

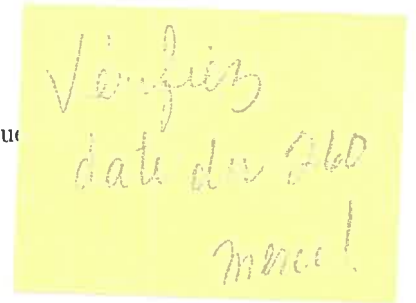
Université de Montréal

Choix des poids de l'estimateur de vraisemblance  
pondérée par rééchantillonnage

par

Joanne Charlebois

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

janvier 2007



QA

3

U54

2007

V.003

**Direction des bibliothèques**

**AVIS**

L'auteur a autorisé l'Université de Montréal à reproduire et diffuser, en totalité ou en partie, par quelque moyen que ce soit et sur quelque support que ce soit, et exclusivement à des fins non lucratives d'enseignement et de recherche, des copies de ce mémoire ou de cette thèse.

L'auteur et les coauteurs le cas échéant conservent la propriété du droit d'auteur et des droits moraux qui protègent ce document. Ni la thèse ou le mémoire, ni des extraits substantiels de ce document, ne doivent être imprimés ou autrement reproduits sans l'autorisation de l'auteur.

Afin de se conformer à la Loi canadienne sur la protection des renseignements personnels, quelques formulaires secondaires, coordonnées ou signatures intégrées au texte ont pu être enlevés de ce document. Bien que cela ait pu affecter la pagination, il n'y a aucun contenu manquant.

**NOTICE**

The author of this thesis or dissertation has granted a nonexclusive license allowing Université de Montréal to reproduce and publish the document, in part or in whole, and in any format, solely for noncommercial educational and research purposes.

The author and co-authors if applicable retain copyright ownership and moral rights in this document. Neither the whole thesis or dissertation, nor substantial extracts from it, may be printed or otherwise reproduced without the author's permission.

In compliance with the Canadian Privacy Act some supporting forms, contact information or signatures may have been removed from the document. While this may affect the document page count, it does not represent any loss of content from the document.

Université de Montréal

Faculté des études supérieures

Ce mémoire intitulé

Choix des poids de l'estimateur de vraisemblance  
pondérée par rééchantillonnage

présenté par

**Joanne Charlebois**

a été évalué par un jury composé des personnes suivantes :

*Alejandro Murua*

---

(président-rapporteur)

*Christian Léger*

---

(directeur de recherche)

*David Haziza*

---

(membre du jury)

Mémoire accepté le:

*20 décembre 2006*

---

## SOMMAIRE

---

La vraisemblance pondérée est une méthode d'estimation qui peut être appliquée quand on dispose de d'autres échantillons reliés à l'échantillon d'intérêt, par exemple des échantillons d'années antérieures ou de régions voisines. Un poids est choisi pour chaque échantillon afin d'incorporer l'information contenue dans l'échantillon et ainsi tenter d'améliorer l'estimation d'un paramètre de la distribution de l'échantillon d'intérêt. Avec un choix approprié de poids, le biais de l'estimateur augmente tandis que sa variance diminue, réduisant ainsi l'erreur quadratique moyenne. Par contre, l'efficacité de la vraisemblance pondérée dépend surtout sur ce choix approprié des poids. Dans ce mémoire on développe le choix de poids par le bootstrap et on compare et contraste cette méthode de choix avec la méthode de la validation croisée développée par Wang et Zidek (2005a).

On développe le choix de poids par la validation croisée et par le bootstrap à partir des mesures d'erreur quadratique moyenne de prévision et d'erreur quadratique moyenne respectivement, et ce pour deux échantillons de données normales ainsi que de données lognormales. On discute certaines propriétés asymptotiques de ces poids, et on effectue des simulations afin de vérifier le comportement des estimateurs résultants.

Pour les données normales, on voit que la théorie nous indique que le poids par la validation croisée converge vers sa valeur optimale lorsque  $n$  devient grand, mais que les simulations indiquent que la convergence est lente et que ce poids est très problématique sans modification. Avec la modification spéciale pour le cas des données normales de Wang et Zidek (2005a) par contre, le poids par la validation croisée performe très bien, en fait on pourrait argumenter qu'il performe le mieux parmi les poids qu'on a calculés lors de la simulation. On voit également que

pour le poids par le bootstrap, afin d'avoir convergence vers la valeur optimale du poids, la théorie nous prescrit une pénalité d'ordre  $n^{-p}$  avec  $0 < p < 1$  au terme du carré du biais dans l'erreur quadratique moyenne, mais qu'en pratique on est mieux de choisir  $p = 0$ , c'est-à-dire de ne pas introduire cette pénalité.

Pour les données lognormales, on voit que la théorie existante sur le poids par la validation croisée a plusieurs trous qui auraient besoin d'être comblés. On voit également que le poids par le bootstrap est calculé d'exactly la même manière que dans le cas de données normales. Les simulations nous montrent de façon claire que le poids bootstrap avec  $p = 0$  est le meilleur poids à utiliser dans ce cas, où l'estimateur de vraisemblance pondérée est plus compliqué que dans le cas de données normales.

On conclut que le choix de poids par le bootstrap a beaucoup d'utilité potentielle comme alternative plus efficace que la validation croisée, et que du travail subséquent est nécessaire afin d'examiner ses propriétés et son comportement avec des données venant de d'autres distributions plus compliquées.

Mots clés : vraisemblance pondérée, validation croisée, bootstrap

## SUMMARY

---

Weighted likelihood is a method of estimation that can be applied when we have other related samples aside from our sample of interest, for example data samples from previous years or from neighbouring regions. Weights are assigned to each sample in order to incorporate the sample's information in the hopes of improving the estimation of a parameter of the distribution of the original sample of interest. With an appropriate choice of weights, the bias of the estimator is increased while its variance is reduced, resulting in a lower mean square error. However, the improvement of the estimator depends strongly on the choice of the weights. In this thesis we develop the choice of weights by the bootstrap and compare and contrast this method with that of cross-validation developed by Wang et Zidek (2005a).

The methods of choice of weights by cross-validation and bootstrap are based on the mean square error of prediction and the mean square error, respectively. We develop these methods for the case of two samples following both normal and lognormal distributions. We discuss certain asymptotic properties of these weights as well as perform simulation studies to examine the behaviour of the resulting estimators.

For the data following normal distributions, we see that the theory implies that the weight converges to its optimal value as  $n$  becomes large, but the simulations show that this convergence is slow and the cross-validation weight used as-is has problems. Using the special modification for the case of normal data developed by Wang et Zidek (2005a), however, this weight performs very well and is arguably the best among the weights calculated in our simulation. We also see that for the choice of weights by the bootstrap, the theory indicates that a penalty of order

$n^{-p}$  with  $0 < p < 1$  is needed on the squared bias term of the mean square error, so that the weight converges to its optimal value. However, in practice we see that it is better to choose  $p = 0$ , that is to not impose such a penalty.

For the data following lognormal distributions, we see that the existing theory about the weights by cross-validation has some gaps that need to be filled. We also see that the bootstrap weights are calculated in exactly the same manner as for the normal data case. Our simulations show clearly that the bootstrap weight with  $p = 0$  is the best choice in this case where the weighted likelihood estimator is more complicated than for normal data.

We conclude that the choice of weights by the bootstrap has great potential as a more efficient alternative to cross-validation, and that subsequent work is needed to examine its properties and behaviour in the case of data samples from other, more complex distributions.

Keywords : weighted likelihood, cross-validation, bootstrap



# TABLE DES MATIÈRES

---

<b>Sommaire</b> .....	iii
<b>Summary</b> .....	v
<b>Liste des figures</b> .....	ix
<b>Liste des tableaux</b> .....	x
<b>Remerciements</b> .....	1
<b>Chapitre 1. Introduction</b> .....	1
1.1. La vraisemblance pondérée.....	1
1.2. La validation croisée.....	3
1.3. Le bootstrap.....	8
1.4. Conventions de notation.....	10
1.5. Cheminement de ce mémoire.....	11
<b>Chapitre 2. Données normales</b> .....	12
2.1. Erreur quadratique moyenne et la valeur optimale de $\lambda$ .....	12
2.2. Choix de $\lambda$ par validation croisée.....	15
2.3. Choix de $\lambda$ par bootstrap.....	24
2.4. Simulations.....	28
<b>Chapitre 3. Données lognormales</b> .....	43
3.1. Erreur quadratique moyenne et la valeur optimale de $\lambda$ .....	44

3.2. Choix de $\lambda$ par validation croisée.....	44
3.3. Choix de $\lambda$ par bootstrap.....	46
3.4. Simulations.....	47
<b>Chapitre 4. Conclusion.....</b>	<b>62</b>
<b>Bibliographie.....</b>	<b>65</b>
<b>Annexe A. Simulations normales.....</b>	<b>A-i</b>
A.1. Programmes.....	A-i
A.2. Histogrammes des huit poids $\hat{\lambda}$ examinés, pour plusieurs valeurs de $n$ et dans les deux cas $\mu_X = \mu_Y$ et $ \mu_X - \mu_Y  = 0,3$ , pour données normales.....	A-xii
A.2.1. $\mu_X = \mu_Y$ .....	A-xii
A.2.2. $ \mu_X - \mu_Y  = 0,3$ .....	A-xvi
<b>Annexe B. Simulations lognormales.....</b>	<b>B-i</b>
B.1. Programmes.....	B-i
B.2. Histogrammes des huit poids $\hat{\lambda}$ examinés, pour plusieurs valeurs de $n$ et dans les deux cas $\mu_X = \mu_Y$ et $ \mu_X - \mu_Y  = 0,3$ , pour données lognormales.....	B-xiv
B.2.1. $\mu_X = \mu_Y$ .....	B-xiv
B.2.2. $ \mu_X - \mu_Y  = 0,3$ .....	B-xviii

## LISTE DES FIGURES

---

- 2.1 Graphiques de  $\frac{EQM_p^*}{EQM_{EVM}}$  en fonction de  $p$  dans le cas de données normales,  $|\mu_X - \mu_Y| = 0,3$  et  $\sigma_X^2 = \sigma_Y^2 = 1$  ..... 36
- 3.1 Exemple de paire de graphiques du critère de la validation croisée  $D$  (“Dvect” à gauche) et de sa dérivée  $\partial D/\partial \lambda$  (“dDvect” à droite) en fonction de  $\lambda$ , lorsque  $n = 20$ ,  $\mu_X \neq \mu_Y$ , et  $\sigma_X^2 = \sigma_Y^2 = 1$ , pour des données lognormales ..... 50
- 3.2 Exemple de paire de graphiques du critère de la validation croisée  $D$  (“Dvect” à gauche) et de sa dérivée  $\partial D/\partial \lambda$  (“dDvect” à droite) en fonction de  $\lambda$ , lorsque  $n = 20$ ,  $\mu_X \neq \mu_Y$ , et  $\sigma_X^2 = \sigma_Y^2 = 1$ , pour des données lognormales ..... 51
- 3.3 Exemple de paire de graphiques du critère de la validation croisée  $D$  (“Dvect” à gauche) et de sa dérivée  $\partial D/\partial \lambda$  (“dDvect” à droite) en fonction de  $\lambda$ , lorsque  $n = 20$ ,  $\mu_X \neq \mu_Y$ , et  $\sigma_X^2 = \sigma_Y^2 = 1$ , pour des données lognormales ..... 52
- 3.4 Exemple de paire de graphiques du critère de la validation croisée  $D$  (“Dvect” à gauche) et de sa dérivée  $\partial D/\partial \lambda$  (“dDvect” à droite) en fonction de  $\lambda$ , lorsque  $n = 20$ ,  $\mu_X \neq \mu_Y$ , et  $\sigma_X^2 = \sigma_Y^2 = 1$ , pour des données lognormales ..... 53

