeur entière,* n *la taille échantillonnale, et* p *le nombre de paramètres à estimer.*

C'est donc dire que l'estimateur

$$\hat{\beta}_{max} \equiv \hat{\beta}_{\left[\frac{n}{2}\right]+\left[\frac{p+1}{2}\right]} \tag{1.7}$$

possède la robustesse maximale. Notez que l'estimateur par moindres carrés tronqués utilisé en pratique est habituellement (1.7) ; mais les MCT sont en réalité une famille d'estimateurs indicée par le paramètre de troncature h où $max \equiv \left[\frac{n}{2}\right] + \left[\frac{p+1}{2}\right] \leq h \leq n$.

Nous nous concentrerons dans ce travail sur l'estimateur (1.7) et sur sa famille (1.6), mais il n'est pas le seul à posséder la robustesse maximale. Par exemple, l'estimateur par moindre médiane des carrés ("Least Median of Squares", LMS), aussi introduit par Rousseeuw (1984), atteint cette borne.

1.2.2. Exemples sur des jeux de données

Afin d'illustrer les résultats obtenus à l'aide des estimateurs par moindres carrés tronqués $\hat{\beta}_h$ (1.6) pour différentes proportions de troncature, nous considérons le modèle suivant :

$$\begin{cases} X_i = V_{1,i} \\ Y_i = V_{2,i} \end{cases} \qquad i = 1, 2, \ldots, n_a$$

$$\begin{cases} X_i = V_{3,i} \\ Y_i = \beta_0 + \beta_1 X_i + \epsilon_i \end{cases} \qquad i = n_a{+}1, \ldots, n$$

$$\begin{aligned} &\text{où } V_1 \sim N(6,\ 0{,}25), \\ &\quad V_2 \sim N(-1,\ 0{,}25), \\ &\quad V_3 \sim U(0{,}5,\ 4{,}5), \\ &\quad \epsilon \sim N(0, 4), \end{aligned} \tag{1.8}$$

$$\beta = (1, 2)', \text{ et } n = 50.$$

Le seul paramètre à déterminer est n_a, soit le nombre de valeurs aberrantes. Les situations $n_a = 0$, $n_a = 10$ et $n_a = 15$ sont illustrées aux figures 1.1, 1.2 et 1.3, respectivement. Les différentes valeurs du paramètre de troncature h utilisées lors du calcul des différents estimateurs sont indiquées dans les légendes (h=50 correspond à l'estimateur par moindres carrés (1.2).)

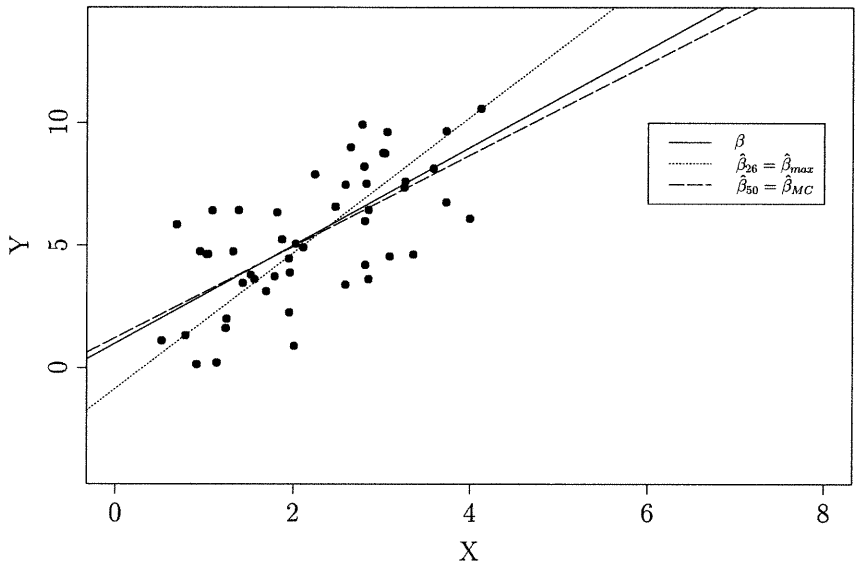


FIGURE 1.1. Modèle (1.8) où $n_a = 0$

Les graphiques ci-dessus nous montrent que l'estimateur $\hat{\beta}_{max}$=$\hat{\beta}_{26}$ (1.7) résiste à 10 ou même à 15 données aberrantes, soit 20% et 30% respectivement. Ceci est en fait l'objet de la proposition 1.2 à la page 7, qui nous indique que nous devrions théoriquement aller jusqu'à $\frac{\left[\frac{n-p}{2}\right]+1}{n} = \frac{\left[\frac{50-2}{2}\right]+1}{50} = 0{,}5$, soit 50% de données aberrantes pour qu'il y ait rupture.

Nous pouvons aussi remarquer que l'estimateur par moindres carrés tronqués ne se brise pas pour certaines valeurs de h supérieures à $max = 26$. Par exemple, lorsque 10 des données sont aberrantes (figure 1.2), $\hat{\beta}_{40}$ résiste à ces dernières. L'estimateur $\hat{\beta}_{41}$ est le premier à briser dans cet exemple. La même remarque s'applique lorsque $n_a = 15$ (figure 1.3) où $\hat{\beta}_{36}$ est le premier à briser sous l'influence des 15 données aberrantes. Tout comme l'estimateur $\hat{\beta}_{26}$ qui est brisé si 25 données

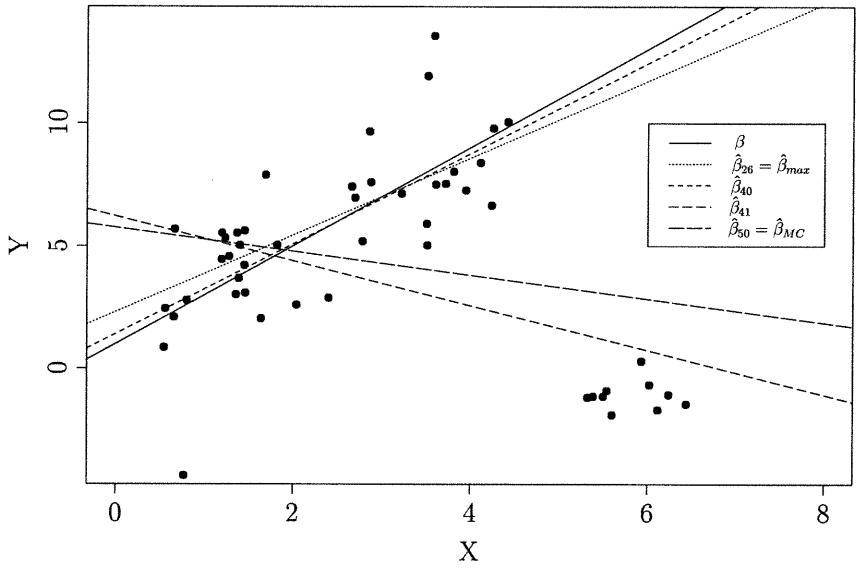


FIGURE 1.2. Modèle (1.8) où $n_a = 10$

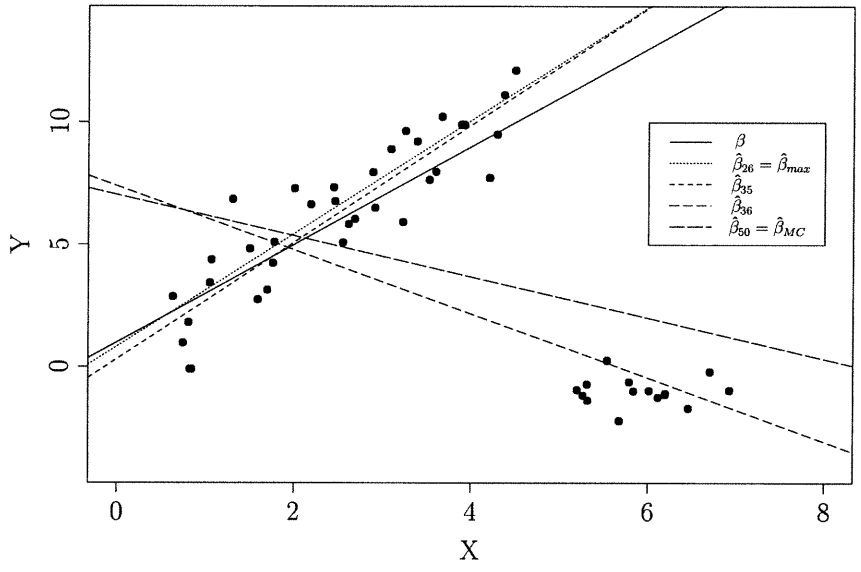


FIGURE 1.3. Modèle (1.8) où $n_a = 15$

ou plus sont aberrantes (proposition 1.2), on peut dire de façon générale que $n-h+1$ données aberrantes ou plus brisent l'estimateur $\hat{\beta}_h$.

Alors pourquoi utiliser systématiquement $\hat{\beta}_{max}$, lorsque 70% ou 80% des données sont non aberrantes ?

1.2.3. Choix du paramètre de troncature

Afin de guider notre choix de la valeur du paramètre h de (1.6) à utiliser, nous étudions le biais et la variance des estimateurs $\hat{\beta}_h$ à l'aide de simulations.

Les jeux de données des simulations 1, 2 et 3 sont générées à partir du modèle (1.8) où $n_a = 0$, $n_a = 10$ et $n_a = 15$ respectivement (voir les figures 1.1, 1.2 et 1.3 pour un exemple de jeu de données de chacun de ces modèles, et les pages 50 à 52 pour une synthèse des différentes simulations.)

Soit $\hat{\beta}_h(i) = \left(\hat{\beta}_{h,0}(i), \hat{\beta}_{h,1}(i)\right)'$ l'estimé par moindres carrés tronqués avec paramètre de troncature h du vecteur $\beta = (\beta_0, \beta_1)' = (1, 2)'$ pour la i^e simulation, $i = 1, 2, \ldots, Nsimul$.

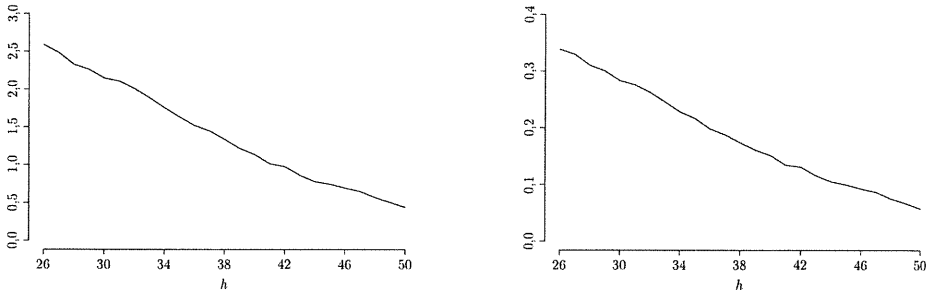
Notons

$$\left(biais(\hat{\beta}_{h,k})\right)^2 = \left(\frac{1}{Nsimul} \sum_{i=1}^{Nsimul} \hat{\beta}_{h,k}(i) - \beta_k\right)^2, \tag{1.9}$$

et

$$var(\hat{\beta}_{h,k}) = \frac{1}{Nsimul - 1} \sum_{i=1}^{Nsimul} (\hat{\beta}_{h,k}(i) - \hat{\beta}_{h,k}(\cdot))^2 \tag{1.10}$$

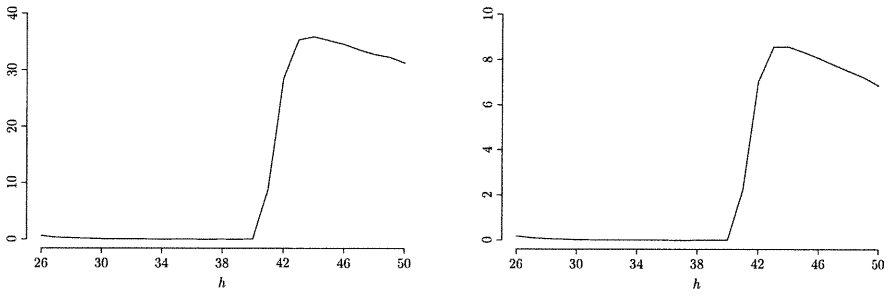
les approximations du biais et de la variance de $\hat{\beta}_{h,k}$ où $h \in \{26, 27, \ldots, 50\}$, $k \in \{0, 1\}$ et $\hat{\beta}_{h,k}(\cdot) = \frac{1}{Nsimul} \sum_{i=1}^{Nsimul} \hat{\beta}_{h,k}(i)$. Les figures 1.4, 1.5 et 1.6 nous montrent les valeurs prises par (1.9) et (1.10) pour les trois simulations.



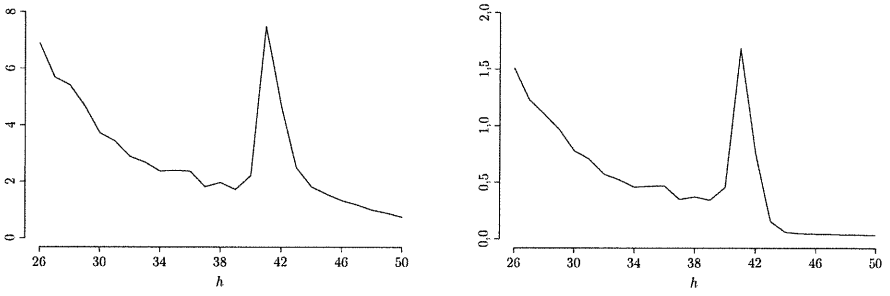
(a) $var(\hat{\beta}_{h,0})$ (b) $var(\hat{\beta}_{h,1})$

FIGURE 1.4. Simulation 1 ($n_a = 0$)

On peut remarquer à la figure 1.4 que la variance diminue lorsque h augmente. Pour cette simulation, le biais carré n'est pas illustré puisqu'il n'est pas significativement différent de zéro ; tous les estimateurs par moindres carrés tronqués sont en fait sans biais pour ce modèle.



(a) $\left(biais(\hat{\beta}_{h,0})\right)^2$ (b) $\left(biais(\hat{\beta}_{h,1})\right)^2$

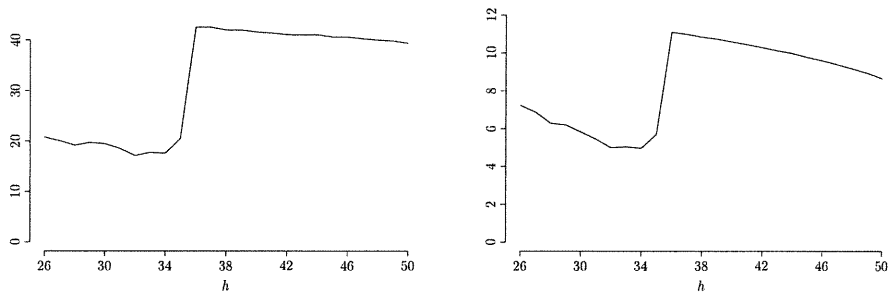


(c) $var(\hat{\beta}_{h,0})$ (d) $var(\hat{\beta}_{h,1})$

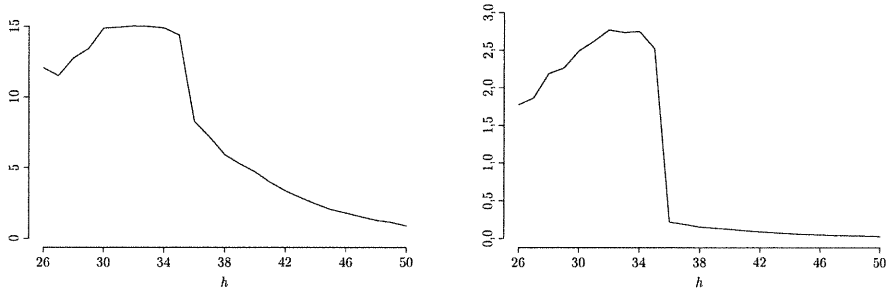
FIGURE 1.5. Simulation 2 ($n_a = 10$)

On remarque à la figure 1.5 que :

- le biais carré est très petit pour h inférieur ou égal au nombre de données non aberrantes, et qu'il augmente de façon importante à partir du moment où au moins une donnée aberrante est incluse ($h = 41$) ;
- tout comme à la simulation 1, la variance diminue lorsque h augmente pour les estimateurs non biaisés ;
- la variance est élevée pour $h = 41$ et très petite pour $h \geq 44$.



(a) $\left(biais(\hat{\beta}_{h,0})\right)^2$ (b) $\left(biais(\hat{\beta}_{h,1})\right)^2$



(c) $var(\hat{\beta}_{h,0})$ (d) $var(\hat{\beta}_{h,1})$

FIGURE 1.6. Simulation 3 ($n_a = 15$)

On remarque à la figure 1.6 que :

- le biais carré est élevé et décroissant pour h inférieur ou égal au nombre de données non aberrantes, et qu'il augmente de façon importante à partir du moment où au moins une donnée aberrante est incluse ($h = 36$) ;
- la variance est élevée et croissante pour h inférieur ou égal au nombre de données non aberrantes, et qu'elle décroît de façon importante lorsqu'une donnée aberrante est incluse ($h = 36$).

Cette dernière simulation nous montre que l'estimation par moindres carrés tronqués, quoique possédant un point de rupture élevé, peut être biaisée par un nombre restreint de données aberrantes. Les estimateurs non robustes discutés en introduction possèdent des fonctions d'influence non bornées, ce qui peut causer un très grand biais lors de l'estimation si une donnée est aberrante. Puisque la

fonction d'influence des moindres carrés tronqués est bornée (voir Rousseeuw & Leroy, 1987, p.191), cet estimateur est à l'abri d'un tel changement. Mais un grand changement borné tel que celui observé à la figure 1.6 reste possible.

Notez qu'on ne peut pas espérer un estimé aux alentours de β si une droite passant par les points aberrants est plus plausible au sens des moindres carrés sur h points que celles passant exclusivement par les points non aberrants. C'est ce qui est arrivé lors de la simulation 3. Pour plusieurs des jeux de données, la plus petite somme de h résidus carrés était atteinte en prenant un ou plusieurs points aberrants, et ce pour plusieurs valeurs de h ; même celles inférieures à 36. C'est pour cette raison que le biais carré (1.9) et la variance (1.10) sont élevés pour $h \leq 35$. Notez que cette situation aurait pu être évitée si la variance des points aberrants avait été plus grande et/ou celle des autres points avait été plus petite.

Nous devons donc toujours garder en tête le fait que l'estimation par moindres carrés tronqués est très robuste, mais faillible.

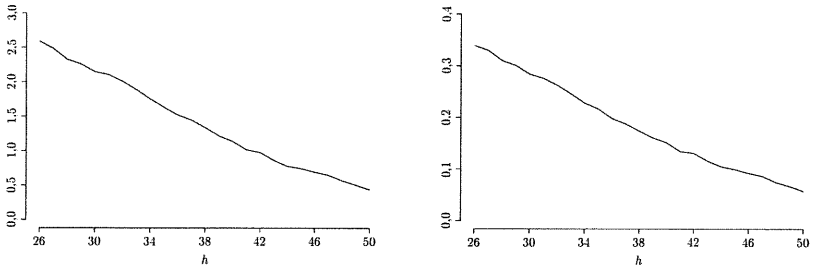
À la lumière des deux premières simulations, la valeur de h qui nous intéresse doit donc :

- être assez petite pour ne pas inclure de données aberrantes qui créeraient un biais important ;
- être assez grande pour bénéficier de la diminution de variance observée.

Afin d'atteindre ce double objectif, l'erreur quadratique moyenne (EQM) peut être utilisée :

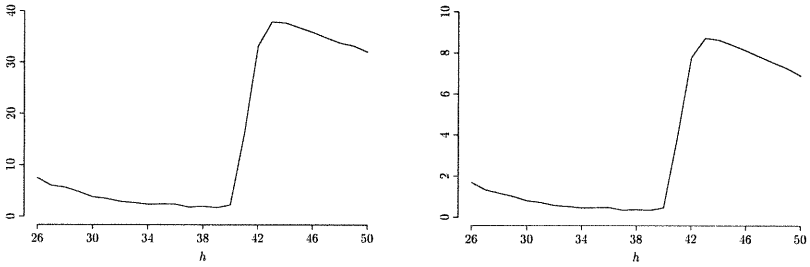
$$\begin{aligned} EQM(\hat{\beta}_{h,k}) &= \mathbb{E}\left[\left(\hat{\beta}_{h,k} - \beta_k\right)^2\right] \\ &= \left(biais(\hat{\beta}_{h,k})\right)^2 + var(\hat{\beta}_{h,k}). \end{aligned} \tag{1.11}$$

Les graphiques suivants nous montrent les valeurs prises par (1.11) pour les trois simulations à l'étude :



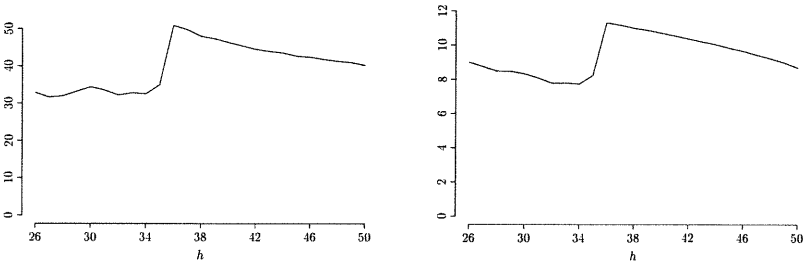
(a) $EQM(\hat{\beta}_{h,0})$ (b) $EQM(\hat{\beta}_{h,1})$

FIGURE 1.7. EQM pour la simulation 1 ($n_a = 0$)



(a) $EQM(\hat{\beta}_{h,0})$ (b) $EQM(\hat{\beta}_{h,1})$

FIGURE 1.8. EQM pour la simulation 2 ($n_a = 10$)



(a) $EQM(\hat{\beta}_{h,0})$ (b) $EQM(\hat{\beta}_{h,1})$

FIGURE 1.9. EQM pour la simulation 3 ($n_a = 15$)

En comparant ces graphiques à ceux du biais au carré et de la variance, on remarque que les EQM sont dominés par la valeur du biais carré lorsqu'au moins une donnée aberrante est incluse dans l'estimateur, et par la variance sinon. Enfin, les minimums semblent être atteints lorsque h est aux alentours du nombre de données non aberrantes (possiblement légèrement inférieur.) Ce choix du h qui minimise l'erreur quadratique moyenne est donc très intéressant. Nous pourrions utiliser $h = 50$ à la simulation 1 et $h = 40$ à la simulation 2 et ainsi obtenir des estimés non biaisés et moins variables que ceux obtenus avec $h = 26$; qui sont néanmoins les plus robustes.

Nous ne considérerons ici que la régression linéaire simple ; nous n'aurons donc à traiter que deux EQM : celui de $\hat{\beta}_{h,0}$ et celui de $\hat{\beta}_{h,1}$. Nous devrons néanmoins tenter de les combiner pour obtenir un critère, et ainsi éviter d'avoir à déterminer un minimum sur β_0, un minimum sur β_1, et choisir un h à partir de ces deux informations. Les méthodes de rééchantillonnage seront utilisées dans ce qui suit pour estimer différents critères basés sur les EQM afin de choisir la valeur de h qui rend l'estimateur $\hat{\beta}_h$ "optimal".

Chapitre 2

RÉÉCHANTILLONNAGE

2.1. INTRODUCTION

En plus d'un estimé ponctuel $\hat{\theta}$ du paramètre θ, les problèmes d'inférence nécessitent souvent l'estimation de différents aspects de la distribution de $\hat{\theta}$. Le rééchantillonnage peut être utilisé à cette fin, en particulier pour l'estimation du biais, de la variance et de l'erreur quadratique moyenne de l'estimateur.

Ce qui rend ces méthodes attrayantes est le fait qu'elles ne nécessitent pas de calculs théoriques qui devraient être refaits pour chaque problème, et qu'elles peuvent être utilisées facilement pour beaucoup de statistiques. Toutefois, la statistique doit être suffisamment lisse ; par exemple, le bootstrap ne fonctionne pas pour le minimum. L'obstacle majeur à l'utilisation routinière du rééchantillonnage a longtemps été le temps de calcul prohibitif ; mais on peut dire que cet obstacle n'en est plus un aujourd'hui.

2.2. ESTIMATION PAR LE PRINCIPE "PLUG-IN"

Afin d'estimer un paramètre $\theta = t(F)$ prenant la forme d'une fonctionnelle statistique de la fonction de répartition F, nous pouvons estimer F par la fonction de répartition expérimentale $\hat{F}$ et utiliser $\hat{\theta} = t(\hat{F})$ comme estimateur. Il s'agit du principe "plug-in".

Définition 2.1 (fonction de répartition expérimentale). *Soit un échantillon* $\mathbf{x} = \{x_1, x_2, \ldots, x_n\}$ *de variables aléatoires indépendantes et identiquement distribuées (i.i.d.) provenant de* F. *La fonction de répartition expérimentale* $\hat{F}$ *est la fonction*

de répartition pour laquelle une probabilité de $1/n$ est attribuée à chaque valeur x_i, $i = 1, 2, \ldots, n$:

$$\hat{F}(x) = \sum_{i=1}^{n} \mathbb{I}_{(-\infty,x]}(x_i),$$

où $\mathbb{I}_A(u)$ est la fonction indicatrice qui vaut 1 si $u \in A$ et 0 sinon.

Définition 2.2 (estimé "plug-in"). *L'estimé "plug-in" du paramètre $\theta = t(F)$ est défini par $\hat{\theta} = t(\hat{F})$ où F et $\hat{F}$ sont les fonctions de répartition et de répartition expérimentale, respectivement.*

Exemple 2.1. *Soit $\{x_1, x_2, \ldots, x_n\}$ un échantillon i.i.d. provenant de F. On estime $\theta = \mathbb{E}_F[X]$ par $\hat{\theta} = \mathbb{E}_{\hat{F}}[X^*] = \frac{1}{n}\sum_{i=1}^{n} x_i = \bar{x}$, où $X \sim F$ et $X^* \sim \hat{F}$.*
On estime aussi $\theta = Var_F(X) = \mathbb{E}_F[X^2] - (\mathbb{E}_F[X])^2$ par

$$\begin{aligned}\hat{\theta} &= Var_{\hat{F}}(X^*) = \mathbb{E}_{\hat{F}}[(X^*)^2] - (\mathbb{E}_{\hat{F}}[X^*])^2 \\ &= \frac{1}{n}\sum_{i=1}^{n} x_i^2 - \bar{x}^2 = \frac{1}{n}\sum_{i=1}^{n} (x_i - \bar{x})^2 . \end{aligned} \tag{2.1}$$

Notez que nous pouvons utiliser d'autres estimateurs que ceux obtenus ci-dessus à l'aide du principe "plug-in". Pour estimer la variance de X, notamment, nous utilisons habituellement l'estimateur sans biais $\hat{\theta} = s(\mathbf{x}) = \frac{1}{n-1}\sum_{i=1}^{n} (x_i - \bar{x})^2$.

Les méthodes de rééchantillonnage sont utilisées entre autres pour étudier des caractéristiques telles que le biais, la variance et l'erreur quadratique moyenne d'estimateurs $\hat{\theta} = s(\mathbf{x})$ (qui peuvent être l'estimateur "plug-in" $t(\hat{F})$), et utilisent elles-mêmes le principe "plug-in".

2.3. Estimations par rééchantillonnage

2.3.1. Estimation de la variance d'un estimateur pour un échantillon i.i.d.

Soit $\theta = t(F)$ une fonctionnelle statistique de F. Nous nous intéressons ici à l'estimation de $Var_F(\hat{\theta})$, la variance de l'estimateur $\hat{\theta} = s(\mathbf{x})$ évalué sur un échantillon i.i.d. $\mathbf{x}$ provenant de F. Notez que $\hat{\theta}$ peut être l'estimé "plug-in" $t(\hat{F})$, mais pas nécessairement.

Définition 2.3 (rééchantillon). *Soient* $\mathbf{x} = \{x_1, x_2, \ldots, x_n\}$ *un échantillon i.i.d. provenant de* F *et* $\hat{F}$ *sa fonction de répartition expérimentale. Un rééchantillon* $\mathbf{x}^* = \{x_1^*, x_2^*, \ldots, x_n^*\}$ *est un échantillon i.i.d. de taille* n *provenant de* $\hat{F}$.

La transformation intégrale de probabilité nous indique comment générer des données selon $\hat{F}$: il suffit de choisir avec remise des données parmi les observations originales $\{x_1, x_2, \ldots, x_n\}$.

Pour obtenir l'estimé par rééchantillonnage de $Var_F(\hat{\theta})$, nous utilisons l'estimé "plug-in" où $\hat{F}$ remplace F :

$$Var_{\hat{F}}(\hat{\theta}^*), \tag{2.2}$$

où $\hat{\theta}^* = s(\mathbf{x}^*)$ est la valeur de l'estimateur $\hat{\theta} = s(\cdot)$ pour un rééchantillon $\mathbf{x}^*$, alors que F a été remplacée par $\hat{F}$. La formule (2.2) est l'estimé idéal par rééchantillonnage de la variance de $\hat{\theta}$.

Exemple 2.2. *Si* $\hat{\theta} = \bar{X}$, *alors*

$$Var_F(\bar{X}) = \frac{Var_F(X)}{n}. \tag{2.3}$$

L'estimé (2.2) prend donc la forme

$$Var_{\hat{F}}(\bar{X}^*) = \frac{Var_{\hat{F}}(X^*)}{n} = \frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n^2}, \quad \textit{voir équation (2.1).}$$

Le cas de la moyenne $\bar{X}$ est exceptionnel puisqu'une formule aussi simple que (2.3) nous donne l'estimateur recherché directement. Un des avantages du rééchantillonnage est qu'il peut être utilisé pour des statistiques beaucoup plus compliquées que la moyenne échantillonnale.

Soit $\hat{\theta}$ la statistique d'intérêt pour laquelle une formule analogue à (2.3) pourrait être difficile à obtenir. Notez que si F était connue, nous pourrions générer B jeux de données à partir de celle-ci et faire l'approximation de $Var_F(\hat{\theta})$ par

$$\frac{1}{B-1} \sum_{b=1}^{B} (\hat{\theta}(b) - \hat{\theta}(\cdot))^2, \tag{2.4}$$

où $\hat{\theta}(b)$ est la valeur de l'estimateur pour le b^e jeu de données et $\hat{\theta}(\cdot) = \frac{1}{B} \sum_{b=1}^{B} \hat{\theta}(b)$. En utilisant la fonction de répartition expérimentale $\hat{F}$ pour remplacer F dans le raisonnement précédent, nous obtenons l'algorithme 2.1.

Algorithme 2.1 (estimation par rééchantillonnage de la variance de $\hat{\theta} = s(\mathbf{x})$).

1. *Générer B rééchantillons $\mathbf{x}^{*1}, \mathbf{x}^{*2}, \ldots, \mathbf{x}^{*B}$ en échantillonnant avec remise n données à partir de $\mathbf{x} = \{x_1, x_2, \ldots, x_n\}$;*
2. *Calculer $\hat{\theta}^*(b) = s(\mathbf{x}^{*b})$, $b = 1, 2, \ldots, B$;*
3. *Estimer $Var_F(\hat{\theta})$ par*

$$\widehat{Var}_B(\hat{\theta}^*) = \frac{1}{B-1}\sum_{b=1}^{B}\left(\hat{\theta}^*(b) - \hat{\theta}^*(\cdot)\right)^2, \text{ où } \hat{\theta}^*(\cdot) = \frac{1}{B}\sum_{b=1}^{B}\hat{\theta}^*(b). \tag{2.5}$$

L'algorithme 2.1 ne nécessite aucun calcul théorique et nous fournit un estimé aussi précis que voulu de $Var_{\hat{F}}(\hat{\theta}^*)$. En effet, la limite de $\widehat{Var}_B(\hat{\theta}^*)$ lorsque B tend vers l'infini est l'estimé idéal par rééchantillonnage de la variance de $\hat{\theta}$ (2.2) (Efron & Tibshirani, 1993, p.47) :

$$\lim_{B\to\infty} \widehat{Var}_B(\hat{\theta}^*) = Var_{\hat{F}}(\hat{\theta}^*) \text{ presque partout.}$$

2.3.2. Autres mesures de performance des estimateurs pour un échantillon i.i.d.

En plus de la variance, le biais et l'erreur quadratique moyenne de l'estimateur $\hat{\theta} = s(\mathbf{x})$ sont souvent d'intérêt. Le rééchantillonnage, et en particulier l'algorithme 2.1 avec une 3^e étape modifiée, sera utilisé dans ce qui suit pour les estimer.

Définition 2.4 (biais et erreur quadratique moyenne). *Soient $\mathbf{x} = \{x_1, x_2, \ldots, x_n\}$ un échantillon i.i.d. provenant de F, $\theta = t(F)$ le paramètre d'intérêt et $\hat{\theta} = s(\mathbf{x})$ l'estimateur de θ. Le biais de $\hat{\theta}$ comme estimateur de θ est la différence entre l'espérance mathématique de $\hat{\theta}$ et le paramètre θ :*

$$\begin{aligned} biais_F = biais_F(\hat{\theta}, \theta) &= \mathbb{E}_F[\hat{\theta}] - \theta \\ &= \mathbb{E}_F[s(\mathbf{x})] - t(F). \end{aligned} \tag{2.6}$$

L'erreur quadratique moyenne (EQM) de $\hat{\theta}$ est l'espérance mathématique de la différence carrée entre l'estimateur $\hat{\theta}$ et le paramètre θ :

$$\begin{aligned} EQM_F &= \mathbb{E}_F[(\hat{\theta} - \theta)^2] \\ &= \mathbb{E}_F[(s(\mathbf{x}) - t(F))^2]. \end{aligned} \tag{2.7}$$

Les estimés idéaux par rééchantillonnage du biais et de l'erreur quadratique moyenne sont obtenus en remplaçant F par $\hat{F}$ dans (2.6) et (2.7) :

$$biais_{\hat{F}} = \mathbb{E}_{\hat{F}}[s(\mathbf{x}^*)] - t(\hat{F}), \tag{2.8}$$

$$EQM_{\hat{F}} = \mathbb{E}_{\hat{F}}[(s(\mathbf{x}^*) - t(\hat{F}))^2]. \tag{2.9}$$

Notez que $\hat{\theta}$ peut être l'estimé "plug-in" $t(\hat{F})$. Dans ce cas, (2.6) et (2.8) sont remplacés par

$$biais_F = \mathbb{E}_F[t(\hat{F})] - t(F) \text{ et} \tag{2.6'}$$

$$biais_{\hat{F}} = \mathbb{E}_{\hat{F}}[t(\hat{F}^*)] - t(\hat{F}), \tag{2.8'}$$

où $\hat{F}^*$ est la fonction de répartition expérimentale du rééchantillon $\mathbf{x}^*$. La même remarque est applicable à l'erreur quadratique moyenne.

Comme dans le cas de la variance, des approximations des estimateurs (2.8) et (2.9) doivent habituellement être faites à l'aide de simulations de Monte Carlo. Nous utilisons pour cela l'algorithme 2.1 où la 3^e étape est remplacée par

$$\widehat{biais}_B(\hat{\theta}^*) = \frac{1}{B}\sum_{b=1}^{B}\hat{\theta}^*(b) - t(\hat{F}) \tag{2.10}$$

dans le cas du biais, et par

$$\widehat{EQM}_B(\hat{\theta}^*) = \frac{1}{B}\sum_{b=1}^{B}(\hat{\theta}^*(b) - t(\hat{F}))^2 \tag{2.11}$$

pour l'erreur quadratique moyenne. En pratique, $t(F)$ peut être difficile à calculer même pour F connu. Dans de tels cas, on utilise $s(\mathbf{x})$ ou un autre estimateur approprié à la place de $t(\hat{F})$ dans les estimés par rééchantillonnage (2.10) et (2.11). La même stratégie est utilisée lorsque le paramètre d'intérêt n'est pas une fonctionnelle.

Notez que les mêmes rééchantillons peuvent être utilisés pour calculer $\widehat{biais}_B(\hat{\theta}^*)$, $\widehat{EQM}_B(\hat{\theta}^*)$ et $\widehat{Var}_B(\hat{\theta}^*)$.

De façon générale, la variance, le biais et l'erreur quadratique moyenne de toute statistique s'exprimant comme une fonctionnelle de F peuvent être estimés par rééchantillonnage en remplaçant F par $\hat{F}$. Dans le cas d'un échantillon

i.i.d., nous utilisons simplement la fonction de répartition expérimentale, mais les méthodes de rééchantillonnage ne se limitent pas à ce type de données.

2.3.3. Autres types de jeux de données

L'étape cruciale de l'algorithme 2.1 est la première, alors qu'il faut estimer F par $\hat{F}$ et générer des données à partir de ce $\hat{F}$.

Notons P le modèle de probabilité qui a généré les données Z. Cette notation plus générale sera utilisée afin de discuter des cas où les données ne sont pas indépendantes et identiquement distribuées (i.i.d.). Il peut s'agir de plusieurs échantillons indépendants, d'une série chronologique, de données censurées à droite, d'une régression, etc. Quelle que soit la forme que prendra P, la méthodologie demeurera la même : estimer P par $\hat{P}$ afin d'obtenir des rééchantillons, et ainsi calculer les estimés par rééchantillonnage voulus ((2.5), (2.10) ou (2.11) par exemple.)

Exemple 2.3 (deux échantillons indépendants). *Notons $Z = (\mathbf{t}, \mathbf{c})$ deux échantillons i.i.d. indépendants représentant un groupe traitement* $\mathbf{t}$ *et un groupe contrôle* $\mathbf{c}$*, de tailles n_t et n_c respectivement. Nous pouvons caractériser le modèle de probabilité P à l'aide de deux fonctions de répartition F_t et F_c, correspondant aux groupes traitement et contrôle respectivement. L'estimé de $P = (F_t, F_c)$ sera donc de la forme $\hat{P} = (\hat{F}_t, \hat{F}_c)$ où $\hat{F}_t$ et $\hat{F}_c$ sont les fonctions de répartition expérimentales correspondant à F_t et à F_c. Un rééchantillon sera obtenu d'un rééchantillon* $\mathbf{t}^*$ *de taille n_t basé sur* $\mathbf{t}$ *et d'un rééchantillon* $\mathbf{c}^*$ *de taille n_c basé du* $\mathbf{c}$ *: $Z^* = (\mathbf{t}^*, \mathbf{c}^*)$.*

Si nous nous intéressons à la différence entre la moyenne μ_t de F_t et μ_c de F_c, les estimés de variance, biais, et erreur quadratique moyenne de la section précédente pourront être calculés pour étudier la distribution de la statistique $\hat{\theta} = s(Z) = \bar{t} - \bar{c}$.

À la section suivante, nous nous intéressons aux estimateurs par rééchantillonnage dans les contextes de régression linéaire.

2.4. RÉÉCHANTILLONNAGE DANS LES CONTEXTES DE RÉGRESSION

Soit le modèle de régression linéaire (1.1) que nous rappelons ici

$$\mathbb{E}[\mathbf{y}|X] = X\beta, \tag{2.12}$$

où $\mathbf{y}_{(n\times 1)}$ est la variable expliquée, $X_{(n\times p)} = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n)' = (\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^p)$ contient la (les) variable(s) explicative(s) et $\beta_{(p\times 1)}$ est le vecteur des paramètres.

On suppose pour le moment que le modèle prend la forme

$$\mathbf{y} = X\beta + \epsilon, \tag{2.13}$$

où ϵ est le vecteur d'erreurs et représente un échantillon i.i.d. de taille n provenant d'une distribution G de moyenne 0.