## LISTE DES TABLEAUX

---

2.1	Probabilité que l'approximation (2.2.9) $\tilde{\lambda}_{CV}$ soit négative pour $ \mu_X - \mu_Y  = 0,3$ et $\sigma_X^2 = \sigma_Y^2 = 1$ . . . . .	19
2.2	Valeur asymptotique des poids par validation croisée . . . . .	22
2.3	Pourcentages de valeurs négatives de $\tilde{\lambda}_{CV}$ lors de sa simulation de taille 1000, pour $ \mu_X - \mu_Y  = 0,3$ et $\sigma_X^2 = 1$ . . . . .	23
2.4	Valeur asymptotique des poids bootstrap quand $\mu_X = \mu_Y$ . . . . .	28
2.5	Valeur asymptotique des poids bootstrap quand $\mu_X \neq \mu_Y$ . . . . .	28
2.6	Espérance et écart type des poids pour des données normales, $\mu_X = \mu_Y$ et $\sigma_X^2 = \sigma_Y^2 = 1$ . . . . .	37
2.7	Espérance et écart type des poids des données normales, $ \mu_X - \mu_Y  = 0,3$ et $\sigma_X^2 = \sigma_Y^2 = 1$ . . . . .	38
2.8	EQM des EVP et écart type des différences au carré entre l'EVP et $\mu_X$ , pour des données normales, $\mu_X = \mu_Y$ et $\sigma_X^2 = \sigma_Y^2 = 1$ . . . . .	39
2.9	EQM des EVP et écart type des différences au carré entre l'EVP et $\mu_X$ , pour des données normales, $ \mu_X - \mu_Y  = 0,3$ et $\sigma_X^2 = \sigma_Y^2 = 1$ . . . . .	40
2.10	Ratios des EQM de l'EVP par rapport à l'EVM pour des données normales, $\mu_X = \mu_Y$ et $\sigma_X^2 = \sigma_Y^2 = 1$ . . . . .	41
2.11	Ratios des EQM de l'EVP par rapport à l'EVM pour des données normales, $ \mu_X - \mu_Y  = 0,3$ et $\sigma_X^2 = \sigma_Y^2 = 1$ . . . . .	42
3.1	Espérance et écart type des poids pour des données lognormales, $\mu_X = \mu_Y = 0$ et $\sigma_X^2 = \sigma_Y^2 = 1$ . . . . .	56

3.2	Espérance et écart type des poids pour des données lognormales, $\mu_X = 0, \mu_Y = 0,3$ et $\sigma_X^2 = \sigma_Y^2 = 1$ .....	57
3.3	EQM des EVP et écart type des différences au carré entre l'EVP et $\mu_X$ , pour des données lognormales, $\mu_X = \mu_Y = 0$ et $\sigma_X^2 = \sigma_Y^2 = 1$ .....	58
3.4	EQM des EVP et écart type des différences au carré entre l'EVP et $\mu_X$ , pour des données lognormales, $\mu_X = 0, \mu_Y = 0,3$ et $\sigma_X^2 = \sigma_Y^2 = 1$ .....	59
3.5	Ratios des EQM de l'EVP par rapport à l'EVM pour des données lognormales, $\mu_X = \mu_Y = 0$ et $\sigma_X^2 = \sigma_Y^2 = 1$ .....	60
3.6	Ratios des EQM de l'EVP par rapport à l'EVM pour des données lognormales, $\mu_X = 0, \mu_Y = 0,3$ et $\sigma_X^2 = \sigma_Y^2 = 1$ .....	61

## REMERCIEMENTS

---

J'aimerais remercier mon directeur de recherche Christian Léger pour m'avoir beaucoup appris, ainsi que pour le support financier. J'aimerais aussi remercier Scott, S.P., Azraël, Possum, Thor et Gretchen pour le support émotionnel.

# Chapitre 1

---

## INTRODUCTION

La méthode de vraisemblance pondérée est utilisée afin de réduire l'erreur quadratique moyenne d'un estimateur, en augmentant son biais tout en diminuant sa variance. On peut faire ceci quand on dispose de d'autres échantillons reliés au nôtre, en mettant un poids sur l'information de chaque échantillon. On espère que l'information additionnelle contenue dans les échantillons reliés améliore notre estimation. L'efficacité de la pondération dépend surtout d'un choix judicieux des poids. Wang et Zidek (2005a) ont montré que les poids peuvent dépendre des données et en particulier ont développé le choix de poids par la méthode de *validation croisée*. Dans ce mémoire on développe le choix des poids par le bootstrap, et on compare et contraste ces deux méthodes.

### 1.1. LA VRAISEMBLANCE PONDÉRÉE

La vraisemblance pondérée est utilisée quand on a  $m$  échantillons

$$\mathbf{X} = \{X_{(1)}, X_{(2)}, \dots, X_{(m)}\}$$

de tailles  $n_1, n_2, \dots, n_m$  respectivement, où

$$X_{(i)} = \{X_{(i)1}, X_{(i)2}, \dots, X_{(i)n_i}\},$$

venant de  $m$  différentes populations. Supposons qu'on s'intéresse à faire de l'inférence sur une de ces populations, disons celle dont provient l'échantillon  $X_{(1)}$ . L'information des autres échantillons peut s'avérer utile si les populations ont un

lien quelconque, et pour cette raison, on aimerait mettre un poids sur l'information recueillie de chaque échantillon afin d'améliorer l'inférence sur la première population. Par exemple, on pourrait s'intéresser à la distribution du nombre de visites à l'hôpital pendant l'été pour des maladies causées par la température élevée. Dans ce cas, les données recueillies à chaque année antérieure pourraient former les autres échantillons; leurs distributions ont une relation probable avec la distribution des données de l'année courante. Alternativement, on pourrait prendre comme échantillons reliés les données provenant de d'autres hôpitaux dans la même région, pour la même année qui nous intéresse.

Si les échantillons  $X_{(1)}, X_{(2)}, \dots, X_{(m)}$  sont distribués selon les densités  $f_1(\cdot; \theta_1), f_2(\cdot; \theta_2), \dots, f_m(\cdot; \theta_m)$  respectivement, et on s'intéresse à estimer le paramètre  $\theta_1$  de la première population, alors la vraisemblance classique est

$$L(X_{(1)}; \theta_1) = \prod_{j=1}^{n_1} f_1(X_{(1)j}; \theta_1),$$

qui nous donne l'estimateur de vraisemblance maximale usuel, qu'on dénote  $\hat{\theta}_1$  suivant Wang et Zidek (2005a). Cette vraisemblance utilise seulement l'information provenant du premier échantillon et ignore complètement l'information contenue dans les autres échantillons.

La vraisemblance pondérée, quant à elle, est définie comme suit (voir Hu et Zidek, 2002) — notons qu'on utilise la valeur  $\theta_1$  pour chacune des  $m$  distributions, et non seulement pour la première distribution, puisqu'on s'intéresse seulement au paramètre de la première population :

$$WL(\mathbf{X}; \theta_1) = \prod_{i=1}^m \prod_{j=1}^{n_i} f_i(X_{(i)j}; \theta_1)^{\lambda_i}$$

où  $\lambda = (\lambda_1, \dots, \lambda_m)$  est le vecteur des poids. On présume toujours que  $\sum_{i=1}^m \lambda_i = 1$  dans ce mémoire. Si les poids sont bien choisis, le rôle du poids  $\lambda_i$  est de quantifier le lien entre l'échantillon  $X_{(i)}$  et l'échantillon  $X_{(1)}$  : si  $\theta_i$  est très semblable à  $\theta_1$ , alors l'information contenue dans l'échantillon  $X_{(i)}$  est très pertinente, et on voudrait que le poids  $\lambda_i$  reflète cette pertinence. Dans ce cas, on voudrait que le poids  $\lambda_i$  soit relativement élevé. Si par contre  $\theta_i$  est "loin" de  $\theta_1$ , le poids  $\lambda_i$



devrait être petit en valeur absolue, puisque l'information contenue dans  $X_{(i)}$  ne va pas nous aider à estimer  $\theta_1$ .

En pratique on travaille surtout avec le logarithme de la vraisemblance pondérée,

$$\log \text{WL}(\mathbf{X}; \theta_1) = \sum_{i=1}^m \sum_{j=1}^{n_i} \lambda_i \log f_i(X_{(i)j}; \theta_1).$$

L'estimateur de vraisemblance pondérée maximale est donc

$$\tilde{\theta}_1 = \arg \sup_{\theta_1 \in \Theta} \text{WL}(\mathbf{X}; \theta_1)$$

qui peut être calculé, dans des cas simples, par la résolution de l'équation

$$\frac{\partial}{\partial \theta_1} \log \text{WL}(\mathbf{X}; \theta_1) = 0.$$

Le critère d'erreur utilisé pour un choix de poids  $\lambda = (\lambda_1, \dots, \lambda_m)$  est l'erreur quadratique moyenne de l'estimateur  $\tilde{\theta}_1$  résultant :

$$\text{EQM}(\lambda) = \text{E}[(\tilde{\theta}_1 - \theta_1)^2].$$

Dans ce mémoire on considère le cas de deux échantillons  $X$  et  $Y$  de taille égale  $n$ . Puisqu'on a seulement deux populations, on utilise la notation  $\theta_X$  au lieu de  $\theta_1$  qu'on a utilisé jusqu'à maintenant. On a alors deux poids  $\lambda_X$  et  $\lambda_Y$  tels que  $\lambda_X + \lambda_Y = 1$ , nous donnant effectivement un seul poids  $\lambda = \lambda_X$ , l'autre étant  $\lambda_Y = 1 - \lambda_X$ .

Les poids  $\lambda$  peuvent être constants, ou comme on traite ici, adaptatifs et basés sur les données. Dans ce mémoire on examine le choix de poids par validation croisée développé par Wang et Zidek (2005a) ainsi que le choix de poids par bootstrap.

## 1.2. LA VALIDATION CROISÉE

L'idée de la validation croisée est de diviser un échantillon en sous-échantillons afin de valider un modèle ou une estimation. L'analyse initiale se fait sur un sous-échantillon, et le restant des données est utilisé comme sous-échantillon validateur afin de vérifier la valeur prédictive du modèle ou de l'estimation.

La méthode de validation croisée, dans sa forme simple, consiste à choisir des observations parmi l'échantillon, observations qui deviennent le sous-échantillon validateur. D'habitude ce sous-échantillon comprend un tiers ou moins des données. Dans la forme la plus simple, l'estimation ou le choix de modèle est fait sur les données "restantes", et la validation est faite une seule fois sur l'unique sous-échantillon validateur, en calculant la mesure de l'erreur de prévision désirée. Par contre, il est plus utile de répéter le processus sur plusieurs sous-échantillons différents, comme dans la méthode de validation croisée *k-fold*.

Dans la validation croisée *k-fold*, l'échantillon est divisé en  $k$  sous-échantillons disjoints. On prend un de ces sous-échantillons comme sous-échantillon validateur pour mesurer l'erreur de prévision de l'estimation calculée à partir des  $k - 1$  autres sous-échantillons. On répète alors ce processus pour chacun des  $k$  sous-échantillons. On prend ensuite la moyenne (ou une autre manière de combiner les résultats) des  $k$  mesures de l'erreur de prévision ainsi obtenues. Quand  $k$ , la taille de l'échantillon original, les  $n$  sous-échantillons disjoints sont simplement les  $n$  échantillons de taille 1 : chaque donnée devient un sous-échantillon validateur. Cette méthode, qu'on explique en détail ci-bas, se nomme *leave-one-out*.

Stone (1974) a commencé l'étude formelle de l'application de la validation croisée au problème de prédiction statistique. Stone (1974) et Geisser (1975) discutent la méthode *k-fold* et utilisent une combinaison linéaire de moyennes de différents sous-échantillons afin d'estimer une moyenne commune. Stone (1977) développe des propriétés asymptotiques de méthodes de validation croisée afin de discuter leurs avantages et désavantages.

La méthode traitée par Wang et Zidek (2005a), qu'on reprend ici, est la méthode *leave-one-out* qui prend la  $i^{\text{ième}}$  observation comme sous-échantillon validateur de taille 1, et ce pour chaque  $i$ ,  $i = 1, \dots, n$ .