Sous les hypothèses précédentes concernant la forme du modèle et l'erreur, l'estimateur par moindres carrés $\hat{\beta}_{MC}$ (1.2) est sans biais pour β. De plus, si la variance des erreurs ϵ est σ_G^2, et si la matrice X est fixe, alors nous avons

$$Cov_G(\hat{\beta}_{MC}) = \sigma_G^2(X'X)^{-1}, \tag{2.14}$$

qui peut être estimé par

$$\widehat{Cov}_G(\hat{\beta}_{MC}) = \hat{\sigma}_G^2(X'X)^{-1}, \tag{2.15}$$

$$\text{où } \hat{\sigma}_G^2 = \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{n} = \frac{\sum_{i=1}^n (y_i - \mathbf{x}_i'\hat{\beta})^2}{n}; \tag{2.16}$$

ou par la version sans biais

$$\widetilde{Cov}_G(\hat{\beta}_{MC}) = \tilde{\sigma}_G^2(X'X)^{-1}, \tag{2.17}$$

$$\text{où } \tilde{\sigma}_G^2 = \frac{\sum_{i=1}^n (y_i - \mathbf{x}_i'\hat{\beta})^2}{n-p}.$$

Les méthodes de rééchantillonnage peuvent être utilisées lors de l'étude de la distribution de l'estimateur $\hat{\beta}$ de β pour le modèle (2.13) (qu'il s'agisse de l'estimateur par moindres carrés ci-dessus, ou d'un autre estimateur), mais aussi pour des modèles plus généraux. En effet, les méthodes peuvent être appliquées dans plusieurs situations alors que l'estimateur n'est pas sous une forme mathématiquement simple, le modèle utilise une fonction de régression non linéaire en β, ou

les erreurs ne sont pas i.i.d. ; des situations pour lesquelles des estimateurs comme (2.15) et (2.17) ne seraient pas simples à obtenir. Sans compter que des estimés de biais, d'erreur quadratique moyenne, d'erreur de prévision, etc. sont également faciles à obtenir par rééchantillonnage.

2.4.1. Rééchantillonnage des résidus

Supposons tout d'abord le modèle de régression (2.13) où $\epsilon \sim G$ est de moyenne 0. Comme il a déjà été mentionné, le modèle de probabilité $P = (G, \beta)$ doit être estimé afin de générer des rééchantillons à partir de $\hat{P} = (\hat{G}, \hat{\beta})$. Si nous connaissions β, nous pourrions estimer G par la fonction de répartition expérimentale du vecteur $\epsilon = \mathbf{y} - X\beta$, comme dans le cas d'un échantillon i.i.d.. Nous allons plutôt utiliser un estimateur $\hat{\beta}$ pour calculer $\hat{\epsilon} = \mathbf{y} - X\hat{\beta}$ et estimer G par la fonction de répartition expérimentale du vecteur $\tilde{\epsilon} = \hat{\epsilon} - \mathbf{1}\hat{\epsilon}_\bullet$ où $\hat{\epsilon}_\bullet = \frac{1}{n}\sum_{i=1}^{n} \hat{\epsilon}_i$. La constante $\hat{\epsilon}_\bullet$ a été soustraite de chaque composante de $\hat{\epsilon}$ afin de centrer les résidus et ainsi nous assurer que $\hat{G}$ soit de moyenne 0. Ayant à notre disposition des estimateurs pour les deux inconnues de P, nous pourrons générer des données selon $\hat{P}$ et étudier la distribution de $\hat{\beta}_{MC} = s(X, \mathbf{y})$ (ou de tout autre estimateur de β) à l'aide des estimateurs par rééchantillonnage ((2.5), (2.10) et (2.11).)

L'algorithme 2.1 de la page 19 peut être adapté au cas du rééchantillonnage par les résidus de $\hat{\beta}_{MC}$ de la façon suivante.

Algorithme 2.2 (estimation de la variance de $\hat{\beta}_{MC} = s(X, \mathbf{y})$ à l'aide du rééchantillonnage des résidus).

1. *Générer les rééchantillons*
   (a) *Estimer β de (2.13) par $\hat{\beta}$;*
   (b) *Générer B rééchantillons $\epsilon^{*1}, \epsilon^{*2}, \ldots, \epsilon^{*B}$ en échantillonnant avec remise parmi les n résidus centrés $\tilde{\epsilon} = \hat{\epsilon} - \mathbf{1}\hat{\epsilon}_\bullet$ où $\hat{\epsilon} = \mathbf{y} - X\hat{\beta}$ et $\hat{\epsilon}_\bullet = \frac{1}{n}\sum_{i=1}^{n} \hat{\epsilon}_i$;*
   (c) *Calculer $\mathbf{y}^{*b} = X\hat{\beta} + \epsilon^{*b}$;*
2. *Calculer $\hat{\beta}^*_{MC}(b) = (X'X)^{-1}X'\mathbf{y}^{*b}$, $b = 1, 2, \ldots, B$;*

3. Estimer $Cov_G(\hat{\beta}_{MC})$ *par*

$$\widehat{Cov}_B(\hat{\beta}^*_{MC}) = \frac{1}{B-1}\sum_{b=1}^{B}\left(\hat{\beta}^*_{MC}(b) - \hat{\beta}^*_{MC}(\cdot)\right)\left(\hat{\beta}^*_{MC}(b) - \hat{\beta}^*_{MC}(\cdot)\right)', \qquad (2.18)$$

$$où\ \ \hat{\beta}^*_{MC}(\cdot) = \frac{1}{B}\sum_{b=1}^{B}\hat{\beta}^*_{MC}(b).$$

Si nous nous intéressons à un autre estimateur que $\hat{\beta}_{MC}$, nous n'avons qu'à utiliser cet estimateur à la 2^e étape. L'estimateur par rééchantillonnage de la covariance de la 3^e étape peut aussi être remplacé par (2.19) ou (2.20) ci-dessous afin d'estimer le biais ou l'erreur quadratique moyenne respectivement :

$$\begin{aligned}\widehat{biais}_B(\hat{\beta}^*_{MC}) &= \frac{1}{B}\sum_{b=1}^{B}\hat{\beta}^*_{MC}(b) - t(\hat{G},\hat{\beta}) \\ &= \frac{1}{B}\sum_{b=1}^{B}\hat{\beta}^*_{MC}(b) - \hat{\beta}, \qquad (2.19)\end{aligned}$$

$$\widehat{EQM}_B(\hat{\beta}^*_{MC}) = \frac{1}{B}\sum_{b=1}^{B}(\hat{\beta}^*_{MC}(b) - \hat{\beta})(\hat{\beta}^*_{MC}(b) - \hat{\beta})'. \qquad (2.20)$$

Notez que puisque les estimateurs dans les contextes de régression sont des vecteurs, nous sommes passés de la variance à la matrice de covariance, que le biais est un vecteur, et que nous avons généralisé la définition de l'erreur quadratique moyenne qui comporte maintenant des termes croisés équivalents aux covariances.

À l'instar de la moyenne d'un échantillon i.i.d., une simulation de Monte Carlo n'est pas nécessaire pour estimer la variance de $\hat{\beta}_{MC}$ dans le cas où la distribution G est centrée en 0 et de variance σ^2_G, à condition que le modèle inclut un terme constant ($\mathbf{x}^1 = \mathbf{1}$), puisque

$$\begin{aligned}Cov_{\hat{G}}(\mathbf{y}^*) &= Cov_{\hat{G}}(X\hat{\beta} + \epsilon^*) \\ &= Cov_{\hat{G}}(\epsilon^*) \\ &= \frac{1}{n}\sum_{i=1}^{n}(\tilde{\epsilon}_i - \tilde{\epsilon}_\bullet)^2 \ \text{ voir (2.1)} \\ &= \frac{1}{n}\sum_{i=1}^{n}\tilde{\epsilon}_i{}^2\end{aligned}$$

$$= \frac{1}{n}\sum_{i=1}^{n} \hat{\epsilon}_i{}^2 \text{ si } s(X,\mathbf{y}) = \hat{\beta}_{MC} \text{ et } \mathbf{x}^1 = \mathbf{1}$$

$$= \hat{\sigma}^2_G, \tag{2.21}$$

donc

$$Cov_{\hat{G}}(\hat{\beta}^*_{MC}) = Cov_{\hat{G}}((X'X)^{-1}X'y^*)$$

$$= \hat{\sigma}^2_G(X'X)^{-1}, \tag{2.22}$$

où $\hat{\epsilon} = \mathbf{y} - X\hat{\beta}$, $\tilde{\epsilon} = \hat{\epsilon} - \hat{\epsilon}_\bullet$, $\hat{\epsilon}_\bullet = \frac{1}{n}\sum_{i=1}^n \hat{\epsilon}_i$, $\tilde{\epsilon}_\bullet = \frac{1}{n}\sum_{i=1}^n \tilde{\epsilon}_i = 0$ et $\hat{\sigma}^2_G$ est défini par (2.16). L'estimé idéal par rééchantillonnage de la variance de $\hat{\beta}_{MC}$ est donc égal à l'estimé biaisé (2.15). Un facteur de correction $\frac{n}{n-p}$ pourrait multiplier l'estimateur par rééchantillonnage de la variance (2.22) ; ce qui produirait l'estimateur sans biais (2.17). Mais Efron & Tibshirani (1993, p.112) ne le conseillent pas en faisant valoir que la variance de l'estimateur biaisé est plus importante que le biais causé par l'omission du facteur $\frac{n}{n-p}$.

Puisque l'estimation de la variance de $\hat{\beta}_{MC}$ du cas ci-dessus ne nécessite pas de simulation de Monte Carlo, l'algorithme 2.2 est plutôt utilisé dans d'autres situations. Nous avons déjà mentionné le cas où l'estimateur d'intérêt de β est différent de $\hat{\beta}_{MC}$, mais nous pouvons aussi l'utiliser, toujours avec modifications appropriées, lorsque la fonction de régression n'est pas linéaire en β par exemple.

Exemple 2.4 (étude de $\hat{\beta}_h$)**.** *Considérons à nouveau le modèle (2.13). Si nous voulons étudier un autre estimateur pour β que $\hat{\beta}_{MC}$, la 2^e étape doit être modifiée en conséquence. Par exemple, si nous voulons étudier la distribution de l'estimateur par moindres carrés tronqués $\hat{\beta}_h$ (1.6) pour un h donné, l'étape 2 de l'algorithme 2.2 sera remplacée par*

$$\hat{\beta}^*_h(b) = \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^{h} (r^{*2}(b,\beta))_{(i)}, \tag{2.23}$$

où $(r^{*2}(b,\beta))_{(1)} \leq (r^{*2}(b,\beta))_{(2)} \leq \ldots \leq (r^{*2}(b,\beta))_{(n)}$ sont les statistiques d'ordre des résidus carrés pour le b^e rééchantillon. Notez que la première étape de l'algorithme 2.2, et en particulier l'estimation de β par $\hat{\beta}$ pour générer les rééchantillons demeure inchangée. Mais en pratique, l'estimateur $\hat{\beta}$ utilisé pour calculer les résidus et dans le calcul du biais et de l'erreur quadratique moyenne peut différer de

l'estimateur "plug-in" ou de l'estimateur à l'étude appliqué aux données originales. Dans les contextes de régression robustes, notamment, un estimateur très robuste tel que $\hat{\beta}_{max}$ (1.7) pourrait être utilisé comme nous le verrons à la section 3.4.3.

L'algorithme 2.2 peut être utilisé dans beaucoup de situations ; mais la façon de générer les rééchantillons dépend du modèle retenu. C'est donc dire que les estimateurs par rééchantillonnage (2.18), (2.19) et (2.20) dépendent de l'adéquation du modèle utilisé pour les calculer. Nous verrons à la section suivante une autre méthode de rééchantillonnage utilisée dans les contextes de régression qui est basée sur des hypothèses beaucoup moins fortes que le rééchantillonnage des résidus de l'algorithme 2.2.

2.4.2. Rééchantillonnage des paires

Le modèle (2.12) peut être formulé de deux façons qui, quoique très semblables, nous poussent à estimer P différemment (voir Davidson & Hinkley, 1997, p.259).

Nous pouvons considérer que la réponse y pour une valeur $\mathbf{x}$ provient d'une distribution $F_{\mathbf{x}}$ de moyenne $\mu(\mathbf{x})$ et variance $\sigma^2(\mathbf{x})$ telle que $\mu(\mathbf{x}) = \mathbf{x}'\beta$. Dans le cas où les erreurs sont indépendantes et identiquement distribuées, nous avons $\sigma^2(\mathbf{x}) = \sigma^2$ et $F_{\mathbf{x}}(y) = G(y - \mu(\mathbf{x}))$ où G, de moyenne 0 et variance σ^2, représente la distribution des erreurs ϵ du modèle (2.13). À la section précédente, nous avons considéré la matrice X fixe et avons estimé G par la fonction de répartition expérimentale des résidus centrés calculés à l'aide d'un estimé $\hat{\beta}$ de β.

Mais nous pouvons considérer que les paires $\mathbf{z}_i = (\mathbf{x}_i, y_i)$ sont indépendantes et identiquement distribuées selon une distribution multivariée P quelconque. Pour estimer P dans ce cas, nous allons utiliser la fonction de répartition expérimentale des paires $(\mathbf{x}_i, y_i)$, i=1, 2, ..., n. Un rééchantillon $Z^* = \{\mathbf{z}^{*1}, \mathbf{z}^{*2}, \ldots, \mathbf{z}^{*n}\}$, qui par définition est un échantillon de taille n provenant de $\hat{P}$, sera obtenu en choisissant avec remise des paires $(\mathbf{x}_i, y_i) = \mathbf{z}_i$ parmi les observations originales $Z = \{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_n\}$. Il s'agit du rééchantillonnage des paires. Notez que nous n'avons pas supposé la linéarité du modèle, et que les matrices X^* ne sont pas égales à X.

Algorithme 2.3 (estimation de la variance de $\hat{\beta}_{MC} = s(X, \mathbf{y})$ à l'aide du rééchantillonnage des paires).

1. *Générer B rééchantillons $Z^{*1}, Z^{*2}, \ldots, Z^{*B}$ en échantillonnant avec remise n données à partir de $Z = \{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_n\} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_n, y_n)\}$;*
2. *Calculer $\hat{\beta}^*_{MC}(b) = (X^{*b'}X^{*b})^{-1}X^{*b'}\mathbf{y}^{*b}$, $b = 1, 2, \ldots, B$;*
3. *Estimer $Cov_G(\hat{\beta}_{MC})$ à l'aide de (2.18).*

À l'instar de l'algorithme 2.2, les 2^e et 3^e étapes de l'algorithme 2.3 peuvent être modifiées pour étudier un autre estimateur que $\hat{\beta}_{MC}$, ou pour estimer le biais ou l'erreur quadratique moyenne de l'estimateur choisi.

2.4.3. Comparaison des méthodes

Les deux méthodes de rééchantillonnage utilisées en régression offrent toutes deux des avantages et inconvénients qui doivent être soupesés pour chaque problème. N'oublions pas que ces méthodes sont particulièrement utiles lorsque l'estimateur $\hat{\beta}_{MC}$ et les estimés de covariance (2.15) et (2.17) ne peuvent être utilisés avec confiance.

Puisque le rééchantillonnage des paires ne suppose aucun modèle paramétrique, il ne dépend pas d'hypothèses sur la forme de celui-ci. En comparaison, le rééchantillonnage des résidus suppose un modèle, qui est utilisé pour calculer $\hat{\beta}$, en plus de supposer que l'erreur entre y_i et la moyenne $\mu_i = \mathbf{x}_i'\beta$ ne dépend pas de $\mathbf{x}_i$ et est en fait i.i.d. pour tout i. Il arrive souvent que ces hypothèses ne sont pas satisfaites en pratique.

2.5. Estimateurs adaptatifs bootstrap

2.5.1. Cas général

Soit $\hat{\theta}_\lambda$ une famille d'estimateurs du paramètre θ indexée par le paramètre λ. Nous allons dans ce qui suit utiliser les données Z afin de choisir la valeur $\hat{\lambda}$ qui optimise la performance de $\hat{\theta}_\lambda$ et ainsi définir l'estimateur $\tilde{\theta}_A$. Puisque les données sont utilisées pour déterminer l'estimateur approprié, nous appelons cette procédure de l'estimation adaptative. Parmi les estimateurs adaptatifs étudiés,

citons la moyenne tronquée avec proportion de troncature λ (Léger & Romano, 1990a), et l'estimation de fonction de densité à l'aide de la méthode du noyau avec fenêtre λ (Léger & Romano, 1990b).

Soit

$$\lambda_{opt} = \underset{\lambda}{\operatorname{argmin}} \mathbb{E}_P \left[g(\hat{\theta}_\lambda, \theta) \right] \tag{2.24}$$

la valeur de λ qui optimise la performance de $\hat{\theta}_\lambda$ où P est le modèle de probabilité, et $\mathbb{E}_P \left[g(\hat{\theta}_\lambda, \theta) \right]$ est une mesure de performance de $\hat{\theta}_\lambda$ qui peut représenter la variance, le biais, l'erreur quadratique moyenne, l'erreur de prévision, etc.

Le modèle P étant inconnu, nous devons utiliser des méthodes d'estimation. Le rééchantillonnage (par exemple les estimateurs (2.5), (2.10) et (2.11)) sont habituellement utilisés à cette fin. L'estimateur par rééchantillonnage de $\mathbb{E}_P \left[g(\hat{\theta}_\lambda, \theta) \right]$ est $\mathbb{E}_{\hat{P}} \left[g(\hat{\theta}^*_\lambda, \hat{\theta}) \right]$; nous avons discuté entre autres les exemples de la variance, du biais et de l'erreur quadratique moyenne dans les contextes d'un échantillon i.i.d. et d'une régression linéaire. L'estimateur par rééchantillonnage de λ_{opt} (2.24) est donc

$$\hat{\lambda} = \underset{\lambda}{\operatorname{argmin}} \mathbb{E}_{\hat{P}} \left[g(\hat{\theta}^*_\lambda, \hat{\theta}) \right] \tag{2.25}$$

Bien que la valeur de $\hat{\lambda}$ soit appelée à changer pour chaque échantillon, l'estimateur adaptatif $\tilde{\theta}_A = \hat{\theta}_{\hat{\lambda}}$ où $\hat{\lambda}$ est donné par (2.25) est bel et bien un estimateur en soi. Nous pourrons ainsi nous intéresser à sa distribution. Puisque rien ne nous garantit par exemple que $Var_P(\tilde{\theta}_A)$ est égale à $Var_P(\hat{\theta}_{\lambda_{opt}})$, nous devrons donc faire une simulation afin de calculer une approximation pour $Var_P(\tilde{\theta}_A)$.

2.5.2. Moindres carrés tronqués adaptatif

La famille d'estimateurs par moindres carrés tronqués $\hat{\beta}_h$ (1.6) est indexée par le paramètre h. Comme il a été question à la page 13, l'erreur quadratique moyenne représente un bon critère de la performance de $\hat{\beta}_{h,k}$ pour $k \in \{0, 1, \ldots, p-1\}$ fixe. Puisque $\hat{\beta}_h$ est un vecteur, il faut choisir un critère qui tienne compte de l'erreur quadratique moyenne de $\hat{\beta}_{h,0}$, de $\hat{\beta}_{h,1}, \ldots$, ainsi que de $\hat{\beta}_{h,p-1}$;

notons ce critère $\mathbb{E}_P\left[g(\hat{\beta}_h, \beta)\right]$. Le critère peut prendre la forme de $\mathbb{E}_P\left[g(\hat{\beta}_h, \beta)\right] = \sum_{i=0}^{p-1}\left(biais(\hat{\beta}_{h,i})\right)^2$, par exemple.

Le but de ce travail est d'étudier l'estimateur adaptatif

$$\tilde{\beta}_A = \hat{\beta}_{\hat{h}}, \tag{2.26}$$

où

$$\hat{h} = \underset{h}{\operatorname{argmin}}\, \mathbb{E}_{\hat{P}}\left[g(\hat{\beta}_h^*, \hat{\beta})\right] \tag{2.27}$$

est l'estimateur par rééchantillonnage de

$$h_{opt} = \underset{h}{\operatorname{argmin}}\, \mathbb{E}_P\left[g(\hat{\beta}_h, \beta)\right]. \tag{2.28}$$

Comme il a été discuté au premier chapitre, l'estimateur $\tilde{\beta}_A$ devra être robuste aux données aberrantes, mais aussi moins variable que $\hat{\beta}_{max}$ (1.7) ; $\mathbb{E}_P\left[g(\hat{\beta}_h, \beta)\right]$ devra donc inclure le biais et la variance du vecteur $\hat{\beta}_h$.

Chapitre 3

ESTIMATEUR DES MOINDRES CARRÉS TRONQUÉS ADAPTATIF PAR RÉÉCHANTILLONNAGE

3.1. INTRODUCTION

Nous nous intéressons dans ce travail à la régression linéaire simple dans des contextes où certaines données sont aberrantes. Les estimateurs $\tilde{\beta}$ habituellement utilisés tels que les moindres carrés tronqués $\hat{\beta}_{max}$ (1.7) et la moindre médiane des carrés (Rousseeuw, 1984), sont très résistants aux données aberrantes, mais sont aussi très variables (voir la figure 1.4 à la page 10 où l'on peut comparer visuellement $\hat{\beta}_{max}=\hat{\beta}_{26}$ à $\hat{\beta}_{MC}=\hat{\beta}_{50}$.) Comme il a déjà été discuté au premier chapitre, l'estimateur que nous cherchons devrait posséder les erreurs quadratiques moyennes pour les estimateurs des paramètres les plus faibles possible.

Nous tenterons d'estimer par rééchantillonnage le paramètre h de l'estimateur par moindres carrés tronqués $\hat{\beta}_h$ (1.6) correspondant à l'estimateur ayant les erreurs quadratiques moyennes les plus faibles ; et ainsi définir un estimateur adaptatif $\tilde{\beta}_A$. Cet estimateur pourrait résister à une grande proportion de données aberrantes, mais serait moins variable que l'estimateur à robustesse maximale $\hat{\beta}_{max}$. En effet, il permettrait de tronquer moins de 50% des données lors du calcul de l'estimateur dans les cas où moins de la moitié des données sont aberrantes.

3.2. Modèles à l'étude

Contrairement à la plupart des articles traitant de régression linéaire robuste, les données aberrantes des modèles que nous étudions à l'aide de simulations ne sont pas simplement dues au terme d'erreur qui ne suit pas une normale de variance constante. Les données aberrantes proviennent plutôt d'un modèle de probabilité complètement différent. Les modèles qui nous intéressent sont les suivants :

$$\begin{cases} X_i = V_{1,i} \\ Y_i = V_{2,i} \end{cases} \qquad i = 1, 2, \ldots, n_a \tag{3.1}$$

$$\begin{cases} X_i = V_{3,i} \\ Y_i = \beta_0 + \beta_1 X_i + \epsilon_i \end{cases} \qquad i = n_a{+}1, \ldots, n \tag{3.2}$$

et peuvent être combinés à l'aide de l'indicatrice I qui sera discutée à la section suivante :

$$\begin{cases} X_i = V_{1,i} I_i + V_{3,i}(1 - I_i) \\ Y_i = V_{2,i} I_i + (\beta_0 + \beta_1 X_i + \epsilon_i)(1 - I_i) \end{cases} \qquad i = 1, 2, \ldots, n \tag{3.3}$$

où $V_1 \sim G_1$, $V_2 \sim G_2$, $V_3 \sim G_3$, $\epsilon \sim G_4$, $I \sim G_5$. L'équation (3.1) est celle du modèle de probabilité des n_a données aberrantes, alors que le modèle linéaire (3.2) représente les $n - n_a$ autres données. Notez que ϵ ne représente que le terme d'erreur de la partie linéaire. Nous nous intéressons à l'estimation de (β_0, β_1), soient les coefficients de la relation linéaire. Les figures 1.1, 1.2 et 1.3 du chapitre 1 illustrent des exemples de jeux de données obtenus du modèle (3.3).

3.2.1. Le nombre de données aberrantes est-il fixe ou aléatoire ?

Supposons que nous observons un jeu de données avec n_a données aberrantes ($I_i = 1$) parmi les n paires (x_i, y_i).

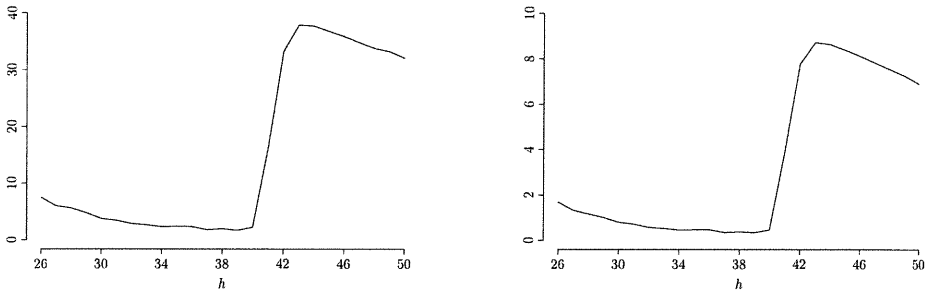
Nous pouvons supposer que le jeu de données observé ne représente qu'une réalisation du phénomène modélisé par (3.3) et que si on recueillait d'autres données à partir de ce même modèle, n_a pourrait très bien être différent. Nous considérons ainsi que I suit une distribution aléatoire. De façon plus précise, si les indicatrices

I_i sont distribuées selon des lois de Bernoulli indépendantes avec probabilité de succès p, alors $n_a = \sum_{i=1}^{n} I_i$ est distribué selon une loi binomiale de paramètres (n, p).

Mais nous pouvons aussi supposer que le nombre de données aberrantes n_a est une constante. Nous ne considérons donc pas tous les jeux de données que nous aurions pu obtenir, mais seulement ceux qui comprennent le même nombre de données aberrantes que celui que nous avons observé. C'est donc dire que n_a données proviennent du modèle (3.1) et les $n-n_a$ restantes du modèle (3.2). Nous considérons sans perte de généralité pour les cas où n_a est fixe ou aléatoire que $I_i = 1$ pour $i = 1, 2, \ldots, n_a$ et $I_i = 0$ pour $i = n_a+1, \ldots, n$.

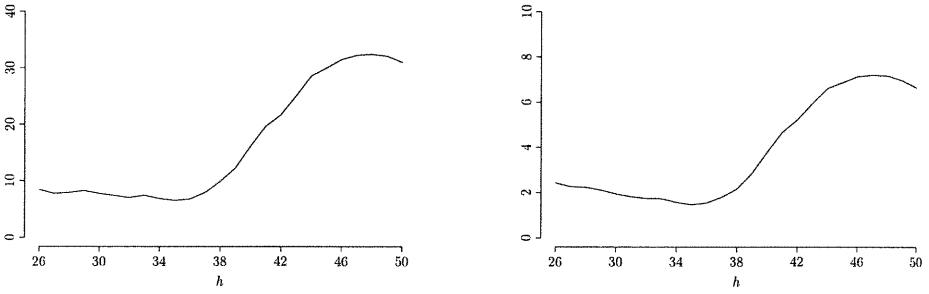
Notez que si nous supposons n_a aléatoire, nous devrons être plus prudents lors de l'inférence puisqu'une autre réalisation de (3.3) pourrait inclure plus de données aberrantes.

La simulation 2 du chapitre 1 est basée sur le modèle (3.3) où $\beta = (1, 2)$, $V_1 \sim N(6,\ 0{,}25)$, $V_2 \sim N(-1,\ 0{,}25)$, $V_3 \sim U(0{,}5,\ 4{,}5)$, $\epsilon \sim N(0, 4)$, $n_a = 10$ (fixe) et $n = 50$ (voir le tableau 4.1 à la page 51). Les erreurs quadratiques moyennes illustrées à la figure 1.8 (page 14) sont reproduites ci-dessous à la figure 3.1. Elle nous montre que les erreurs quadratiques moyennes diminuent entre $h = max = 26$ et $h = 40$ pour augmenter abruptement à $h = 41$, alors qu'au moins une donnée aberrante doit être incluse dans le calcul de $\hat{\beta}_h$. La simulation 2a est pour sa part basée sur le modèle (3.3) où $n_a \sim Bin(50,\ 0{,}2)$, alors que toutes les autres distributions et paramètres sont identiques à ceux de la simulation 2. Notez que $\mathbb{E}[n_a] = 10$ ici aussi. Dans ce cas, les erreurs quadratiques moyennes (voir figure 3.2 ci-dessous) atteignent plutôt leurs minimums aux alentours de $h = 35$ et augmentent beaucoup moins rapidement après ce point. Les modèles des simulations 2 et 2a sont donc différents à un point tel que l'estimateur qui semble optimal est différent dans les deux cas. Nous devrons donc en tenir compte lors du rééchantillonnage.



(a) $EQM(\hat{\beta}_{h,0})$ (b) $EQM(\hat{\beta}_{h,1})$

FIGURE 3.1. EQM pour la simulation 2 ($n_a = 10$)



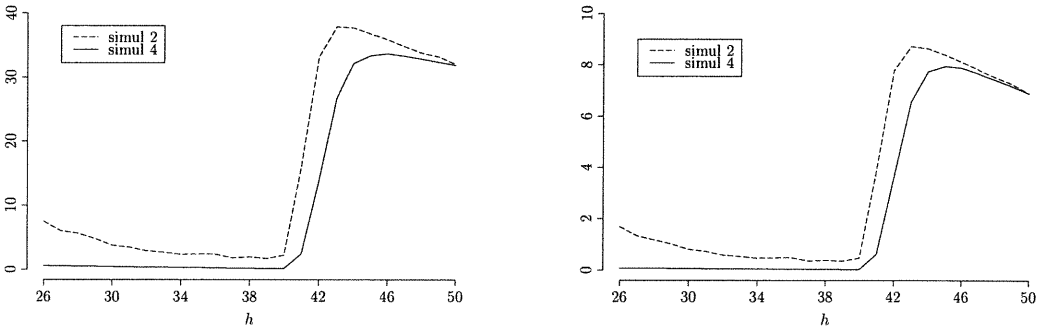
(a) $EQM(\hat{\beta}_{h,0})$ (b) $EQM(\hat{\beta}_{h,1})$

FIGURE 3.2. EQM pour la simulation 2a ($n_a \sim Bin(50,\ 0{,}2)$)

3.2.2. Exemples de modèles

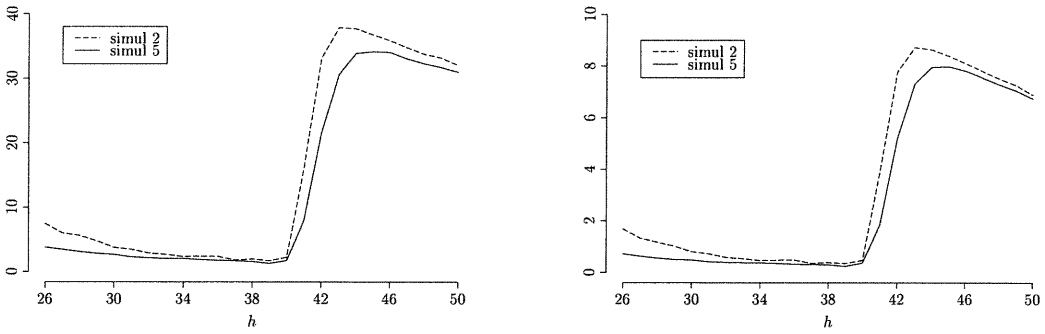
Les modèles (3.1) et (3.2) et *a fortiori* le modèle combiné (3.3) sont très généraux. En plus du nombre de valeurs aberrantes n_a qui peut être fixe ou aléatoire, les lois et paramètres de V_1 à V_3 et de ϵ peuvent varier. Nous avons déjà examiné trois simulations où n_a=0, n_a=10 ou n_a=15 est fixe ; voir figures 1.7, 1.8 (reproduite à la figure 3.1) et 1.9 à la page 14. Voici d'autres modèles pour lesquels n_a=10 est fixe et où un seul paramètre diffère de ceux utilisés à la simulation 2 (voir le tableau 4.1 à la page 51 pour les descriptions complètes.) Nous nous intéressons ici à l'influence sur les erreurs quadratiques moyennes de la modification de quelques paramètres du modèle (3.3).

De façon générale, si la variance de ϵ diminue, les erreurs quadratiques moyennes des différents estimateurs seront plus petites (voir la simulation 4 à la figure 3.3 où $\epsilon \sim N(0,1)$) ; il en sera de même si les données aberrantes sont plus variables (voir la simulation 5 à la figure 3.4 où $V_1 \sim N(6,\ 1)$ et $V_2 \sim N(-1,\ 1)$.) C'est donc dire que les estimateurs par moindres carrés tronqués, et en particulier les plus robustes, sont plus performants pour ces modèles.



(a) $EQM(\hat{\beta}_{h,0})$ (b) $EQM(\hat{\beta}_{h,1})$

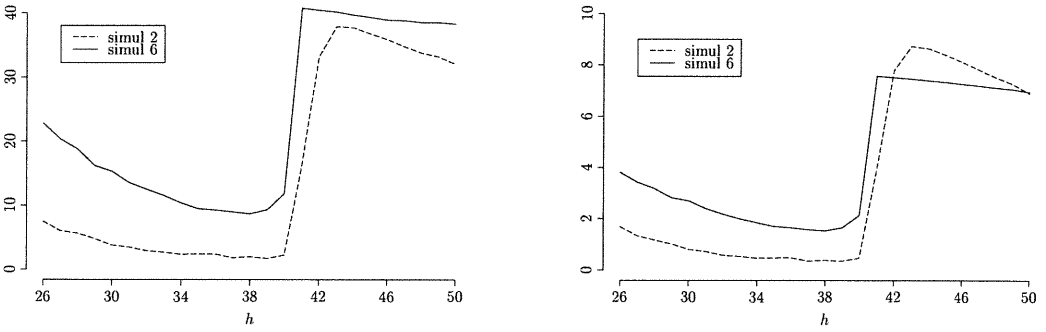
FIGURE 3.3. EQM pour la simulation 4 ($\epsilon \sim N(0,1)$)



(a) $EQM(\hat{\beta}_{h,0})$ (b) $EQM(\hat{\beta}_{h,1})$

FIGURE 3.4. EQM pour la simulation 5 ($V_1 \sim N(6,\ 1)$ et $V_2 \sim N(-1,\ 1)$)

À la simulation 6, nous avons déplacé le centre des données aberrantes de $(6, -1)$ vers $(10, -1)$. Ceci a eu pour effet d'augmenter l'erreur quadratique moyenne de la plupart des estimateurs, ainsi que de creuser l'écart entre les estimateurs les plus robustes et les autres estimateurs non biaisés (voir figure 3.5.)



(a) $EQM(\hat{\beta}_{h,0})$ (b) $EQM(\hat{\beta}_{h,1})$

FIGURE 3.5. EQM pour la simulation 6 ($V_1 \sim N(10,\ 0{,}25)$ et $V_2 \sim N(-1,\ 0{,}25)$)

3.3. SOLUTIONS DE LA LITTÉRATURE

Comme il a déjà été mentionné, la plupart des articles traitant de régression linéaire robuste se concentrent sur des données aberrantes provenant du terme

d'erreur ϵ du modèle (3.2) lors de leurs simulations. Ces modèles peuvent être exprimés à l'aide des modèles (3.1) et (3.2), où les distributions de V_1 et de V_3 doivent être la même, alors que celle de V_2 doit prendre la forme $\beta_0 + \beta_1 X_i + \delta_i$ où les δ_i possèdent une distribution différente de celle des ϵ_i :

$$\begin{cases} X_i = V_{1,i} \\ Y_i = \beta_0 + \beta_1 X_i + \delta_i \end{cases} \quad i = 1, 2, \ldots, n_a \tag{3.4}$$

$$\begin{cases} X_i = V_{1,i} \\ Y_i = \beta_0 + \beta_1 X_i + \epsilon_i \end{cases} \quad i = n_a{+}1, \ldots, n \tag{3.5}$$

où $V_1 \sim G_1$, $\epsilon \sim G_4$ et $\delta \sim G_6$.

Atkinson & Cheng (1999) ont utilisé la technique de "forward search" (voir Atkinson & Riani, 2000) pour choisir le paramètre h de l'estimateur par moindres carrés tronqués. Les modèles considérés sont de la forme (3.4) et (3.5) où $V_1 \sim U(0, 10)$, $\epsilon \sim N(0, 1)$, $\delta \sim N(12, 1)$ et n_a est fixe. Remarquez que la moyenne, et non pas la variance des erreurs change d'un groupe à l'autre.

Le "forward search" débute avec un petit ensemble de données auquel on ajoute (habituellement) une donnée à chaque étape de l'algorithme jusqu'à ce que toutes les données aient été incluses. À chaque étape, différentes statistiques et tests sont calculés. Le fonctionnement de la méthode repose sur le fait que l'introduction d'une donnée influente est signalée par des changements abrupts dans les statistiques calculées à chaque étape (Atkinson & Riani, 2000, p.32). Le pourcentage de troncature "idéal" est donc celui de la dernière étape avant qu'il y ait un "grand" changement dans les statistiques. Mais aucune indication n'est donnée pour discerner un changement significatif d'un changement dû au hasard. L'intuition et l'expérience sont donc nécessaires au bon fonctionnement de cette méthode.

Nous ne pourrons donc pas comparer notre estimateur adaptatif à l'estimateur par moindres carrés tronqués obtenu par "forward search" puisque ce dernier nécessiterait une intervention subjective lors de chaque simulation, alors que nous connaissons la valeur cherchée, et que la technique peut difficilement être automatisée.

Nous avons néanmoins étudié le modèle utilisé par Atkinson & Cheng (1999) aux simulations 7 et 7a.

3.4. Notre solution

Soit $\tilde{\beta}_A$=$\hat{\beta}_{\hat{h}}$ (2.26), où $\hat{h}$=$argmin_h \mathbb{E}_{\hat{P}} \left[g(\hat{\beta}^*_h, \hat{\beta}) \right]$ (2.27) estime h_{opt}=$argmin_h \mathbb{E}_P \left[g(\hat{\beta}_h, \beta) \right]$ (2.28), l'estimateur adaptatif par moindres carrés tronqués de β du modèle (3.2) ou (3.3). Comme il a déjà été argumenté au chapitre 1, nous désirons un estimateur tel que les erreurs quadratiques moyennes de ses composantes soient les plus petites possible ; nous devrons donc définir un critère de performance $\mathbb{E}_P \left[g(\hat{\beta}_h, \beta) \right]$ qui tienne compte de ces dernières. Nous devrons aussi nous interroger sur la façon d'adapter le rééchantillonnage à de telles situations.

3.4.1. Critères

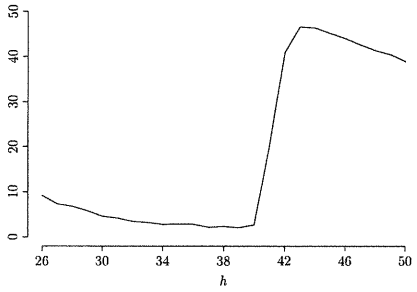
Puisque les différents estimateurs $\hat{\beta}_h$, où $h = max, \ldots, n$ sont des vecteurs (de dimension 2 dans le problème à l'étude), nous devons définir un ou des critères qui prennent en considération les erreurs quadratiques moyennes de $\hat{\beta}_{h,0}$ et de $\hat{\beta}_{h,1}$. Lors de la phase d'expérimentation, nous avons appliqué à la matrice d'erreur quadratique moyenne généralisée des quantités habituellement utilisées pour résumer les matrices de variance covariance :

- $EQM(\hat{\beta}_{h,0})$;
- $EQM(\hat{\beta}_{h,1})$;
- $EQM(\hat{\beta}_{h,0}) + EQM(\hat{\beta}_{h,1})$, soit la trace de la matrice d'EQM ; (3.6)
- $EQM(\hat{\beta}_{h,0}) * EQM(\hat{\beta}_{h,1}) - \left(EQM(\hat{\beta}_{h,0}, \hat{\beta}_{h,1}) \right)^2$, soit le déterminant de la matrice d'EQM ; (3.7)

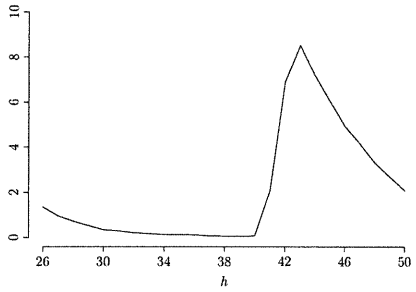
$$\bullet \quad EQM(\hat{y}_h(x)) = EQM(\hat{\beta}_{h,0} + \hat{\beta}_{h,1}x) \\ = EQM(\hat{\beta}_{h,0}) + 2xEQM(\hat{\beta}_{h,0}, \hat{\beta}_{h,1}) \\ +x^2 EQM(\hat{\beta}_{h,1}), \tag{3.8}$$

soit l'EQM de l'espérance au point x.