Pour être plus précis, on a deux échantillons de variables aléatoires

$$\begin{aligned} X_1, X_2, \dots, X_n &\stackrel{\text{i.i.d.}}{\sim} f_X(X; \theta_X) \\ Y_1, Y_2, \dots, Y_n &\stackrel{\text{i.i.d.}}{\sim} f_Y(Y; \theta_Y) \end{aligned}$$

où i.i.d. dénote que les données sont indépendantes et identiquement distribuées, et où les deux échantillons  $X$  et  $Y$  sont présumés indépendants l'un de l'autre. À noter que Wang et Zidek (2005a) ont permis une corrélation  $\rho$  entre les observations  $X_i$  et  $Y_i$ ,  $i = 1, \dots, n$ , c'est-à-dire une corrélation pour la paire  $(X_i, Y_i)$  mais l'indépendance entre les paires. Le paramètre  $\theta_X$  est le paramètre d'intérêt. Notons que dans ce mémoire on fait toujours l'hypothèse que les  $X$  et les  $Y$  suivent des distributions de la même famille.

On fait l'hypothèse que le paramètre  $\theta_X$  et  $E[X_i]$  sont liés via une fonction  $\phi$  telle que  $\phi(\theta_X) = E[X_i]$ .

Avec la méthode *leave-one-out*, on utilise toutes les observations sauf la  $i^{\text{ième}}$  observation de chaque échantillon, afin de prédire  $X_i$ .

Pour faire ceci, on calcule l'estimateur de vraisemblance pondérée maximale  $\tilde{\theta}_X^{(-i)}$  à partir des sous-échantillons  $X^{(-i)} = \{X_1, \dots, X_{i-1}, X_{i+1}, \dots, X_n\}$  et  $Y^{(-i)} = \{Y_1, \dots, Y_{i-1}, Y_{i+1}, \dots, Y_n\}$ , c'est-à-dire on enlève la  $i^{\text{ième}}$  observation de chacun des deux échantillons. Notons qu'on utilise en général la notation  $(-i)$  pour indiquer que la  $i^{\text{ième}}$  observation a été enlevée d'un échantillon, ou enlevée lors du calcul d'un estimateur. Notons aussi que  $\tilde{\theta}_X^{(-i)}$  (ainsi que  $\tilde{\theta}_X$ ) dépend du poids  $\lambda$ , ce qui n'est pas indiqué explicitement afin d'alléger la notation.

Ensuite on prédit  $X_i$ , qui est traitée comme une observation future, c'est-à-dire indépendante des observations utilisées pour la prédire, via  $\phi(\tilde{\theta}_X^{(-i)})$ .

La mesure d'erreur résultante qui nous intéresse est l'erreur quadratique moyenne de prévision,

$$\text{EQMP} = E \left[ \sum_{i=1}^n (X_{future} - \phi(\tilde{\theta}_X))^2 \right] \quad (1.2.1)$$

où  $X_{future}$  est une observation de loi  $f_X(\cdot, \theta_X)$  indépendante de  $X_1, \dots, X_n$  et de  $Y_1, \dots, Y_n$

On répète le processus décrit pour chacune des  $n$  observations, ce qui nous donne l'estimateur de l'erreur quadratique moyenne de prévision par la validation croisée suivante :

$$D(\lambda) = \frac{1}{n} \sum_{i=1}^n (X_i - \phi(\tilde{\theta}_X^{(-i)}))^2 \quad (1.2.2)$$

Pour plus de détails, voir Wang et Zidek (2005a, pp. 464-467). On aimerait trouver le poids  $\lambda$  qui minimise l'estimateur de l'erreur quadratique moyenne de prévision par la validation croisée  $D$ .

Si l'espérance d'une observation  $E[X_i]$  est une fonction linéaire du paramètre d'intérêt  $\theta_X$ , alors  $\bar{\theta}_X$  est une combinaison linéaire des estimateurs de vraisemblance maximale de chaque population. Dans ce cas on peut dériver la forme analytique du poids qui est optimal par rapport au critère  $D$ , ce qui rend l'application de la validation croisée relativement facile comparé à d'autres distributions pour lesquelles  $E[X_i]$  est une fonction non linéaire de  $\theta_X$ . Plusieurs familles exponentielles qui sont utilisées fréquemment, telles les distributions normales, binomiales, exponentielles, ou Poisson, satisfont ce critère.

Jusqu'à maintenant, nous avons traité  $\theta_X$  comme un scalaire. De façon plus générale, les paramètres peuvent être représentés par  $\theta_X = (\mu_X, \tau_X)$  où  $\mu_X$  est le paramètre d'intérêt,  $\tau_X$  est un scalaire ou un vecteur de paramètres de nuisance et  $E[X_i] = \phi(\theta_X)$ . Par abus de notation, nous allons parler indifféremment de  $\theta_X$  ou de  $\mu_X$ .

Prenons le cas de données normales, qu'on traite en détail dans le Chapitre 2, en guise d'illustration. On s'intéresse au paramètre de localisation de la distribution  $N(\mu_X, \sigma_X^2)$  de l'échantillon  $X$ . On a alors  $\phi(\theta_X) = \theta_X$  et  $\phi(\theta_Y) = \theta_Y$ , et le logarithme de la vraisemblance de  $\theta_X = \mu_X$  est

$$\log L(X; \theta_X) = \sum_{i=1}^n \left( -\log(\sqrt{2\pi}\sigma_X) - \frac{(X_i - \mu_X)^2}{2\sigma_X^2} \right)$$

et en posant sa dérivée partielle par rapport à  $\mu_X$  égale à zéro, on obtient l'estimateur usuel de vraisemblance maximale  $\hat{\mu}_X = \bar{X}$ . De manière similaire pour la distribution  $N(\mu_Y, \sigma_Y^2)$  de  $Y$  on a l'estimateur de vraisemblance maximale  $\hat{\theta}_Y = \hat{\mu}_Y = \bar{Y}$ .

Le logarithme de la vraisemblance pondérée est

$$\begin{aligned} \log WL(X, Y; \theta_X) &= \lambda \sum_{i=1}^n \left( -\log(\sqrt{2\pi}\sigma_X) - \frac{(X_i - \mu_X)^2}{2\sigma_X^2} \right) \\ &\quad + (1 - \lambda) \sum_{i=1}^n \left( -\log(\sqrt{2\pi}\sigma_X) - \frac{(Y_i - \mu_X)^2}{2\sigma_X^2} \right). \end{aligned}$$

Sa dérivée par rapport à  $\mu_X$  est

$$\frac{\partial}{\partial \mu_X} \log \text{WL}(X, Y; \theta_X) = \lambda \sum_{i=1}^n \frac{(X_i - \mu_X)}{\sigma_X^2} + (1 - \lambda) \sum_{i=1}^n \frac{(Y_i - \mu_X)}{\sigma_X^2},$$

alors en posant cette expression égale à zéro, on obtient facilement l'estimateur de vraisemblance pondérée de  $\theta_X$  :

$$\begin{aligned} \tilde{\theta}_X &= \tilde{\mu}_X \\ &= \lambda \bar{X} + (1 - \lambda) \bar{Y}. \end{aligned}$$

La mesure d'erreur quadratique moyenne de prévision devient

$$\begin{aligned} D(\lambda) &= \frac{1}{n} \sum_{i=1}^n (X_i - \phi(\tilde{\theta}_X^{(-i)}))^2 \\ &= \frac{1}{n} \sum_{i=1}^n (X_i - \phi(\tilde{\mu}_X^{(-i)}))^2 \\ &= \frac{1}{n} \sum_{i=1}^n [X_i - (\lambda \bar{X}^{(-i)} + (1 - \lambda) \bar{Y}^{(-i)})]^2 \end{aligned} \quad (1.2.3)$$

où  $\bar{X}^{(-i)} = \frac{1}{n-1} \sum_{j \neq i} X_j$  et  $\bar{Y}^{(-i)} = \frac{1}{n-1} \sum_{j \neq i} Y_j$ . Puisque  $\phi$  est une fonction linéaire de la moyenne, on peut trouver une solution explicite à  $\partial D / \partial \lambda = 0$  afin de minimiser  $D$ , comme on verra dans le Chapitre 2.

Pour le cas de données lognormales traité dans le Chapitre 3, la fonction  $\phi$  n'est plus linéaire. On s'intéresse au paramètre  $\mu_X$  de la distribution  $\text{LN}(\mu_X, \sigma_X^2)$  de l'échantillon  $X$ . On a alors  $\phi(\theta_X) = E[X_i] = e^{\theta_X + \frac{\sigma_X^2}{2}}$  et  $\phi(\theta_Y) = E[Y_i] = e^{\theta_Y + \frac{\sigma_Y^2}{2}}$ . Le logarithme de la vraisemblance de  $\theta_X = \mu_X$  est

$$\log L(X; \theta_X) = \sum_{i=1}^n \left( -\log(\sqrt{2\pi}\sigma_X) - \frac{(\log(X_i) - \mu_X)^2}{2\sigma_X^2} - \log(X_i) \right)$$

et en posant sa dérivée partielle par rapport à  $\mu_X$  égale à zéro, on obtient l'estimateur de vraisemblance maximale

$$\hat{\theta}_X = \frac{1}{n} \sum_{i=1}^n \log(X_i). \quad (1.2.4)$$

Le logarithme de la vraisemblance pondérée est

$$\begin{aligned} \log \text{WL}(X, Y; \theta_X) &= \lambda \sum_{i=1}^n \left( -\log(\sqrt{2\pi}\sigma_X) - \frac{(\log(X_i) - \mu_X)^2}{2\sigma_X^2} - \log(X_i) \right) \\ &\quad + (1 - \lambda) \sum_{i=1}^n \left( -\log(\sqrt{2\pi}\sigma_X) - \frac{(\log(Y_i) - \mu_X)^2}{2\sigma_X^2} - \log(Y_i) \right). \end{aligned}$$

Sa dérivée par rapport à  $\mu_X$  est

$$\frac{\partial}{\partial \mu_X} \log \text{WL}(X, Y; \theta_X) = \lambda \sum_{i=1}^n \frac{(\log(X_i) - \mu_X)}{\sigma_X^2} + (1 - \lambda) \sum_{i=1}^n \frac{(\log(Y_i) - \mu_X)}{\sigma_X^2},$$

et en posant cette expression égale à zéro, on obtient l'estimateur de vraisemblance pondérée de  $\theta_X$  :

$$\tilde{\theta}_X = \tilde{\mu}_X \tag{1.2.5}$$

$$= \lambda \sum_{i=1}^n \frac{\log(X_i)}{n} + (1 - \lambda) \sum_{i=1}^n \frac{\log(Y_i)}{n}. \tag{1.2.6}$$

Puisque la fonction  $\phi$  n'est pas linéaire en  $\mu_X$ , la situation va être plus compliquée en ce qui concerne  $D$  et la minimisation de  $D$ , ce qu'on verra en détail dans le Chapitre 3.

La validation croisée décrite ici est une méthode parmi les différentes techniques de rééchantillonnage. Efron (1983) affirme qu'une autre méthode de rééchantillonnage, le bootstrap, fonctionne mieux dans plusieurs cas. Au lieu d'analyser des sous-ensembles des données avec la validation croisée, le bootstrap analyse des sous-échantillons *aléatoires* des données. On passe maintenant à l'explication du bootstrap.

### 1.3. LE BOOTSTRAP

Le bootstrap d'Efron (1979) est une alternative à la validation croisée pour trouver des poids adaptatifs de vraisemblance pondérée.

L'idée du bootstrap est d'estimer la distribution  $F$  de nos données  $X_1, X_2, \dots, X_n$  par  $\tilde{F}$ , et ensuite d'estimer la caractéristique de la distribution d'une statistique qui nous intéresse par le calcul de la quantité équivalente à partir d'échantillons i.i.d. de  $\tilde{F}$ . Le bootstrap que nous appliquons est le bootstrap *non paramétrique*,

où  $\hat{F}_n = \frac{1}{n} \sum_{i=1}^n I\{X_i \leq x\}$  est la fonction de répartition expérimentale de l'échantillon de  $n$  données (où  $I$  dénote la fonction indicatrice). Créer des échantillons i.i.d. de taille  $n$  de  $\hat{F}_n$  consiste à rééchantillonner avec remise de notre échantillon de données original  $X_1, X_2, \dots, X_n$ , où chaque observation  $X_i$  a la même probabilité  $\frac{1}{n}$  de sélection. On utilise la notation conventionnelle qui met une étoile sur les quantités bootstrap, alors l'échantillon bootstrap se note  $X_1^*, X_2^*, \dots, X_n^*$  et la moyenne bootstrap est  $\bar{X}^* = \frac{1}{n} \sum_{i=1}^n X_i^*$ , par exemple. Le bootstrap sert surtout à estimer des caractéristiques de la distribution d'une statistique, comme la variance ou le biais, et ainsi l'erreur quadratique moyenne d'un estimateur. Pour plus de détails, voir Shao et Tu (1995, section 1.4).

Le bootstrap est particulièrement utile parce qu'il permet d'estimer directement l'erreur quadratique moyenne d'un estimateur de  $\theta_X$  sans passer par la prévision de données  $X_i$ , et ainsi sans avoir à passer par l'erreur quadratique moyenne de prévision. Comme on a vu dans la section 1.2, la validation croisée est basée sur la prévision et on utilise l'erreur quadratique moyenne de prévision donnée dans l'équation (1.2.1). Avec le bootstrap, on peut estimer directement l'erreur quadratique moyenne de l'estimateur de vraisemblance pondérée :

$$\begin{aligned} \text{EQM}(\lambda) &= E[(\tilde{\theta}_X - \theta_X)^2] \\ &= \text{Var}(\tilde{\theta}_X) + \text{Biais}^2(\tilde{\theta}_X) \end{aligned}$$

Pour cette raison on a l'intuition que le bootstrap pourrait s'avérer plus précis et plus efficace que la validation croisée comme outil pour choisir les poids de vraisemblance pondérée, idée qu'on explore ici pour les cas de données normales et lognormales.

Dans le cas de données normales considéré au Chapitre 2, on va utiliser la méthode du bootstrap afin d'estimer chaque terme dans l'expression suivante de l'erreur quadratique moyenne :

$$\begin{aligned} \text{EQM}(\lambda) &= \lambda^2 \text{Var}(\bar{X}) + (1 - \lambda)^2 \text{Var}(\bar{Y}) + 2\lambda(1 - \lambda) \text{Cov}(\bar{X}, \bar{Y}) \\ &\quad + (1 - \lambda)^2 (\mu_X - \mu_Y)^2, \end{aligned} \tag{1.3.1}$$

qui est la forme prise par l'erreur quadratique moyenne dans le cas de données normales.

Le bootstrap peut estimer l'erreur quadratique moyenne elle-même sans estimer chaque terme individuellement, mais on ne prend pas cette route puisqu'on va introduire un facteur pénalisant le dernier terme de l'équation (1.3.1), comme on verra plus tard dans la section 2.3.

#### 1.4. CONVENTIONS DE NOTATION

On mentionne quelques conventions de notation qu'on va suivre dans ce mémoire.

L'estimateur de vraisemblance maximale et l'estimateur de vraisemblance pondérée, les estimateurs qui maximisent les quantités  $L$  et  $WL$  de la section 1.1, seront notés  $EVM = \hat{\theta}$  et  $EVP = \tilde{\theta}$  respectivement.

Deux types de convergence vont nous concerner : la convergence en probabilité et la convergence en loi. Une suite de variables aléatoires  $X_1, X_2, \dots, X_n, \dots$  converge en probabilité vers  $X$ , noté  $X_n \xrightarrow{\text{Pr}} X$ , si

$$\lim_{n \rightarrow \infty} \Pr[|X_n - X| \geq \varepsilon] = 0$$

pour tout  $\varepsilon > 0$ .

Une suite de variables aléatoires  $X_1, X_2, \dots, X_n, \dots$ , suivant les distributions  $F_1, F_2, \dots, F_n, \dots$  respectivement, converge en loi vers  $X$  qui suit la distribution  $F$ , noté  $X_n \xrightarrow{L} X$ , si

$$\lim_{n \rightarrow \infty} F_n(x) = F(x)$$

pour tout point de continuité  $x$  de  $F$ .

On utilise aussi la notation *grand-O* qui est définie comme suit : pour des fonctions  $f$  et  $g$  en  $n$ ,  $f(n) = O(g(n))$  lorsque  $n \rightarrow \infty$  quand il existe  $n_0$  et une constante  $M$  tels que

$$|f(n)| \leq M|g(n)|$$

pour tout  $n > n_0$ .



En version stochastique, on a la définition suivante pour une suite de variables aléatoires  $X_n$  :  $X_n = O_P(1)$  quand pour tout  $\varepsilon > 0$ , il y a une constante  $M < \infty$  telle que

$$\Pr[|X_n| > M] < \varepsilon$$

pour tout  $n$ . On écrit  $X_n = O_P(f(n))$  quand

$$\frac{X_n}{f(n)} = O_P(1).$$

## 1.5. CHEMINEMENT DE CE MÉMOIRE

On a maintenant la base nécessaire pour procéder à l'examen du cas de données normales dans le Chapitre 2, et du cas de données lognormales dans le Chapitre 3.

Dans le Chapitre 2, on commence par le développement de la théorie du cas de données suivant une distribution normale : examen de l'erreur quadratique moyenne et de sa minimisation, le choix de  $\lambda$  par validation croisée et les propriétés asymptotiques de ce poids, y compris une discussion des problèmes avec ce poids et des améliorations utilisées par Wang et Zidek (2005a,b). Ensuite, on développe le choix du poids bootstrap et ses propriétés. La description et l'analyse de nos simulations et de leurs liens avec la théorie complètent le chapitre. Une démarche similaire est suivie dans le Chapitre 3, pour le cas particulier des données suivant une distribution lognormale.

La discussion générale et la conclusion forment le Chapitre 4, et finalement en annexe on trouve les programmes informatiques utilisés lors des simulations ainsi qu'une sélection de graphiques illustratifs.

# Chapitre 2

---

## DONNÉES NORMALES

Dans ce chapitre, nous allons considérer le cas de données normales. On a deux échantillons indépendants d'observations indépendantes et identiquement distribuées,  $X_i \sim N(\mu_X, \sigma_X^2)$  et  $Y_i \sim N(\mu_Y, \sigma_Y^2)$  pour  $i = 1, \dots, n$ , et on veut estimer le paramètre  $\theta_X = \mu_X$  de la première distribution. Comme on a vu dans la section 1.2, l'estimateur de vraisemblance maximale est simplement la moyenne,  $\bar{X}$  de la première population, et l'estimateur de vraisemblance pondérée est donc  $\hat{\mu}_X = \lambda\bar{X} + (1 - \lambda)\bar{Y}$ .

### 2.1. ERREUR QUADRATIQUE MOYENNE ET LA VALEUR OPTIMALE DE $\lambda$

L'erreur quadratique moyenne de l'estimateur de vraisemblance pondérée de  $\mu_X$  est

$$\begin{aligned} \text{EQM}(\lambda) &= \text{Var}(\lambda\bar{X} + (1 - \lambda)\bar{Y}) + \text{Biais}^2(\lambda\bar{X} + (1 - \lambda)\bar{Y}) \\ &= \lambda^2\text{Var}(\bar{X}) + (1 - \lambda)^2\text{Var}(\bar{Y}) + 2\lambda(1 - \lambda)\text{Cov}(\bar{X}, \bar{Y}) + \\ &\quad ((1 - \lambda)(\mu_Y - \mu_X))^2 \\ &= \lambda^2\frac{\sigma_X^2}{n} + (1 - \lambda)^2\frac{\sigma_Y^2}{n} + 0 + (1 - \lambda)^2(\mu_X - \mu_Y)^2 \\ &= \lambda^2\left(\frac{\sigma_X^2 + \sigma_Y^2}{n} + (\mu_X - \mu_Y)^2\right) - 2\lambda\left(\frac{\sigma_Y^2}{n} + (\mu_X - \mu_Y)^2\right) \\ &\quad + \frac{\sigma_Y^2}{n} + (\mu_X - \mu_Y)^2 \end{aligned} \tag{2.1.1}$$

On dérive par rapport à  $\lambda$  afin de trouver le poids  $\lambda$  qui minimise l'erreur quadratique moyenne.

$$\frac{\partial \text{EQM}}{\partial \lambda} = 2\lambda \left( \frac{\sigma_X^2 + \sigma_Y^2}{n} + (\mu_X - \mu_Y)^2 \right) - 2 \left( \frac{\sigma_Y^2}{n} + (\mu_X - \mu_Y)^2 \right)$$

En posant égal à 0, on obtient la valeur de  $\lambda$  qui minimise l'erreur quadratique moyenne,

$$\begin{aligned} \lambda_{opt} &= \frac{\frac{\sigma_Y^2}{n} + (\mu_X - \mu_Y)^2}{\frac{\sigma_X^2 + \sigma_Y^2}{n} + (\mu_X - \mu_Y)^2} \\ &= 1 - \frac{\frac{\sigma_X^2}{n}}{\frac{\sigma_X^2 + \sigma_Y^2}{n} + (\mu_X - \mu_Y)^2} \end{aligned} \quad (2.1.2)$$

Quand  $\mu_X \neq \mu_Y$  on voit que

$$1 - \lambda_{opt} = \frac{\sigma_X^2}{n(\mu_X - \mu_Y)^2} + O(n^{-2}).$$

Lorsque  $n \rightarrow \infty$ ,  $\lambda_{opt} \rightarrow 1$ , ce qui confirme l'intuition qu'asymptotiquement on veut mettre tout le poids sur le premier groupe.

Notons que quand  $\mu_X = \mu_Y$  la valeur optimale est de

$$\lambda_{opt} = 1 - \frac{\sigma_X^2}{\sigma_X^2 + \sigma_Y^2} = \frac{\sigma_Y^2}{\sigma_X^2 + \sigma_Y^2}. \quad (2.1.3)$$

Prenons un moment pour explorer le cas d'un  $\lambda = \lambda_0$  fixe, qui sera utile pour expliquer certaines découvertes lors des simulations.

Notons que l'erreur quadratique moyenne de l'estimateur de vraisemblance maximale se trouve en posant  $\lambda = 1$  dans l'équation (2.1.1) :

$$\begin{aligned} \text{EQM}(1) &= \left( \frac{\sigma_X^2 + \sigma_Y^2}{n} \right) - 2 \left( \frac{\sigma_Y^2}{n} \right) + \frac{\sigma_Y^2}{n} \\ &= \frac{\sigma_X^2}{n}. \end{aligned}$$

Si on a un  $\lambda_0$  fixe et si  $\mu_X = \mu_Y$ , alors l'équation (2.1.1) devient

$$\text{EQM}(\lambda_0) = \lambda_0^2 \left( \frac{\sigma_X^2 + \sigma_Y^2}{n} \right) - 2\lambda_0 \left( \frac{\sigma_Y^2}{n} \right) + \frac{\sigma_Y^2}{n}.$$

On a le ratio des erreurs quadratiques moyennes

$$\frac{\text{EQM}(\lambda_0)}{\text{EQM}(1)} = \lambda_0^2 \left( \frac{\sigma_X^2 + \sigma_Y^2}{\sigma_X^2} \right) - 2\lambda_0 \left( \frac{\sigma_Y^2}{\sigma_X^2} \right) + \frac{\sigma_Y^2}{\sigma_X^2}.$$

Pour  $0 < \lambda < 1$  il est facile de vérifier que ce ratio prend son maximum à une des bornes de l'intervalle. Quand  $\lambda = 0$  on a un ratio de  $\frac{\sigma_Y^2}{\sigma_X^2}$ , et quand  $\lambda = 1$  le ratio est de  $\frac{\sigma_X^2 + \sigma_Y^2}{\sigma_X^2} - \frac{\sigma_Y^2}{\sigma_X^2} = 1$ . Si on fait l'hypothèse que  $\sigma_X^2 = \sigma_Y^2$  (ce qu'on fera lors des simulations), on a alors que le ratio est inférieur à 1 pour tout  $0 < \lambda_0 < 1$ . Donc, quand les moyennes et variances sont égales, un tel poids fixe nous donne toujours une amélioration par rapport à l'estimateur de vraisemblance maximale. Ceci est intuitif quand on considère que l'estimateur de vraisemblance maximale utilise  $n$  observations, tandis que l'estimateur de vraisemblance en utilise  $2n$  qui viennent essentiellement de la même distribution dans ce cas.