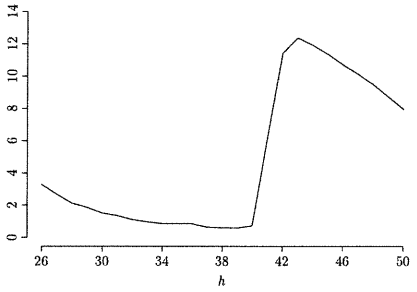
Les valeurs prises par les critères (3.6), (3.7) ainsi que le critère (3.8) évalué au point $x=\mu_x$ pour les simulations 2 et 2a, alors que *Nsimul*=500, sont illustrées aux figures 3.6 et 3.7.



(a) critère (3.6)

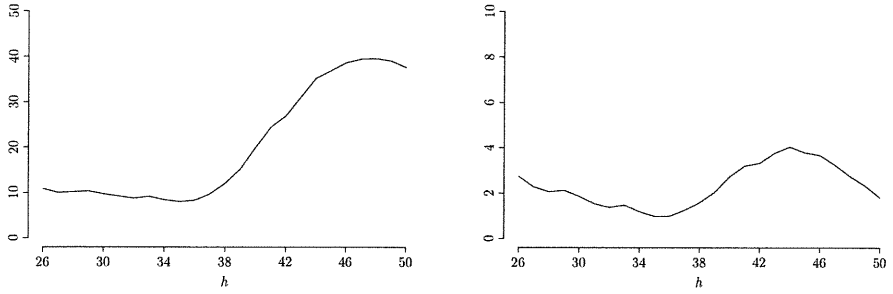


(b) critère (3.7)



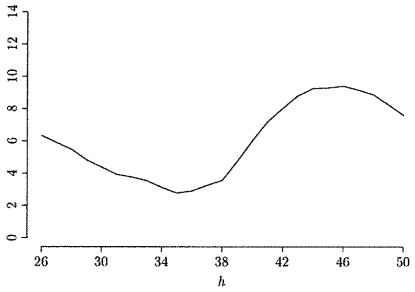
(c) critère (3.8) avec $x=\mu_x$

FIGURE 3.6. Critères pour la simulation 2



(a) critère (3.6)

(b) critère (3.7)



(c) critère (3.8) avec $x{=}\mu_x$

FIGURE 3.7. Critères pour la simulation 2a

Comme pour les simulations 2 et 2a ci-dessus, les minima des différents critères sont habituellement atteints pour des valeurs de h semblables pour les modèles étudiés ; mais ceci n'est pas toujours le cas.

Notez que les équivariances de régression et d'échelle des moindres carrés tronqués (Rousseeuw & Leroy, 1987, p.132) font que ces critères sont multipliés par c^2 ou c^4 si $cy + \mathbf{xv}$ (où c est une constante et $\mathbf{v}$ un vecteur colonne) est utilisé à la place de y ; les minima sont donc inchangés. La propriété d'équivariance affine (*ibid.*) fait que les biais, variances et erreurs quadratiques moyennes de $\hat{\beta}_h$ se trouvent modifiés si on remplace $\mathbf{x}$ par $\mathbf{x}'A$, pour une matrice non singulière A de dimension $p \times p$. Mais $\hat{y} = \mathbf{x}'\hat{\beta}$ est inchangé. Le critère (3.8) sera donc inchangé, alors que (3.6) et (3.7) seront modifiés d'une façon qui ne conserve pas nécessairement les minima. Ces deux critères dépendront donc de l'échelle de $\mathbf{x}$. Puisqu'il est invariant aux différentes transformations pour toutes les valeurs de

x, nous nous concentrerons dans la suite sur le critère (3.8), soit l'erreur quadratique moyenne de l'estimation de y que nous évaluerons au point μ_x, et que nous estimerons par rééchantillonnage au point $\bar{x}$:

$$\begin{aligned} EQM(\hat{y}_h(x)) = & \frac{1}{Nsimul} \sum_{i=1}^{Nsimul} (\hat{\beta}_{h,0}(i) - \beta_0)^2 \\ & + 2x \frac{1}{Nsimul} \sum_{i=1}^{Nsimul} (\hat{\beta}_{h,0}(i) - \beta_0)(\hat{\beta}_{h,1}(i) - \beta_1) \\ & + x^2 \frac{1}{Nsimul} \sum_{i=1}^{Nsimul} (\hat{\beta}_{h,1}(i) - \beta_1)^2, \end{aligned} \tag{3.9}$$

où h=$max, \ldots, n$, $\left(\hat{\beta}_{h,0}(i), \hat{\beta}_{h,1}(i)\right)'$ est l'estimateur par moindres carrés tronqués avec paramètre h de $(\beta_0, \beta_1)'$ pour le i^e jeu de données, et *Nsimul* est le nombre de jeux de données simulés.

3.4.2. Rééchantillonnage appliqué au problème présent

Les méthodes de rééchantillonnage nécessitent l'estimation de P, soit le modèle de probabilité à l'origine des données. Ce modèle est ici donné par (3.1) et (3.2) ou par (3.3), et comporte 2 paramètres inconnus (β_0, β_1), qui sont les paramètres d'intérêt, en plus des 5 fonctions de répartition (G_1 à G_5.)

Si le nombre de valeurs aberrantes n_a est aléatoire, nous avons bel et bien que les n données à notre disposition sont indépendantes et identiquement distribuées selon le modèle commun (3.3). Par contre, si n_a est considéré fixe, les données proviennent de deux modèles distincts. Nous devrons donc tenter de reproduire cela dans notre façon de rééchantillonner.

Puisque le rééchantillonnage des paires dans un contexte de régression ne fait que supposer que les données sont générées de façon indépendante et qu'elles sont identiquement distribuées selon P, nous utiliserons cette méthode si n_a est considéré aléatoire. Le rééchantillonnage par les résidus n'est pas possible, du moins dans sa forme habituelle, puisque y ne s'exprime pas sous la forme d'une moyenne plus un terme d'erreur homoscédastique (Davidson & Hinkley, 1997, p.326). Intuitivement, si nous utilisions un estimateur $\hat{\beta}$ robuste pour calculer les résidus $\hat{\epsilon}$

et que nous rééchantillonnions les résidus centrés (voir algorithme 2.2, page 23), les grands résidus provenant des valeurs aberrantes pourraient être additionnés aux $X\hat{\beta}$ correspondant aux données non aberrantes. Par exemple, si les supports de V_1 et V_3 diffèrent, ceci aurait pour effet de générer des valeurs aberrantes où elles n'auraient pas pu être générées à partir du modèle de probabilité P.

Si n_a est considéré fixe, nous devrons tenter de séparer les deux groupes, rééchantillonner de façon distincte dans chacun des deux groupes, et mettre le tout en commun.

Soient

$$r_i = y_i - (\hat{\beta}_{max,0} + x_i\hat{\beta}_{max,1}), \quad i = 1, 2, \ldots, n$$

le résidu au point i, où $\hat{\beta}_{max}$ est l'estimé de β par moindres carrés tronqués possédant la robustesse maximale (voir proposition 1.2, p.7) et $\hat{\sigma}$ un estimé robuste d'écart type du terme d'erreur (voir la section 4.2.2 à la page 55 pour la description du $\hat{\sigma}$ de la fonction ltsreg de S-Plus 6.0.)

Nous considérons aberrant le point i si $|r_i|/\hat{\sigma} > 2{,}5$; il s'agit du critère utilisé par Rousseeuw et Leroy dans le programme de régression robuste PROGRESS (voir Rousseeuw & Leroy, 1987, p.238). Ce critère nous permettra donc de former deux groupes et de générer un rééchantillon de taille $\hat{n}_a$ à partir des $\hat{n}_a$ paires telles que $|r_i|/\hat{\sigma} > 2{,}5$, et un rééchantillon de taille $n-\hat{n}_a$ à partir des autres couples. Dans le cas du groupe aberrant, le rééchantillonage par les paires devra être fait, alors que le rééchantillonnage par les résidus ou les paires pourrait être fait dans le second groupe. Nous avons néanmoins retenu le rééchantillonnage des paires dans les deux groupes. Un rééchantillon de taille n sera ainsi formé en unissant les deux rééchantillons précédents. Après avoir estimé l'appartenance des paires aux deux modèles de probabilité, cette façon de rééchantillonner estime donc par rééchantillonnage des paires les modèles de probabilité (3.1) et (3.2) séparément.

Notez que ces façons de rééchantillonner sont générales et peuvent être appliquées à des données provenant de plusieurs modèles différents, peu importe les formes prises par ceux-ci. Il faut simplement s'assurer dans le cas où on considère

n_a fixe que la séparation des données entre aberrantes et non aberrantes est efficace ; alors que le rééchantillonnage des paires lorsque n_a est considéré aléatoire ne pose pas de tel problème.

3.4.3. Algorithme

L'estimation de l'erreur quadratique moyenne de l'espérance au point μ_x pour h fixe à l'aide du rééchantillonnage des paires est basée sur l'algorithme 2.3 (page 27). La façon de générer les B rééchantillons de la première étape a été discutée ci-dessus ; l'estimateur de la 2^e étape est l'estimateur par moindres carrés tronqués avec paramètre h que nous notons $\hat{\beta}_h^*(b)$ pour le b^e rééchantillon ; et la fonction estimée à la 3^e étape est

$$\begin{aligned}
\widehat{EQM}_B(\hat{y}_h^*(\bar{x})) &= \widehat{EQM}_B(\hat{\beta}_{h,0}^*) + 2\bar{x}\widehat{EQM}_B(\hat{\beta}_{h,0}^*, \hat{\beta}_{h,1}^*) \\
&\quad + \bar{x}^2\widehat{EQM}_B(\hat{\beta}_{h,1}^*) \\
&= \frac{1}{B}\sum_{b=1}^{B}(\hat{\beta}_{h,0}^*(b) - \hat{\beta}_0)^2 \\
&\quad + 2\bar{x}\frac{1}{B}\sum_{b=1}^{B}(\hat{\beta}_{h,0}^*(b) - \hat{\beta}_0)(\hat{\beta}_{h,1}^*(b) - \hat{\beta}_1) \\
&\quad + \bar{x}^2\frac{1}{B}\sum_{b=1}^{B}(\hat{\beta}_{h,1}^*(b) - \hat{\beta}_1)^2, \qquad (3.10)
\end{aligned}$$

où $\hat{\beta} = \hat{\beta}_{max}$ (1.7) est l'estimateur par moindres carrés tronqués avec robustesse maximale calculé sur l'échantillon original. Remarquez que l'estimation par rééchantillonnage de l'erreur quadratique moyenne de l'espérance au point $x = \bar{x} = \mathbb{E}_{\hat{F}}[X^*]$ est utilisée pour estimer le critère original d'erreur quadratique moyenne de l'espérance au point $x = \mu_x = \mathbb{E}_F[X]$.

Comme il a été discuté à l'exemple 2.4 de la page 25, utiliser $\hat{\beta} = \hat{\beta}_h$, que nous savons biaisé entre autres lorsque h est supérieur au nombre de données aberrantes, produirait des estimés par rééchantillonnage du biais et de l'erreur quadratique moyenne de $\hat{\beta}_h$ faussés. Par exemple, l'estimateur par rééchantillonnage du biais des moindres carrés $\frac{1}{B}\sum_{b=1}^{B}\hat{\beta}_{h=n}^*(b) - \hat{\beta}_{h=n}$ sera nul même si le

modèle contient des valeurs influentes, puisque $\hat{\beta}_{h=n}$ est lui-même biaisé. L'estimateur $\hat{\beta}_{max}$ étant de robustesse maximale, nous l'utiliserons dans (3.10) pour toutes les valeurs de h.

Algorithme 3.1 (estimation de l'EQM de l'espérance au point μ_x pour définir l'estimateur adaptatif $\tilde{\beta}_A$ (2.26) à l'aide du rééchantillonnage des paires).

0. Calculer $\hat{\beta}_{max}$ sur le jeu de données original Z ;

1. **Si $\mathbf{n_a}$ est aléatoire**, *générer B rééchantillons $Z^{*1}, Z^{*2}, \ldots, Z^{*B}$ en échantillonnant avec remise n données à partir de $Z = \{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_n\} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_n, y_n)\}$;*

1'. **Si $\mathbf{n_a}$ est fixe**

(a) Calculer $r_i = y_i - (\hat{\beta}_{max,0} + x_i \hat{\beta}_{max,1})$ $i = 1, 2, \ldots, n$;

(b) Calculer $\hat{\sigma}$, un estimé robuste d'écart type du terme d'erreur ;

(c) Calculer

$$\hat{I}_i = \begin{cases} 1 & si \ \frac{|r_i|}{\hat{\sigma}} > 2{,}5 \\ 0 & sinon \end{cases}$$

et $\hat{n}_a = \sum_{i=1}^{n} \hat{I}_i$;

(d) Générer B rééchantillons $Z^{*1}, Z^{*2}, \ldots, Z^{*B}$ de taille n composés

- *d'un rééchantillon de taille $\hat{n}_a$ obtenu en échantillonnant avec remise parmi les $\hat{n}_a$ $\mathbf{z}_i$ tels que $\hat{I}_i = 1$;*
- *et d'un rééchantillon de taille $n - \hat{n}_a$ obtenu en échantillonnant avec remise parmi les $n - \hat{n}_a$ $\mathbf{z}_i$ tels que $\hat{I}_i = 0$;*

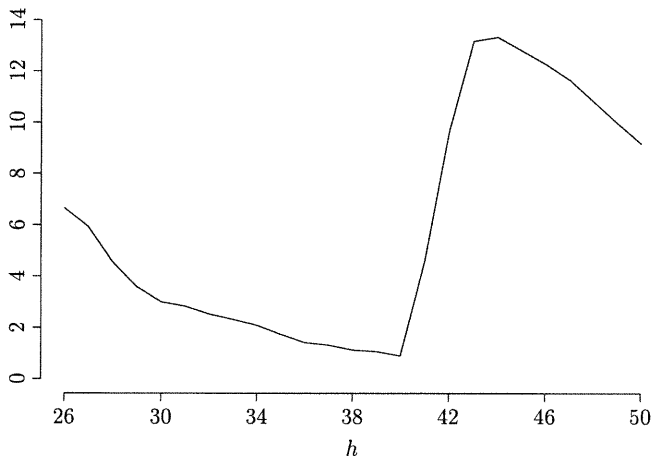
2. Calculer l'estimateur par moindres carrés tronqués $\hat{\beta}^*_h(b)$ pour $h = max, \ldots, n$ et $b = 1, 2, \ldots, B$;

3. Estimer $EQM(\hat{y}_h(\mu_x))$ à l'aide de $\widehat{EQM}_B(\hat{y}^*_h(\bar{x}))$ (3.10) pour $h = max, \ldots, n$;

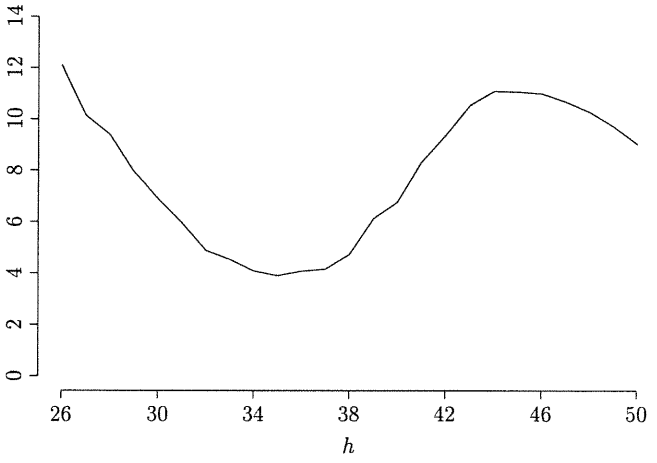
4. Définir $\tilde{\beta}_A = \hat{\beta}_{\hat{h}}$, où $\hat{h} = \text{argmin}_h \widehat{EQM}_B(\hat{y}^*_h(\bar{x}))$.

L'algorithme 3.1 a été utilisé pour estimer par rééchantillonnage la valeur de l'EQM de l'espérance au point μ_x pour un jeu de données semblable à celui de la figure 1.2 à la page 9, où $n_a = 10$ (simulation 2.) Le nombre B de rééchantillons utilisé fut de 200 et nous avons considéré n_a tour à tour fixe et aléatoire. À la

figure 3.8, les estimations par rééchantillonnage $\widehat{EQM}_B(\hat{y}_h^*(\bar{x}))$ (3.10) du critère $EQM(\hat{y}_h(\mu_x))$ (3.8) sont illustrés.



(a) $\widehat{EQM}_B(\hat{y}_h^*(\bar{x}))$ si $\mathbf{n_a}$ **considéré fixe**



(b) $\widehat{EQM}_B(\hat{y}_h^*(\bar{x}))$ si $\mathbf{n_a}$ **considéré aléatoire**

FIGURE 3.8. Estimations par rééchantillonnage de $EQM(\hat{y}_h(\mu_x))$ pour un jeu de données où $n_a = 10$

On remarque que pour ce jeu de données, les estimations par rééchantillonnage fonctionnent très bien. En particulier, les deux façons utilisées pour estimer le modèle de probabilité P semblent bien adaptées. Les valeurs prises par l'estimateur adaptatif pour le jeu de données à l'étude sont $\tilde{\beta}_A = \hat{\beta}_{40}$ si n_a est considéré fixe, et $\tilde{\beta}_A = \hat{\beta}_{35}$ si n_a est considéré aléatoire.

Nous sommes donc en mesure d'estimer par rééchantillonnage l'EQM de l'espérance au point μ_x pour $\hat{\beta}_h$, h=$max, \ldots, n$ afin de définir $\tilde{\beta}_A$ (2.26) pour un jeu de données Z provenant d'un modèle de probabilité P (3.3). Puisque nous nous intéressons au comportement de l'estimateur adaptatif $\tilde{\beta}_A$, nous devrons donc le calculer pour plusieurs jeux de données provenant du même modèle de probabilité P ; et nous devrons utiliser différents modèles de probabilité P. Tout comme nous avons utilisé des approximations par simulation pour le biais, la variance, l'erreur quadratique moyenne des estimateurs $\hat{\beta}_h$, ainsi que l'erreur quadratique moyenne de l'espérance au point μ_x, nous utiliserons ces approximations pour étudier l'estimateur $\tilde{\beta}_A$.

3.4.4. Point de rupture de l'estimateur adaptatif

Comme il a déjà été question au premier chapitre, le point de rupture d'un estimateur représente le pourcentage minimal de données aberrantes nécessaire pour créer un biais non borné à cet estimateur (voir définition 1.1 à la page 6.) La fraction $\frac{n-h+1}{n}$ représente le point de rupture des moindres carrés tronqués de paramètre h ($\frac{\left[\frac{n-p}{2}\right]+1}{n}$ pour l'estimateur à robustesse maximale lorsque h= $\left[\frac{n}{2}\right] + \left[\frac{p+1}{2}\right]$, et $1/n$ pour les moindres carrés.) Mais qu'en est-il de l'estimateur adaptatif $\tilde{\beta}_A$=$\hat{\beta}_{\hat{h}}$ où $\hat{h}$= argmin$_h$ $\widehat{EQM}_B(\hat{y}^*_h(\bar{x}))$.

Si pour tout jeu de données incluant une donnée aberrante arbitraire, le choix bootstrap idéal du paramètre h, argmin$_h$ $EQM_{\hat{P}}(\hat{y}^*_h(\bar{x})) \in \{max, \ldots, n-1\}$, alors le point de rupture de l'estimateur $\tilde{\beta}_A$ est supérieur à $1/n$ (i.e. $\epsilon_n(\tilde{\beta}_A) > 1/n$), puisqu'une seule donnée aberrante n'est pas suffisante pour briser l'estimateur choisi. De la même façon, si argmin$_h$ $EQM_{\hat{P}}(\hat{y}^*_h(\bar{x})) \in \{max, \ldots, n-2\}$ pour tout jeu de données incluant deux données aberrantes arbitraires, $\epsilon_n(\tilde{\beta}_A) > 2/n$, et ainsi de suite.

Notez que nous nous intéressons ici aux estimés idéaux par rééchantillonnage. Si le nombre de rééchantillons B est fixé, il est possible que le point de rupture de l'estimateur adaptatif soit inférieur au cas où $B \to \infty$; mais cette probabilité est quantifiable et très petite.

Puisque nous n'avons pas de formule explicite pour $\hat{\beta}_h$, nous allons étudier dans cette section un autre estimateur adaptatif. Cet estimateur devra choisir parmi deux estimateurs simples : la moyenne et la médiane d'un échantillon univarié. Le critère $\mathbb{E}_{\hat{P}}\left[g(\hat{\theta}^*_\lambda, \hat{\theta})\right]$ à minimiser (voir équation (2.25), page 28) sera l'erreur quadratique moyenne, où $\hat{\theta}$ représente la médiane de l'échantillon original. Ce choix a été fait pour imiter l'estimateur $\tilde{\beta}_A$ qui utilise $\hat{\theta}$=$\hat{\beta}_{max}$ pour tous les estimateurs par moindres carrés tronqués.

Soit l'estimateur

$$\tilde{Z} = \begin{cases} med(Z) & \text{si } \mathbb{E}_{\hat{P}}\left[(med(Z^*) - med(Z))^2\right] < \mathbb{E}_{\hat{P}}\left[(\bar{Z}^*) - med(Z))^2\right] \\ \bar{Z} & \text{sinon} \end{cases}$$

où Z est un échantillon univarié et $med(\cdot)$ représente la médiane. Notez que $\epsilon_n(\bar{Z})$=$1/n$ et $\epsilon_n(med(Z))$= $\left[\frac{n+1}{2}\right]/n$ sont les points de rupture des deux estimateurs composant $\tilde{Z}$.

Si pour tout jeu de données incluant une donnée aberrante arbitraire $\mathbb{E}_{\hat{P}}\left[(med(Z^*) - med(Z))^2\right] < \mathbb{E}_{\hat{P}}\left[(\bar{Z}^*) - med(Z))^2\right]$, alors $\tilde{Z}$=$med(Z)$ et $\epsilon_n(\tilde{Z}) > 1/n$ (puisque $\epsilon_n(med(Z)) > 1/n$) ; sinon $\epsilon_n(\tilde{Z}) \leq 1/n$.

Soit $Z = \left\{X_{(1)}, X_{(2)}, \ldots, X_{(i-1)}, X_{(i+1)}, \ldots, X_{(n)}, Y\right\}$ un jeu de données comprenant n valeurs distinctes où la statistique d'ordre $X_{(i)}$ a été remplacée par Y où $|Y| \to \infty$, n est impair et $\hat{P}$ la fonction de répartition expérimentale de l'échantillon Z.

Les estimés idéaux par rééchantillonnage prennent la forme suivante :

$$\begin{aligned}\mathbb{E}_{\hat{P}}\left[(med(Z^*)-med(Z))^2\right] &= \sum_{j=1}^{i-1} p_j(X_{(j)}-med(Z))^2 + \sum_{j=i+1}^{n} p_{j-1}(X_{(j)}-med(Z))^2 \\ &\quad + p_Y(Y-med(Z))^2 \\ &= Y^2 p_Y + O(Y) \\ &= Y^2\left[\sum_{k=0}^{(n-1)/2}\binom{n}{k}\left(\frac{n-1}{n}\right)^k\left(\frac{1}{n}\right)^{n-k}\right] + O(Y),\end{aligned}$$

où $p_j = \mathbb{P}(med(Z^*) = X_{(j)})$

$$= \sum_{k=0}^{(n-1)/2}\binom{n}{k}\left[\left(\frac{j-1}{n}\right)^k\left(\frac{n-j+1}{n}\right)^{n-k} - \left(\frac{j}{n}\right)^k\left(\frac{n-j}{n}\right)^{n-k}\right] \quad (3.11)$$

$$p_Y = \mathbb{P}(med(Z^*) = Y) = \sum_{k=0}^{(n-1)/2}\binom{n}{k}\left(\frac{n-1}{n}\right)^k\left(\frac{1}{n}\right)^{n-k}$$

(voir Efron & Tibshirani, 1993, p.16)

et $\mathbb{E}_{\hat{P}}\left[(med(Z^*)-med(Z))^2\right] - Y^2 p_Y = O(Y)$

puisque $\lim_{Y\to\infty} Y^{-1}\left(\mathbb{E}_{\hat{P}}\left[(med(Z^*)-med(Z))^2\right] - Y^2 p_Y\right) = a,$

où a est une constante,

$$\begin{aligned}\mathbb{E}_{\hat{P}}\left[\left(\bar{Z}^*-med(Z)\right)^2\right] &= Var_{\hat{P}}(\bar{Z}^*) + \left(\mathbb{E}_{\hat{P}}\left[\bar{Z}^*-med(Z)\right]\right)^2 \\ &= \frac{1}{n^2}\left(\sum_{\substack{j=1\\ j\neq i}}^{n}(X_{(j)}-\bar{Z})^2 + (Y-\bar{Z})^2\right) + \left(\bar{Z}-med(Z)\right)^2 \\ &= Y^2\left[\frac{1}{n^2}\left(2-\frac{1}{n}\right)\right] + O(Y),\end{aligned}$$

par simples manipulations algébriques.

Donc $\tilde{Z} = med(Z) \Leftrightarrow \text{diff}(n) \equiv \left[\sum_{k=0}^{(n-1)/2}\binom{n}{k}\left(\frac{n-1}{n}\right)^k\left(\frac{1}{n}\right)^{n-k}\right] - \left[\frac{1}{n^2}\left(2-\frac{1}{n}\right)\right] < 0.$
La figure 3.9 illustre les valeurs prises par $\text{diff}(n)$ pour les nombres impairs de 3 à 399.

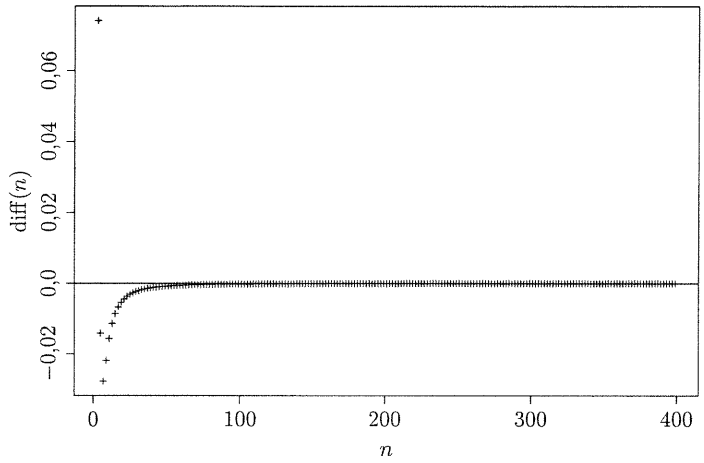


FIGURE 3.9. Valeur de diff(n) en fonction de n

Nous avons donc que pour tous les cas étudiés, sauf $n = 3$, le point de rupture de l'estimateur adaptatif $\tilde{Z}$ est supérieur à $1/n$. L'estimateur $\tilde{Z}$ semble donc plus robuste que $\bar{Z}$ si n est impair et $n \geq 5$. Les cas où n est pair devraient donner le même résultat, la principale différence étant que $med(Z^*)$ n'est pas nécessairement égale à un élément de l'échantillon Z. La généralisation aux jeux de données incluant des données égales est pour sa part triviale.

Afin d'étudier les cas où plusieurs données sont aberrantes, on suppose que ces dernières sont toutes égales. On pourra dire que l'estimateur adaptatif sera rompu par des valeurs aberrantes arbitraires s'il rompt dans de tels cas. En utilisant la formule (3.11) dans le cas de la médiane et en mettant en évidence tous les termes en Y^2 pour la moyenne, nous arrivons aux résultats suivants dans le cas n=5 :

$$\mathbb{E}_{\hat{P}}\left[(med(Z^*) - med(Z))^2\right] = 0{,}05792Y^2 + O(Y),$$
$$\mathbb{E}_{\hat{P}}\left[\bar{Z}^* - med(Z))^2\right] = 0{,}072Y^2 + O(Y),$$

si une donnée est aberrante, et

$$\mathbb{E}_{\hat{P}}\left[(med(Z^*) - med(Z))^2\right] = 0{,}31744Y^2 + O(Y),$$
$$\mathbb{E}_{\hat{P}}\left[\bar{Z}^* - med(Z))^2\right] = 0{,}208Y^2 + O(Y),$$

si deux données sont aberrantes.

C'est donc dire que dans ce dernier cas, $\tilde{Z} = \bar{Z}$ et que le point de rupture de $\tilde{Z}$ est plus petit ou égal à 2/5. Les résultats de tels calculs pour d'autres valeurs de n sont reportés au tableau 3.1 (les commandes Mathematica, ainsi que les résultats détaillés sont disponibles à l'annexe A.)

n	Borne supérieure du point de rupture de $\tilde{Z}$	Point de rupture de $med(Z)$
3	1/3	2/3
5	2/5	3/5
7	3/7	4/7
9	3/9	5/9
11	4/11	6/11
13	5/13	7/13
15	6/15	8/15
17	7/17	9/17
19	8/19	10/19
21	9/21	11/21

TABLEAU 3.1. Étude du point de rupture de $\tilde{Z}$

L'estimateur adaptatif $\tilde{Z}$ est donc plus robuste que $\bar{Z}$, mais moins robuste que $med(Z)$.

Bien que nous n'ayons aucune preuve, nous croyons que le même phénomène se produit pour $\tilde{\beta}_A$ qui devrait être plus robuste que $\hat{\beta}_{MC}$, mais moins robuste que $\hat{\beta}_{max}$.

3.5. CONCLUSION

Le modèle (3.3) que nous utilisons étant très général, nous présenterons au chapitre suivant les résultats obtenus avec l'estimateur adaptatif $\tilde{\beta}_A$ pour plusieurs modèles. Puisque notre estimateur adaptatif se veut une alternative moins variable à l'estimateur par moindres carrés tronqués à robustesse maximale, nous nous concentrerons principalement sur la comparaison des estimateurs $\tilde{\beta}_A$ (2.26) et $\hat{\beta}_{max}$ (1.7).

Chapitre 4

SIMULATIONS

Nous étudierons dans ce chapitre le comportement de l'estimateur adaptatif $\tilde{\beta}_A$ pour plusieurs modèles différents. Les erreurs quadratiques moyennes des différents estimateurs par moindres carrés tronqués, ainsi que de l'estimateur adaptatif, seront utilisées afin de comparer les estimateurs entre eux. Nous pourrons ainsi comparer à l'aide d'efficacités relatives le comportement de l'estimateur adaptatif à celui des estimateurs $\hat{\beta}_{max}$, $\hat{\beta}_n$=$\hat{\beta}_{MC}$ et enfin $\hat{\beta}_{h_{opt}}$. Des approximations basées sur 500 jeux de données seront utilisées pour calculer ces quantités.

4.1. Modèles testés

Voici le modèle général qui a été utilisé pour les différentes simulations :

$$\begin{cases} X_i = V_{1,i} I_i + V_{3,i}(1 - I_i) \\ Y_i = V_{2,i} I_i + (\beta_0 + \beta_1 X_i + \epsilon_i)(1 - I_i) \end{cases} \qquad i = 1, 2, \ldots, n \tag{4.1}$$

où $V_1 \sim N(\mu_{x_a},\ \sigma_a^2)$,

$V_2 \sim N(-1,\ \sigma_a^2)$,

$V_3 \sim U(0{,}5,\ 4{,}5)$, (4.2)

$\epsilon \sim N(0, \sigma_\epsilon^2)$,

$n_a = \sum_{i=1}^{n} I_i \sim Bin(n, p)$ (aléatoire) ou fixe,

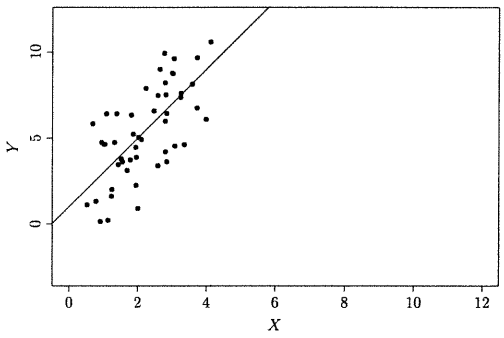
$\beta = (1, 2)'$ et $n = 50$.

Les six modèles principaux à l'étude se distinguent par l'emplacement (éloigné ou non) et la variance (grande ou petite) des valeurs aberrantes, la variance des erreurs de la partie linéaire (petite ou grande) et enfin par le nombre de données aberrantes qui peut être aléatoire (probabilité 0,2 ou 0,3) ou fixe (égal à 0, 10 ou 15.) Le tableau 4.1 résume le tout :

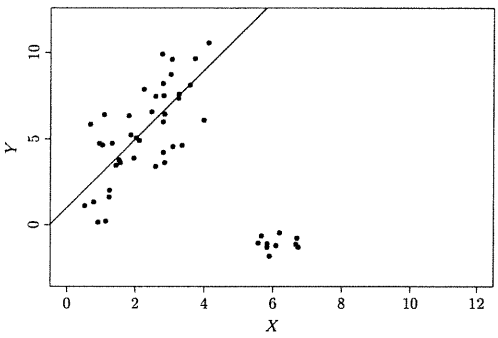
Simulation	μ_{x_a}	σ_a^2	σ_ϵ^2	p	n_a
				n_a aléatoire	n_a fixe
1	6	0,25	4		0
2	6	0,25	4		10
2a	6	0,25	4	0,2	
3	6	0,25	4		15
3a	6	0,25	4	0,3	
4	6	0,25	1		10
4a	6	0,25	1	0,2	
5	6	1	4		10
5a	6	1	4	0,2	
6	10	0,25	4		10
6a	10	0,25	4	0,2	

TABLEAU 4.1. Modèles à l'étude

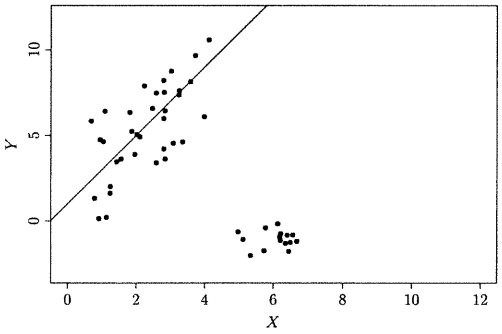
Notez que pour ces modèles, le paramètre h qui correspond à l'estimateur par moindres carrés tronqués à robustesse maximale (1.7) est $max = \left[\frac{n}{2}\right] + \left[\frac{p+1}{2}\right] = 26$. Afin de saisir visuellement la différence entre les différents modèles, la figure 4.1 illustre des exemples de jeux de données provenant des 6 modèles où n_a est fixe.



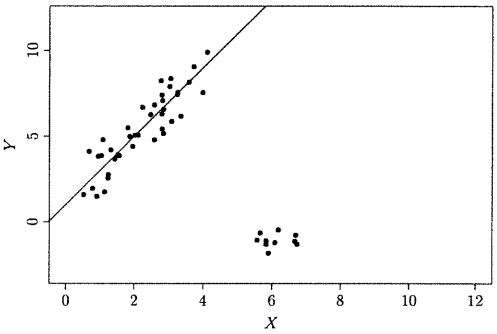
(a) Simulation 1



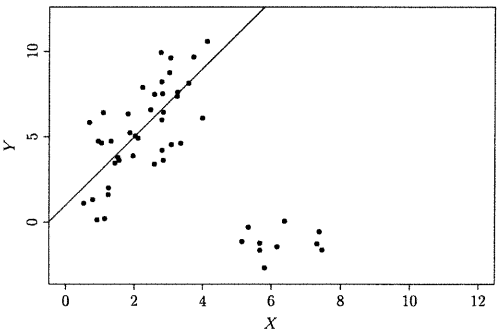
(b) Simulation 2



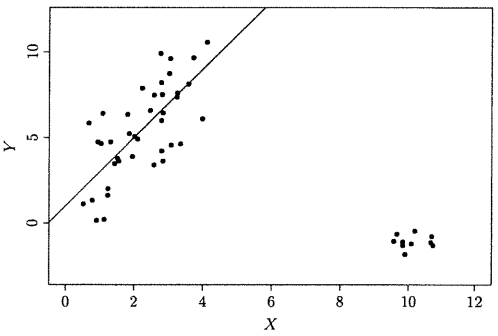
(c) Simulation 3



(d) Simulation 4



(e) Simulation 5



(f) Simulation 6

FIGURE 4.1. Exemples de jeux de données pour les simulations 1 à 6

Aux simulations 7 et 8, deux autres modèles visuellement différents des précédents seront étudiés; ceux-ci représentent d'autres situations où des données aberrantes peuvent être engendrées.

Les simulations 7 et 7a sont basées sur le modèle de l'article de Atkinson & Cheng (1999) discuté à la page 36. Dans ce cas, toutes les données suivent le modèle linéaire, les X_i et l'erreur proviennent d'une distribution unique, tandis que l'ordonnée à l'origine est fonction du groupe d'appartenance :

$$Y_i = \beta_0 + 12I_i + \beta_1 X_i + \epsilon_i,\ i = 1, 2, \ldots, n \tag{4.3}$$

où $X \sim U(0,\ 10)$,

$\epsilon \sim N(0, 1)$,

$n_a = \sum_{i=1}^{n} I_i \sim Bin(n,\ 0{,}2)$ (aléatoire) ou fixe (10),

$\beta = (1, 2)'$ et $n = 50$.

Ce modèle peut être exprimé sous la forme du modèle général (4.1) où les distributions de V_1 et de V_3 sont identiques, alors que $V_{2,i} = \beta_0 + 12 + \beta_1 X_i + \epsilon_i$; les autres paramètres et distributions étant ceux de (4.2). Voici un exemple d'un jeu de données généré à partir du modèle (4.3) :

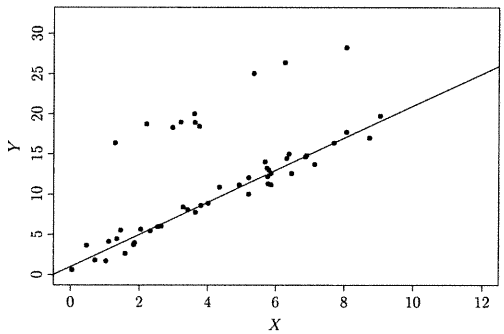


FIGURE 4.2. Exemple simulation 7

Finalement, aux simulations 8 et 8a, un modèle linéaire avec erreur ϵ suivant une distribution normale contaminée est étudié :

$$Y_i = \beta_0 + \beta_1 X_i + \epsilon_i,\ i = 1, 2, \ldots, n$$

où $X \sim U(0{,}5,\ 4{,}5)$,

$$\epsilon \sim \begin{cases} N(0,\ 100) \text{ si } i = 1, 2, \ldots, n_a \\ N(0,\ 4) \text{ si } i = n_a{+}1, \ldots, n \end{cases}$$

$$n_a = \sum_{i=1}^{n} I_i \sim Bin(n, 0{,}2) \text{ (aléatoire) ou fixe,}$$

$\beta = (1, 2)'$ et $n = 50$.