Par contre, si  $\mu_X \neq \mu_Y$ , le ratio des erreurs quadratiques moyennes est

$$\begin{aligned} \frac{\text{EQM}(\lambda_0)}{\text{EQM}(1)} &= \frac{n}{\sigma_X^2} \left[ \lambda_0^2 \left( \frac{\sigma_X^2 + \sigma_Y^2}{n} + (\mu_X - \mu_Y)^2 \right) - 2\lambda_0 \left( \frac{\sigma_Y^2}{n} + (\mu_X - \mu_Y)^2 \right) \right. \\ &\quad \left. + \frac{\sigma_Y^2}{n} + (\mu_X - \mu_Y)^2 \right] \\ &= \lambda_0^2 \left( \frac{\sigma_X^2 + \sigma_Y^2}{\sigma_X^2} + n \frac{(\mu_X - \mu_Y)^2}{\sigma_X^2} \right) - 2\lambda_0 \left( \frac{\sigma_Y^2}{\sigma_X^2} + n \frac{(\mu_X - \mu_Y)^2}{\sigma_X^2} \right) \\ &\quad + \frac{\sigma_Y^2}{\sigma_X^2} + n \frac{(\mu_X - \mu_Y)^2}{\sigma_X^2} \\ &\longrightarrow \infty \text{ lorsque } n \rightarrow \infty. \end{aligned}$$

Parce que le terme correspondant au carré du biais est de l'ordre  $O(n)$ , quand les moyennes sont différentes, l'estimateur de vraisemblance pondérée va rapidement devenir pire que l'estimateur de vraisemblance maximale lorsque  $n$  augmente. Ceci est intuitif puisque le poids  $\lambda_0$  est fixe et la moitié des observations qu'on utilise viennent d'une distribution avec une moyenne différente de celle qu'on estime. La taille échantillonnale  $n_0$  pour laquelle le ratio est supérieur à 1 pour  $n \geq n_0$  dépend de la magnitude de la différence entre les moyennes, ainsi que la valeur exacte du poids  $\lambda_0$  fixe.

## 2.2. CHOIX DE $\lambda$ PAR VALIDATION CROISÉE

L'estimateur par validation croisée vise à minimiser la fonction d'écart  $D$ , qui prend la forme de l'équation (1.2.3) dans le cas présent :

$$D(\lambda) = \frac{1}{n} \sum_{i=1}^n (X_i - \lambda \bar{X}^{(-i)} - (1 - \lambda) \bar{Y}^{(-i)})^2$$

En posant égal à zéro la dérivée de cette fonction, on trouve la valeur

$$\hat{\lambda}_{CV} = 1 - \frac{\sum_{i=1}^n (\bar{X}^{(-i)} - \bar{Y}^{(-i)}) (\bar{X}^{(-i)} - X_i)}{\sum_{i=1}^n (\bar{X}^{(-i)} - \bar{Y}^{(-i)})}$$

Wang et Zidek (2005a, Lemme 2.1) montrent que ceci est équivalent à

$$\hat{\lambda}_{CV} = 1 - \frac{S_2}{S_1} \tag{2.2.1}$$

$$= 1 - \frac{\frac{n}{(n-1)^2} (\hat{\sigma}_x^2 - \widehat{\text{Cov}})}{\frac{n(n-2)}{(n-1)^2} (\bar{X} - \bar{Y})^2 + \frac{1}{n(n-1)^2} \sum_{i=1}^n (X_i - Y_i)^2} \tag{2.2.2}$$

où  $\hat{\sigma}_X^2 = \frac{1}{n} \sum_{i=1}^n (X_i - \bar{X})^2$  et  $\widehat{\text{Cov}} = \frac{1}{n} \sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y})$ .

On considère le numérateur  $S_2$  en premier. On a, par le théorème limite centrale (voir par exemple Serfling, 1980, p. 28),

$$\sqrt{n}(\hat{\sigma}_X^2 - \sigma_X^2) \xrightarrow{L} N(0, \tau_1^2)$$

et il s'ensuit  $\hat{\sigma}_X^2 = \sigma_X^2 + O_P(n^{-1/2})$ ; de manière similaire,

$$\sqrt{n}(\widehat{\text{Cov}} - \text{Cov}) \xrightarrow{L} N(0, \tau_2^2),$$

où la valeur des  $\tau^2$  n'est pas importante pour notre développement. Puisque les deux échantillons sont indépendants, la vraie covariance est  $\text{Cov} = 0$ , ce qui entraîne  $\widehat{\text{Cov}} = O_P(n^{-1/2})$ . On a donc

$$\begin{aligned} S_2 &= \frac{n}{(n-1)^2} (\sigma_X^2 + O_P(n^{-1/2}) + 0 + O_P(n^{-1/2})) \\ &= \frac{n}{(n-1)^2} (\sigma_X^2 + O_P(n^{-1/2})) \\ &= \frac{1}{n} (\sigma_X^2 + O_P(n^{-1/2})) \\ &= \frac{1}{n} \sigma_X^2 + O_P(n^{-3/2}). \end{aligned} \tag{2.2.3}$$

Examinons le dénominateur  $S_1$ . On note que

$$X_i = \mu_X + \sigma_X Z_{X_i}, \quad Y_i = \mu_Y + \sigma_Y Z_{Y_i}$$

où  $Z_{X_i} \stackrel{\text{i.i.d.}}{\sim} N(0, 1)$  indépendant de  $Z_{Y_i} \stackrel{\text{i.i.d.}}{\sim} N(0, 1)$ . On a aussi que

$$\bar{X} - \bar{Y} = \mu_X - \mu_Y + \sqrt{\frac{\sigma_X^2 + \sigma_Y^2}{n}} Z \quad (2.2.4)$$

où  $Z \sim N(0, 1)$ .

Le dénominateur est donc égal à

$$\begin{aligned} S_1 &= \frac{n(n-2)}{(n-1)^2} (\bar{X} - \bar{Y})^2 + \frac{1}{n(n-1)^2} \sum_{i=1}^n (X_i - Y_i)^2 \\ &= \frac{n(n-2)}{(n-1)^2} \left( \mu_X - \mu_Y + \frac{\sqrt{\sigma_X^2 + \sigma_Y^2}}{\sqrt{n}} Z \right)^2 \\ &\quad + \frac{1}{(n-1)^2} \left( \frac{1}{n} \sum_{i=1}^n (\mu_X - \mu_Y + \sigma_X Z_{X_i} - \sigma_Y Z_{Y_i})^2 \right) \\ &= \frac{n(n-2)}{(n-1)^2} \left( (\mu_X - \mu_Y)^2 + \frac{2(\mu_X - \mu_Y)\sqrt{\sigma_X^2 + \sigma_Y^2}}{\sqrt{n}} Z + \frac{\sigma_X^2 + \sigma_Y^2}{n} Z^2 \right) \\ &\quad + O_P(n^{-2}) \\ &= \left( 1 - \frac{1}{(n-1)^2} \right) \left( (\mu_X - \mu_Y)^2 + \frac{2(\mu_X - \mu_Y)\sqrt{\sigma_X^2 + \sigma_Y^2}}{\sqrt{n}} Z + \frac{\sigma_X^2 + \sigma_Y^2}{n} Z^2 \right) \\ &\quad + O_P(n^{-2}) \\ &= (\mu_X - \mu_Y)^2 + \frac{2(\mu_X - \mu_Y)\sqrt{\sigma_X^2 + \sigma_Y^2}}{\sqrt{n}} Z + \frac{\sigma_X^2 + \sigma_Y^2}{n} Z^2 + O_P(n^{-2}) \quad (2.2.5) \end{aligned}$$

Donc, pour  $n$  grand,

$$\begin{aligned} \frac{S_2}{S_1} &= \frac{\frac{\sigma_X^2}{n} + O_P(n^{-3/2})}{(\mu_X - \mu_Y)^2 + \frac{2(\mu_X - \mu_Y)\sqrt{\sigma_X^2 + \sigma_Y^2}}{\sqrt{n}} Z + \frac{\sigma_X^2 + \sigma_Y^2}{n} Z^2 + O_P(n^{-2})} \quad (2.2.6) \\ &= \begin{cases} \frac{\sigma_X^2}{n(\mu_X - \mu_Y)^2} + O_P(n^{-3/2}) & \text{si } \mu_X \neq \mu_Y, \\ \frac{\sigma_X^2}{\sigma_X^2 + \sigma_Y^2} \frac{1}{Z^2} + O_P(n^{-3/2}) & \text{si } \mu_X = \mu_Y. \end{cases} \end{aligned}$$

Examinons les deux cas. Quand  $\mu_X \neq \mu_Y$  on a que  $\frac{S_2}{S_1} \xrightarrow{\text{Pr}} 0$  lorsque  $n \rightarrow \infty$ , ce qui est équivalent à  $\hat{\lambda}_{CV} \xrightarrow{\text{Pr}} 1$ , c'est-à-dire mettre tout le poids sur la première population.

Notons que quand  $\mu_X \neq \mu_Y$ ,

$$\begin{aligned} \frac{1 - \hat{\lambda}_{CV}}{1 - \lambda_{opt}} &= \frac{\frac{\sigma_X^2}{n(\mu_X - \mu_Y)^2} + O_P(n^{-3/2})}{\frac{\sigma_X^2}{n(\mu_X - \mu_Y)^2} + O(n^{-2})} \\ &= 1 + O_P(n^{-1/2}). \end{aligned}$$

Par contre quand  $\mu_X = \mu_Y$  on a

$$1 - \hat{\lambda}_{CV} = \frac{\sigma_X^2}{\sigma_X^2 + \sigma_Y^2} \frac{1}{Z^2} + O_P(n^{-3/2}) \quad (2.2.7)$$

où  $Z$  est une variable aléatoire de loi  $N(0, 1)$ . Puisque  $E[\frac{1}{Z^2}] = \infty$ , l'espérance de  $1 - \hat{\lambda}_{CV}$  est donc infinie. Ainsi, la méthode de validation croisée ne fonctionne pas dans ce cas.

Wang et Zidek (2005a, Proposition 2.2) démontrent que si  $\mu_X \neq \mu_Y$ , alors

$$\hat{\lambda}_{CV} \xrightarrow{\text{Pr}} 1.$$

Ils affirment aussi que dans le cas  $\mu_X = \mu_Y$ , la limite asymptotique de  $\hat{\lambda}_{CV}$  n'existe pas.

Comme solution au problème de l'espérance infinie soulevée dans l'équation (2.2.7), Wang et Zidek (2005a, p.469) proposent l'addition d'une petite constante positive  $\delta$  au dénominateur  $S_1$  de l'équation (2.2.1). Ce dénominateur devient asymptotiquement

$$S'_1 = \delta + \frac{\sigma_X^2 + \sigma_Y^2}{n} Z^2 + O_P(n^{-2})$$

et donc l'estimateur est

$$\begin{aligned} 1 - \hat{\lambda}'_{CV} &= \frac{\frac{\sigma_X^2}{n} + O_P(n^{-3/2})}{\delta + \frac{\sigma_X^2 + \sigma_Y^2}{n} Z^2 + O_P(n^{-2})} \\ &= \frac{\sigma_X^2}{n\delta} + O_P(n^{-3/2}). \end{aligned} \quad (2.2.8)$$

Le problème de l'espérance infinie est réglée, mais  $1 - \hat{\lambda}'_{CV}$  tend vers 0 en probabilité lorsque  $n$  tend vers l'infini, et non vers la valeur optimale de  $\frac{\sigma_X^2}{\sigma_X^2 + \sigma_Y^2}$ , donc cette solution n'est pas complètement adéquate.