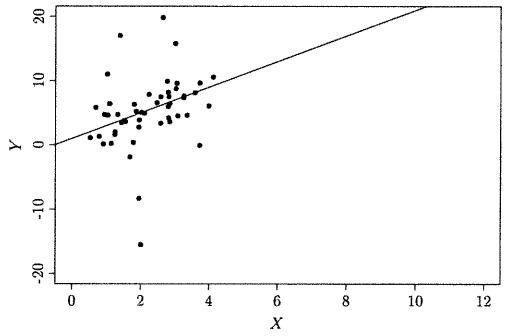


FIGURE 4.3. Exemple simulation 8

4.2. IMPLANTATION DE LA MÉTHODE

Lors des différentes simulations, dont les programmes S-Plus (version 6.0) sont disponibles à l'annexe B, nous avons utilisé *Nsimul*=500 jeux de données simulés et B=200 rééchantillons dans chaque cas. Par opposition à l'estimation des bornes d'intervalles de confiance, un tel nombre de rééchantillons est suffisant pour l'estimation de l'erreur quadratique moyenne.

4.2.1. Estimation de β par moindres carrés tronqués

Puisqu'aucune formule analytique ne nous donne la valeur prise par l'estimateur par moindres carrés tronqués pour h fixé, un algorithme d'approximation doit être utilisé.

L'estimateur $\hat{\beta}_h$ (1.6) est la valeur de β qui minimise la somme des h plus petits résidus carrés. Le seul algorithme exact connu nécessite la formation de tous les sous-ensembles de taille h, le calcul de la droite de régression par moindres carrés et de la somme des carrés des h résidus pour chacun d'eux. La droite de régression du sous-ensemble avec la somme des carrés la plus petite est l'estimateur par moindres carrés tronqués de paramètre h. Selon Hawkins (1994, p.187), cet algorithme devient impraticable lorsque n est plus élevé qu'une douzaine ou deux, ou si plus de deux ou trois donnnées doivent être tronquées.

Pour calculer $\hat{\beta}_h$, nous utilisons la fonction ltsreg de Splus version 6.0 pour Unix. L'algorithme utilisé est de type génétique (voir Burns, 1992) où différentes solutions correspondent à l'estimation par moindres carrés appliquée à différents sous-ensembles de taille h. Plusieurs solutions sont examinées, et celle qui possède la somme des h résidus carrés la plus petite est retenue. Puisque seulement un certain nombre de sous-ensembles sont formés, cet algorithme est approximatif. L'implantation de l'algorithme permet la modification de certains paramètres; nous avons tout de même utilisé les valeurs par défaut.

4.2.2. Estimateur robuste $\hat{\sigma}$

Nous avons utilisé l'estimé robuste d'écart type fourni par la fonction ltsreg de S-Plus 6.0. Cet estimateur est calculé de la façon suivante :

- $s^0 = c(n)\sqrt{\frac{1}{max}\sum_{i=1}^{max}(r^2)_{(i)}}$

- $w_i = \begin{cases} 1 & \text{si } \frac{|r_i|}{s^0} \leq 2{,}5 \\ 0 & \text{sinon} \end{cases} \quad i = 1, 2, \ldots, n\,;$

- $\hat{\sigma} = \sqrt{\frac{\sum_{i=1}^{n} w_i r_i^2}{\sum_{i=1}^{n} w_i - p}},$

où $c(n) = \left[1 - \frac{2n}{max}\Phi^{-1}\left(\frac{max+n}{2n}\right)\phi\left(\Phi^{-1}\left(\frac{max+n}{2n}\right)\right)\right]^{-1/2}$ est un facteur de correction (voir Rousseeuw & Hubert, 1997), Φ et ϕ représente la fonction de répartition et la densité d'une variable aléatoire $N(0,1)$, $max = \left[\frac{n}{2}\right] + \left[\frac{p+1}{2}\right]$, r_i sont les résidus obtenus de $\hat{\beta}_{max}$ et p est le nombre de paramètres à estimer.

4.3. MESURES DE PERFORMANCE DES ESTIMATEURS ADAPTATIFS

En plus des estimés par rééchantillonnage de l'erreur quadratique moyenne de l'espérance en $\bar{x}$ obtenus des estimateurs par moindres carrés tronqués pour chaque jeu de données, les 500 jeux de données simulés nous permettront de calculer des approximations aux biais, variances et erreurs quadratiques moyennes des deux composantes de $\hat{\beta}_h$ et de $\tilde{\beta}_A$. Des graphiques illustrant ces quantités pour toutes les simulations sont disponibles à l'annexe C (page 111.) Nous nous intéresserons en premier lieu à l'erreur quadratique moyenne de l'espérance au point $x{=}\mu_x$ de $\hat{\beta}_h$, $(h{=}max, \ldots, n)$ et de $\tilde{\beta}_A$ calculés à partir des approximations d'erreurs quadratiques moyennes précédentes.

Afin de juger de la qualité de l'estimateur par rééchantillonnage $\widehat{EQM}_B(\hat{y}_h^*(\overline{x}))$ de $EQM(\hat{y}_h(\mu_x))$ pour déterminer le h optimal, nous comparerons $EQM(\hat{y}_h(\mu_x))$ à la moyenne de ses 500 estimés par rééchantillonnage. Notez que l'intérêt est dans le paramètre h correspondant au minimum de $\widehat{EQM}_B(\hat{y}_h^*(\overline{x}))$ pour chacun des jeux de données qui définit $\tilde{\beta}_A$, plutôt que dans la courbe moyenne ; mais cette dernière permet de visualiser les résultats obtenus par rééchantillonnage. À titre d'exemple, voici les erreurs quadratiques moyennes de l'espérance à la moyenne des différents estimateurs de la simulation 2 qui seront discutés à la section 4.4.

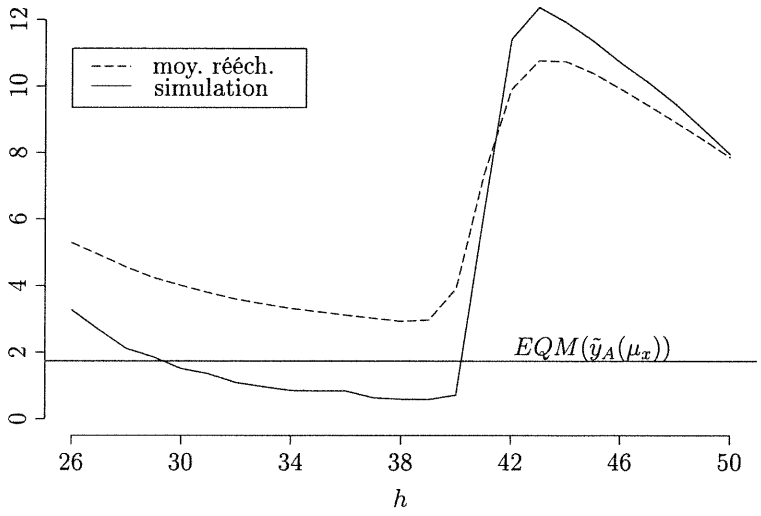


FIGURE 4.4. EQM de l'espérance à la moyenne pour la simulation 2

La courbe pleine représente l'approximation par simulation basée sur 500 jeux de données de l'erreur quadratique moyenne de l'espérance au point μ_x, $EQM(\hat{y}_h(\mu_x))$ (3.9) pour les différents estimateurs par moindres carrés tronqués. La courbe pointillée est la moyenne des 500 estimations par rééchantillonnage de l'erreur quadratique moyenne de l'espérance au point $\bar{x}$, $\widehat{EQM}_B(\hat{y}_h^*(\bar{x}))$ (3.10) pour les mêmes estimateurs. Et finalement, la droite horizontale est l'approximation par simulation de l'erreur quadratique moyenne de l'espérance au point μ_x pour l'estimateur adaptatif $\tilde{\beta}_A$, $EQM(\tilde{y}_A(\mu_x)) = EQM(\tilde{\beta}_{A,0} + \mu_x\tilde{\beta}_{A,1})$.

Le graphique précédent nous permet de comparer l'estimateur adaptatif $\tilde{\beta}_A$ aux autres estimateurs, mais il nous permet aussi de visualiser les résultats obtenus par rééchantillonnage. C'est pour cette raison que ce type de graphique a été utilisé pour étudier le fonctionnement de la méthode lors du développement. Il peut en effet aider à diagnostiquer les raisons du mauvais fonctionnement de l'estimateur adaptatif. Des graphiques semblables mais illustrant soit le biais carré, la variance ou l'erreur quadratique moyenne de $\hat{\beta}_{h,0}$ et de $\hat{\beta}_{h,1}$ peuvent aider à investiguer plus à fond les résultats obtenus pour un modèle particulier (voir annexe C.)

Afin d'étudier la performance de l'estimateur adaptatif $\tilde{\beta}_A$, nous allons illustrer différemment certains éléments du graphique précédent. On s'intéresse premièrement aux efficacités de chacun des estimateurs ($\tilde{\beta}_A$ et $\hat{\beta}_h$ où $h \in \{max, \ldots, n\}$) par rapport à $\hat{\beta}_{h_{opt}}$:

$$Eff_A = \frac{EQM(\hat{y}_{h_{opt}}(\mu_x))}{EQM(\tilde{y}_A(\mu_x))}, \tag{4.4}$$

$$Eff(h) = \frac{EQM(\hat{y}_{h_{opt}}(\mu_x))}{EQM(\hat{y}_h(\mu_x))}. \tag{4.5}$$

Puisqu'on désire un estimateur adaptatif avec l'erreur quadratique moyenne de l'espérance en μ_x la plus petite possible, on désire une efficacité par rapport à h_{opt}, soit Eff_A, la plus grande possible. Notez que pour tous les modèles, il y aura toujours un paramètre h tel que $Eff(h)$=1 ; il s'agit du paramètre optimal

par rapport à notre mesure de performance, h_{opt}. Nous nous intéresserons principalement à la performance de $\tilde{\beta}_A$ par rapport à $\hat{\beta}_{max}=\hat{\beta}_{26}$, $\hat{\beta}_{MC}=\hat{\beta}_{50}$ et $\hat{\beta}_{h_{opt}}$:

$$\frac{Eff_A}{Eff(26)} = \frac{EQM(\hat{y}_{26}(\mu_x))}{EQM(\tilde{y}_A(\mu_x))}, \tag{4.6}$$

$$\frac{Eff_A}{Eff(50)} = \frac{EQM(\hat{y}_{50}(\mu_x))}{EQM(\tilde{y}_A(\mu_x))}, \tag{4.7}$$

$$\frac{Eff_A}{Eff(h_{opt})} = \frac{EQM(\hat{y}_{h_{opt}}(\mu_x))}{EQM(\tilde{y}_A(\mu_x))} = Eff_A, \tag{4.8}$$

où $\hat{y}_h(\mu_x) = \hat{\beta}_{h,0}+\mu_x\hat{\beta}_{h,1}$, $\tilde{y}_A(\mu_x) = \tilde{\beta}_{A,0}+\mu_x\tilde{\beta}_{A,1}$ et les EQM sont calculés à l'aide de l'équation (3.9) où $\hat{\beta}_{h,0}$ et $\hat{\beta}_{h,1}$ sont remplacés par l'estimateur d'intérêt ($\hat{\beta}_{26}$, $\hat{\beta}_{50}$ ou $\tilde{\beta}_A$.) Puisque les calculs des EQM de (4.6), (4.7) et (4.8) basés sur (3.9) sont en fait des moyennes, des tests-t pairés basés sur les *Nsimul* jeux de données seront faits pour tester l'hypothèse d'égalité des différentes efficacités d'intérêt (Eff_A versus $Eff(26)$, $Eff(50)$ et $Eff(h_{opt})$.)

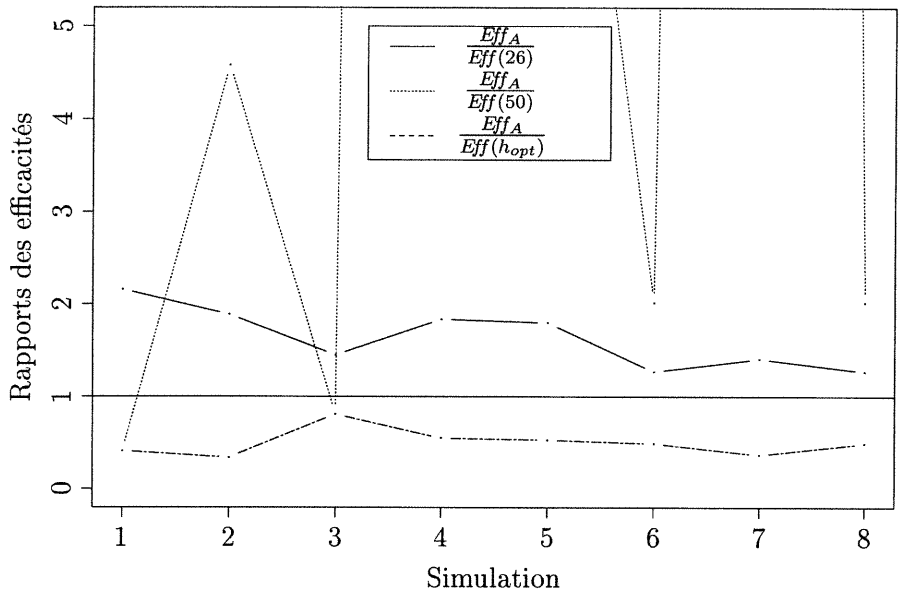
4.4. Résultats des simulations

Le tableau 4.2 et la figure 4.5 résument les résultats des différentes simulations dont les résultats graphiques complets sont disponibles à partir de la page 64.

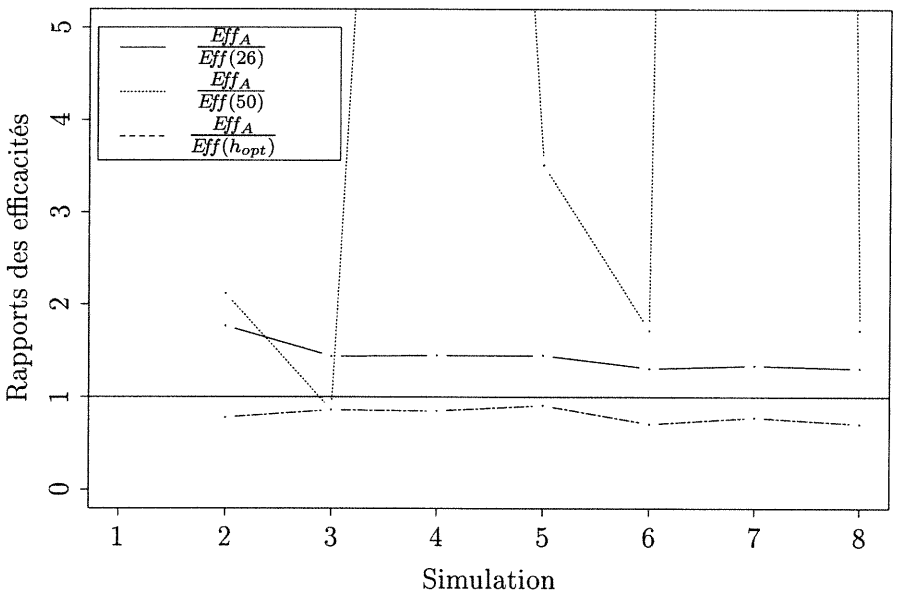
La colonne h_{opt} du tableau 4.2 indique le paramètre h optimal au sens de l'erreur quadratique moyenne de l'espérance au point μ_x tel que déterminé par simulation. Les trois autres colonnes donnent les valeurs prises par (4.6), (4.7) et (4.8) ainsi que les valeurs-p des tests pairés d'égalité du numérateur et du dénominateur. Puisque nous souhaitons une efficacité élevée pour notre estimateur adaptatif, les rapports supérieurs à 1 sont dans la direction désirée. Notez que puisque $Eff(h_{opt}) \geq Eff_A$, les valeurs-p associées à ce rapport sont unilatérales, tandis que les autres valeurs-p sont bilatérales.

Simulation	h_{opt}	$\frac{Eff_A}{Eff(26)}$ (p-bi)		$\frac{Eff_A}{Eff(50)}$ (p-bi)		$\frac{Eff_A}{Eff(h_{opt})} = Eff_A$ (p-uni)	
1	50	2,16	0	0,41	0	0,41	0
2	39	1,89	0	4,58	0	0,34	0
2a	35	1,77	0	2,12	0	0,78	0
3	50	1,45	0	0,81	0	0,81	0
3a	50	1,44	0	0,86	0	0,86	0
4	40	1,84	0	108,35	0	0,55	0
4a	31	1,45	0,04	21,12	0	0,85	0,24
5	39	1,80	0	10,42	0	0,53	0
5a	34	1,45	0	3,51	0	0,91	0,2
6	38	1,27	0	2,02	0	0,49	0
6a	36	1,31	0	1,72	0	0,71	0
7	40	1,41	0	80,85	0	0,37	0
7a	34	1,34	0	82,62	0	0,78	0
8	38	1,27	0	2,02	0	0,49	0
8a	36	1,31	0	1,72	0	0,71	0

TABLEAU 4.2. Comparaison des EQM de l'espérance au point μ_x



(a) n_a fixe



(b) n_a aléatoire

FIGURE 4.5. Comparaison des efficacités

Au sens de l'erreur quadratique moyenne de l'espérance évaluée en μ_x, on voit que l'estimateur adaptatif, lui-même défini comme l'estimateur par moindres carrés tronqués qui minimise l'estimateur par rééchantillonnage de l'erreur quadratique moyenne de l'espérance en μ_x, est bel et bien meilleur que $\hat{\beta}_{max}$ et $\hat{\beta}_{MC}$ dans presque toutes les situations étudiées. L'estimateur adaptatif est donc une alternative intéressante, au sens de l'EQM de l'espérance en μ_x, à l'estimateur par moindres carrés et à l'estimateur par moindres carrés tronqués de robustesse maximale, ce dernier étant habituellement utilisé pour se protéger contre une proportion importante de données aberrantes.

Exception faite des simulations 4a et 5a, où on ne peut pas détecter de différences significatives, la dernière colonne du tableau 4.2 et la figure 4.5 nous montrent que, tel que prévu pour des échantillons de petite taille, $\tilde{\beta}_A$ est statistiquement moins efficace que l'estimateur par moindres carrés tronqués avec paramètre optimal h_{opt}.

Une approche du type minimax pour évaluer la performance de l'estimateur adaptatif $\tilde{\beta}_A$ consiste à calculer et comparer la polyefficacité des différents estimateurs. La polyefficacité d'un estimateur sur un ensemble de modèles est l'efficacité minimale atteinte par cet estimateur sur l'ensemble des modèles (Tukey, 1979). Les figures 4.6, 4.7 et 4.8 illustrent la polyefficacité des différents estimateurs par moindres carrés tronqués, ainsi que de l'estimateur adaptatif, pour les simulations présentées jusqu'à maintenant.

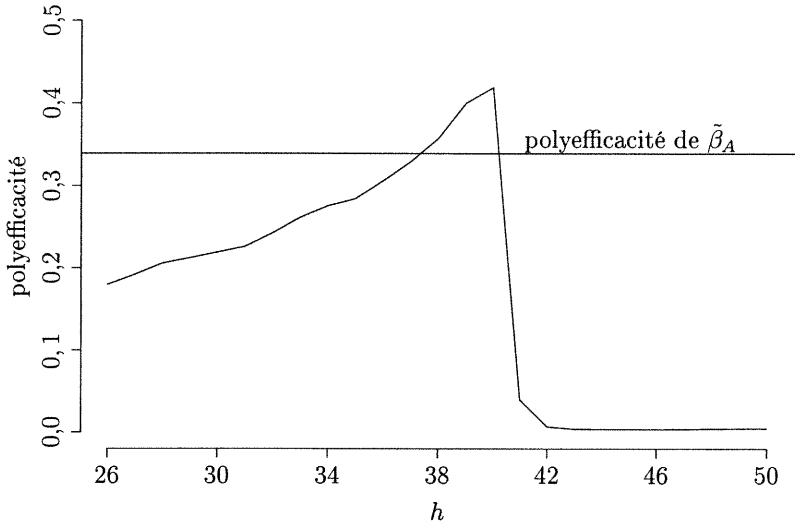


FIGURE 4.6. Polyefficacité des différents estimateurs pour les 8 modèles où n_a est fixe

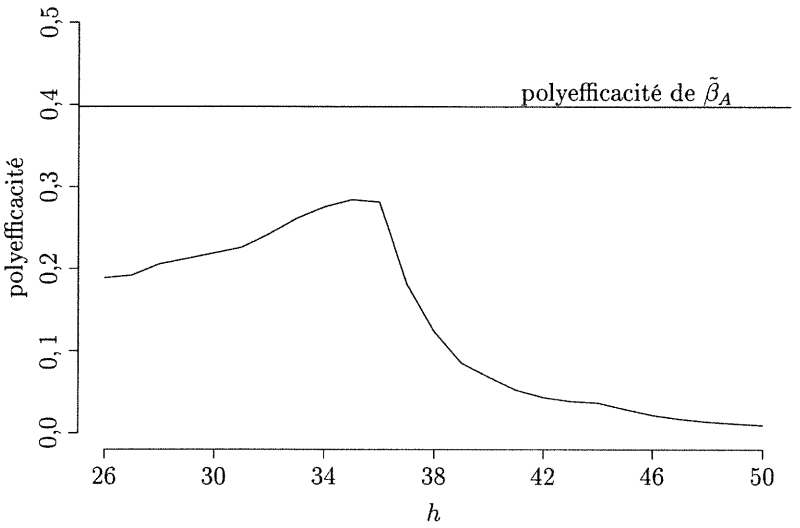


FIGURE 4.7. Polyefficacité des différents estimateurs pour les 7 modèles où n_a est aléatoire

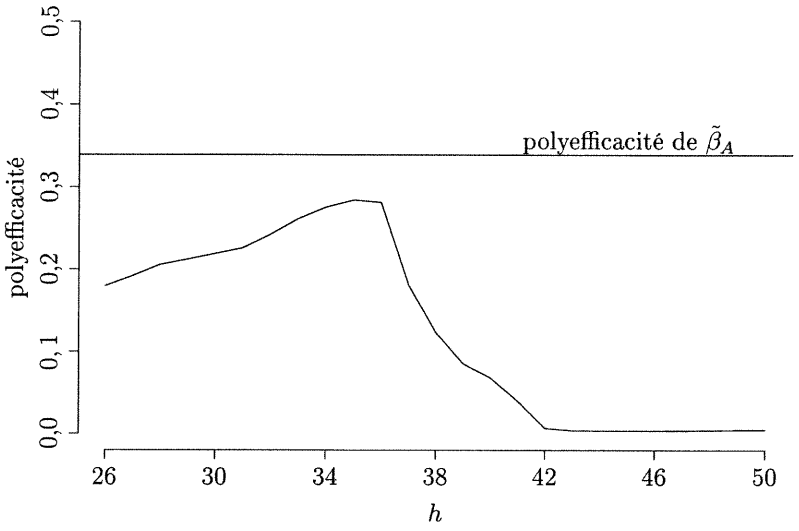
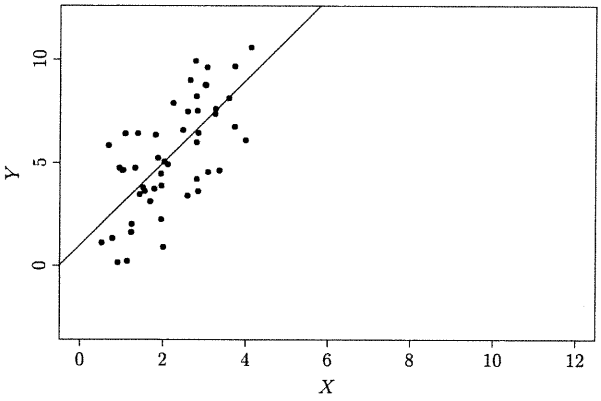


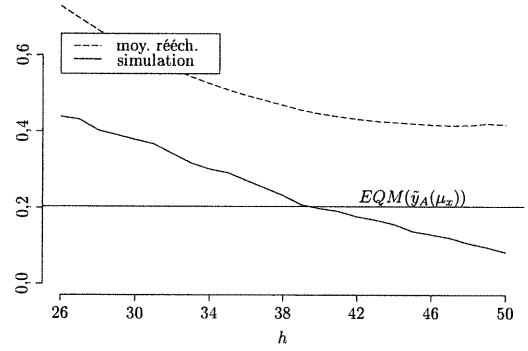
FIGURE 4.8. Polyefficacité des différents estimateurs pour l'ensemble des 15 modèles étudiés

Lorsque nous ne considérons que les modèles où n_a est fixe, l'estimateur $\hat{\beta}_{40}$ est celui qui possède la polyefficacité la plus élevée, mais l'estimateur adaptatif $\tilde{\beta}_A$ n'est pas très loin. Par contre, lorsqu'on considère les modèles où n_a est aléatoire ou simplement tous les modèles étudiés, l'efficacité minimale de $\tilde{\beta}_A$ est supérieure à l'efficacité minimale de tous les estimateurs par moindres carrés tronqués pour ces mêmes modèles. Notez que les estimateurs adaptatifs sont calculés en sachant si n_a est fixe ou aléatoire. Mais comme il en sera question à la section 4.5, l'efficacité de $\tilde{\beta}_A$ dépend très peu de cette information.

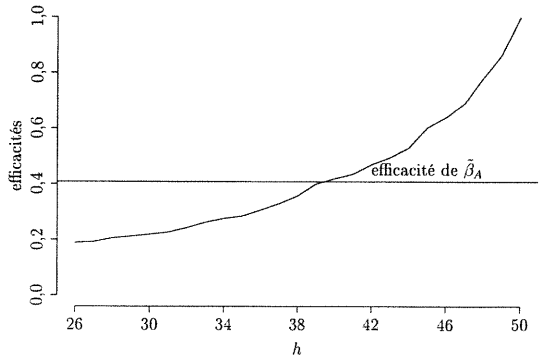
Les figures 4.9 à 4.24 illustrent les résultats des différentes simulations. En plus de l'efficacité qui est illustrée pour tous les estimateurs, on s'intéresse à la comparaison de la moyenne des 500 estimations par rééchantillonnage $\widehat{EQM}_B(\hat{y}_h^*(\bar{x}))$ de $EQM(\hat{y}_h(\mu_x))$ et de l'approximation par simulation de $EQM(\hat{y}_h(\mu_x))$, ainsi qu'à l'approximation de $EQM(\tilde{y}_A(\mu_x))$, tels qu'illustrés à la figure 4.4 de la page 56.



(a) Exemple simulation 1



(b) EQM de l'espérance à la moyenne (simulation 1)



(c) Efficacités (simulation 1)

FIGURE 4.9. Simulation 1

Puisque le modèle de la simulation 1 ne comporte aucune donnée aberrante, tous les estimateurs par moindres carrés tronqués sont sans biais pour β, et $\hat{\beta}_{MC}$=$\hat{\beta}_{50}$ est le moins variable. L'estimateur $\hat{\beta}_{MC}$ est aussi celui qui possède l'erreur quadratique de l'espérance au point μ_x le plus petit (voir figure 4.9(b).) L'histogramme suivant nous montre que le paramètre $\hat{h}$ choisi par l'algorithme 3.1 (voir page 43) a été $\hat{h}$=50 dans environ la moitié des 500 simulations, alors que des valeurs plus petites ont été choisies pour l'autre moitié des jeux de données.

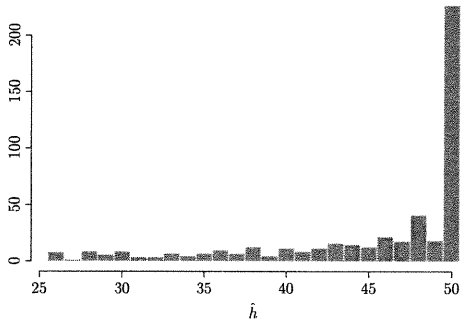
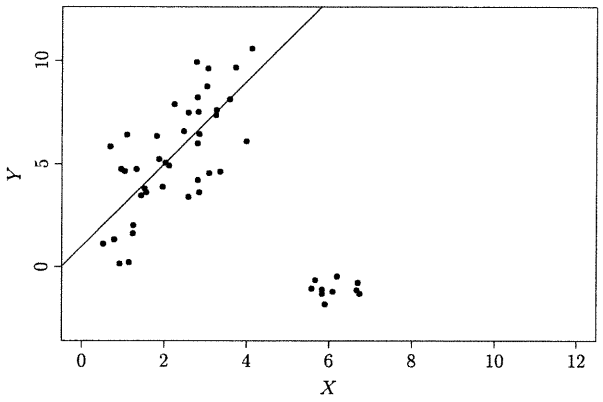


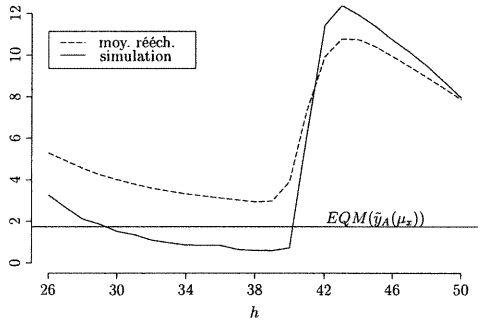
FIGURE 4.10. Choix de $\hat{h}$ (2.27) pour les 500 jeux de données de la simulation 1

L'efficacité de $\tilde{\beta}_A$ telle que définie par (4.4) est donc inférieure à celle de $\hat{\beta}_{MC}$ définie par (4.5) où h=50, mais il s'agit du prix à payer lors de l'utilisation d'estimateurs robustes lorsqu'aucune donnée n'est aberrante. Notons que $\tilde{\beta}_A$ est néanmoins significativement plus efficace que $\hat{\beta}_{max}$=$\hat{\beta}_{26}$.

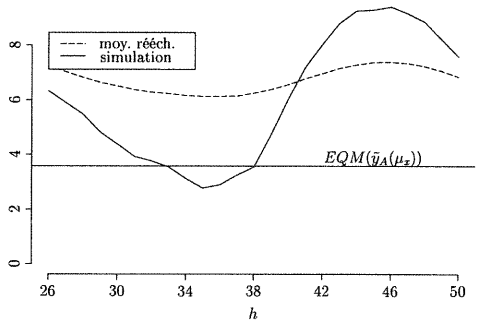
L'utilisation de notre estimateur adaptatif dans un cas où aucune donnée n'est aberrante est donc une alternative intéressante à l'estimateur par moindres carrés tronqués à robustesse maximale, tandis que les moindres carrés demeurent le meilleur choix.



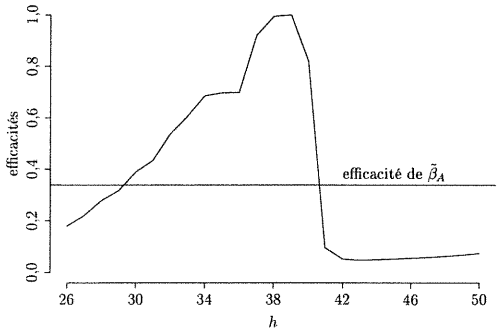
(a) Exemple simulation 2



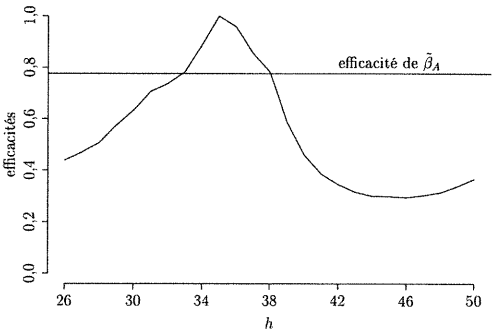
(b) EQM de l'espérance à la moyenne (simulation 2)



(c) EQM de l'espérance à la moyenne (simulation 2a)



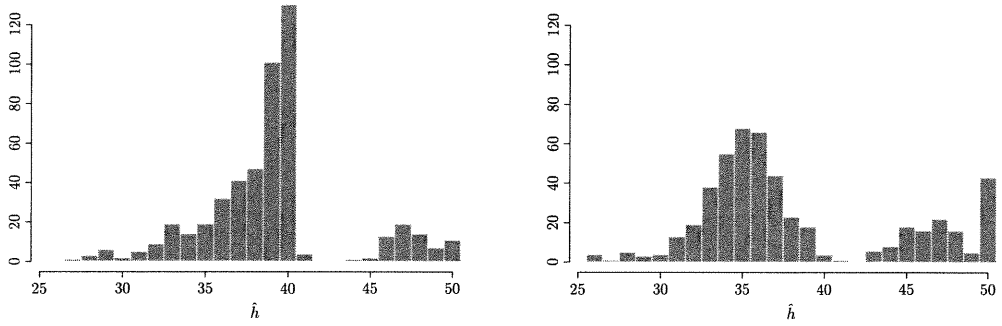
(d) Efficacités (simulation 2)



(e) Efficacités (simulation 2a)

FIGURE 4.11. Simulations 2 et 2a

Pour les simulations 2 et 2a, les paramètres optimaux sont 39 et 35 pour n_a fixe et aléatoire respectivement. Les indices choisis par l'algorithme pour les différents jeux de données sont illustrés à la figure 4.12.

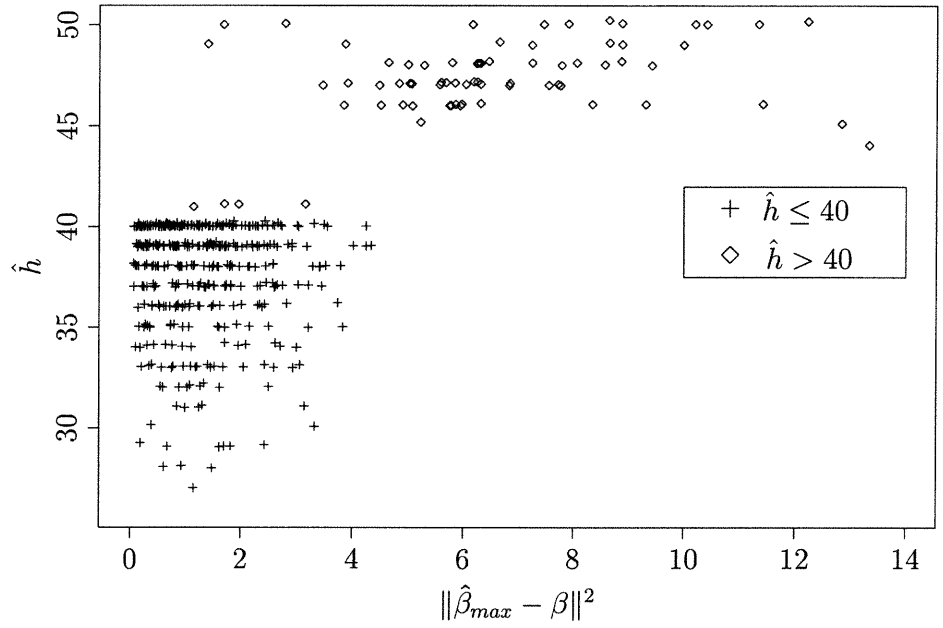


(a) simulation 2 (b) simulation 2a

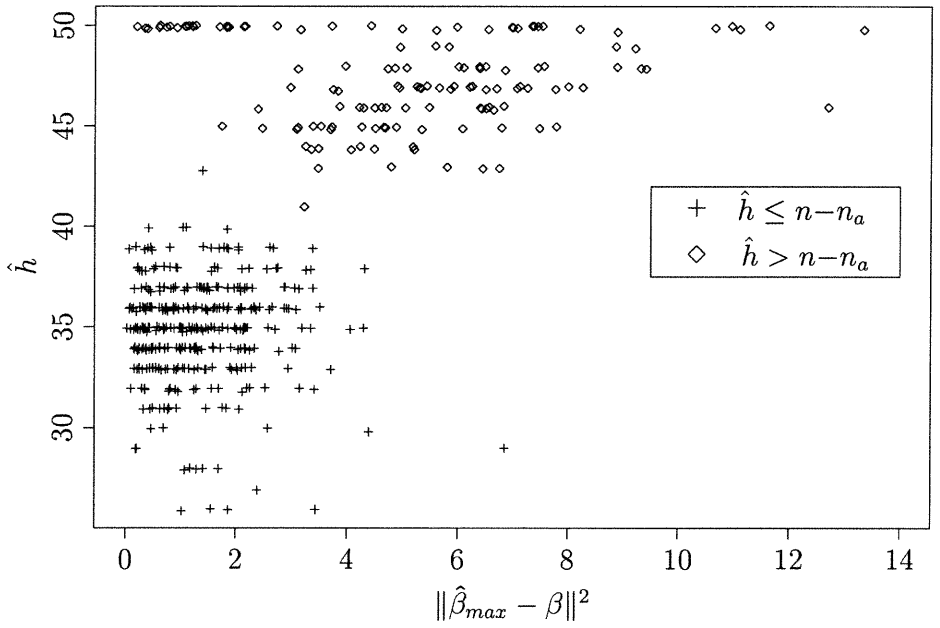
FIGURE 4.12. Choix de $\hat{h}$ (2.27) pour les 500 jeux de données des simulations 2 et 2a

Dans le cas de la simulation 2, les choix de $\hat{h}$>40 (qui donnent tous des estimateurs $\tilde{\beta}_A$=$\hat{\beta}_{\hat{h}}$ biaisés) ont été causés la plupart du temps par des valeurs de $\hat{\beta}_{max}$ influencées par les données aberrantes. Il s'agit en effet de simulations pour lesquelles l'estimateur par moindres carrés tronqués à robustesse maximale appliqué au jeu de données original a rompu. Les valeurs de $\widehat{EQM}_B(\hat{y}_h^*(\bar{x}))$ (3.10) pour tous les h ont donc été influencés. La figure 4.13(a) illustre le choix de l'algorithme en fonction de la norme euclidienne entre le vecteur β et l'estimateur $\hat{\beta}_{max}$ pour la simulation 2. Les + correspondent aux jeux de données simulés pour lesquels l'estimateur adaptatif est un estimateur par moindres carrés tronqués $\hat{\beta}_h$ (1.6) tel que le paramètre h est inférieur ou égal au nombre de données provenant du modèle linéaire (40 pour la simulation 2) ; alors que les $\diamond$ correspondent aux jeux de données pour lesquels l'estimateur adaptatif incluait au moins une donnée aberrante. On peut facilement remarquer que les "mauvais choix" ($\diamond$) sont habituellement associés aux jeux de données pour lesquels l'estimateur $\hat{\beta}_{max}$ est éloigné de β.

Dans le cas où n_a est aléatoire (simulation 2a), une étude des cas où $\hat{h}>n-n_a$ montre le même phénomène, alors que les $\diamond$ sont souvent dans la partie de droite de la figure 4.13(b).



(a) Simulation 2



(b) Simulation 2a

FIGURE 4.13. Choix de $\hat{h}$ (2.27) en fonction de $\hat{\beta}_{max}$ pour les simulations 2 et 2a

L'estimateur adaptatif tel que nous l'avons défini doit donc être basé sur un estimateur de β le plus robuste possible, puisque cet estimateur est utilisé dans l'estimation de l'erreur quadratique moyenne de l'espérance. Le choix de $\hat{\beta}_{max}$ qui a déjà été justifié pour de telles raisons est donc le meilleur choix que nous puissions faire, bien que cet estimateur ne soit pas infaillible.