En fait, en pratique,  $\hat{\lambda}_{CV}$  a des problèmes même lorsque  $\mu_X \neq \mu_Y$ . Wang et Zidek (2005a, pp. 476-477) précisent que les valeurs dans leurs simulations sont

obtenues en utilisant un “terme de correction” afin de régler un problème qu’ils disent “d’instabilité numérique”. Comme on verra dans la section 2.4, le problème avec  $\hat{\lambda}_{CV}$  est sérieux. Leur terme de correction mène à un poids qu’on nomme  $\hat{\lambda}_{CVmod}$ . On prend cette opportunité pour discuter de sa raison d’être et du terme de correction lui-même.

On verra dans la section 2.4 que le poids  $\hat{\lambda}_{CV}$  peut prendre des valeurs négatives pour des valeurs de  $n \leq 300$ , ce qui n’a pas d’interprétation pratique intuitive, et qui mène à un estimateur peu performant.

Examinons de plus près comment on peut se trouver avec des poids négatifs dans le cas  $\mu_X \neq \mu_Y$ , cas dans lequel Wang et Zidek (2005a) ont développé leur Proposition 2.2 mentionnée ci-haut. De l’équation (2.2.6) on avait l’expression suivante pour  $\hat{\lambda}_{CV}$  :

$$1 - \frac{S_2}{S_1} = 1 - \frac{\frac{\sigma_X^2}{n} + O_P(n^{-3/2})}{(\mu_X - \mu_Y)^2 + \frac{2(\mu_X - \mu_Y)\sqrt{\sigma_X^2 + \sigma_Y^2}}{\sqrt{n}}Z + \frac{\sigma_X^2 + \sigma_Y^2}{n}Z^2 + O_P(n^{-2})}$$

Il est à noter que pour des petits  $n$ , le terme en  $\mu_X - \mu_Y$  peut facilement dominer  $S_1$ , au lieu du terme en  $(\mu_X - \mu_Y)^2$ , selon la réalisation de la variable aléatoire  $Z$ . Si on met de côté tous les termes d’ordre  $O_P(n^{-1})$  au numérateur ainsi qu’au dénominateur de l’expression précédente, on obtient l’expression plus simple

$$\bar{\lambda}_{CV} = 1 - \frac{\sigma_X^2}{n(\mu_X - \mu_Y)^2 + 2\sqrt{n}(\mu_X - \mu_Y)\sqrt{\sigma_X^2 + \sigma_Y^2}Z}. \quad (2.2.9)$$

On a simulé des histogrammes de cette nouvelle statistique simplifiée, basés sur 10000 valeurs de  $Z$ . L’approximation est qualitativement correcte; les histogrammes ont à peu près la même allure que les vrais histogrammes de  $\hat{\lambda}_{CV}$ . Utilisons l’approximation  $\bar{\lambda}_{CV}$  de l’équation (2.2.9) afin de pouvoir comprendre de manière plus simple les poids négatifs.

On a résolu l’équation (2.2.9) afin de trouver les valeurs de  $Z$  menant à un  $\bar{\lambda}_{CV}$  négatif. On a ensuite substitué les valeurs  $\sigma_X^2 = \sigma_Y^2 = 1$  et  $|\mu_X - \mu_Y| = 0,3$  qu’on prend lors des simulations, et on a calculé la probabilité que la valeur de  $Z \sim N(0, 1)$  soit dans l’intervalle menant à une valeur négative, pour plusieurs



$n$	intervalle où $\tilde{\lambda}_{CV} < 0$	$\Pr[\tilde{\lambda}_{CV} < 0]$
10	$Z < 0,037$	0,514
20	$Z < -0,210$	0,416
50	$Z < -0,583$	0,279
100	$Z < -0,942$	0,173
200	$Z < -1,416$	0,078
300	$Z < -1,769$	0,038
400	$Z < -2,062$	0,019

TAB. 2.1. Probabilité que l'approximation (2.2.9)  $\tilde{\lambda}_{CV}$  soit négative pour  $|\mu_X - \mu_Y| = 0,3$  et  $\sigma_X^2 = \sigma_Y^2 = 1$

valeurs de  $n$ . Ces probabilités nous donnent donc un estimé de la probabilité qu'on se trouve avec un  $\hat{\lambda}_{CV}$  négatif pour chaque  $n$ . Ces probabilités diminuent lorsque  $n$  augmente et sont résumées dans le tableau 2.1.

Wang et Zidek (2005b) tentent d'expliquer et d'améliorer cette situation, dans une prépublication. Ils font plusieurs suggestions pour corriger le problème, notamment de tronquer les poids négatifs à 0, ou, celle que nous poursuivons ici, l'ajout d'une variable aléatoire  $\delta(n, X, Y)$  au dénominateur  $S_1$ , comme ils ont fait pour leur article original Wang et Zidek (2005a).

L'ajout d'une constante  $\delta$ , mentionné plus tôt dans le contexte du problème asymptotique quand  $\mu_X = \mu_Y$ , n'aide pas cette nouvelle situation en pratique, parce que si la constante est trop petite, elle n'a aucun effet, tandis que si elle est trop grande, elle va forcer  $\hat{\lambda}_{CV}$  à être très près de 1. Wang et Zidek (2005b) développent alors une variable aléatoire  $\delta(n, X, Y)$  qui est ajoutée à  $S_1$  afin d'obtenir  $\hat{\lambda}_{CVmod}$ .

Puisque le problème se manifeste lorsque la différence empirique entre les deux moyennes est trop petite, le  $\delta(n, X, Y)$  choisi devrait en tenir compte. La valeur

proposée par Wang et Zidek (2005b) est

$$\begin{aligned}\delta(n, X, Y) &= \frac{\hat{\sigma}_X^2}{2n} + \frac{\hat{\sigma}_X^2}{\left(1 + \frac{|\bar{X} - \bar{Y}|}{\sqrt{\frac{\hat{\sigma}_X^2 + \hat{\sigma}_Y^2}{n}}}\right) \log(n)} \\ &= \frac{\hat{\sigma}_X^2}{2n} + \frac{\hat{\sigma}_X^2}{(1 + |t|) \log(n)}\end{aligned}$$

où  $t = \frac{\bar{X} - \bar{Y}}{\sqrt{\frac{\hat{\sigma}_X^2 + \hat{\sigma}_Y^2}{n}}}$  est une statistique  $t$  basée sur les deux échantillons, sauf qu'on prend  $\hat{\sigma}^2$  au lieu du  $s^2$  habituel. On aimerait comparer le comportement asymptotique de  $\hat{\lambda}_{CVmod}$  à  $\lambda_{opt}$  et à celui de  $\hat{\lambda}_{CV}$  développé plus tôt dans cette section.

Examinons d'abord le comportement de  $\frac{1}{1+|t|}$ . Il faut considérer les deux cas,  $\mu_X = \mu_Y$  et  $\mu_X \neq \mu_Y$ . On a

$$\begin{aligned}|t| &= \frac{\sqrt{n}|\bar{X} - \bar{Y}|}{\sqrt{\hat{\sigma}_X^2 + \hat{\sigma}_Y^2}} \\ &= \left| \frac{\sqrt{n}}{\sqrt{\hat{\sigma}_X^2 + \hat{\sigma}_Y^2}}(\mu_X - \mu_Y) + \sqrt{\frac{\sigma_X^2 + \sigma_Y^2}{\hat{\sigma}_X^2 + \hat{\sigma}_Y^2}} Z \right| \text{ où } Z \sim N(0, 1) \\ &= \left| \sqrt{\frac{n}{\sigma_X^2 + \sigma_Y^2 + O_P(n^{-1/2})}}(\mu_X - \mu_Y) + \sqrt{\frac{\sigma_X^2 + \sigma_Y^2}{\sigma_X^2 + \sigma_Y^2 + O_P(n^{-1/2})}} Z \right| \\ &= \left| \sqrt{\frac{n}{\sigma_X^2 + \sigma_Y^2} + O_P(n^{1/2})}(\mu_X - \mu_Y) + \sqrt{1 + O_P(n^{-1/2})} Z \right| \\ &= \left| \left( \sqrt{\frac{n}{\sigma_X^2 + \sigma_Y^2}} + O_P(1) \right) (\mu_X - \mu_Y) + (1 + O_P(n^{-1/2})) Z \right| \\ &= \left| \left( \sqrt{\frac{n}{\sigma_X^2 + \sigma_Y^2}} + O_P(1) \right) (\mu_X - \mu_Y) + Z + O_P(n^{-1/2}) \right| \\ &= \begin{cases} |Z + O_P(n^{-1/2})| = O_P(1) & \text{si } \mu_X = \mu_Y. \\ O_P(n^{1/2}) & \text{si } \mu_X \neq \mu_Y, \end{cases}\end{aligned}$$

On a donc

$$\begin{aligned} \frac{1}{1+|t|} &= \begin{cases} \frac{1}{1+O_P(1)} & \text{si } \mu_X = \mu_Y. \\ \frac{1}{1+O_P(n^{1/2})} & \text{si } \mu_X \neq \mu_Y, \end{cases} \\ &= \begin{cases} O_P(1) & \text{si } \mu_X = \mu_Y. \\ O_P(n^{-1/2}) & \text{si } \mu_X \neq \mu_Y, \end{cases} \end{aligned} \quad (2.2.10)$$

Maintenant on examine  $S_1 + \delta(n, X, Y)$ . Quand  $\mu_X = \mu_Y$ , on a

$$\begin{aligned} S_1 + \delta(n, X, Y) &= S_1 + \frac{1}{2n} \left( \sigma_X^2 + O_P(n^{-1/2}) \right) + \frac{\sigma_X^2 + O_P(n^{-1/2})}{(1+|t|)\log(n)} \\ &= S_1 + \frac{\sigma_X^2}{2n} + O_P(n^{-3/2}) + \left( \frac{\sigma_X^2}{\log(n)} + O_P(n^{-1/2} \log^{-1}(n)) \right) \left( \frac{1}{1+|t|} \right) \\ &= \left( \frac{\sigma_X^2}{\log(n)} + O_P(n^{-1/2} \log^{-1}(n)) \right) \left( \frac{1}{1+|t|} \right) + \frac{\sigma_X^2 + \sigma_Y^2}{n} Z^2 + \frac{\sigma_X^2}{2n} \\ &\quad + O_P(n^{-3/2}) \\ &= \frac{\sigma_X^2}{\log(n)} \left( \frac{1}{1+|t|} \right) + O_P(n^{-1/2} \log^{-1}(n)) \\ &= O_P(\log^{-1}(n)), \end{aligned} \quad (2.2.11)$$

et quand  $\mu_X \neq \mu_Y$ , on a

$$\begin{aligned} S_1 + \delta(n, X, Y) &= S_1 + \frac{1}{2n} \left( \sigma_X^2 + O_P(n^{-1/2}) \right) + \frac{\sigma_X^2 + O_P(n^{-1/2})}{(1+|t|)\log(n)} \\ &= S_1 + \frac{\sigma_X^2}{2n} + O_P(n^{-3/2}) + \left( \frac{\sigma_X^2}{\log(n)} + O_P(n^{-1/2} \log^{-1}(n)) \right) \left( \frac{1}{1+|t|} \right) \\ &= (\mu_X - \mu_Y)^2 + \frac{2(\mu_X - \mu_Y)\sqrt{\sigma_X^2 + \sigma_Y^2}}{\sqrt{n}} Z \\ &\quad + \left( \frac{\sigma_X^2}{\log(n)} + O_P(n^{-1/2} \log^{-1}(n)) \right) \left( \frac{1}{1+|t|} \right) \\ &\quad + \frac{\sigma_X^2 + \sigma_Y^2}{n} Z^2 + \frac{\sigma_X^2}{2n} + O_P(n^{-2}) \\ &= (\mu_X - \mu_Y)^2 + O_P(n^{-1/2}), \end{aligned} \quad (2.2.12)$$

où  $Z \sim N(0, 1)$ , en utilisant le développement pour  $S_1$  de l'équation (2.2.5), et en appliquant l'équation (2.2.10).