Remarquez en terminant que l'efficacité de $\tilde{\beta}_A$ est supérieure dans le cas aléatoire (voir figures 4.11(d) et 4.11(e)), ce qui est le cas dans toutes les simulations, mais ceci est dû au fait que les erreurs quadratiques moyennes de l'espérance sont plus élevées lorsque n_a est aléatoire. Donc un même écart entre les estimateurs $\tilde{\beta}_A$ et $\hat{\beta}_{h_{opt}}$ mène à une plus grande efficacité lorsque n_a est aléatoire puisque le point de référence est plus élevé.

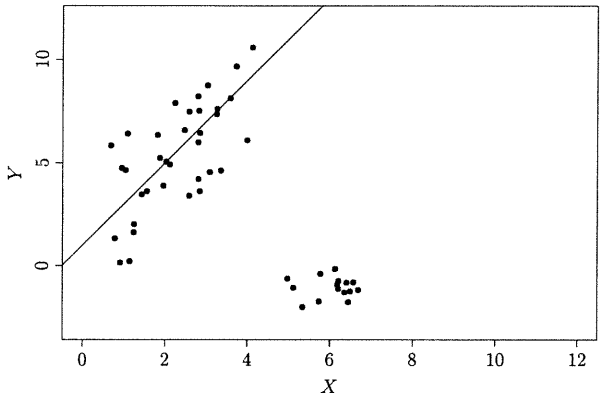
Aux simulations 3 et 3a, le nombre de données aberrantes est passé de 10 à 15, ou de 20% à 30% comparativement aux simulations 2 et 2a. Le tableau 4.2 et les figures 4.14(b) et 4.14(c) nous montrent que h_{opt}=50 pour ces deux simulations. Ceci est pour le moins surprenant lorsqu'on regarde la figure 4.14(a). Les figures 1.6(a) et 1.6(b) du chapitre 1 illustrant le biais carré de $\hat{\beta}_{h,0}$ et $\hat{\beta}_{h,1}$ pour la simulation 3 sont reproduites à la figure 4.15. Il s'agit de la principale cause pour expliquer des erreurs quadratiques moyennes de l'espérance en μ_x aussi élevées pour les valeurs de h inférieures ou égale à 35.

Comme il a été mentionné au chapitre 1, les biais sont élevés pour tous les estimateurs, incluant $\hat{\beta}_{max}$ à $\hat{\beta}_{35}$ qui devraient être sans biais s'ils n'étaient pas affectés par les valeurs aberrantes. C'est donc dire que ces estimateurs ont rompu pour plusieurs des jeux de données. Notez que nous ne contredisons pas ici la proposition 1.2 (page 7) qui affirme que le point de rupture de $\hat{\beta}_{max}$ est $\frac{\left[\frac{n-p}{2}\right]+1}{n} = 0{,}5$. L'estimateur $\hat{\beta}_{max}$ a rompu pour plusieurs jeux de données, n'incluant que 30% de données aberrantes, mais pas de façon arbitraire en causant un biais infini. Et à l'instar des simulations 2 et 2a, un estimé $\hat{\beta}_{max}$ de β influencé par les données aberrantes implique des estimations par rééchantillonnage de $EQM(\hat{y}(\mu_x))$ influencées ; ce qui se répercute sur l'estimateur adaptatif $\tilde{\beta}_A$.

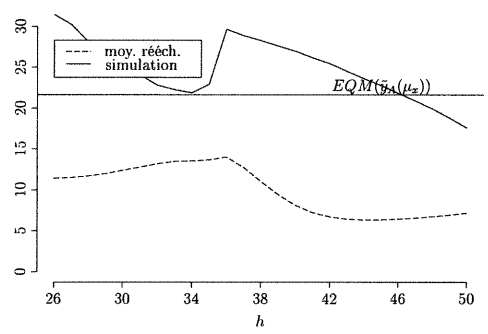
Encore une fois, cette simulation nous montre que l'utilisation d'un estimateur à robustesse maximale ne donne pas toujours les meilleurs résultats, même lorsqu'un grand nombre de données sont aberrantes et influentes. Pour les modèles à l'étude aux simulations 3 et 3a, l'estimateur adaptatif est meilleur, au sens de l'erreur quadratique moyenne de la moyenne en μ_x, que $\hat{\beta}_{max}$ (voir figures 4.14(b) et 4.14(c)) ; et la différence est statistiquement significative (voir tableau 4.2 page 59.)

D'un autre côté, le minimum local des erreurs quadratiques moyennes de l'espérance en μ_x aux alentours de h=34 et le minimum global en h=50 nous poussent à regarder notre critère de plus près. La figure 4.16, où sont illustrés $\hat{y}_{26}(\mu_x)$ et $\hat{y}_{50}(\mu_x)$ pour un jeu de données de la simulation 3, nous montre que ces deux quantités peuvent être relativement près l'une de l'autre. C'est-à-dire que bien que les pentes des deux estimateurs soient très différentes, les espérances en μ_x le sont beaucoup moins.

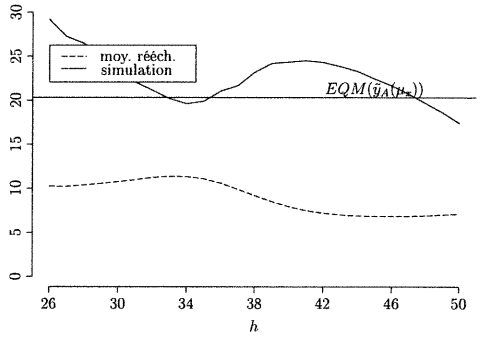
Ce problème, qui n'est pas propre à ces deux simulations, aurait pu être évité en utilisant par exemple la moyenne de l'erreur quadratique moyenne de l'espérance évaluée en des points plus extrêmes.



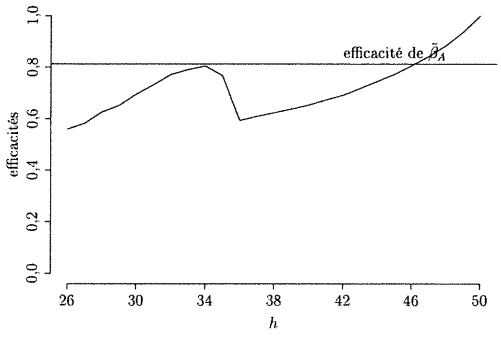
(a) Exemple simulation 3



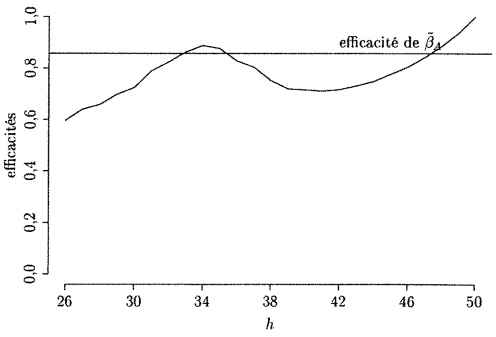
(b) EQM de l'espérance à la moyenne (simulation 3)



(c) EQM de l'espérance à la moyenne (simulation 3a)

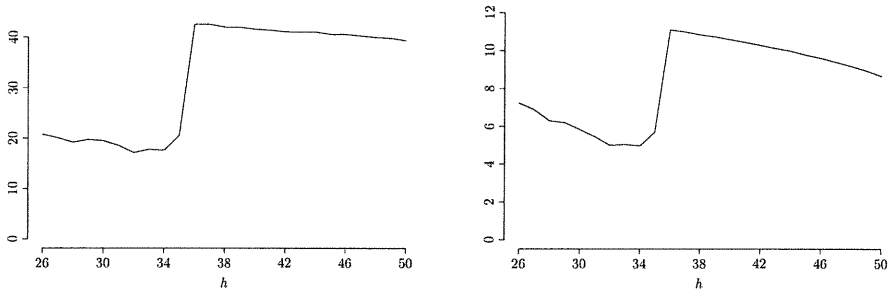


(d) Efficacités (simulation 3)



(e) Efficacités (simulation 3a)

FIGURE 4.14. Simulations 3 et 3a



(a) $\left(biais(\hat{\beta}_{h,0})\right)^2$ (b) $\left(biais(\hat{\beta}_{h,1})\right)^2$

FIGURE 4.15. Biais carré de $\hat{\beta}_{h,0}$ et $\hat{\beta}_{h,1}$ pour la simulation 3

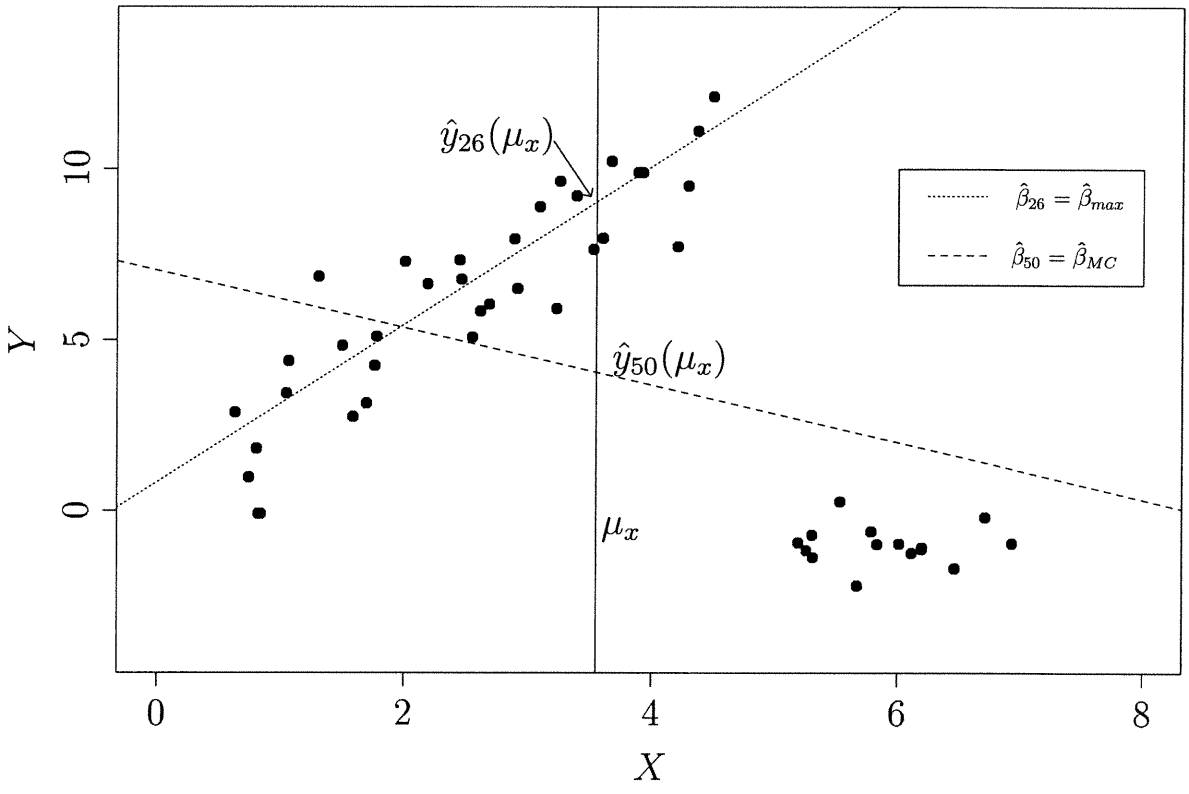


FIGURE 4.16. Étude du critère d'EQM de l'espérance au point μ_x

Les simulations 4 et 4a diffèrent des simulations 2 et 2a seulement au niveau de la variance de ϵ qui est de 1 au lieu de 4. Les paramètres optimaux sont $h_{opt} = 40$ et $h_{opt} = 31$ pour les modèles fixe et aléatoire respectivement.

Puisque la distinction entre les données aberrantes et celles de la partie linéaire est plus nette qu'aux simulations 2 et 2a, $\hat{\beta}_{max}$ rompt beaucoup moins souvent, comme en témoignent les histogrammes de la figure 4.18, qui peuvent être comparés à ceux de la figure 4.12. Il en résulte aussi des efficacités plus grandes qu'aux simulations 2 et 2a (voir figure 4.17.)

Pour la simulation 4a, notons finalement que $\tilde{\beta}_A$ n'est pas statistiquement plus efficace que $\hat{\beta}_{26}$, mais pas non plus statistiquement moins efficace que $\hat{\beta}_{h_{opt}}$ (voir tableau 4.2.)

À l'instar des modèles des simulations 4 et 4a, ceux des simulations 5 et 5a, dont les résultats sont illustrés à la figure 4.19, causent moins de ruptures de $\hat{\beta}_{max}$ que les simulations 2 et 2a (comparez par exemple les valeurs de $EQM(\hat{y}_{max}(\mu_x))$ des figures 4.11(b) et 4.19(b).) Dans le cas présent, c'est la variance des données aberrantes qui est passée de 0,25 à 1. Puisque la variance des données aberrantes aux simulations 2 et 2a était petite, les résidus correspondant à ces points étaient donc souvent petits lorsque la droite de régression $\hat{\beta}_{max}$ passait près d'eux. Ce phénomène étant beaucoup plus rare aux simulations 5 et 5a, $\hat{\beta}_{max}$ rompt moins souvent.

Les valeurs de h_{opt} sont ici 39 et 34 pour les modèles fixe et aléatoire respectivement, ce qui est normal pour 10 données aberrantes (fixe ou en moyenne.)

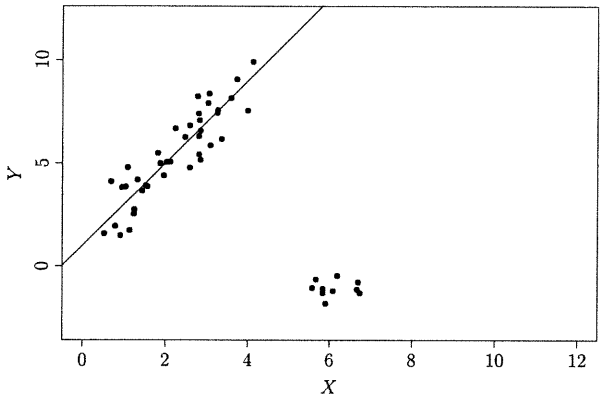
Notons que l'efficacité de 0,91 de $\tilde{\beta}_A$ dans le cas aléatoire n'est pas statistiquement inférieure à 1.

Pour les simulations 6 et 6a, le centre des données aberrantes a été déplacé de (6, -1) vers (10, -1), alors que tous les autres paramètres sont identiques à ceux des simulations 2 et 2a.

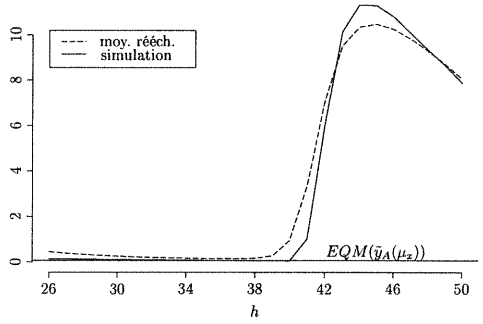
Les valeurs de h_{opt} de 38 et 36 ainsi que les graphiques d'efficacités de la figure 4.20 nous portent à croire que l'estimateur $\tilde{\beta}_A$ a donné de bons résultats pour les deux modèles, alors que la moyenne des estimations par rééchantillonnage semble indiquer une anomalie principalement à la simulation 6a. Les histogrammes des

$\hat{h}$ choisis de la figure 4.21 nous montrent en effet que des valeurs supérieures à $n-n_a$ ont été choisies à plusieurs reprises. L'étude des biais, variances et erreurs quadratiques moyennes illustrées aux figures 4.22 et C.11 nous montre que les estimés par rééchantillonnage des biais et variances suivent bien les approximations par simulations, ce qui est moins vrai pour les erreurs quadratiques moyennes.

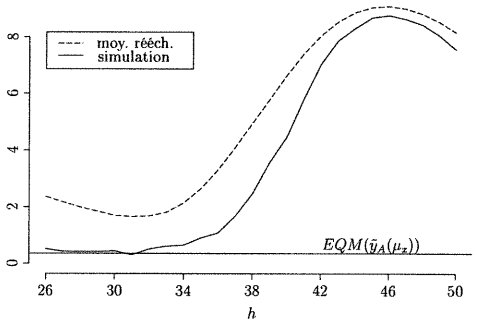
L'emploi de $\tilde{\beta}_A$ est donc préférable (toujours au sens de l'EQM de l'espérance en μ_x) à l'utilisation systématique de $\hat{\beta}_{max}$ ou $\hat{\beta}_{MC}$, mais cet estimateur ne donne pas de bons résultats à tout coup. Tout comme aux simulations 2 et 2a, la figure 4.23 nous montre que ces résultats peuvent être expliqués en bonne partie par des estimations $\hat{\beta}_{max}$ qui ont rompu ($\diamond$ dans la partie de droite des deux figures.)



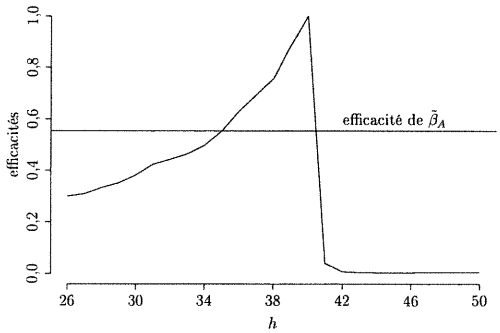
(a) Exemple simulation 4



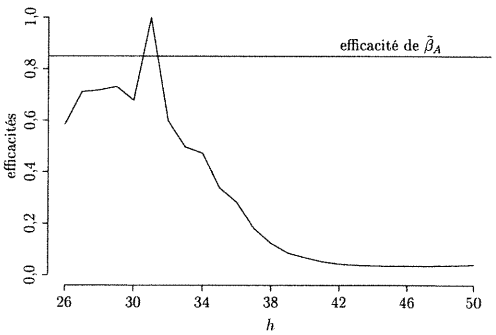
(b) EQM de l'espérance à la moyenne (simulation 4)



(c) EQM de l'espérance à la moyenne (simulation 4a)

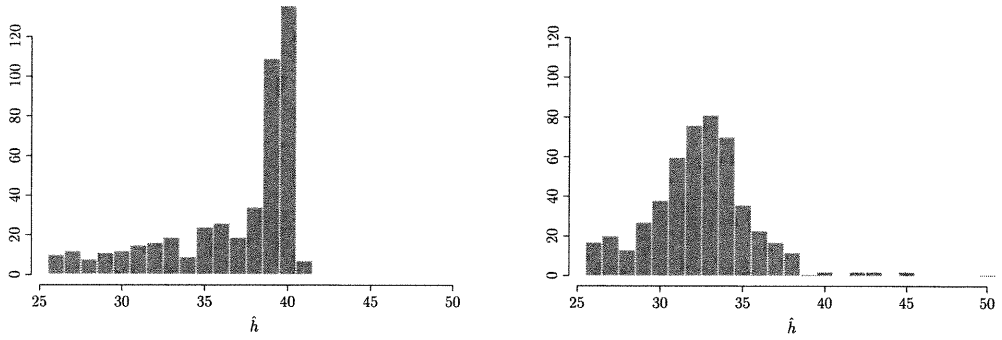


(d) Efficacités (simulation 4)



(e) Efficacités (simulation 4a)

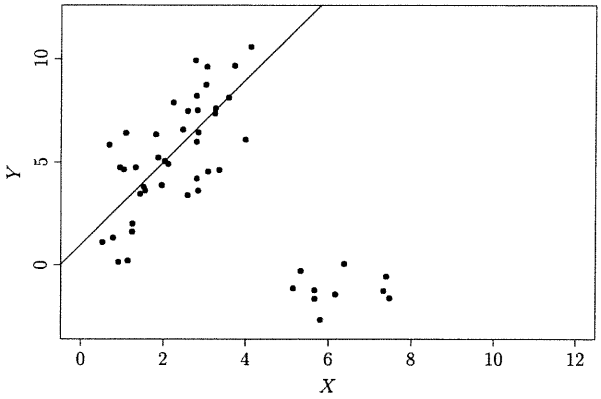
FIGURE 4.17. Simulations 4 et 4a



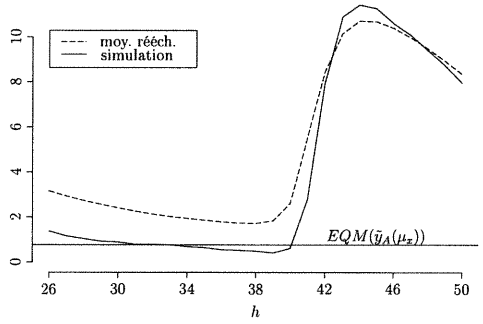
(a) simulation 4

(b) simulation 4a

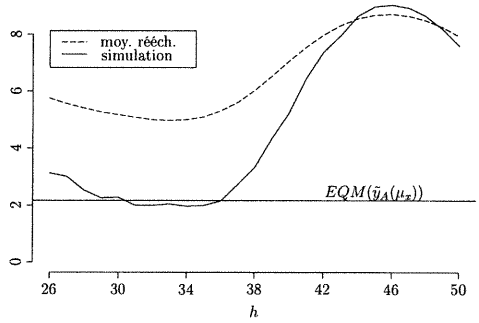
FIGURE 4.18. Choix de $\hat{h}$ (2.27) pour les 500 jeux de données des simulations 4 et 4a



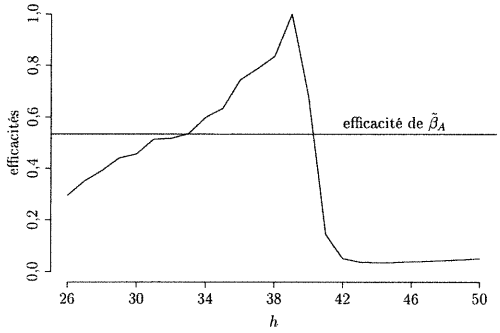
(a) Exemple simulation 5



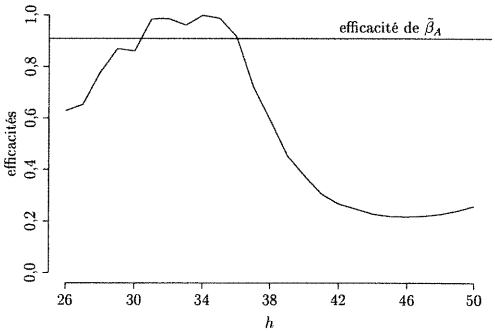
(b) EQM de l'espérance à la moyenne (simulation 5)



(c) EQM de l'espérance à la moyenne (simulation 5a)

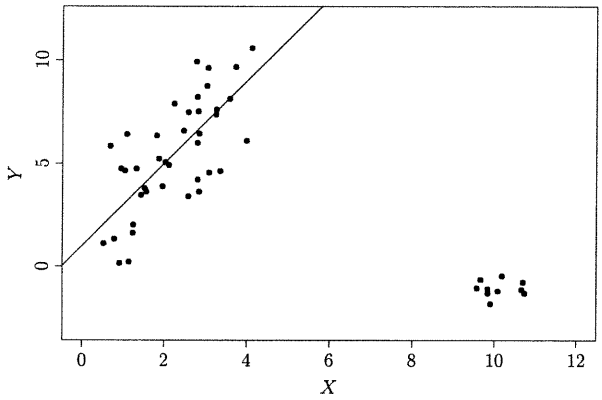


(d) Efficacités (simulation 5)

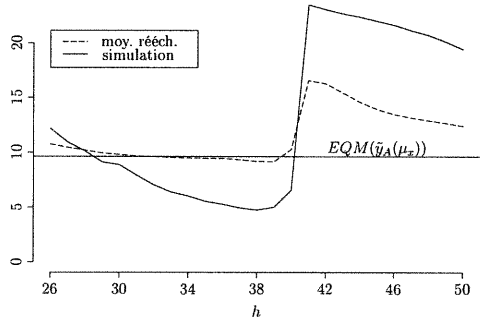


(e) Efficacités (simulation 5a)

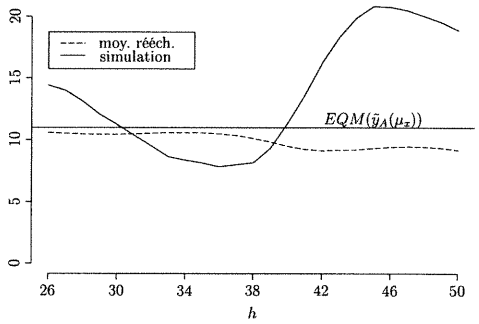
FIGURE 4.19. Simulations 5 et 5a



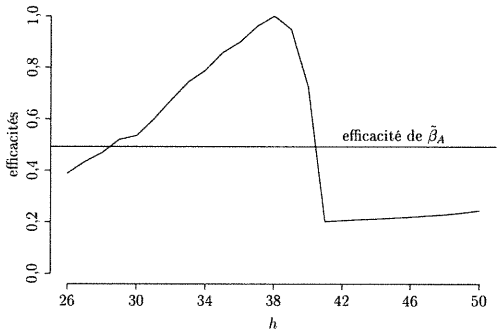
(a) Exemple simulation 6



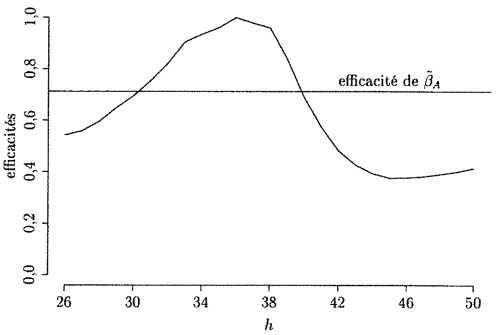
(b) EQM de l'espérance à la moyenne (simulation 6)



(c) EQM de l'espérance à la moyenne (simulation 6a)

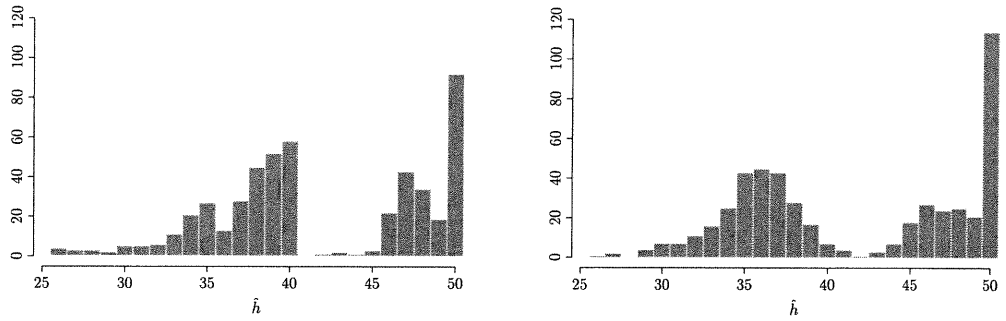


(d) Efficacités (simulation 6)



(e) Efficacités (simulation 6a)

FIGURE 4.20. Simulations 6 et 6a



(a) simulation 6 (b) simulation 6a

FIGURE 4.21. Choix de $\hat{h}$ (2.27) pour les 500 jeux de données des simulations 6 et 6a

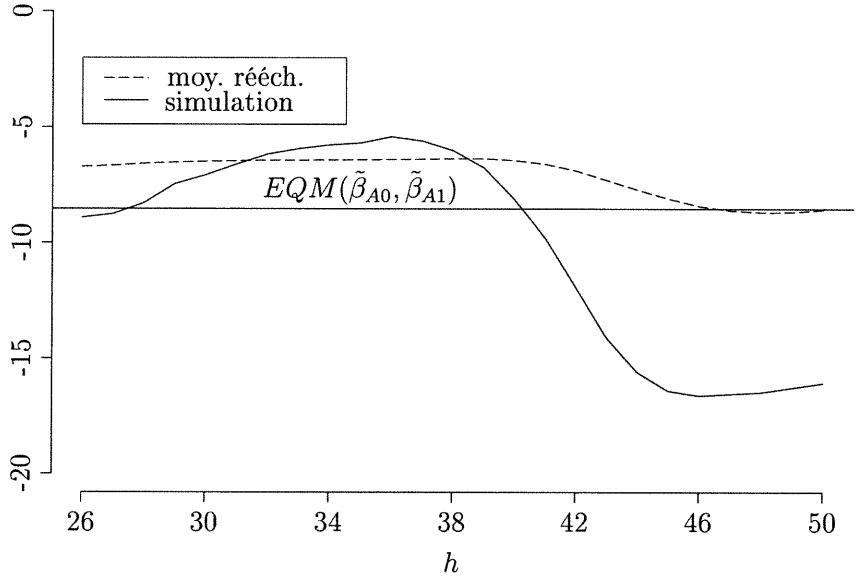
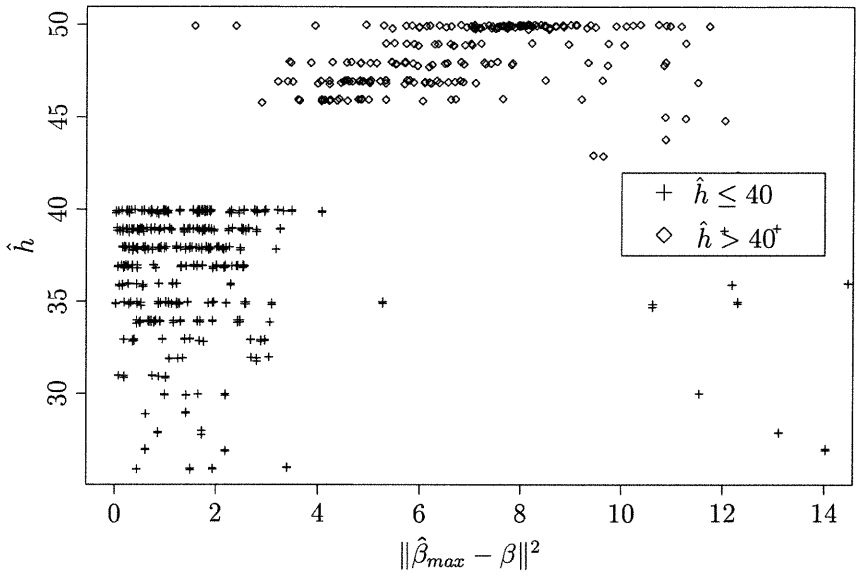
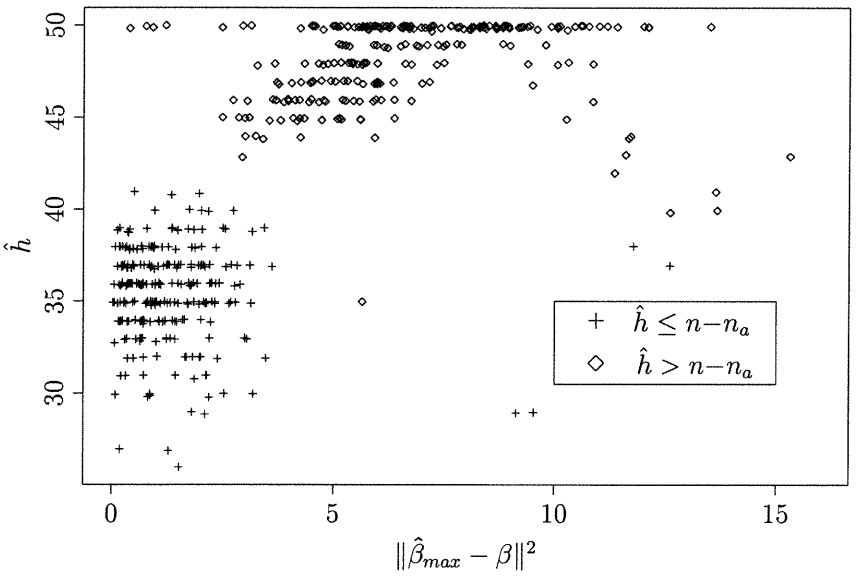


FIGURE 4.22. $EQM(\hat{\beta}_{h,0}, \hat{\beta}_{h,1})$ pour la simulation 6a



(a) Simulation 6

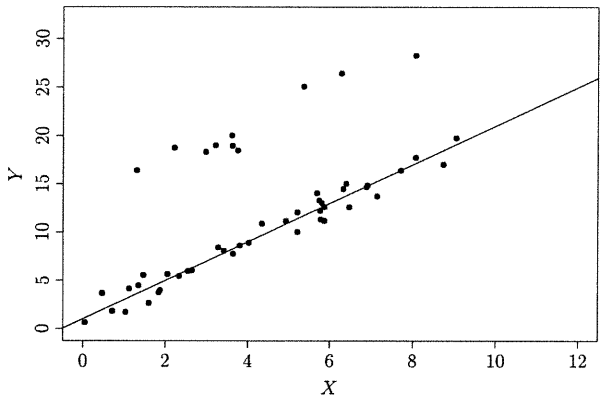


(b) Simulation 6a

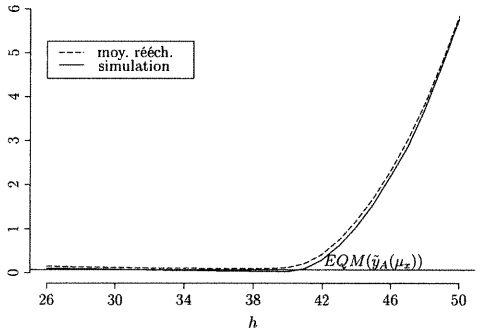
FIGURE 4.23. Choix de $\hat{h}$ (2.27) en fonction de $\hat{\beta}_{max}$ pour les simulations 6 et 6a

Les modèles utilisés aux simulations 7 et 7a sont très différents des précédents ; ils ont été étudiés pour tester notre estimateur adaptatif sur d'autres types de modèles. Le tableau 4.2 nous indique que h_{opt} vaut 40 et 34 pour les modèles où n_a est fixe et aléatoire respectivement. L'estimateur adaptatif est encore ici statistiquement plus efficace que l'estimateur par moindres carrés tronqués à robustesse maximale dans les deux cas. Les graphiques du critère et de l'efficacité sont disponibles à la figure 4.24.

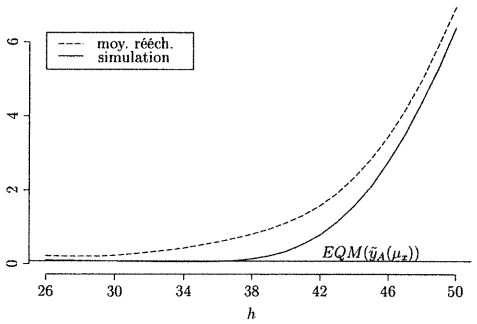
Finalement, les modèles des simulations 8 et 8a comportent des valeurs aberrantes pouvant se trouver à n'importe quelle ordonnée et ce pour toutes les valeurs de x comprises entre 0,5 et 4,5. Les résultats obtenus sont encore une fois très intéressants, alors que l'estimateur adaptatif surpasse $\hat{\beta}_{max}$ lorsque n_a est fixe et aléatoire (voir figure 4.25.) Dans ces cas, bien que les erreurs ϵ soient i.i.d. et distribuées selon une normale contaminée, il est préférable au sens de l'erreur quadratique moyenne d'omettre quelques observations. En effet, comme l'illustrent les figures C.14 et C.15 des pages 127 et 128, la variance des estimateurs par moindres carrés tronqués diminue à mesure que le nombre de données incluses h augmente, mais remonte à partir du moment où les observations provenant de la composante la plus variable de ϵ doivent être incluses. C'est pourquoi nous obtenons h_{opt}=38 et h_{opt}=36 pour les modèles où n_a est fixe et aléatoire respectivement.



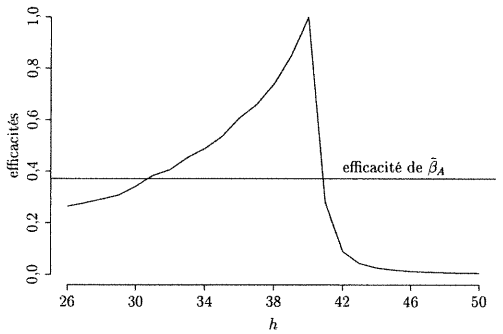
(a) Exemple simulation 7



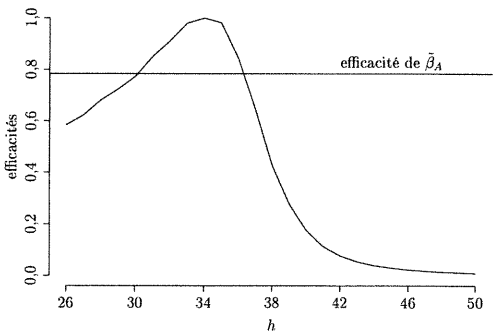
(b) EQM de l'espérance à la moyenne (simulation 7)



(c) EQM de l'espérance à la moyenne (simulation 7a)

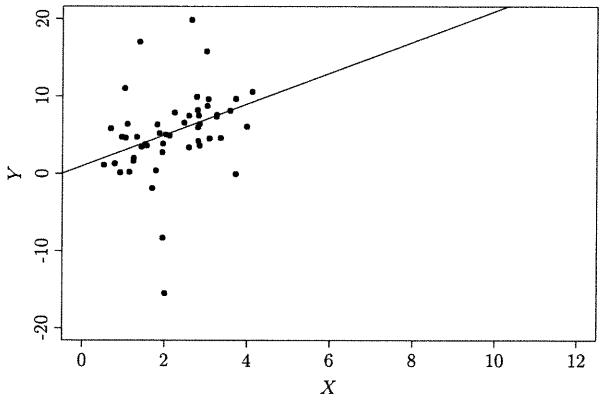


(d) Efficacités (simulation 7)

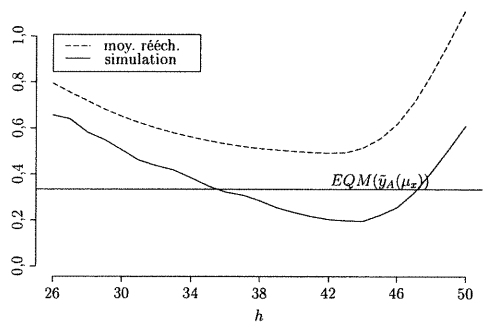


(e) Efficacités (simulation 7a)

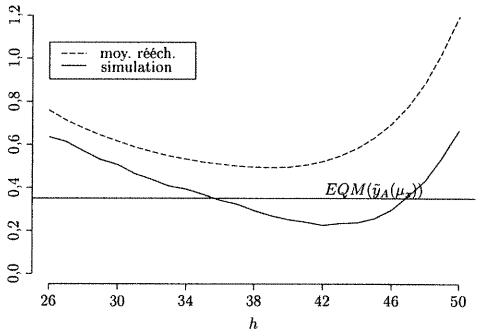
FIGURE 4.24. Simulations 7 et 7a



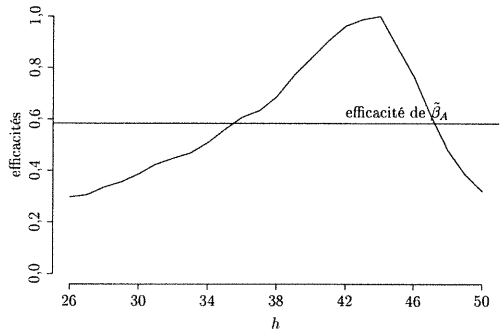
(a) Exemple simulation 8



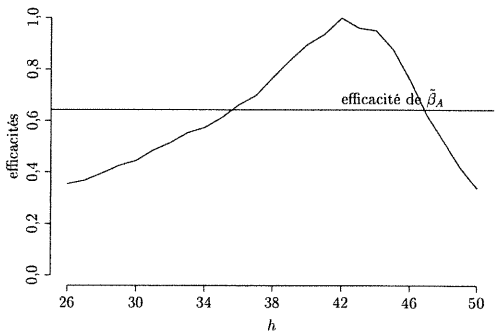
(b) EQM de l'espérance à la moyenne (simulation 8)



(c) EQM de l'espérance à la moyenne (simulation 8a)



(d) Efficacités (simulation 8)



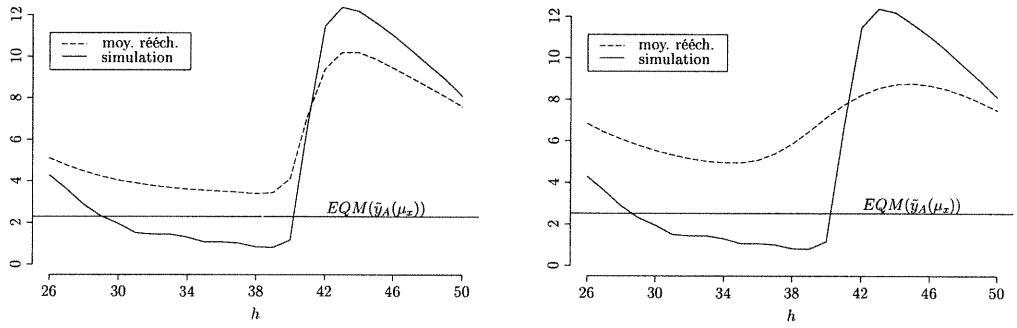
(e) Efficacités (simulation 8a)

FIGURE 4.25. Simulations 8 et 8a

4.5. DOIT-ON SAVOIR SI LE NOMBRE DE DONNÉES ABERRANTES EST FIXE OU ALÉATOIRE ?

Jusqu'à maintenant, nous avons toujours choisi une méthode de rééchantillonnage en sachant si n_a est fixe ou aléatoire. Nous allons dans ce qui suit générer des données selon les modèles des simulations 2 et 2a, puis appliquer les deux méthodes de rééchantillonnage dans chacun des cas.