Cas	$\lambda_{opt}$ asy.	$\hat{\lambda}_{CV}$	$\hat{\lambda}'_{CV}$	$\hat{\lambda}_{CVmod}$
$\mu_X = \mu_Y$	$1 - \frac{\sigma_X^2}{\sigma_X^2 + \sigma_Y^2}$	$1 - \frac{\sigma_X^2}{\sigma_X^2 + \sigma_Y^2} \frac{1}{Z^2} + O_P(n^{-3/2})$	$1 + O_P(n^{-1})$	$1 + O_P(n^{-1} \log(n))$
$\mu_X \neq \mu_Y$	1	$1 + O_P(n^{-1})$	$1 + O_P(n^{-1})$	$1 + O_P(n^{-1})$

TAB. 2.2. Valeur asymptotique des poids par validation croisée

Le développement pour  $S_2$  vient de l'équation (2.2.3),  $S_2 = \frac{1}{n}\sigma_X^2 + O_P(n^{-3/2})$ . Prenons le cas  $\mu_X = \mu_Y$ . On obtient alors de l'équation (2.2.11),

$$\begin{aligned}
1 - \hat{\lambda}_{CVmod} &= \frac{S_2}{S_1 + \delta(n, X, Y)} \\
&= \frac{\frac{\sigma_X^2}{n} + O_P(n^{-3/2})}{O_P(\log^{-1}(n))} \\
&= O_P(n^{-1} \log(n)).
\end{aligned}$$

Dans le cas  $\mu_X \neq \mu_Y$ , on obtient de l'équation (2.2.12)

$$\begin{aligned}
1 - \hat{\lambda}_{CVmod} &= \frac{S_2}{S_1 + \delta(n, X, Y)} \\
&= \frac{\frac{\sigma_X^2}{n} + O_P(n^{-3/2})}{(\mu_X - \mu_Y)^2 + O_P(n^{-1/2})} \\
&= \frac{\sigma_X^2}{n(\mu_X - \mu_Y)^2} + O_P(n^{-3/2}).
\end{aligned}$$

Donc, dans les deux cas  $\mu_X \neq \mu_Y$  et  $\mu_X = \mu_Y$ ,  $\frac{S_2}{S_1 + \delta(n, X, Y)} \xrightarrow{\text{Pr}} 0$  lorsque  $n \rightarrow \infty$ . Ce comportement asymptotique concorde avec la valeur optimale dans le cas de  $\mu_X \neq \mu_Y$ , mais pas dans le cas  $\mu_X = \mu_Y$  où la valeur optimale est de  $\frac{\sigma_Y^2}{\sigma_X^2 + \sigma_Y^2}$ . Rappelons-nous que la même chose est survenue avec  $\hat{\lambda}'_{CV}$  après l'ajout du  $\delta$  constant requis au dénominateur dans l'équation (2.2.8). Le tableau 2.2 résume en présentant la valeur asymptotique de chaque poids développé dans cette section ainsi que le poids asymptotique optimal dans les deux cas.

On verra dans la section 2.4 que  $\hat{\lambda}_{CVmod}$  améliore grandement les estimations par rapport à  $\hat{\lambda}_{CV}$ , et en fait performe très bien. Par contre, comme le mentionnent Wang et Zidek (2005b), le terme  $\delta(n, X, Y)$  reste un peu arbitraire et sans justification théorique définitive. Mais on peut au moins voir comment ça

aide à résoudre le problème de poids négatifs qu'on a avec  $\hat{\lambda}_{CV}$ , en reprenant la même sorte d'analyse par approximation qu'on a fait avec l'équation (2.2.9) pour le cas de moyennes différentes : si on enlève cette fois les termes d'ordre  $O_P(n^{-1/2} \log^{-1}(n))$  du numérateur  $S_2$  et du dénominateur  $S_1 + \delta(n, X, Y)$  de l'équation (2.2.12), on obtient

$$\begin{aligned} 1 - \tilde{\lambda}_{CVmod} &= \frac{\frac{\sigma_X^2}{n}}{(\mu_X - \mu_Y)^2 + \frac{2(\mu_X - \mu_Y)\sqrt{\sigma_X^2 + \sigma_Y^2}}{\sqrt{n}}Z + \frac{\sigma_X^2}{\log(n)}\left(\frac{1}{1+|t|}\right)} \\ &= \frac{\sigma_X^2}{n(\mu_X - \mu_Y)^2 + 2\sqrt{n}(\mu_X - \mu_Y)\sqrt{\sigma_X^2 + \sigma_Y^2}Z + \frac{n}{\log(n)}\sigma_X^2\left(\frac{1}{1+|t|}\right)}. \end{aligned}$$

En comparant à l'équation (2.2.9), on voit que maintenant on a le terme additionnel en  $\frac{1}{1+|t|}$ , qui est de l'ordre de  $O_P\left(\frac{n^{1/2}}{\log(n)}\right)$  puisque  $\frac{1}{1+|t|} = O_P(n^{-1/2})$ . On a fait une simulation, à partir de 1000  $Z$  et  $t$ , des valeurs de  $\tilde{\lambda}_{CVmod}$  pour  $|\mu_X - \mu_Y| = 0,3$  et  $\sigma_X^2 = \sigma_Y^2 = 1$ , pour plusieurs valeurs de  $n$ , et on a calculé le pourcentage de poids négatifs parmi les poids simulés. Les résultats de cette simulation se trouvent dans le tableau 2.3. On voit que le pourcentage de valeurs négatives est beaucoup plus petit que les probabilités que  $\tilde{\lambda}_{CV}$  soit négatif, données dans le tableau 2.1. À noter que cette simulation de  $\tilde{\lambda}_{CVmod}$  n'est pas une approximation correcte puisqu'elle traite  $Z$  et  $t$  comme variables aléatoires indépendantes tandis qu'elles sont en réalité dépendantes. Par contre, cette simulation nous donne une indication de la raison pour laquelle  $\delta(n, X, Y)$  résout le problème des poids négatifs : le nouveau terme d'ordre  $O_P\left(\frac{n^{1/2}}{\log(n)}\right)$  au dénominateur de l'équation (2.2.12) rend  $\hat{\lambda}_{CVmod}$  beaucoup plus stable par rapport à  $\hat{\lambda}_{CV}$ .

On conclut cette section en mentionnant qu'on dénote l'estimateur résultant du choix de poids par validation croisée par

$$EVP_{CV} = \hat{\lambda}_{CV}\bar{X} + (1 - \hat{\lambda}_{CV})\bar{Y},$$

et de manière similaire on définit  $EVP_{CVmod}$ .

$n$	Pourcentage de $\tilde{\lambda}_{CVmod} < 0$
10	0,087
20	0,041
50	0,012
100	0,003
200	0
300	0
400	0

TAB. 2.3. Pourcentages de valeurs négatives de  $\tilde{\lambda}_{CV}$  lors de sa simulation de taille 1000, pour  $|\mu_X - \mu_Y| = 0,3$  et  $\sigma_X^2 = 1$

### 2.3. CHOIX DE $\lambda$ PAR BOOTSTRAP

Comme on a vu dans la section 1.3, on utilise le bootstrap pour estimer chaque terme dans l'expression de l'erreur quadratique moyenne,

$$EQM(\lambda) = \lambda^2 \text{Var}(\bar{X}) + (1-\lambda)^2 \text{Var}(\bar{Y}) + 2\lambda(1-\lambda) \text{Cov}(\bar{X}, \bar{Y}) + (1-\lambda)^2 (\mu_X - \mu_Y)^2.$$

On a

$$\begin{aligned} \text{Var}^*(\bar{X}^*) &= \frac{1}{n^2} \sum_{i=1}^n \text{Var}^*(X_i^*) \\ &= \frac{1}{n} \text{Var}^*(X_1^*) \\ &= \frac{1}{n} \left( \sum_{i=1}^n (X_i - \bar{X})^2 \text{Pr}(X_1^* = X_i) \right) \\ &= \frac{1}{n} \left( \frac{1}{n} \sum_{i=1}^n (X_i - \bar{X})^2 \right) \\ &= \frac{1}{n} \hat{\sigma}_X^2, \end{aligned}$$

et de la même façon,  $\text{Var}^*(\bar{Y}^*) = \frac{1}{n} \hat{\sigma}_Y^2$ .

On applique le bootstrap de manière dépendante, c'est-à-dire les échantillons  $X$  et  $Y$  sont tirés en même temps. Cette façon de faire est conservatrice. Pour

cette raison,  $X_i^*$  et  $Y_i^*$  sont corrélées. Le terme de la covariance est donc

$$\begin{aligned}
\text{Cov}^*(\bar{X}^*, \bar{Y}^*) &= \frac{1}{n^2} \sum_{i=1}^n \sum_{j=1}^n \text{Cov}^*(X_i^*, Y_j^*) \\
&= \frac{1}{n^2} \left( n \text{Cov}^*(X_1^*, Y_1^*) \right) \\
&= \frac{1}{n} \sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y}) \Pr(X_1^* = X_i, Y_1^* = Y_i) \\
&= \frac{1}{n^2} \sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y}) \\
&= \frac{1}{n} \widehat{\text{Cov}}.
\end{aligned}$$

Dans le dernier terme de l'expression de l'erreur quadratique moyenne, on remplace  $\mu_X$  et  $\mu_Y$  par leurs estimés bootstrap,  $E^*[X^*] = \bar{X}$  et  $E^*[Y^*] = \bar{Y}$  respectivement. On a donc

$$\begin{aligned}
\text{EQM}^*(\lambda) &= \lambda^2 \text{Var}^*(\bar{X}^*) + (1 - \lambda)^2 \text{Var}^*(\bar{Y}^*) + 2\lambda(1 - \lambda) \text{Cov}^*(\bar{X}^*, \bar{Y}^*) \\
&\quad + (1 - \lambda)^2 (E^*[X^*] - E^*[Y^*])^2 \\
&= \lambda^2 \frac{\hat{\sigma}_X^2}{n} + (1 - \lambda)^2 \frac{\hat{\sigma}_Y^2}{n} + 2\lambda(1 - \lambda) \frac{\widehat{\text{Cov}}}{n} + (1 - \lambda)^2 (\bar{X} - \bar{Y})^2.
\end{aligned}$$

Notons l'ordre de grandeur des termes. Les deux premiers termes en  $\frac{\hat{\sigma}_X^2}{n}$  et  $\frac{\hat{\sigma}_Y^2}{n}$  sont d'ordre de grandeur  $O_P(n^{-1})$ . Le troisième terme, en  $\frac{\widehat{\text{Cov}}}{n}$ , est d'ordre  $O_P(n^{-3/2})$ , puisque  $\widehat{\text{Cov}} = O_P(n^{-1/2})$ . Finalement, si  $\mu_X \neq \mu_Y$ , le dernier terme, le carré du biais, est d'ordre  $O_P(1)$  puisque  $(\bar{X} - \bar{Y})^2 = (\mu_X - \mu_Y)^2 + O_P(n^{-1/2})$ . Ce dernier terme domine ainsi l'expression dans ce cas.

Si par contre on a  $\mu_X = \mu_Y$ , alors le vrai biais est de 0. Si on laisse le terme du biais tel quel, on aurait, de l'équation (2.2.4)  $(\bar{X} - \bar{Y})^2 = \frac{\sigma_X^2 + \sigma_Y^2}{n} Z^2 = O_P(n^{-1})$ . Ce terme serait donc de même ordre  $O_P(n^{-1})$  que les deux premiers termes et ainsi pourrait apporter une influence non désirée. Dans ce cas, on aimerait pénaliser le terme du biais pour qu'il converge vers 0 plus rapidement, afin qu'il n'ait pas d'influence importante sur notre minimisation de l'erreur quadratique moyenne.