Si n_a est fixe (simulation 2), voici les erreurs quadratiques moyennes de l'espérance à la moyenne pour les estimateurs par moindres carrés tronqués et pour les estimateurs adaptatifs, déterminés en considérant n_a fixe et aléatoire :



(a) Rééchantillonnage pour n_a fixe

(b) Rééchantillonnage pour n_a aléatoire

FIGURE 4.26. EQM de l'espérance à la moyenne pour n_a fixe (simulation 2)

On remarque que la moyenne des estimations par rééchantillonnage lorsque n_a est considéré fixe suit beaucoup mieux l'approximation par simulation de $EQM(\hat{y}_h(\mu_x))$. Mais, ce qui est plus important, les erreurs quadratiques moyennes de l'espérance en μ_x de $\tilde{\beta}_A$ sont très semblables dans les deux cas. La figure 4.27 nous montre les $\hat{h}$ choisis par les deux techniques de rééchantillonnage auxquels a été ajouté un bruit afin de discerner les points superposés.

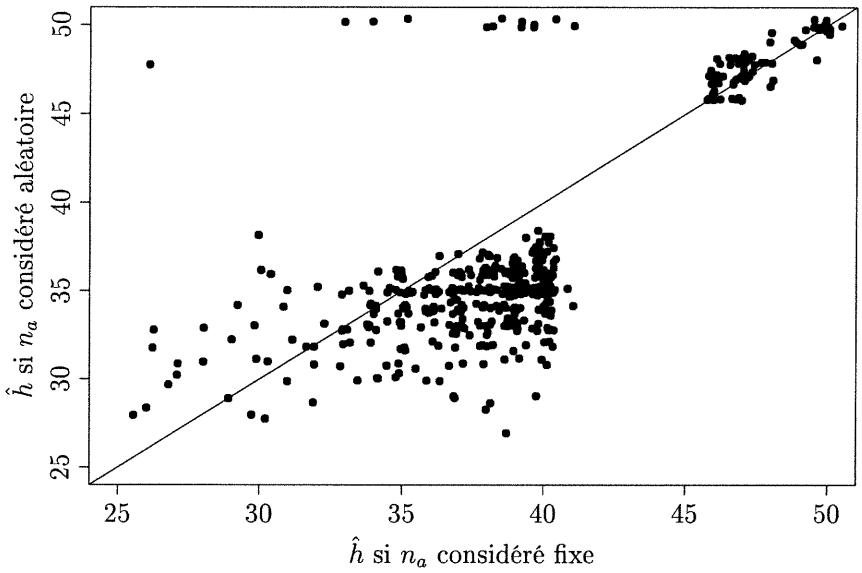
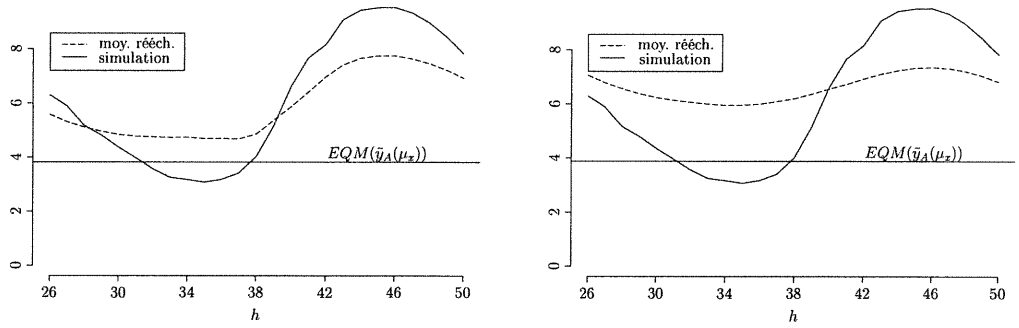


FIGURE 4.27. Indices choisis par les deux algorithmes pour n_a fixe (simulation 2)

On remarque que les résultats obtenus du rééchantillonnage pour n_a considéré aléatoire sont habituellement plus prudents que pour n_a considéré fixe. On peut aussi noter que quelques simulations ont donné $\hat{h} = 50$ pour n_a considéré aléatoire et $\hat{h} \leq 40$ pour n_a considéré fixe ; ce qui a certainement contribué à donner un EQM de l'espérance légèrement plus élevé dans le cas où n_a est considéré aléatoire.

Dans le cas où n_a est aléatoire (simulation 2a), on peut dire que les moyennes des estimations obtenues par les deux types de rééchantillonnage ne suivent pas très bien la courbe d'erreur quadratique moyenne de l'espérance au point μ_x (voir figure 4.28.) Alors que la performance de $\tilde{\beta}_A$ est encore ici semblable dans les deux cas (figure 4.28), et les choix de $\hat{h}$ sont plus prudents dans le cas où n_a est considéré aléatoire (voir figure 4.29.)



(a) Rééchantillonnage pour n_a fixe (b) Rééchantillonnage pour n_a aléatoire

FIGURE 4.28. EQM de l'espérance à la moyenne pour n_a aléatoire (simulation 2a)

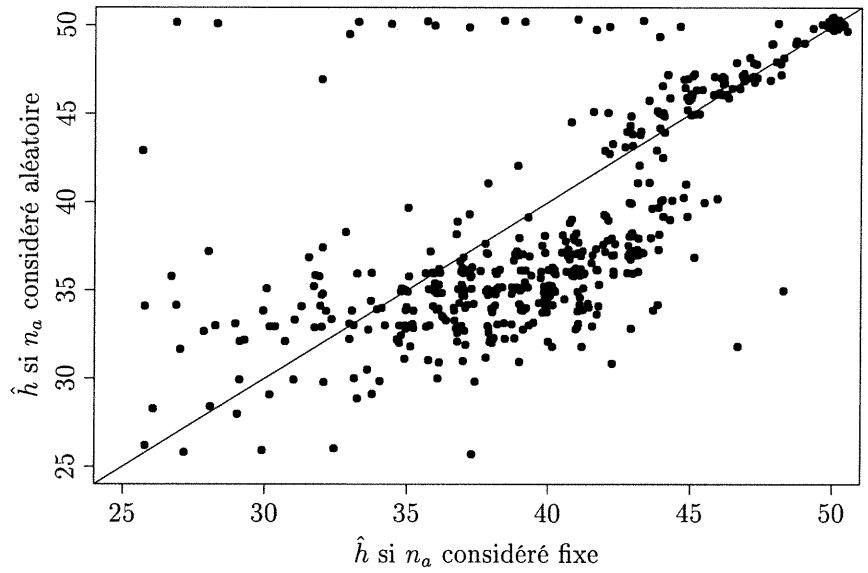


FIGURE 4.29. Indices choisis par les deux algorithmes pour n_a aléatoire (simulation 2a)

Les approximations d'erreurs quadratiques moyennes de l'espérance au point μ_x précédentes nous montrent que les deux méthodes de rééchantillonnage appliquées aux mêmes jeux de données donnent des résultats très semblables ; et

ce peu importe que n_a soit fixe ou aléatoire. Les différences observées à la section précédente entre les efficacités des estimateurs adaptatifs où n_a est considéré fixe et aléatoire sont donc dues à la différence entre les modèles, plutôt qu'aux algorithmes de rééchantillonnage. C'est donc dire que le rééchantillonnage des paires, normalement utilisé lorsque nous considérons que les n données sont i.i.d., pourrait être utilisé pour tous les jeux de données.

4.6. Discussion

Les résultats des simulations précédentes nous indiquent que l'estimateur $\tilde{\beta}_A$ est meilleur que l'estimateur par moindres carrés tronqués $\hat{\beta}_{max}$ dans une variété de situations. Les situations étudiées vont du modèle linéaire ne comprenant aucune donnée aberrante, à des modèles pour lesquels l'estimateur à robustesse maximale $\hat{\beta}_{max}$ éprouve de la difficulté à estimer β sans se rompre.

Nous avons ainsi observé des efficacités pour l'estimateur adaptatif (voir l'équation (4.4) à la page 57) entre 0,34 à la simulation 2 et 0,91 à la simulation 5a. Dans ce dernier cas $EQM(\hat{y}_{h_{opt}}(\mu_x))$ et $EQM(\tilde{y}_A(\mu_x))$ basés sur 500 simulations ne sont pas statistiquement différents. De plus, la polyefficacité de $\tilde{\beta}_A$ basée sur les 15 modèles étudiés (0,34) est supérieure à celle de tous les estimateurs par moindres carrés tronqués.

Puisque les estimations par rééchantillonnage du biais et surtout de l'erreur quadratique moyenne de l'espérance dépendent de $\hat{\beta}_{max}$ (voir (3.10)), la méthode utilisée pour déterminer par rééchantillonnage le paramètre $\hat{h}$ à utiliser pour l'estimateur adaptatif fonctionne bien si l'estimateur $\hat{\beta}_{max}$ ne brise pas. Nous avons illustré aux figures 4.13 et 4.23 que les choix de $\hat{h}$ inappropriés étaient habituellement causés par des estimations $\hat{\beta}_{max}$ éloignées de β. Mais il est important de noter que même pour ces modèles, l'estimateur adaptatif est plus performant que les moindres carrés tronqués de robustesse maximale.

Une autre cause d'erreur dont nous avons discutée est le fait que notre critère est évalué au centre des données, alors que les valeurs prédites à l'aide des différents estimateurs par moindres carrés tronqués peuvent être semblables pour de telles valeurs. Le critère $EQM(\hat{y}_h(x))$ (3.8), soit l'erreur quadratique moyenne de

l'espérance au point x a été retenu pour son équivariance pour tout x. Le choix arbitraire de l'évaluer en $x{=}\mu_x$ pourrait être remplacé par la moyenne du critère évalué en $x{=}x_{(n)}$ et en $x{=}x_{(1)}$, ce qui conserverait l'équivariance.

Finalement, nous avons comparé les deux estimations possibles du modèle de probabilité P qui ont été utilisées pour obtenir des rééchantillons. Nous avons constaté que les deux estimations donnent des résultats très semblables lorsque n_a est fixe et aléatoire, alors que le rééchantillonnage des paires (jusqu'ici utilisé lorsque nous considérons n_a aléatoire) donne des valeurs $\hat{h}$ habituellement inférieures à celles obtenues en estimant d'abord le groupe d'appartenance des données.

CONCLUSION

Afin de réduire l'influence de points aberrants en régression linéaire, des estimateurs robustes, tels que les moindres carrés tronqués, sont habituellement utilisés. L'estimateur par moindres carrés tronqués avec paramètre de troncature h correspond au vecteur $\hat{\beta}_h$ minimisant la somme des h plus petits résidus carrés. L'estimateur à robustesse maximale correspond au choix de h le plus petit possible, qui est tout juste supérieur à la moitié des données. Cet estimateur étant très variable, nous avons proposé d'estimer le paramètre h_{opt} correspondant au choix optimal, qui est aussi sans biais pour β, mais moins variable.

Nous avons choisi de minimiser l'erreur quadratique moyenne de l'espérance à la moyenne $\hat{y}_h(\mu_x) = \mu_x'\hat{\beta}_h$, estimé par l'estimateur par rééchantillonnage de l'erreur quadratique moyenne de $\bar{x}'\hat{\beta}_h^*$. L'estimation du modèle de probabilité ayant généré les données a pris pour sa part deux formes, dépendant de notre perception du modèle. Si nous considérions le nombre de données aberrantes (n_a) comme aléatoire, c'est-à-dire que les n données étaient i.i.d., nous avons échantillonné avec remise n données parmi les n originales. Si nous considérions n_a comme fixe, c'est-à-dire que les données provenaient de plusieurs modèles, nous avons tenté de reproduire cela dans notre façon de former les rééchantillons. Nous estimions tout d'abord l'appartenance des points aux différents modèles, pour ensuite rééchantillonner séparément dans les différents sous-groupes ainsi formés.

Nous nous sommes intéressés à une grande variété de modèles, allant de situations ne comportant aucune donnée aberrante, à d'autres où les moindres carrés tronqués à robustesse maximale avaient de la difficulté à ne pas rompre sous l'influence des données aberrantes. Pour tous les modèles étudiés, nous sommes arrivés à la conclusion que l'erreur quadratique moyenne de l'espérance en μ_x

de l'estimateur adaptatif, obtenu de 200 rééchantillons, est inférieure à celle des moindres carrés tronqués à robustesse maximale. De plus, la polyefficacité de $\tilde{\beta}_A$ calculée sur les 15 modèles étudiés est supérieure à celle de tous les estimateurs par moindres carrés tronqués. Le coût de la formation des rééchantillons et du calcul des estimés d'erreur quadratique moyenne est donc largement compensé par l'obtention d'un estimateur plus performant que celui habituellement utilisé pour limiter l'influence des données aberrantes.

Lors de la comparaison du modèle où n_a est fixe à celui où n_a est aléatoire, nous avons réalisé que l'efficacité de l'estimateur $\tilde{\beta}_A$ calculé en ayant cette information est supérieure lorsque n_a est aléatoire. Mais ceci est dû aux erreurs quadratiques moyennes qui sont supérieures dans ces cas ; un même écart entre $EQM(\hat{y}_{h_{opt}}(\mu_x))$ et $EQM(\tilde{y}_A(\mu_x))$ résultant en une plus grande efficacité. De plus, l'étude de la performance de la méthode de rééchantillonnage normalement utilisée pour n_a fixe lorsque n_a est aléatoire, et vice-versa nous a montré que les deux méthodes fonctionnent bien dans les deux situations.

Puisque les estimateurs par moindres carrés tronqués ne s'expriment pas sous forme analytique, l'étude du point de rupture de l'estimateur adaptatif $\tilde{\beta}_A$ est relativement complexe. Nous avons étudié théoriquement un estimateur adaptatif qui prenait la valeur de la moyenne ou de la médiane d'un échantillon univarié selon que l'estimation par rééchantillonnage de l'erreur quadratique moyenne de $\bar{x}$ ou de $med(x)$ était inférieure. Les résultats obtenus montrent que cet estimateur est plus robuste que la moyenne, mais moins robuste que la médiane. Bien que nous n'ayons pas considéré de points aberrants arbitraires, ou du moins très éloignés des autres données, les graphiques illustrant les biais carrés à l'annexe C montrent que l'estimateur adaptatif $\tilde{\beta}_A$ est plus robuste que l'estimateur par moindres carrés.

Nous n'avons considéré dans ce mémoire que le cas le plus simple de régression linéaire, la régression linéaire simple. Dans ce cas, le graphique des données permet d'identifier visuellement les données aberrantes, ce qui permet de choisir un paramètre h égal au nombre de données suivant la relation linéaire. L'estimateur adaptatif peut être utile dans ce contexte si un grand nombre de jeux de données

doivent être étudiés de façon automatisée. Mais plus important, la généralisation de la méthode à plusieurs variables explicatives ne devrait pas poser de problème majeur, si ce n'est de l'augmentation du temps de calcul nécessaire pour estimer $\hat{\beta}_h$. Dans de tels contextes, l'estimateur adaptatif pourrait pallier à l'absence de représentation visuelle complète des données.

Annexe A

ÉTUDE DU POINT DE RUPTURE DE L'ESTIMATEUR ADAPTATIF

Commandes Mathematica pour obtenir les tableaux A.1, A.2, et A.3 :

Coefficient en Y^2 de $\mathbb{E}_{\hat{P}}\left[(med(Z^*) - med(Z))^2\right]$ qui est la somme des *aber* p_j correspondant aux données aberrantes.

```
med[n_, aber_]:=Sum[Sum[Binomial[n,k]*( ((j-1)/n)^k * ((n-j+1)/n)^(n-k) -
(j/n)^k * ((n-j)/n)^(n-k)),{k,0,(n-1)/2}],{j,n-aber+1,n}]
```

Coefficient en Y^2 de $\mathbb{E}_{\hat{P}}\left[\left(\bar{Z}^* - med(Z)\right)^2\right]$ lorsque *aber* données sont aberrantes (on suppose sans perte de généralité qu'il s'agit des données $\{n-aber+1, \ldots, n\}$) :

$$\begin{aligned}
\mathbb{E}_{\hat{P}}\left[\left(\bar{Z}^* - med(Z)\right)^2\right] &= Var_{\hat{P}}(\bar{Z}^*) + \left[\mathbb{E}_{\hat{P}}\left[\bar{Z}^* - med(Z)\right]\right]^2 \\
&= \frac{1}{n^2}\left\{\sum_{j=1}^{n-aber}\left[X_{(j)} - \bar{Z}\right]^2 + aber\left[Y - \bar{Z}\right]^2\right\} + \left[\bar{Z} - med(Z)\right]^2 \\
&= \frac{1}{n^2}\sum_{j=1}^{n-aber}\left[X_{(j)} - \frac{\left(\sum_{k=1}^{n-aber} X_{(k)} + aberY\right)}{n}\right]^2 + \\
&\quad \frac{aber}{n^2}\left[Y - \frac{\left(\sum_{k=1}^{n-aber} X_{(k)} + aberY\right)}{n}\right]^2 + \\
&\quad \left[\frac{\left(\sum_{k=1}^{n-aber} X_{(k)} + aberY\right)}{n} - med(Z)\right]^2
\end{aligned}$$

```
moy[x_, n_, aber_]:=
(1/n^2)*(Sum[(x[j]- (Sum[x[k],{k,1,n-aber}]+aber*Y)/n)^2,{j,1,n-aber}] +
aber*(Y-(Sum[x[k],{k,1,n-aber}]+aber*Y)/n)^2)+
((Sum[x[k],{k,1,n-aber}]+aber*Y)/n - med)^2
```

Manipulations pour obtenir les tableaux.

```
tablemoy=Table[N[Coefficient[moy[x, i, j],Y^2]], {i,3,21,2}, {j,1,i/2}]
tablemed=Table[N[med[i,j]], {i,3,21,2}, {j,1,i/2}]

TableForm[tablemoy]
TableForm[tablemed]

medsiT=TableForm[Table[N[Coefficient[moy[x, i, j],Y^2]]>N[med[i,j]],
{i,3,21,2}, {j,1,i/2}]]
```

	nombre de données aberrantes									
n	1	2	3	4	5	6	7	8	9	10
3	0,259259									
5	0,05792	0,31744								
7	0,01015	0,108274	0,3469							
9	0,00144928	0,0303733	0,144846	0,365507						
11	0,000174092	0,00720712	0,0512466	0,172747	0,378631					
13	0,0000180257	0,001479	0,0157041	0,0706523	0,194924	0,38853				
15	1,63896 e-6	0,000267106	0,00423975	0,0254504	0,0882316	0,213103	0,396341			
17	1,32785 e-7	0,0000430482	0,00102225	0,00818641	0,0356237	0,104068	0,228361	0,402709		
19	9,69902 e-9	6,26154 e-6	0,000222569	0,00237769	0,0129774	0,0457901	0,118351	0,241407	0,40803	
21	6,44863 e-10	8,29675 e-7	0,0000441621	0,000629321	0,00430569	0,0183302	0,055723	0,13128	0,25273	0,412563

TABLEAU A.1. Coefficient en Y^2 de $\mathbb{E}_{\hat{P}}\left[(med(Z^*) - med(Z))^2\right]$

	nombre de données aberrantes									
n	1	2	3	4	5	6	7	8	9	10
3	0,185185									
5	0,072	0,208								
7	0,0379009	0,110787	0,218659							
9	0,0233196	0,0685871	0,135802	0,224966						
11	0,0157776	0,0465815	0,0924117	0,153268	0,229151					
13	0,0113792	0,0336823	0,0669094	0,111061	0,166136	0,232135				
15	0,00859259	0,0254815	0,0506667	0,0841481	0,125926	0,176	0,23437			
17	0,00671687	0,0199471	0,0396906	0,0659475	0,0987177	0,138001	0,183798	0,236108		
19	0,00539437	0,0160373	0,0319289	0,053069	0,0794576	0,111095	0,147981	0,190115	0,237498	
21	0,00442717	0,0131735	0,0262391	0,0436238	0,0653277	0,0913508	0,121693	0,156355	0,195335	0,238635

TABLEAU A.2. Coefficient en Y^2 de $\mathbb{E}_{\hat{P}}\left[\left(\bar{Z}^* - med(Z)\right)^2\right]$

	nombre de données aberrantes									
n	1	2	3	4	5	6	7	8	9	10
3	Faux									
5	Vrai	Faux								
7	Vrai	Vrai	Faux							
9	Vrai	Vrai	Faux	Faux						
11	Vrai	Vrai	Vrai	Faux	Faux					
13	Vrai	Vrai	Vrai	Vrai	Faux	Faux				
15	Vrai	Vrai	Vrai	Vrai	Vrai	Faux	Faux			
17	Vrai	Vrai	Vrai	Vrai	Vrai	Vrai	Faux	Faux		
19	Vrai	Vrai	Vrai	Vrai	Vrai	Vrai	Vrai	Faux	Faux	
21	Vrai	Vrai	Vrai	Vrai	Vrai	Vrai	Vrai	Vrai	Faux	Faux

TABLEAU A.3. $\tilde{Z} = med(Z)$?

Annexe B

PROGRAMMES S-PLUS

Afin d'éviter les boucles dans la programmation, le programme *prog.1* initialise les différents paramètres, charge toutes les fonctions S-Plus nécessaires contenues dans le fichier *funsplus*, et appelle la fonction *simulation* qui génère deux jeux de données et effectue tous les calculs à partir de ceux-ci. Par la suite, 249 réalisations du programme *prog.2* appelant la fonction *simulation* nous fourniront les résultats pour 498 autres jeux de données. La fonction *prog.3* est enfin utilisée pour résumer et illustrer les résultats.

Script UNIX

```
#!/bin/sh

Splus < prog.1 > Out/prog.1.out 2>Out/prog.1.warn
Splus < prog.2 > Out/prog.2.1.out 2>Out/prog.2.1.warn
Splus < prog.2 > Out/prog.2.2.out 2>Out/prog.2.2.warn
Splus < prog.2 > Out/prog.2.3.out 2>Out/prog.2.3.warn
Splus < prog.2 > Out/prog.2.4.out 2>Out/prog.2.4.warn
.
.
.
Splus < prog.2 > Out/prog.2.247.out 2>Out/prog.2.247.warn
Splus < prog.2 > Out/prog.2.248.out 2>Out/prog.2.248.warn
Splus < prog.2 > Out/prog.2.249.out 2>Out/prog.2.249.warn
Splus TRUNC_AUDIT 100
Splus < prog.3 > Out/prog.3.out 2>Out/prog.3.warn
```

prog.1

```
date()

# J'initialise les différents paramètres:
seed_c(53, 24,  1, 56, 62,  2, 27, 40, 52 ,13, 38,  3)
```

```
ncas_50
maxbreak_ floor((ncas+2+1)/2)
nbh_ncas - maxbreak +1
outlier_10
ord_1
slope_2
ordtrans_ord
slopetrans_slope
sigma_2.0
sigmaout_0.5   #0.5
muxout_6
muyout_-1
xmin_0.5
xmax_4.5
mux_((ncas-outlier)*(xmin + (xmax-xmin)/2) + outlier*muxout)/ncas
sigmabruit_0.005  # je l'ai diminue...
ncover_array(seq(maxbreak,ncas))
nsimul_2 #10
nboot_200  #200
fixal_"fixe"   # fixe ou aléatoire

source("./funsplus")
temp_simulation(seed,ncas,maxbreak,outlier,slope,ord,sigma,sigmaout,
                sigmabruit,nsimul,nboot,fixal,muxout,muyout,xmin,xmax,nbh)
seed_temp$oldseed

temp2_array(unlist(temp$resul, recursive=F))

jj_1; kk_2; ll_3
for (i in 2:nsimul){
        jj_ c(jj, 1 + (i-1)*3)
        kk_c(kk, jj[i]+1)
        ll=c(ll, jj[i]+2)
}


eqmsim_temp2[jj]
eqmarray_array(unlist(eqmsim), dim=c(nbh,11,nsimul),
  dimnames=list(seq(maxbreak,ncas), c("NCOVER","biais02","biais12",
  "var0","var1","eqm0","eqm1","eqm01","crit2","crit3","crit4"),
  seq(1,nsimul)))
meaneqm_apply(eqmarray,c(1,2),mean)

########################################

adaptsim_temp2[kk]
adaptarray_array(unlist(adaptsim), dim=c(15,nsimul),
  dimnames=list(c("e0", "b0e0", "b1e0", "e1", "b0e1", "b1e1",
  "c2", "b0c2", "b1c2", "c3", "b0c3", "b1c3", "c4", "b0c4", "b1c4"),
   seq(1,nsimul)))
```

```
#######################################

betahatsim_temp2[ll]
betaarray_array(unlist(betahatsim), dim=c(2,nbh,nsimul),
  dimnames=list(c("beta0" ,"beta1"), seq(maxbreak,ncas), seq(1,nsimul)))

seed
date()
```

prog.2

```
date()
seed

temp_simulation(seed,ncas,maxbreak,outlier,slope,ord,sigma,sigmaout,
                sigmabruit,nsimul,nboot,fixal,muxout,muyout,xmin,xmax,nbh)
seed_temp$oldseed

temp2_array(unlist(temp$resul, recursive=F))

dimension1_dim(adaptarray)

eqmsim_temp2[jj]
eqmarraytemp_array(unlist(eqmsim), dim=c(nbh,11,nsimul),
  dimnames=list(seq(maxbreak,ncas), c("NCOVER","biais02","biais12",
  "var0","var1","eqm0","eqm1","eqm01","crit2","crit3","crit4"),
  seq(1,nsimul)))
meaneqmtemp_apply(eqmarraytemp,c(1,2),mean)

meaneqm_meanmat(meaneqm, meaneqmtemp, dimension1[2]%/%nsimul)

#######################################


adaptsim_temp2[kk]
adaptarraytemp_array(unlist(adaptsim), dim=c(15,dimension1[2]+nsimul),
  dimnames=list(c("e0", "b0e0", "b1e0", "e1", "b0e1", "b1e1",
  "c2", "b0c2", "b1c2", "c3", "b0c3", "b1c3", "c4", "b0c4", "b1c4"),
   seq(1,dimension1[2]+nsimul)))

for (i in 1:dimension1[2]){
        adaptarraytemp[,i]_adaptarray[,i]
}
adaptarray_adaptarraytemp


#######################################

betahatsim_temp2[ll]
betaarraytemp_array(unlist(betahatsim),
```

```
        dim=c(2, nbh, dimension1[2]+nsimul),
        dimnames=list(c("beta0" ,"beta1"), seq(maxbreak,ncas), seq(1,nsimul)))
for (i in 1:dimension1[2]){
        betaarraytemp[,,i]_betaarray[,,i]
}
betaarray_betaarraytemp

dput(meaneqm, file="./Test/meaneqm")
dput(adaptarray, file="./Test/adaptarray")
dput(betaarray, file="./Test/betaarray")


seed
date()
```

prog.3

```
eqmbeta_mesuresimv2(betaarray,ordtrans,slopetrans,maxbreak,ncas,nbh,mux)
eqmadapt_mesureadaptv2(adaptarray,ordtrans,slopetrans)

#graphmoyv2(meaneqm,eqmbeta,eqmadapt,"./Résultats/moyenne.ps",maxbreak,
#       ncas,nbh)
#funhistv2(adaptarray,"./Résultats/hist.ps",maxbreak,ncas)

graph(meaneqm,eqmbeta,eqmadapt,"./Résultats/graph.ps",maxbreak,
        ncas,nbh)
```

funsplus

```
simulation_function(seed, ncas, maxbreak, outlier, slope, ord, sigma,
                sigmaout, sigmabruit, nsimul, nboot, fixal, muxout, muyout,
                xmin, xmax, nbh){

        .Random.seed<<-seed

        data_funsimul(ncas, outlier, slope, ord, sigma, sigmaout, nsimul,
                fixal, muxout, muyout, xmin, xmax)
        resul_lapply(data,funltsboot,nboot,ncas,ncover,maxbreak,sigmabruit,
                outlier,fixal,nbh)

        oldseed<-.Random.seed

        return(list(oldseed=oldseed, resul=resul))
}

meanmat_function(matpasse, matmaint, simpasse){
        matmoyenne_(simpasse*matpasse + matmaint)/(simpasse + 1)
        return(matmoyenne)
}

simultest_function(i, slope, ord, sigma, sigmaout, ncas, muxout, muyout,
```

```
                xmin, xmax){

                test1_NULL
                if(i==0){
                        test1[1]_ xmin + (xmax-xmin)*runif(1)
                        test1[2]_ (ord + slope*test1[1]) + rnorm(1,mean=0,sd=sigma)
                }
                else {
                        test1[1]_ muxout + rnorm(1, mean=0, sd=sigmaout)
                        test1[2]_ muyout + rnorm(1, mean=0, sd=sigmaout)
                }
        return(test1)
}

funsimul_function(ncas, outlier, slope, ord, sigma, sigmaout, nsimul, fixal,
                muxout, muyout, xmin, xmax){
        if(fixal=="fixe") loutlier_rep(outlier,nsimul) else
                loutlier_rbinom(nsimul, ncas, outlier/ncas)
        aber_NULL
        for (j in 1:nsimul){
                aber_c(aber,rep(0,ncas-loutlier[j]),rep(1,loutlier[j]))
                }
        aber_array(aber)
        datatest_t(apply(aber,1,simultest,slope,ord,sigma,sigmaout,ncas,
                muxout,muyout,xmin,xmax))
        # la commande suivante place tous les chiffres entre ""
        datatestsplit_split(data.frame(datatest),(seq(1,(ncas*nsimul))-1)%/%ncas+1)
        return(datatestsplit)
}

funreg_function(i, datain){
        out_ltsreg(datain[,2]~ datain[,1], quan=i)$coefficients
        return(out)
}

# enlever maxbreak
funlts_function(data, maxbreak, ncover){
        betalts_apply(ncover,1,funreg,data)
        return(betalts)
}

mesuresv2_function(betaboot, betahat, maxbreak, ncas, meanx, nbh){

        m1_maxbreak-1

        eqm_array(dim=c(nbh,11))
        dimnames(eqm)_list(seq(maxbreak,ncas),c("NCOVER","biais02","biais12",
                "var0","var1","eqm0","eqm1","eqm01","crit2","crit3","crit4"))
        eqm[,"NCOVER"]_seq(maxbreak,ncas)
        for (i in maxbreak:ncas){
```

```
                eqm[i-m1,"biais02"]_
                (mean(betaboot[1,i-m1,])-betahat[1,1])**2
                eqm[i-m1,"biais12"]_
                (mean(betaboot[2,i-m1,])-betahat[2,1])**2
                eqm[i-m1,"var0"]_var(betaboot[1,i-m1,])
                eqm[i-m1,"var1"]_var(betaboot[2,i-m1,])
                eqm[i-m1,"eqm0"]_eqm[i-m1,"biais02"] + eqm[i-m1,"var0"]
                eqm[i-m1,"eqm1"]_eqm[i-m1,"biais12"] + eqm[i-m1,"var1"]
                eqm[i-m1,"eqm01"]_
                        mean((betaboot[1,i-m1,]-betahat[1,1])*
                        (betaboot[2,i-m1,]-betahat[2,1]))

                eqm[i-m1,"crit2"]_eqm[i-m1,"eqm0"] +
                        (meanx**2)*eqm[i-m1,"eqm1"] + 2*meanx*eqm[i-m1,"eqm01"]
                eqm[i-m1,"crit3"]_eqm[i-m1,"eqm0"] + eqm[i-m1,"eqm1"]
                eqm[i-m1,"crit4"]_eqm[i-m1,"eqm0"]*eqm[i-m1,"eqm1"] -
                        (eqm[i-m1,"eqm01"])**2
        }
        return(eqm)
}

funadaptv2_function(eqm, betahat, maxbreak){

        m1_maxbreak-1

        indicee0_eqm[,"NCOVER"][eqm[,"eqm0"]==min(eqm[,"eqm0"])]
        indicee1_eqm[,"NCOVER"][eqm[,"eqm1"]==min(eqm[,"eqm1"])]
        indicec2_eqm[,"NCOVER"][eqm[,"crit2"]==min(eqm[,"crit2"])]
        indicec3_eqm[,"NCOVER"][eqm[,"crit3"]==min(eqm[,"crit3"])]
        indicec4_eqm[,"NCOVER"][eqm[,"crit4"]==min(eqm[,"crit4"])]


        adapt_c(
        indicee0, betahat[1,indicee0-m1], betahat[2,indicee0-m1],
        indicee1, betahat[1,indicee1-m1], betahat[2,indicee1-m1],
        indicec2,  betahat[1,indicec2-m1],  betahat[2,indicec2-m1],
        indicec3,  betahat[1,indicec3-m1],  betahat[2,indicec3-m1],
        indicec4,  betahat[1,indicec4-m1],  betahat[2,indicec4-m1])

        return(adapt)
}

funltsboot_function(datain, nboot, ncas, ncover, maxbreak, sigmabruit,
                outlier, fixal, nbh){

        cat("Nouvelle simulation", "\n")

        #datain_data$"2"
        #plot(datain[,1], datain[,2])
        betahat_funlts(datain,maxbreak,ncover)
```

```
        #abline(betahat[1,1], betahat[2,1])

        if(fixal=="fixe"){
                # la commande suivante ne donne pas seulement un
                # chiffre, mais datain[, 2]
                sigmastar_ltsreg(datain[,2]~ datain[,1], quan=maxbreak)$scale
                # on pourrait envisager d'utiliser LMS au lieu de LTS26 pour ne pas
                # favoriser ce dernier...
                res_cbind(seq(1,ncas),
                        datain[,2] - (betahat[1,26-25] +
                        betahat[2,26-25]*datain[,1]))
                resaugmente_cbind(res,abs(res[,2]/sigmastar))
                outl_resaugmente[,1][resaugmente[,3]>2.5]
                cat("voici les pts aberrants selon sigmastar:", outl, "\n")
                cat(outl, "\n", file="./Out/aberrants", append=T)
                ok_resaugmente[,1][resaugmente[,3]<=2.5]
                indices_NULL
                i_1
                while(i<=nboot){
                        indices_c(indices,
                        sample(ok, length(ok), replace=T),
                        sample(outl, length(outl), replace=T))
                        i_i+1
                }
        } else indices_sample(seq(1,ncas),ncas*nboot,replace=T)
        bruit1_rnorm(ncas*nboot,mean=0,sd=sigmabruit)
        bruit2_rnorm(ncas*nboot,mean=0,sd=sigmabruit)
        xboot_datain[indices,1] + bruit1
        yboot_datain[indices,2] + bruit2
        # la ligne suivante est assez longue et elle cree aussi des ""
        databoot_split(data.frame(cbind(xboot,yboot)),
        (seq(1,ncas*nboot)-1)%/%ncas+1)
        # la ligne la plus longue...
        betaboot_lapply(databoot,funlts,maxbreak,ncover)

        betaboot2_array(unlist(betaboot), dim=c(2,nbh,nboot),
        dimnames=list(c("beta0","beta1"), seq(maxbreak,ncas), seq(1,nboot)))

        meanx_mean(datain[,1])
        eqm_mesuresv2(betaboot2, betahat, maxbreak, ncas, meanx, nbh)

        adapt_funadaptv2(eqm, betahat, maxbreak)

        return(list(eqm=eqm, adapt=adapt, betahat=betahat))
}

mesuresimv2_function(betaarray, ord, slope, maxbreak, ncas, nbh, mux){

        m1_maxbreak-1
        nsimulttl_dim(betaarray)[3]
```

```
        eqm_array(dim=c(nbh,17))

        dimnames(eqm)_list(seq(maxbreak,ncas),c("NCOVER","biais02","biais12",
                "var0","var1","eqm0","eqm1","eqm01","varbiais02",
                "varbiais12","varvar0","varvar1","vareqm0","vareqm1",
                "crit2","crit3","crit4"))
        eqm[,"NCOVER"]_seq(maxbreak,ncas)

        for (h in maxbreak:ncas){
                meanb0_mean(betaarray[1,h-m1,])
                meanb1_mean(betaarray[2,h-m1,])
                meanb02_mean(betaarray[1,h-m1,]**2)
                meanb12_mean(betaarray[2,h-m1,]**2)

                eqm[h-m1,"biais02"]_(mean(betaarray[1,h-m1,]) - ord)**2
                eqm[h-m1,"biais12"]_(mean(betaarray[2,h-m1,]) - slope)**2
                eqm[h-m1,"var0"]_var(betaarray[1,h-m1,])
                eqm[h-m1,"var1"]_var(betaarray[2,h-m1,])
                eqm[h-m1,"eqm0"]_1*eqm[h-m1,"biais02"] + 1*eqm[h-m1,"var0"]
                eqm[h-m1,"eqm1"]_1*eqm[h-m1,"biais12"] + 1*eqm[h-m1,"var1"]
                eqm[h-m1,"eqm01"]_mean((betaarray[1,h-m1,] - ord)*
                        (betaarray[2,h-m1,] - slope))

                eqm[h-m1,"varbiais02"]_0
                eqm[h-m1,"varbiais12"]_0
                eqm[h-m1,"varvar0"]_
                        sum((-2*meanb0*(betaarray[1,h-m1,] - meanb0) +
                        betaarray[1,h-m1,]**2 - meanb02)**2)/nsimulttl**2
                eqm[h-m1,"varvar1"]_
                        sum((-2*meanb1*(betaarray[2,h-m1,] - meanb1) +
                        betaarray[2,h-m1,]**2 - meanb12)**2)/nsimulttl**2
                eqm[h-m1,"vareqm0"]_var((betaarray[1,h-m1,]-ord)**2)/nsimulttl
                eqm[h-m1,"vareqm1"]_var((betaarray[2,h-m1,]-slope)**2)/nsimulttl

                eqm[h-m1,"crit2"]_eqm[h-m1,"eqm0"] + 2*(mux)*eqm[h-m1,"eqm01"]+
                ((mux)**2)*eqm[h-m1,"eqm1"]
                eqm[h-m1,"crit3"]_eqm[h-m1,"eqm0"] + eqm[h-m1,"eqm1"]
                eqm[h-m1,"crit4"]_eqm[h-m1,"eqm0"]*eqm[h-m1,"eqm1"] -
                        (eqm[h-m1,"eqm01"])**2
        }
        return(eqm)
}

# ajouter mux dans les variables utilisees
mesureadaptv2_function(adaptarray,ord,slope){

        nsimulttl_dim(adaptarray)[2]

        eqme0_array(dim=c(1,12))
```

```
eqme1_array(dim=c(1,12))
eqmc2_array(dim=c(1,12))
eqmc3_array(dim=c(1,12))
eqmc4_array(dim=c(1,12))

dimnames(eqme0)_list(1,c("biais02","biais12","var0","var1",
"eqm0","eqm1","eqm01","vareqm0","vareqm1","crit2","crit3","crit4"))
dimnames(eqme1)_list(1,c("biais02","biais12","var0","var1",
"eqm0","eqm1","eqm01","vareqm0","vareqm1","crit2","crit3","crit4"))
dimnames(eqmc2)_list(1,c("biais02","biais12","var0","var1",
"eqm0","eqm1","eqm01","vareqm0","vareqm1","crit2","crit3","crit4"))
dimnames(eqmc3)_list(1,c("biais02","biais12","var0","var1",
"eqm0","eqm1","eqm01","vareqm0","vareqm1","crit2","crit3","crit4"))
dimnames(eqmc4)_list(1,c("biais02","biais12","var0","var1",
"eqm0","eqm1","eqm01","vareqm0","vareqm1","crit2","crit3","crit4"))

#résultats pour EQM sur b0:
eqme0[1,"biais02"]_(mean(adaptarray["b0e0",])-ord)**2
eqme0[1,"biais12"]_(mean(adaptarray["b1e0",])-slope)**2
eqme0[1,"var0"]_var(adaptarray["b0e0",])
eqme0[1,"var1"]_var(adaptarray["b1e0",])
eqme0[1,"eqm0"]_eqme0[1,"biais02"] + eqme0[1,"var0"]
eqme0[1,"eqm1"]_eqme0[1,"biais12"] + eqme0[1,"var1"]
eqme0[1,"eqm01"]_mean((adaptarray["b0e0",]-ord)*
        (adaptarray["b1e0",]-slope))
eqme0[1,"vareqm0"]_var((adaptarray["b0e0",]-ord)**2)/nsimulttl
eqme0[1,"vareqm1"]_var((adaptarray["b1e0",]-slope)**2)/nsimulttl
eqme0[1,"crit2"]_eqme0[1,"eqm0"] + 2*mux*eqme0[1,"eqm01"]
        + (mux)**2 * eqme0[1,"eqm1"]
eqme0[1,"crit3"]_eqme0[1,"eqm0"] + eqme0[1,"eqm1"]
eqme0[1,"crit4"]_(eqme0[1,"eqm0"]*eqme0[1,"eqm1"]) -
        (eqme0[1,"eqm01"])**2
#résultats pour EQM sur b1:
eqme1[1,"biais02"]_(mean(adaptarray["b0e1",])-ord)**2
eqme1[1,"biais12"]_(mean(adaptarray["b1e1",])-slope)**2
eqme1[1,"var0"]_var(adaptarray["b0e1",])
eqme1[1,"var1"]_var(adaptarray["b1e1",])
eqme1[1,"eqm0"]_eqme1[1,"biais02"] + eqme1[1,"var0"]
eqme1[1,"eqm1"]_eqme1[1,"biais12"] + eqme1[1,"var1"]
eqme1[1,"eqm01"]_mean((adaptarray["b0e1",]-ord)*
        (adaptarray["b1e1",]-slope))
eqme1[1,"vareqm0"]_var((adaptarray["b0e1",]-ord)**2)/nsimulttl
  eqme1[1,"vareqm1"]_var((adaptarray["b1e1",]-slope)**2)/nsimulttl
eqme1[1,"crit2"]_eqme1[1,"eqm0"] + 2*mux*eqme1[1,"eqm01"]+
(mux)**2 * eqme1[1,"eqm1"]
eqme1[1,"crit3"]_eqme1[1,"eqm0"] + eqme1[1,"eqm1"]
eqme1[1,"crit4"]_(eqme1[1,"eqm0"]*eqme1[1,"eqm1"]) -
        (eqme1[1,"eqm01"])**2
#résultats pour critere2:
eqmc2[1,"biais02"]_(mean(adaptarray["b0c2",])-ord)**2
```

```
        eqmc2[1,"biais12"]_(mean(adaptarray["b1c2",])-slope)**2
        eqmc2[1,"var0"]_var(adaptarray["b0c2",])
        eqmc2[1,"var1"]_var(adaptarray["b1c2",])
        eqmc2[1,"eqm0"]_eqmc2[1,"biais02"] + eqmc2[1,"var0"]
        eqmc2[1,"eqm1"]_eqmc2[1,"biais12"] + eqmc2[1,"var1"]
        eqmc2[1,"eqm01"]_mean((adaptarray["b0c2",]-ord)*
                (adaptarray["b1c2",]-slope))
        eqmc2[1,"vareqm0"]_var((adaptarray["b0c2",]-ord)**2)/nsimulttl
        eqmc2[1,"vareqm1"]_var((adaptarray["b1c2",]-slope)**2)/nsimulttl
        eqmc2[1,"crit2"]_eqmc2[1,"eqm0"] + 2*mux*eqmc2[1,"eqm01"]+
        (mux)**2 * eqmc2[1,"eqm1"]
        eqmc2[1,"crit3"]_eqmc2[1,"eqm0"] + eqmc2[1,"eqm1"]
        eqmc2[1,"crit4"]_(eqmc2[1,"eqm0"]*eqmc2[1,"eqm1"]) -
                (eqmc2[1,"eqm01"])**2
        #résultats pour critere3:
        eqmc3[1,"biais02"]_(mean(adaptarray["b0c3",])-ord)**2
        eqmc3[1,"biais12"]_(mean(adaptarray["b1c3",])-slope)**2
        eqmc3[1,"var0"]_var(adaptarray["b0c3",])
        eqmc3[1,"var1"]_var(adaptarray["b1c3",])
        eqmc3[1,"eqm0"]_eqmc3[1,"biais02"] + eqmc3[1,"var0"]
        eqmc3[1,"eqm1"]_eqmc3[1,"biais12"] + eqmc3[1,"var1"]
        eqmc3[1,"eqm01"]_mean((adaptarray["b0c3",]-ord)*
                (adaptarray["b1c3",]-slope))
        eqmc3[1,"vareqm0"]_var((adaptarray["b0c3",]-ord)**2)/nsimulttl
        eqmc3[1,"vareqm1"]_var((adaptarray["b1c3",]-slope)**2)/nsimulttl
        eqmc3[1,"crit2"]_eqmc3[1,"eqm0"] + 2*mux*eqmc3[1,"eqm01"]
                + (mux)**2 * eqmc3[1,"eqm1"]
        eqmc3[1,"crit3"]_eqmc3[1,"eqm0"] + eqmc3[1,"eqm1"]
        eqmc3[1,"crit4"]_(eqmc3[1,"eqm0"]*eqmc3[1,"eqm1"]) -
                (eqmc3[1,"eqm01"])**2
        #résultats pour critere4:
        eqmc4[1,"biais02"]_(mean(adaptarray["b0c4",])-ord)**2
        eqmc4[1,"biais12"]_(mean(adaptarray["b1c4",])-slope)**2
        eqmc4[1,"var0"]_var(adaptarray["b0c4",])
        eqmc4[1,"var1"]_var(adaptarray["b1c4",])
        eqmc4[1,"eqm0"]_eqmc4[1,"biais02"] + eqmc4[1,"var0"]
        eqmc4[1,"eqm1"]_eqmc4[1,"biais12"] + eqmc4[1,"var1"]
        eqmc4[1,"eqm01"]_mean((adaptarray["b0c4",]-ord)*
                (adaptarray["b1c4",]-slope))
        eqmc4[1,"vareqm0"]_var((adaptarray["b0c4",]-ord)**2)/nsimulttl
        eqmc4[1,"vareqm1"]_var((adaptarray["b1c4",]-slope)**2)/nsimulttl
        eqmc4[1,"crit2"]_eqmc4[1,"eqm0"] + 2*mux*eqmc4[1,"eqm01"] +
        (mux)**2 * eqmc4[1,"eqm1"]
        eqmc4[1,"crit3"]_eqmc4[1,"eqm0"] + eqmc4[1,"eqm1"]
        eqmc4[1,"crit4"]_(eqmc4[1,"eqm0"]*eqmc4[1,"eqm1"]) -
                (eqmc4[1,"eqm01"])**2

        return(list(eqme0=eqme0, eqme1=eqme1, eqmc2=eqmc2,
                eqmc3=eqmc3, eqmc4=eqmc4))
}
```

```
sousgraphmoyv2_function(type, eqmbeta, meaneqm, eqmadapt, mesure, lab,
                titre, nbh){

        if(type=="ordinaire") {
                varmesure_paste("var",mesure, sep="")
                tsplot(eqmbeta[,mesure],
                eqmbeta[,mesure]-2*sqrt(eqmbeta[,varmesure]),
                eqmbeta[,mesure]+2*sqrt(eqmbeta[,varmesure]),
                meaneqm[,mesure], lty=c(1,2,2,3), axes=F, main=titre)
                abline(h=eqmadapt$eqmc2[1,mesure])
                if(mesure=="eqm0" | mesure=="eqm1"){
                        abline(h=eqmadapt$eqmc2[1,mesure]+
                        2*sqrt(eqmadapt$eqmc2[1,varmesure]), lty=2)
                        abline(h=eqmadapt$eqmc2[1,mesure]-
                        2*sqrt(eqmadapt$eqmc2[1,varmesure]), lty=2)
                }
                axis(1,at=seq(1,nbh+3,4), labels=lab); axis(2)
        }
        if(type=="nouveau"){
                varmesure_paste("var",mesure, sep="")
                tsplot(eqmbeta[,mesure],
                meaneqm[,mesure], lty=c(1,3), axes=F, main=titre)
                axis(1,at=seq(1,nbh+3,4), labels=lab); axis(2)
                abline(h=eqmadapt$eqmc2[1,mesure])
        }
}

graphmoyv2_function(meaneqm, eqmbeta, eqmadapt, maxbreak, ncas, nbh){

        #postscript(file=finfichier, setfont=ps.setfont.latin1)

        lab1_seq(maxbreak,ncas+3,4)

        par(mfrow=c(3,2))
        sousgraphmoyv2("ordinaire",eqmbeta,meaneqm,eqmadapt,"biais02",lab1,
                "biais carré b0",nbh)
        sousgraphmoyv2("ordinaire",eqmbeta,meaneqm,eqmadapt,"biais12",lab1,
                "biais carré b1",nbh)
        sousgraphmoyv2("ordinaire",eqmbeta,meaneqm,eqmadapt,"var0",lab1,
                "var b0",nbh)
        sousgraphmoyv2("ordinaire",eqmbeta,meaneqm,eqmadapt,"var1",lab1,
                "var b1",nbh)
        sousgraphmoyv2("ordinaire",eqmbeta,meaneqm,eqmadapt,"eqm0",lab1,
                "eqm b0",nbh)
        sousgraphmoyv2("ordinaire",eqmbeta,meaneqm,eqmadapt,"eqm1",lab1,
                "eqm b1",nbh)

        par(mfrow=c(2,2))
```

```
        sousgraphmoyv2("nouveau",eqmbeta,meaneqm,eqmadapt,"eqm01",lab1,
                "eqm01",nbh)
        sousgraphmoyv2("nouveau",eqmbeta,meaneqm,eqmadapt,"crit2",lab1,
                "critere 2",nbh)
        sousgraphmoyv2("nouveau",eqmbeta,meaneqm,eqmadapt,"crit3",lab1,
                "critere 3",nbh)
        sousgraphmoyv2("nouveau",eqmbeta,meaneqm,eqmadapt,"crit4",lab1,
                "critere 4",nbh)
        #dev.off()
}

funhistv2_function(adaptarray, maxbreak, ncas){

        #postscript(file=fintitre, setfont=ps.setfont.latin1)

        ymax_max(c(hist(adaptarray["e0",], breaks=seq(maxbreak-0.5,ncas+0.5),
        plot=F)$counts,
        hist(adaptarray["e1",], breaks=seq(maxbreak-0.5,ncas+0.5), plot=F)$counts,
        hist(adaptarray["c2",], breaks=seq(maxbreak-0.5,ncas+0.5), plot=F)$counts,
        hist(adaptarray["c3",], breaks=seq(maxbreak-0.5,ncas+0.5), plot=F)$counts,
        hist(adaptarray["c4",], breaks=seq(maxbreak-0.5,ncas+0.5), plot=F)$counts ))

        par(mfrow=c(2,2))
        hist(adaptarray["e0",], breaks=seq(maxbreak-0.5,ncas+0.5),
        ylim=c(0,ymax), xlab="indice", main="indices choisis par EQM sur b0")
        hist(adaptarray["e1",], breaks=seq(maxbreak-0.5,ncas+0.5),
        ylim=c(0,ymax), xlab="indice", main="indices choisis par EQM sur b1")

        plot(adaptarray["e0",]+rnorm(dim(adaptarray)[2],0,0.1),
        adaptarray["e1",],pch="*", xlim=c(maxbreak,ncas),
        ylim=c(maxbreak,ncas), xlab="indices choisis sur beta0",
        ylab="indices choisis sur beta1",main="Indices choisis par
        EQM");
        abline(0,1)

        par(mfrow=c(2,2))
        hist(adaptarray["c2",], breaks=seq(maxbreak-0.5,ncas+0.5),
        ylim=c(0,ymax), xlab="indice", main="indices choisis par critere 2")
        hist(adaptarray["c3",], breaks=seq(maxbreak-0.5,ncas+0.5),
        ylim=c(0,ymax), xlab="indice", main="indices choisis par critere 3")
        hist(adaptarray["c4",], breaks=seq(maxbreak-0.5,ncas+0.5),
        ylim=c(0,ymax), xlab="indice", main="indices choisis par critere 4")

        #dev.off()
}

graph_function(meaneqm, eqmbeta, eqmadapt, finfichier, maxbreak, ncas,
                nbh){

        postscript(file=finfichier, setfont=ps.setfont.latin1)
```

```
        graphmoyv2(meaneqm, eqmbeta, eqmadapt, maxbreak, ncas, nbh)
        funhistv2(adaptarray, maxbreak, ncas)

        dev.off()
}
```

Annexe C

BIAIS, VARIANCES ET EQM

Les courbes pleines des graphiques de cette annexe correspondent aux approximations par simulations basées sur 500 jeux de données du biais carré, de la variance, et de l'erreur quadratique moyenne des deux composantes des différents estimateurs par moindres carrés tronqués ($\hat{\beta}_{h,0}$ et $\hat{\beta}_{h,1}$.) Puisque les *Nsimul* valeurs de $\hat{\beta}_{h,k}$, $h \in \{max, \ldots, n\}, k \in \{0,1\}$ ont été conservées, des estimés de la variance de $var(\hat{\beta}_{h,k})$ et de $EQM(\hat{\beta}_{h,k})$ furent calculés.

La méthode delta non paramétrique (Davidson & Hinkley, 1997) fut utilisée pour estimer la variance de $var(\hat{\beta}_{h,k}), h \in \{max, \ldots, n\}, k \in \{0,1\}$:

$$\begin{aligned} Var^{DN}\left(var(\hat{\beta}_{h,k})\right) &= Var^{DN}\left(\mathbb{E}_F\left[\left(\hat{\beta}_{h,k} - \mathbb{E}_F\left[\hat{\beta}_{h,k}\right]\right)^2\right]\right) \\ &= Var^{DN}\left(\mathbb{E}_F\left[\hat{\beta}_{h,k}^2\right] - \left(\mathbb{E}_F\left[\hat{\beta}_{h,k}\right]\right)^2\right). \end{aligned}$$

Posons $t_1(F) = \mathbb{E}_F\left[\hat{\beta}_{h,k}\right]$ et $t_2(F) = \mathbb{E}_F\left[\hat{\beta}_{h,k}^2\right]$,
donc $Var^{DN}\left(var(\hat{\beta}_{h,k})\right) = Var^{DN}\left(t_2(F) - (t_1(F))^2\right)$.

Théorème C.1 (exercice 10, Davidson & Hinkley (1997), p. 63). *Si* $t(F) = \int a(x)dF(x)$, *alors la fonction d'influence de* T *prend la forme suivante* $L_{t(F)}(y) = a(y) - t(F)$.

Posons $t_r(F) = \int x^r dF(x)$, donc $L_{t_r(F)}(y) = y^r - t_r(F)$.

Théorème C.2 (Davidson & Hinkley (1997), p. 48). *Si* $t(F) = g\left(t_1(F), \ldots, t_m(F)\right)$, *alors* $L_{t(F)}(y) = \sum_{i=1}^{m} \frac{\partial g}{\partial t_i(F)} L_{t_i(F)}(y)$.

Posons $t(F) = g\left(t_1(F), t_2(F)\right) = t_2(F) - (t_1(F))^2$,

donc $L_{t(F)}(y) = \frac{\partial g}{\partial t_1(F)} L_{t_1(F)}(y) + \frac{\partial g}{\partial t_2(F)} L_{t_2(F)}(y)$

$= -2t_1(F)(y - t_1(F)) + y^2 - t_2(F).$

$\Rightarrow l(y) \equiv L_{t(\hat{F})}(y)$

$= -2t_1(\hat{F})(y - t_1(\hat{F})) + y^2 - t_2(\hat{F})$

$\Rightarrow l_j \equiv l(y_j)$

$= -2t_1(\hat{F})(y_j - t_1(\hat{F})) + y_j^2 - t_2(\hat{F})$

Finalement,

$Var^{DN}\left(t_2(F) - (t_1(F))^2\right)$

$= \frac{1}{(Nsimul)^2} \sum_{j=1}^{Nsimul} l_j^2$ (éq 2.36, Davidson & Hinkley (1997), p. 47)

$= \frac{1}{(Nsimul)^2} \sum_{j=1}^{Nsimul} \left(-2t_1(\hat{F})(y_j - t_1(\hat{F})) + y_j^2 - t_2(\hat{F})\right)^2$

$= \frac{1}{(Nsimul)^2} \sum_{j=1}^{Nsimul} \left(-2\hat{\beta}_{h,k}(\cdot)\left(\hat{\beta}_{h,k}(j) - \hat{\beta}_{h,k}(\cdot)\right) + \left(\hat{\beta}_{h,k}(j)\right)^2 - \hat{\beta}_{h,k}^2(\cdot)\right)^2$

où $\hat{\beta}_{h,k}(\cdot) = \frac{1}{Nsimul} \sum_{i=1}^{Nsimul} \hat{\beta}_{h,k}(i)$ et $\hat{\beta}_{h,k}^2(\cdot) = \frac{1}{Nsimul} \sum_{i=1}^{Nsimul} \left(\hat{\beta}_{h,k}(i)\right)^2$.

La variance de $EQM(\hat{\beta}_{h,k})$ est pour sa part estimée de la façon suivante :

$$\begin{aligned}
\widehat{Var}\left(EQM(\hat{\beta}_{h,k})\right) &= \widehat{Var}\left(\frac{1}{Nsimul} \sum_{j=1}^{Nsimul} \left(\hat{\beta}_{h,k}(j) - \beta_k\right)^2\right) \\
&= \frac{1}{(Nsimul)^2} \sum_{j=1}^{Nsimul} \widehat{Var}\left(\left(\hat{\beta}_{h,k}(j) - \beta_k\right)^2\right) \\
&= \frac{1}{Nsimul} \widehat{Var}\left(\left(\hat{\beta}_{h,k}(1) - \beta_k\right)^2\right).
\end{aligned}$$

À l'aide des deux formules précédentes, des intervalles de $\pm$ 2 écarts type centrés sur les approximations par simulations sont illustrés.

Les droites horizontales sont les approximations par simulations du biais carré, de la variance, et de l'erreur quadratique moyenne des deux composantes de l'estimateur adaptatif $\tilde{\beta}_A$. Des intervalles de $\pm$ 2 écarts type centrés sur les approximations des erreurs quadratiques moyennes sont illustrés.

Finalement, les courbes pointillées illustrent la moyenne des 500 estimations par rééchantillonnage des biais carrés, variances et erreurs quadratiques moyennes

des estimateurs par moindres carrés tronqués. Étant donné le nombre très élevé de $\hat{\beta}^*_{h,k}(i)$, les estimations de variance n'ont pas été faites.

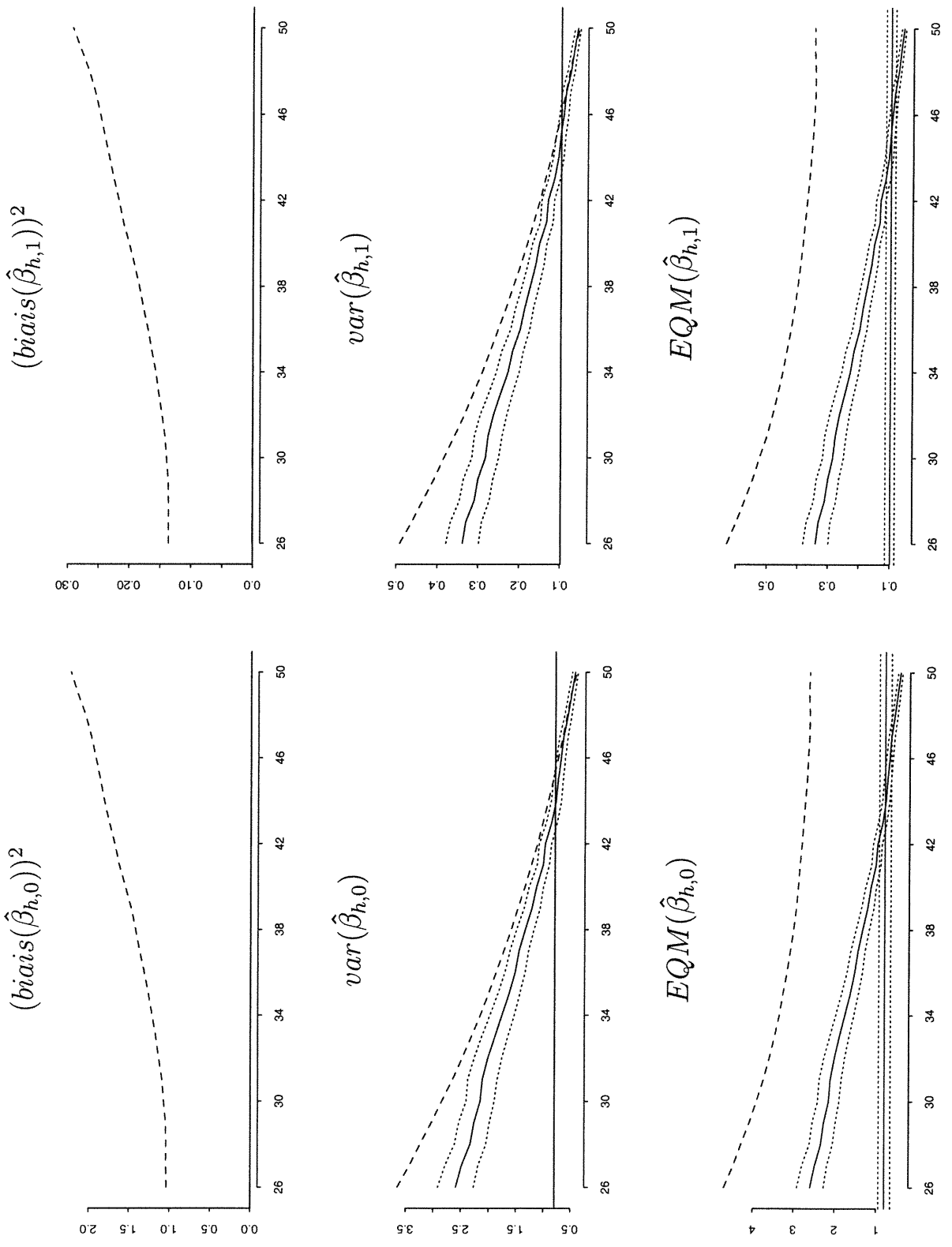


FIGURE C.1. Simulation 1

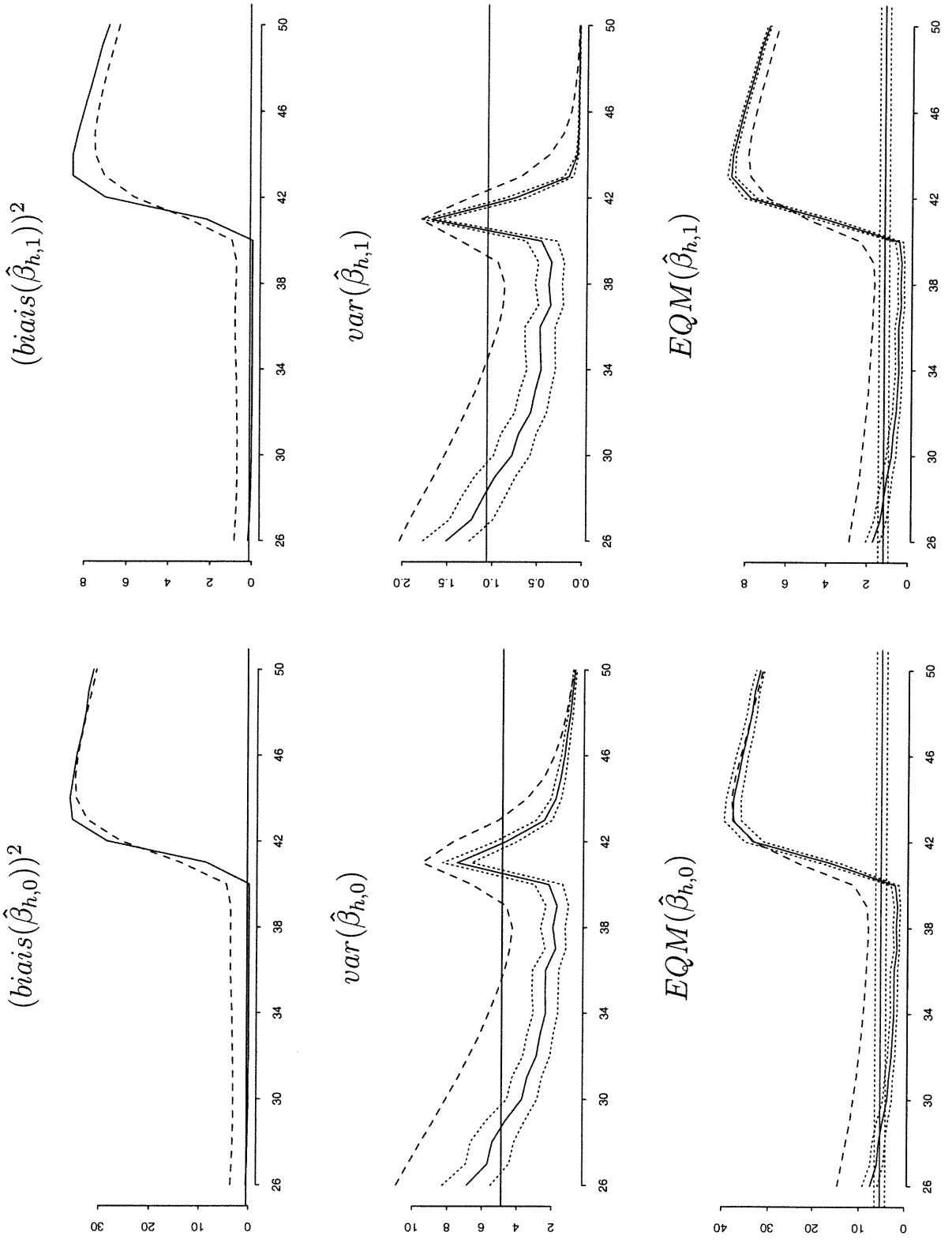


FIGURE C.2. Simulation 2

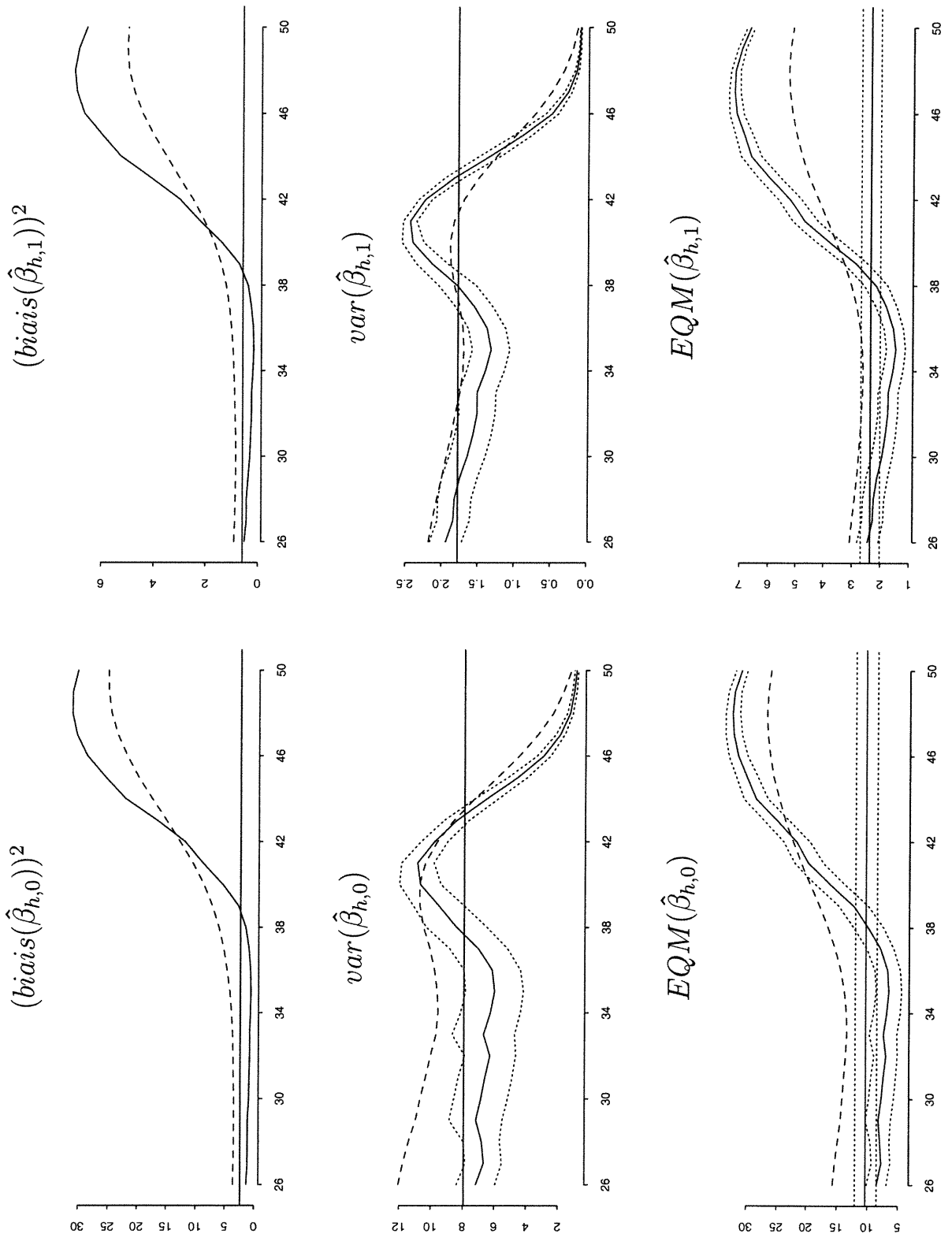


FIGURE C.3. Simulation 2a

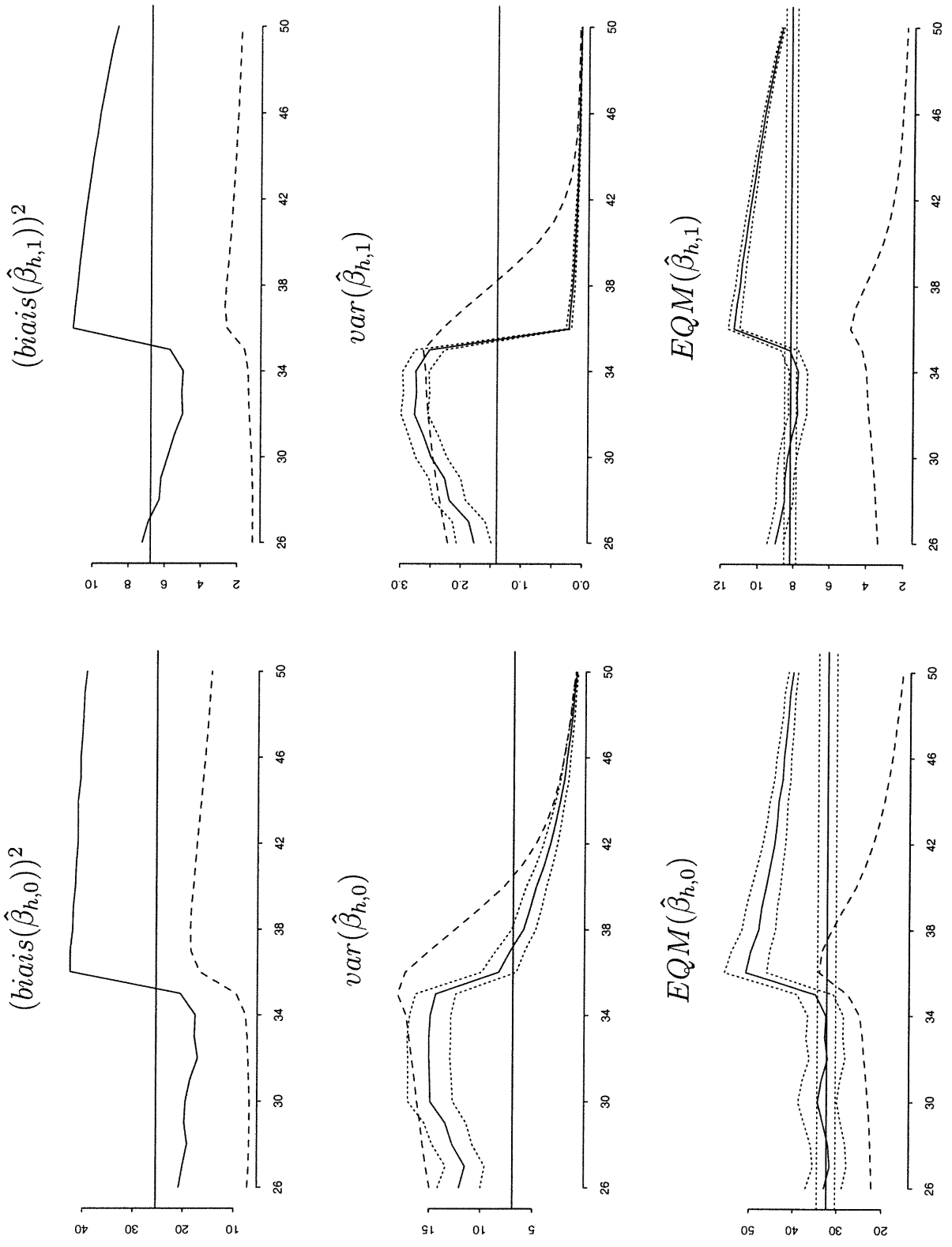


FIGURE C.4. Simulation 3

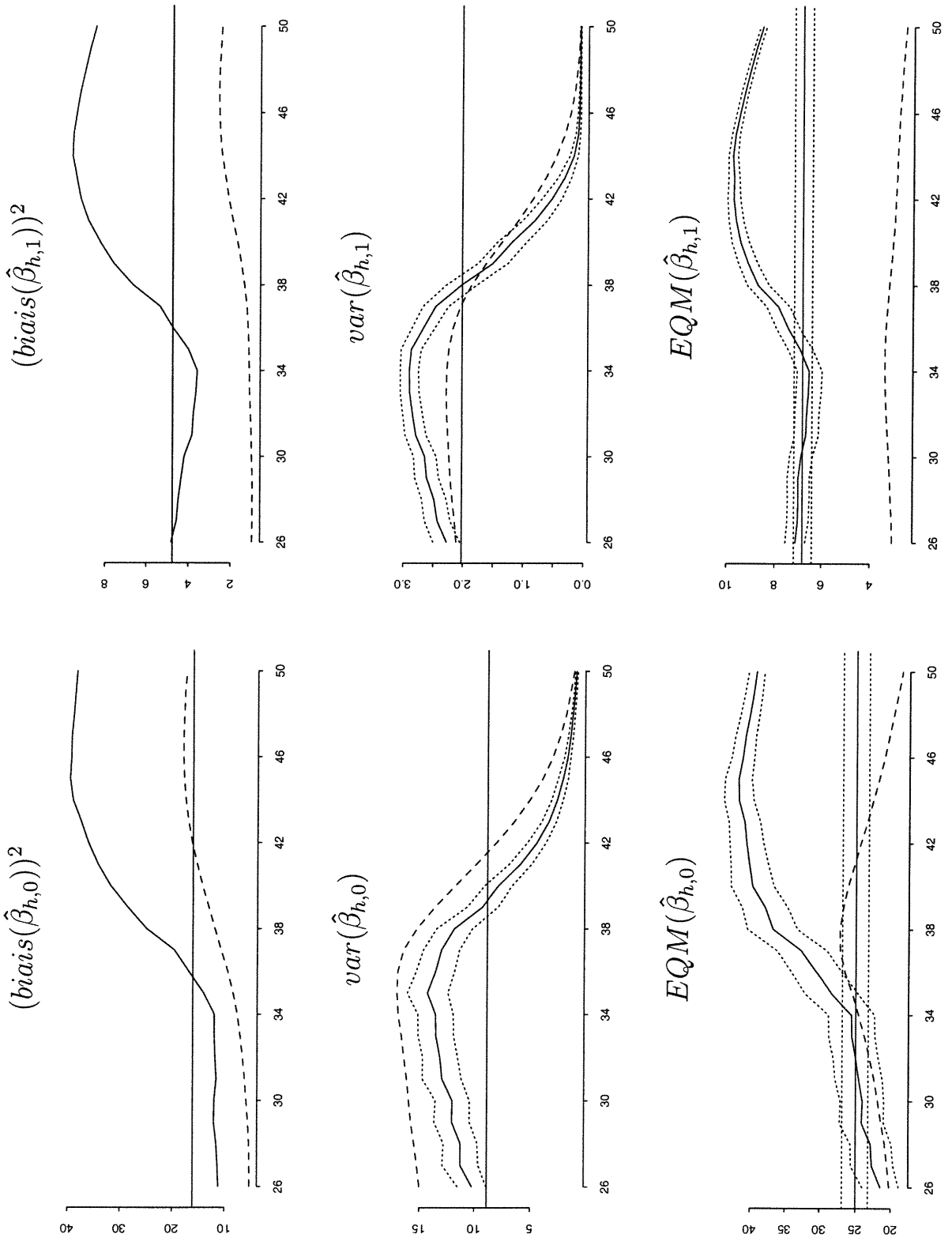


FIGURE C.5. Simulation 3a

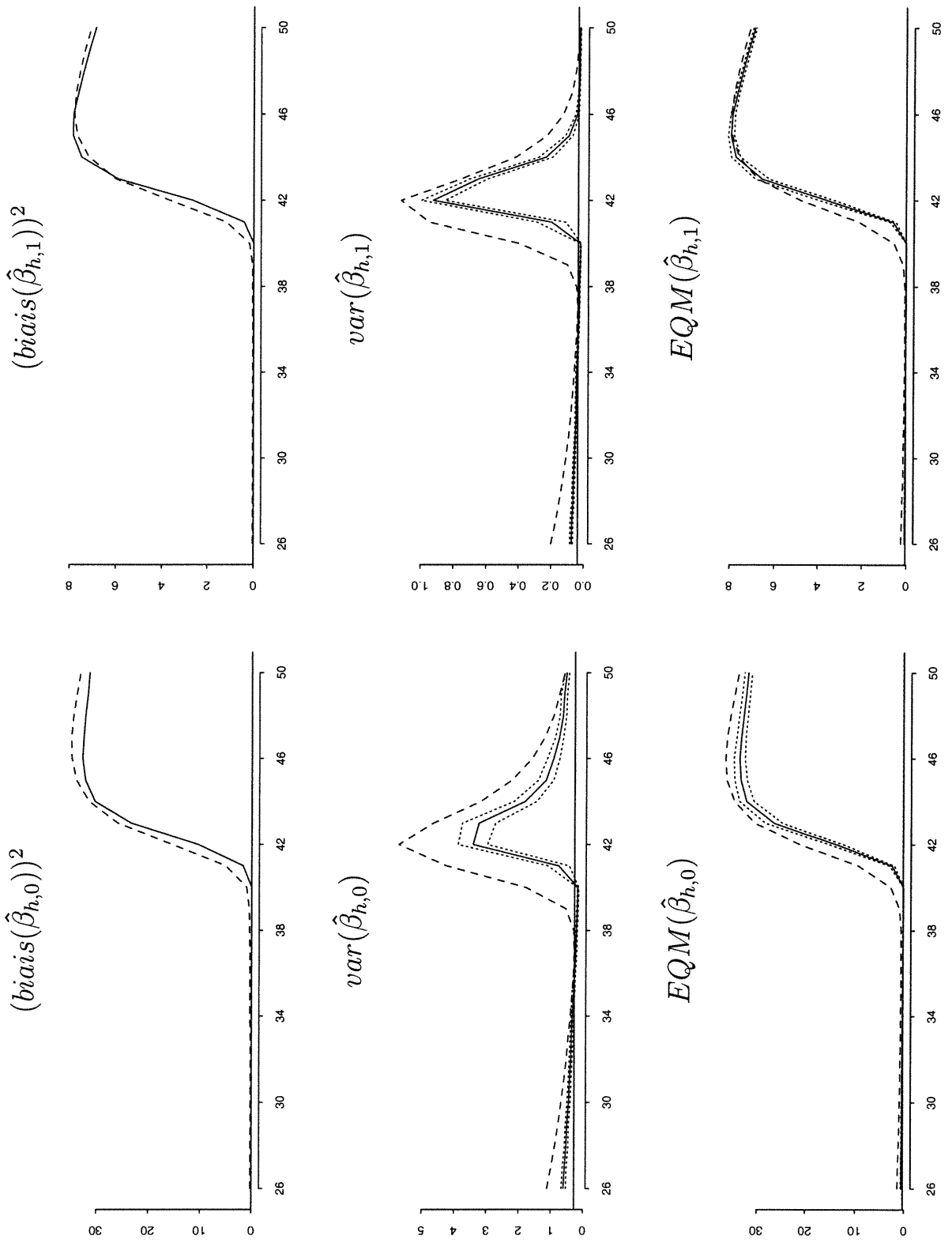


FIGURE C.6. Simulation 4

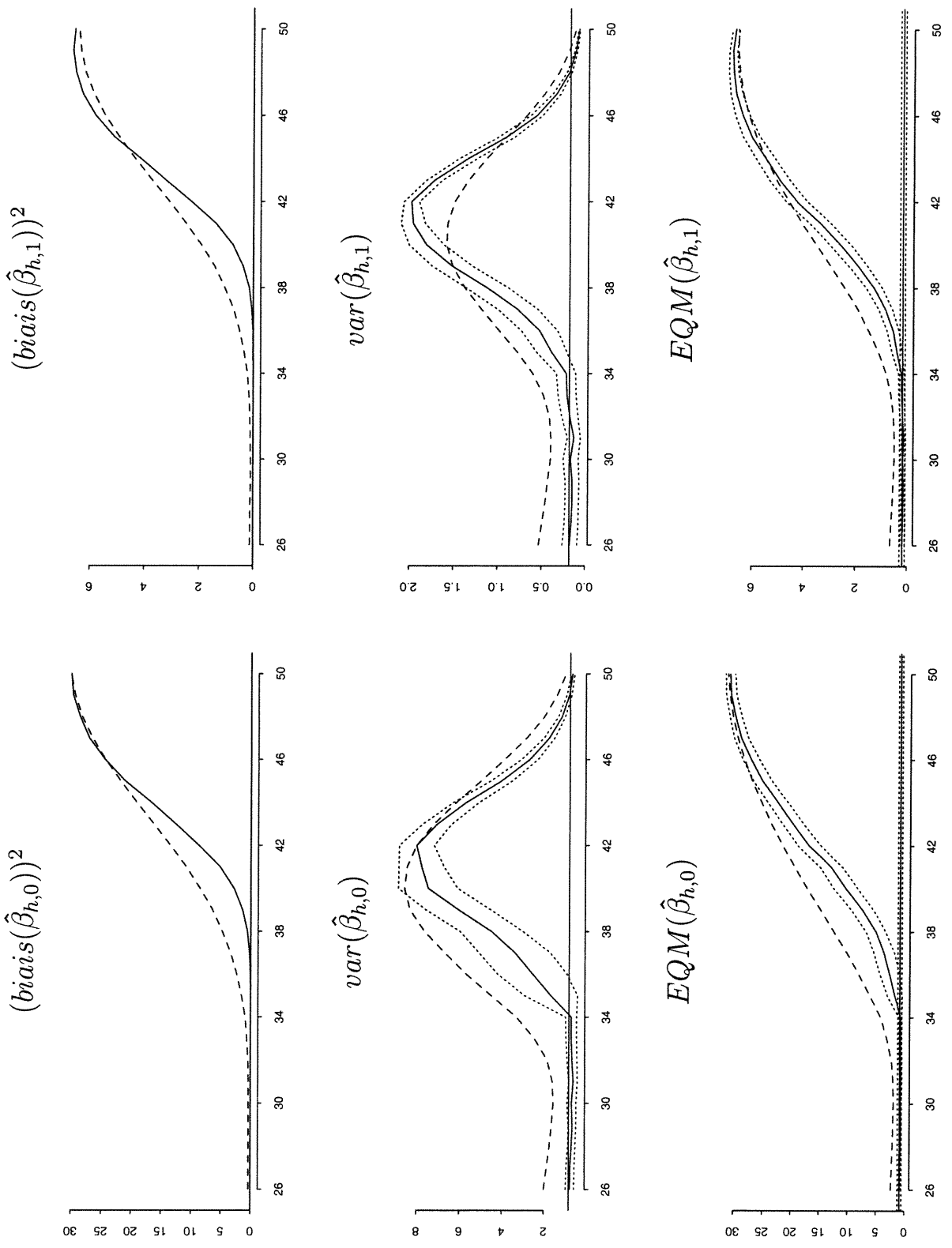


FIGURE C.7. Simulation 4a

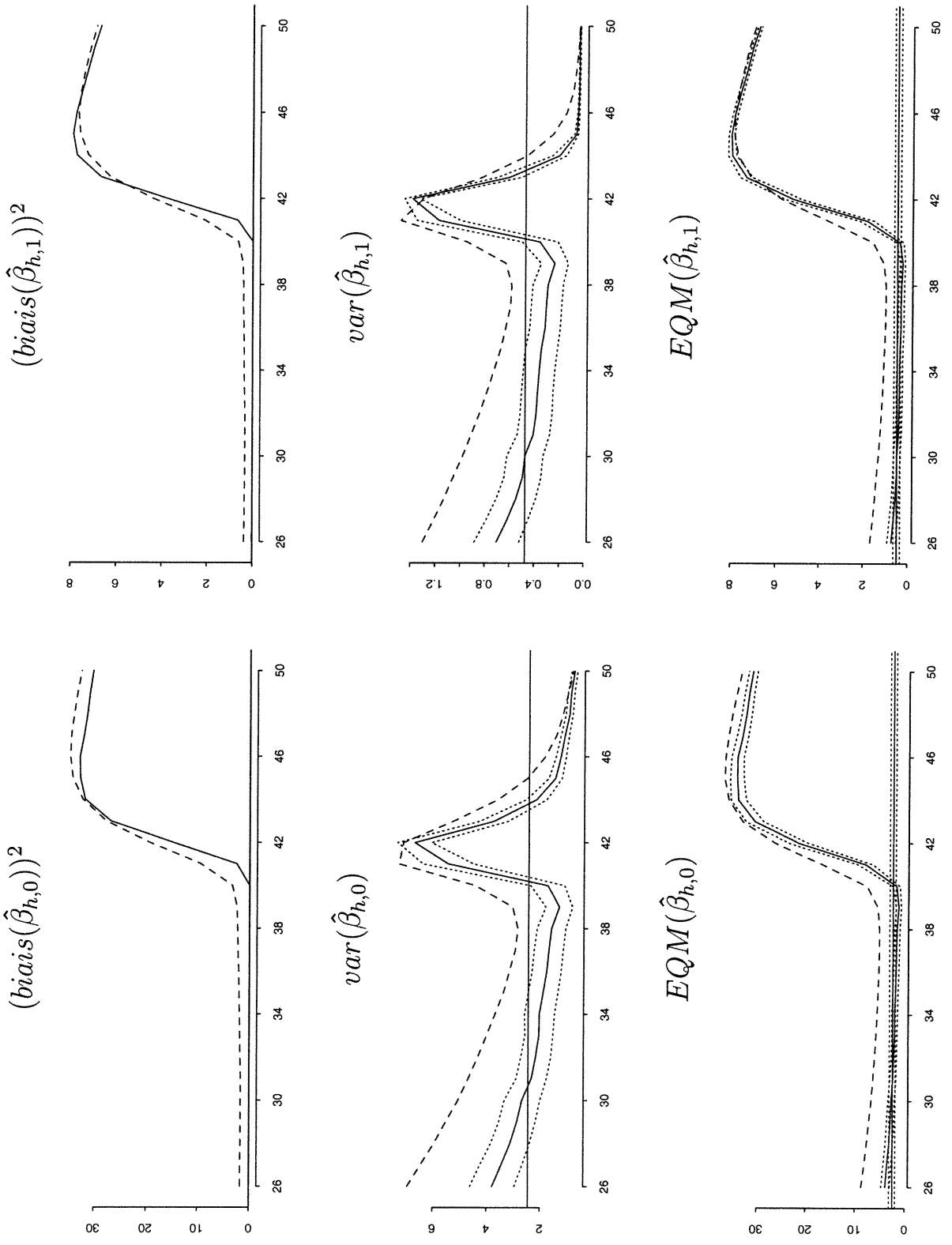


FIGURE C.8. Simulation 5

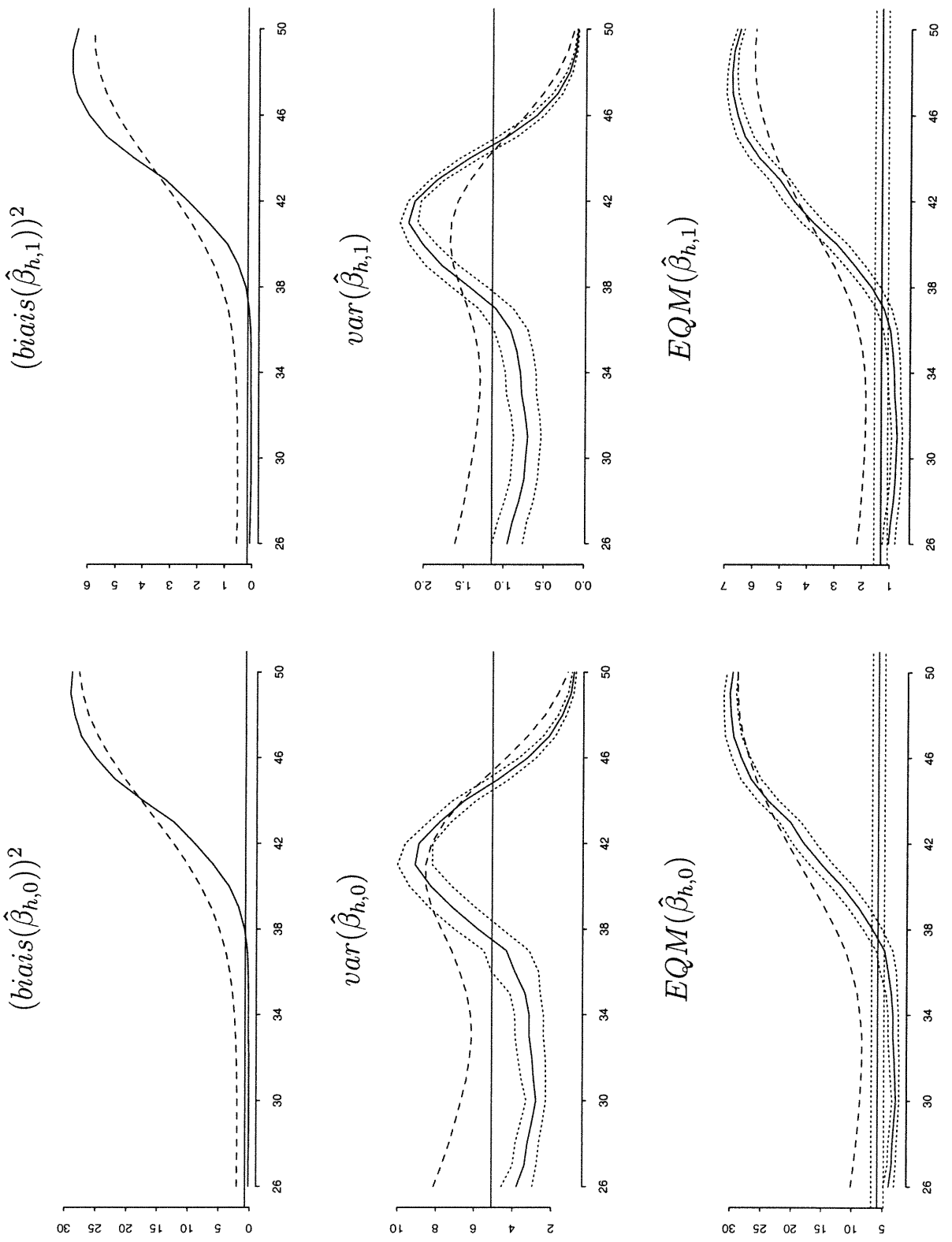


FIGURE C.9. Simulation 5a

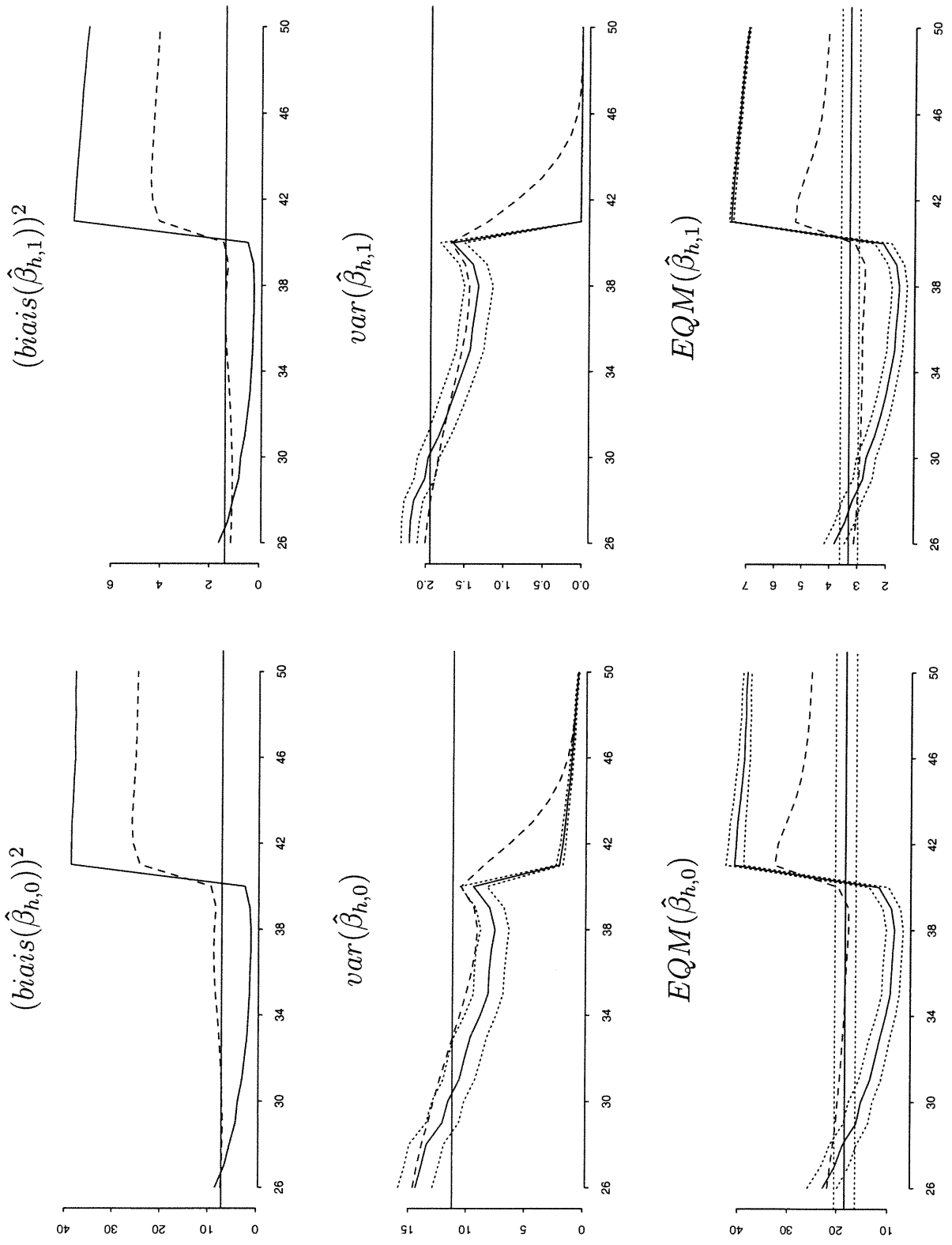


FIGURE C.10. Simulation 6

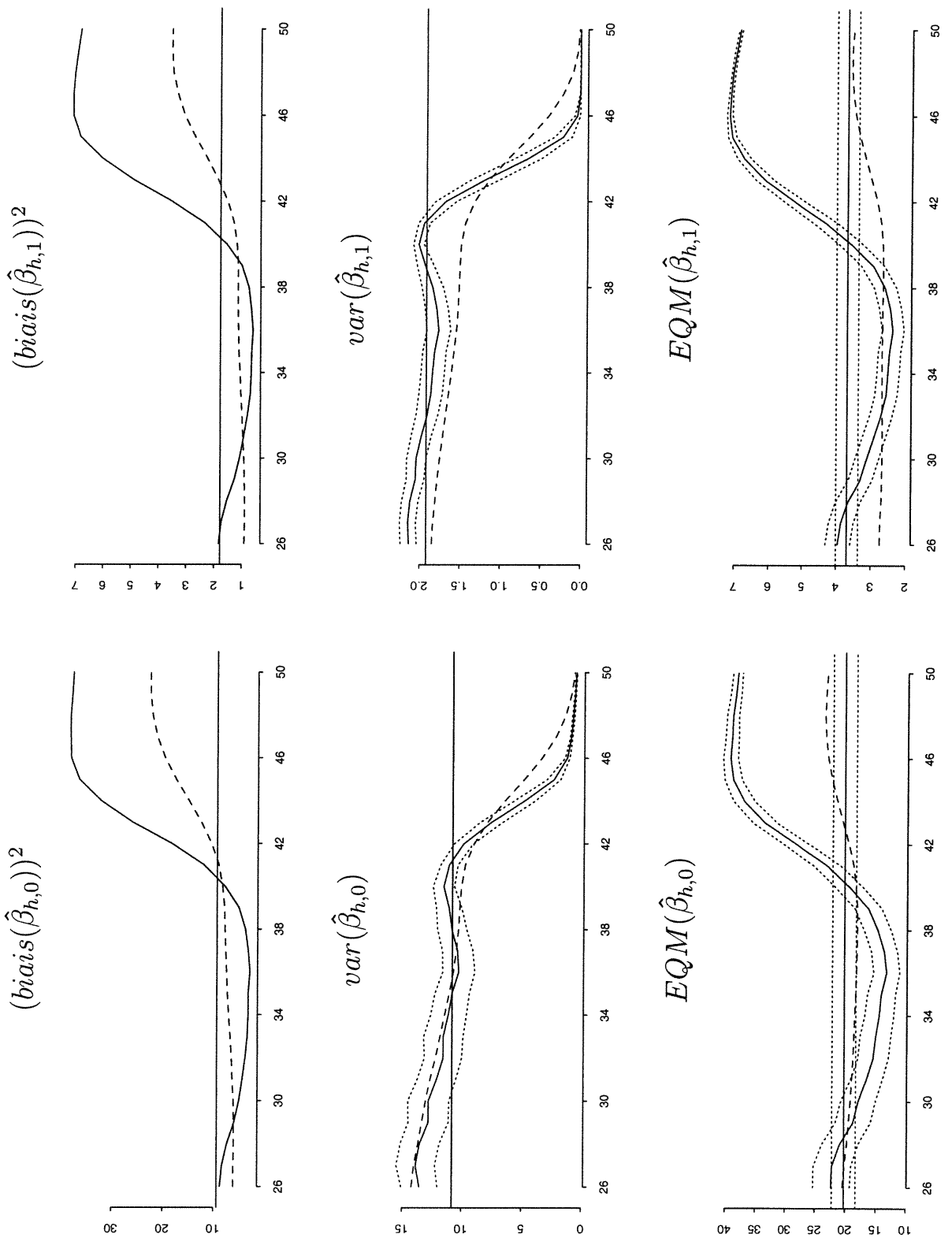


FIGURE C.11. Simulation 6a

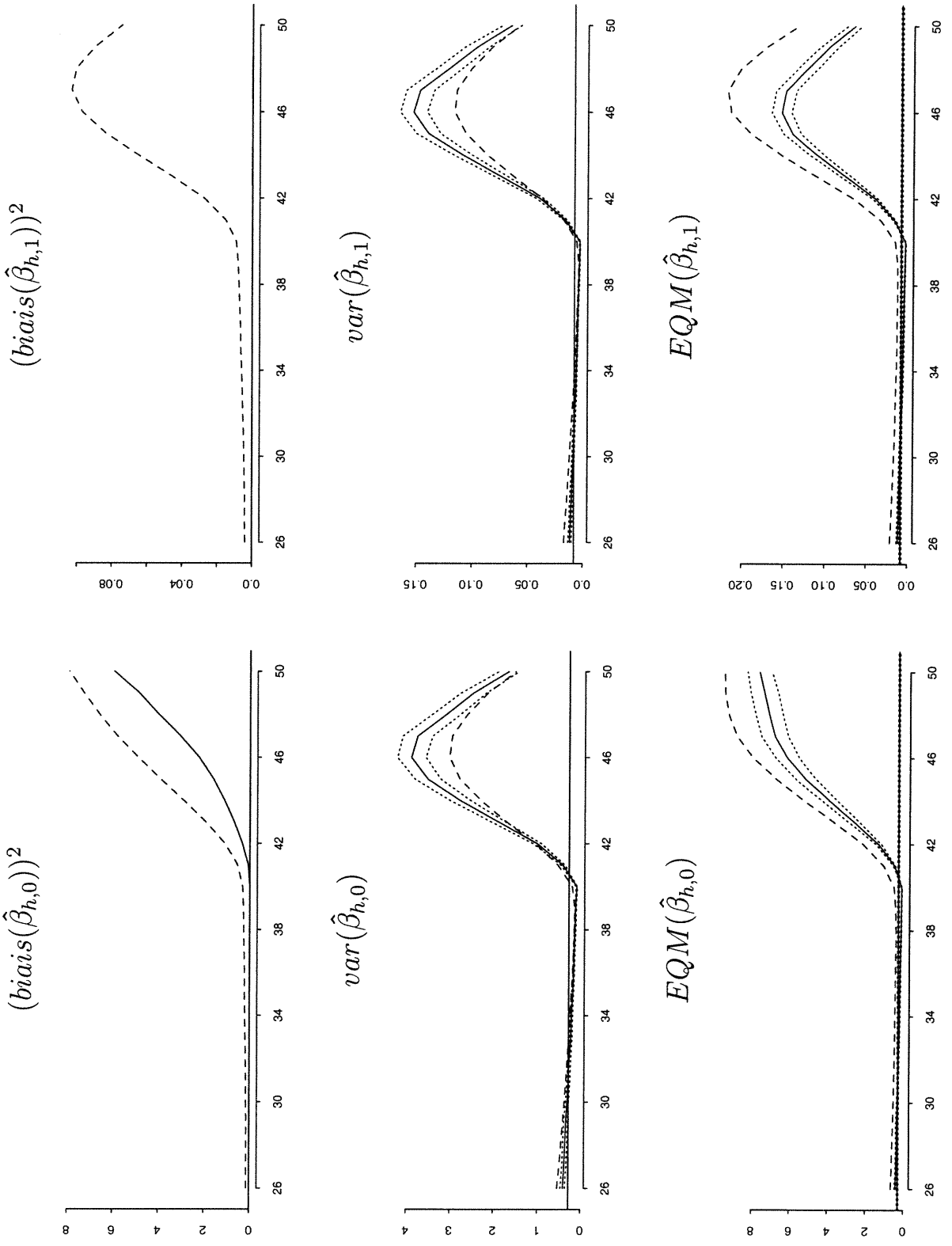


FIGURE C.12. Simulation 7

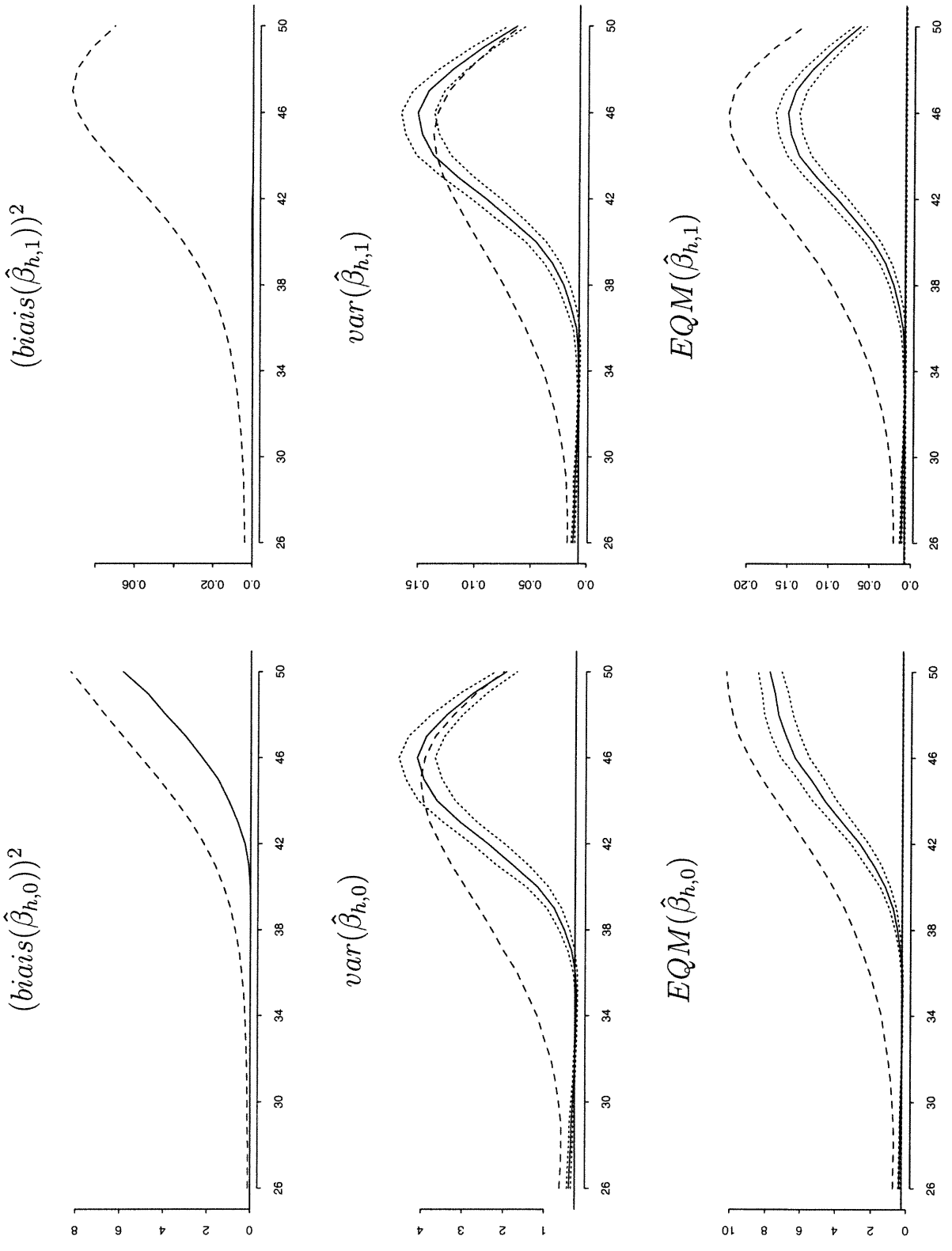


FIGURE C.13. Simulation 7a

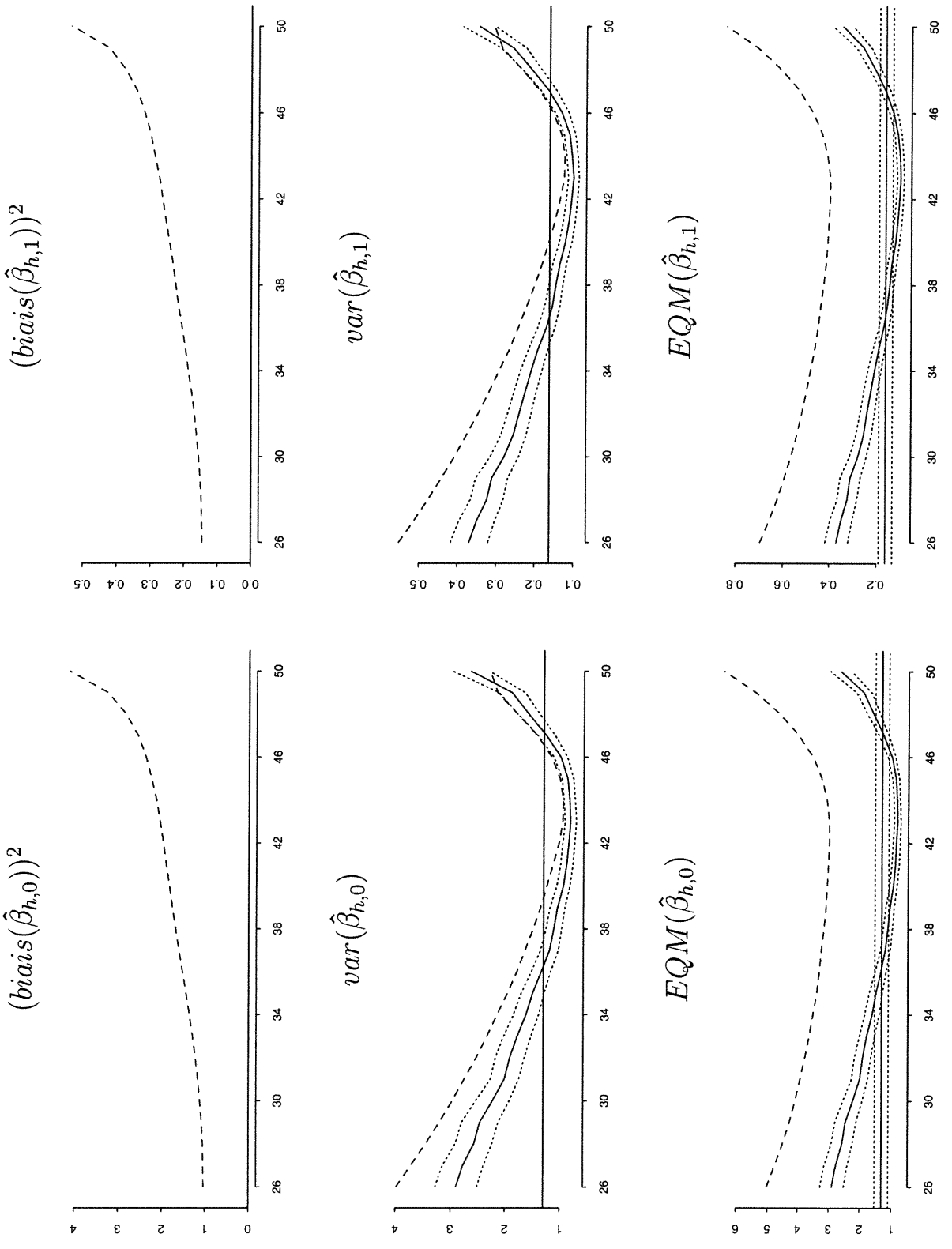


FIGURE C.14. Simulation 8

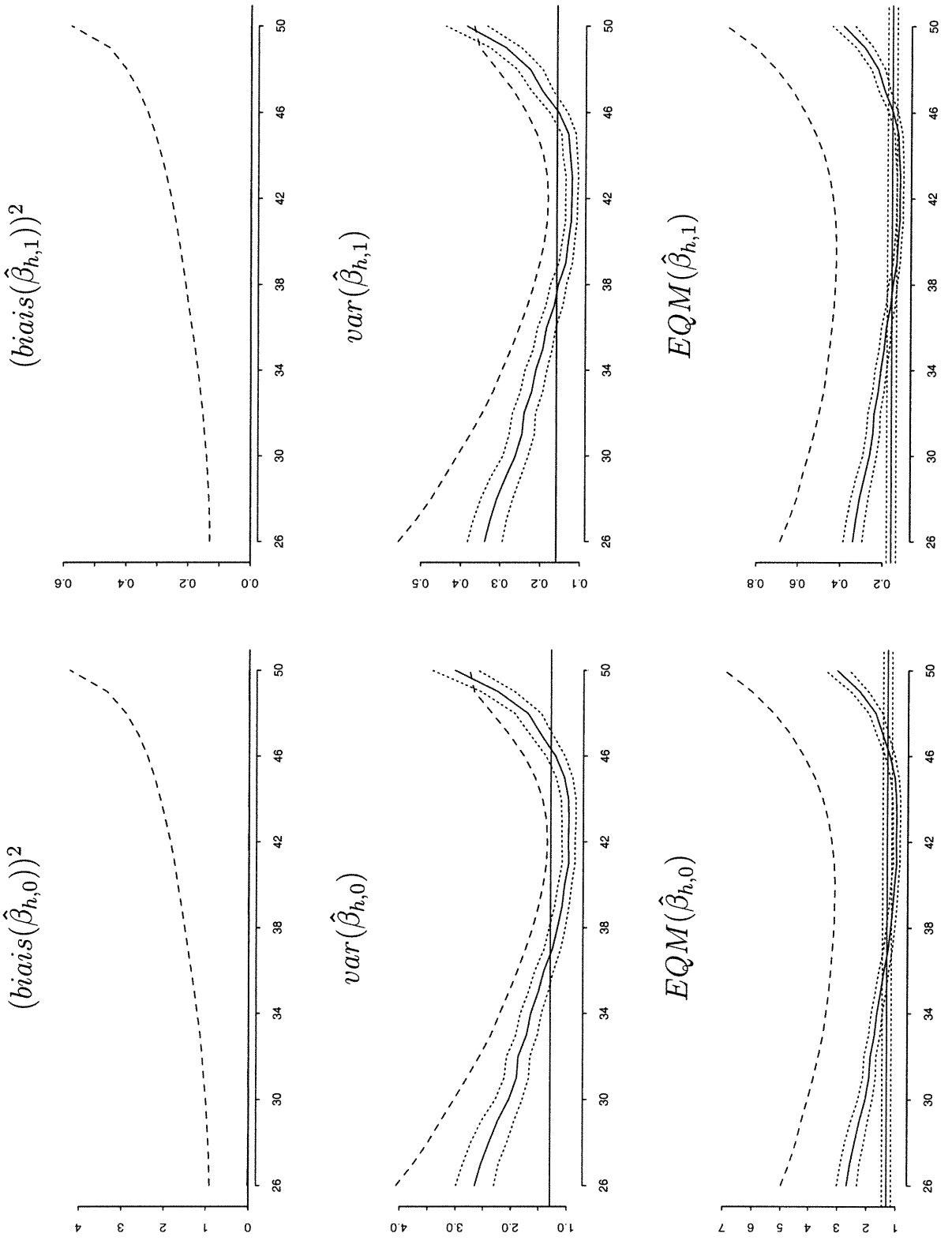


FIGURE C.15. Simulation 8a

Bibliographie

ATKINSON, A. C. & CHENG, T.-C. (1999). Computing least trimmed squares regression with the forward search. *Statistics and Computing* **9**, 251–263.

ATKINSON, A. C. & RIANI, M. (2000). *Robust Diagnostic Regression Analysis.* New York : Springer-Verlag.

BURNS, P. J. (1992). A genetic algorithm for robust regression estimation. Statistical Science Technical Note.

DAVIDSON, A. C. & HINKLEY, D. V. (1997). *Bootstrap Methods and their Applications.* Cambridge University Press.

DONOHO, D. L. & HUBER, P. J. (1983). The notion of breakdown point. In *A Festschrift for Erich Lehmann*, P. Bickel, K. Doksum & J. L. Hodges, Jr., eds. Wadsworth, Belmont, CA.

EDGEWORTH, F. Y. (1887). On observations relating to several quantities. *Hermathena* **6**, 279–285.

EFRON, B. & TIBSHIRANI, R. J. (1993). *An Introduction to the Bootstrap.* New York : Chapman & Hall.

HAMPEL, F. R., RONCHETTI, E. M., ROUSSEEUW, P. J. & STAHEL, W. A. (1985). *Robust Statistics : The Approach Based on Influence Functions.* New York : Wiley.

HAWKINS, D. M. (1994). The feasible solution algorithm for least trimmed squares regression. *Computational Statistics & Data Analysis* **17**, 185–196.

HUBER, P. J. (1972). Robust statistics : A review. *The Annals of Mathematical Statistics* **43**, 1041–1067.

HUBER, P. J. (1973). Robust regression : Asymptotics, conjectures and monte carlo. *Annals of Statistics* **1**, 799–821.

LE CAM, L. (1986). The central limit theorem around 1935. *Statistical Science* **1**, 78–96.

LÉGER, C. & ROMANO, J. P. (1990a). Bootstrap adaptive estimation : The trimmed-mean example. *La Revue Canadienne de Statistique* **18**, 297–314.

LÉGER, C. & ROMANO, J. P. (1990b). Bootstrap choice of tuning parameters. *Annals of the Institute of Statistical Mathematics* **142**, 709–735.

ROUSSEEUW, P. J. (1984). Least median of squares regression. *Journal of the American Statistical Association* **79**, 871–880.

ROUSSEEUW, P. J. & HUBERT, M. (1997). Recent developments in PROGRESS. In L_1-*Statistical Procedures and Related Topics*, Y. Dodge, ed., vol. 31 of *IMS Lecture Notes - Monograph Series*. Inst. Math. Statist. (Hayward).

ROUSSEEUW, P. J. & LEROY, A. M. (1987). *Robust Regression and Outlier Detection*. New York : Wiley.

TUKEY, J. W. (1979). Study of robustness by simulation : Particularly improvement by adjustment and combination. In *Robustness in statistics*, R. Launer & G. Wilkinson, eds. New York : Academic Press.