Par contre, si  $\mu_X \neq \mu_Y$ , on veut que ce terme de biais continue à dominer les autres, afin de ne pas gravement sous-estimer le biais et ainsi l'erreur quadratique moyenne, alors il ne faut pas trop pénaliser le terme. On introduit un facteur

$c_n = O(n^{-p})$  au dernier terme; il faut trouver une valeur de  $c_n$  qui atteint le compromis désiré (et donc on discute du choix de  $p$  plus loin). Notre formule de  $\text{EQM}_p^*$  devient, avec  $c_n^{-p}$ ,

$$\text{EQM}_p^*(\lambda) = \lambda^2 \frac{\hat{\sigma}_X^2}{n} + (1-\lambda)^2 \frac{\hat{\sigma}_Y^2}{n} + 2\lambda(1-\lambda) \frac{\widehat{\text{Cov}}}{n} + (1-\lambda)^2 n^{-p} (\bar{X} - \bar{Y})^2.$$

En dérivant par rapport à  $\lambda$ , on obtient

$$\begin{aligned} \frac{\partial \text{EQM}_p^*}{\partial \lambda} &= 2\lambda \frac{\hat{\sigma}_X^2}{n} - 2(1-\lambda) \frac{\hat{\sigma}_Y^2}{n} + 2(1-2\lambda) \frac{\widehat{\text{Cov}}}{n} - 2(1-\lambda) n^{-p} (\bar{X} - \bar{Y})^2 \\ &= 2\lambda \left( \frac{\hat{\sigma}_X^2}{n} + \frac{\hat{\sigma}_Y^2}{n} - \frac{2\widehat{\text{Cov}}}{n} + n^{-p} (\bar{X} - \bar{Y})^2 \right) + \frac{2\widehat{\text{Cov}}}{n} - 2 \frac{\hat{\sigma}_Y^2}{n} \\ &\quad - 2n^{-p} (\bar{X} - \bar{Y})^2 \end{aligned}$$

et notre estimé bootstrap de  $\lambda_{opt}$  suit en posant l'équation précédente égale à 0 :

$$\begin{aligned} \hat{\lambda}_p^* &= \frac{\frac{\hat{\sigma}_Y^2}{n} - \frac{\widehat{\text{Cov}}}{n} + n^{-p} (\bar{X} - \bar{Y})^2}{\frac{\hat{\sigma}_X^2}{n} + \frac{\hat{\sigma}_Y^2}{n} - \frac{2\widehat{\text{Cov}}}{n} + n^{-p} (\bar{X} - \bar{Y})^2} \\ &= 1 - \frac{\frac{\hat{\sigma}_X^2}{n} - \frac{\widehat{\text{Cov}}}{n}}{\frac{\hat{\sigma}_X^2}{n} + \frac{\hat{\sigma}_Y^2}{n} - \frac{2\widehat{\text{Cov}}}{n} + n^{-p} (\bar{X} - \bar{Y})^2} \\ &= 1 - \frac{\hat{\sigma}_X^2 - \widehat{\text{Cov}}}{\hat{\sigma}_X^2 + \hat{\sigma}_Y^2 - 2\widehat{\text{Cov}} + n^{1-p} (\bar{X} - \bar{Y})^2} \end{aligned} \quad (2.3.1)$$

Afin de choisir  $p$  pour que ceci converge à la bonne valeur de  $\lambda_{opt}$  il faut considérer les deux cas  $\mu_X = \mu_Y$  et  $\mu_X \neq \mu_Y$ .

Considérons le cas  $\mu_X = \mu_Y$ . On veut alors que  $\hat{\lambda}_p^* \xrightarrow{\text{Pr}} 1 - \frac{\sigma_X^2}{\sigma_X^2 + \sigma_Y^2}$  comme on a vu dans l'équation (2.1.3). Dans ce cas, on a  $\bar{X} - \bar{Y} = \frac{\sqrt{\sigma_X^2 + \sigma_Y^2}}{\sqrt{n}} Z$  où  $Z \sim N(0, 1)$ .



En se rappelant que  $\widehat{\text{Cov}} = O_P(n^{-1/2})$ , l'équation (2.3.1) nous donne

$$\begin{aligned}
\hat{\lambda}_p^* &= 1 - \frac{\hat{\sigma}_X^2 - \widehat{\text{Cov}}}{\hat{\sigma}_X^2 + \hat{\sigma}_Y^2 - 2\widehat{\text{Cov}} + n^{1-p}(\bar{X} - \bar{Y})^2} \\
&= 1 - \frac{\hat{\sigma}_X^2 - \widehat{\text{Cov}}}{\hat{\sigma}_X^2 + \hat{\sigma}_Y^2 - 2\widehat{\text{Cov}} + n^{1-p} \left( \frac{\sigma_X^2 + \sigma_Y^2}{n} Z^2 \right)} \\
&= 1 - \frac{\hat{\sigma}_X^2 - \widehat{\text{Cov}}}{\hat{\sigma}_X^2 + \hat{\sigma}_Y^2 - 2\widehat{\text{Cov}} + n^{-p}(\sigma_X^2 + \sigma_Y^2)Z^2} \\
&= 1 - \frac{\sigma_X^2 + O_P(n^{-1/2})}{\sigma_X^2 + \sigma_Y^2 + O_P(n^{-1/2}) + n^{-p}(\sigma_X^2 + \sigma_Y^2)Z^2} \\
&\xrightarrow{\text{Pr}} \begin{cases} 1 - \frac{\sigma_X^2}{\sigma_X^2 + \sigma_Y^2} & \text{si } p > 0, \\ 1 - \frac{\sigma_X^2}{\sigma_X^2 + \sigma_Y^2} \left( \frac{1}{1+Z^2} \right) & \text{si } p = 0, \\ 1 & \text{si } p < 0. \end{cases}
\end{aligned}$$

Dans le cas  $p = 0$ ,  $\hat{\lambda}_0^*$  converge vers une quantité aléatoire. Puisque  $E\left[\frac{1}{1+Z^2}\right] \approx 0,65$ , dans ce cas  $1 - \hat{\lambda}_0^*$  converge vers une variable aléatoire dont l'espérance est plus petite que ce qu'on voudrait.

Dans le cas  $\mu_X \neq \mu_Y$ , on veut que  $\hat{\lambda}_p^* \xrightarrow{\text{Pr}} 1$  comme on a vu dans la section 2.1. Puisque l'équation (2.2.4) nous montre que  $(\bar{X} - \bar{Y})^2 = O_P(1)$  dans ce cas, l'équation (2.3.1) converge en probabilité vers

$$\begin{aligned}
\hat{\lambda}_p^* &= 1 - \frac{\hat{\sigma}_X^2 - \widehat{\text{Cov}}}{\hat{\sigma}_X^2 + \hat{\sigma}_Y^2 - 2\widehat{\text{Cov}} + n^{1-p}(\bar{X} - \bar{Y})^2} \\
&= 1 - \frac{\sigma_X^2 + O_P(n^{-1/2})}{\sigma_X^2 + \sigma_Y^2 + O_P(n^{-1/2}) + (\mu_X - \mu_Y)^2 n^{1-p} + O_P(n^{1/2-p})} \\
&\xrightarrow{\text{Pr}} \begin{cases} 1 & \text{si } p < 1, \\ 1 - \frac{\sigma_X^2}{\sigma_X^2 + \sigma_Y^2 + (\mu_X - \mu_Y)^2} & \text{si } p = 1. \\ 1 - \frac{\sigma_X^2}{\sigma_X^2 + \sigma_Y^2} & \text{si } p > 1. \end{cases}
\end{aligned}$$

$\lambda_{opt} \text{ asy.}$	$p < 0$	$p = 0$	$p > 0$
$1 - \frac{\sigma_X^2}{\sigma_X^2 + \sigma_Y^2}$	$1 + O_P(n^p)$	$1 - \frac{\sigma_X^2}{\sigma_X^2 + \sigma_Y^2} \left( \frac{1}{1+Z^2} \right) + O_P(n^{-1/2})$	$1 - \frac{\sigma_X^2}{\sigma_X^2 + \sigma_Y^2} + O_P(n^{-\min\{1/2, p\}})$

TAB. 2.4. Valeur asymptotique des poids bootstrap quand  $\mu_X = \mu_Y$ 

$\lambda_{opt} \text{ asy.}$	$p < 1$	$p = 1$	$p > 1$
1	$1 + O_P(n^{1-p})$	$1 - \frac{\sigma_X^2}{\sigma_X^2 + \sigma_Y^2 + (\mu_X - \mu_Y)^2} + O_P(n^{-1/2})$	$1 - \frac{\sigma_X^2}{\sigma_X^2 + \sigma_Y^2} + O_P(n^{-\min\{1/2, p-1\}})$

TAB. 2.5. Valeur asymptotique des poids bootstrap quand  $\mu_X \neq \mu_Y$ 

Les tableaux 2.4 et 2.5 résument les cas en termes du  $\hat{\lambda}_p^*$  asymptotique. On voit qu'il faut  $0 < p < 1$  pour que notre estimation fonctionne bien asymptotiquement dans les deux cas.

On va dénoter l'estimateur résultant de ce choix de poids par bootstrap par

$$EVP_p^* = \hat{\lambda}_p^* \bar{X} + (1 - \hat{\lambda}_p^*) \bar{Y}.$$

On procède maintenant à une simulation du cas de données normales. On verra que, bien qu'on vient de voir que selon la théorie on a besoin de choisir  $p > 0$ , en pratique  $p = 0$  est le choix de  $p$  qui mène aux meilleures estimations en général, c'est-à-dire quand on ignore si les moyennes des deux échantillons sont égales ou non.

## 2.4. SIMULATIONS

Des simulations ont été effectuées afin d'examiner et de comparer la performance des estimateurs par validation croisée et bootstrap. Les simulations ont été faites à l'aide du logiciel R. Les programmes se trouvent dans l'annexe A.

Deux échantillons de taille  $n$  ont été tirés des distributions  $N(\mu_X, \sigma_X^2)$  et  $N(\mu_Y, \sigma_Y^2)$  respectivement. Les poids  $\hat{\lambda}_{CV}$ ,  $\hat{\lambda}_{CVmod}$ , et  $\hat{\lambda}_p^*$  pour plusieurs valeurs de  $p$  ont été calculés, ainsi que les estimateurs de vraisemblance pondérée  $EVP_{CV}$ ,  $EVP_{CVmod}$ ,  $EVP_p^*$  résultants et leur erreur quadratique moyenne. L'estimateur de vraisemblance maximale et son erreur quadratique moyenne ont aussi été calculés pour fin de comparaison. Ces étapes ont été répétées 1000 fois, et les moyenne et écart type de chaque quantité d'intérêt calculés.

Un estimateur  $\hat{\lambda}_W^*$  a aussi été créé basé sur un test de Welch de  $H_0 : \mu_X = \mu_Y$  versus  $H_1 : \mu_X \neq \mu_Y$ . Puisque la théorie nous suggère un  $p$  près de zéro sous  $H_1$  mais un  $p$  plus élevé sous  $H_0$ , cet estimateur choisit  $p = 0,05$  si la valeur- $p$  du test Welch est inférieure à 50%, mais  $p = 0,5$  sinon. La raison pour le niveau très élevé de ce test est que la théorie de la section 2.3 nous suggère un  $p$  élevé lorsque  $\mu_X = \mu_Y$ , mais un  $p$  près de 0 quand  $\mu_X \neq \mu_Y$ . Si on se trompe en mettant un  $p$  élevé lorsqu'en fait les moyennes sont différentes, le prix à payer va être lourd, comme on confirmera dans cette section. Donc, on prend un niveau de 50% pour le test de Welch afin de s'assurer de ne pas faire cette erreur coûteuse trop souvent. Cet estimateur de Welch a aussi été calculé avec le choix de  $p$  alternatif  $p = 0,25$  mais les résultats sont très semblables à ceux pour  $p = 0,5$ .

Les simulations ont été faites pour  $\mu_X = \mu_Y$ , avec plusieurs valeurs de  $n$  entre 10 et 150, et pour  $|\mu_X - \mu_Y| = 0,3$  avec plusieurs valeurs de  $n$  entre 10 et 500, ainsi que quelques valeurs de  $n$  plus élevées. Toutes les simulations ont été faites pour  $\sigma_X^2 = 1 = \sigma_Y^2$ .

Des exemples d'histogrammes des divers estimateurs, pour des  $n$  entre 10 et 150, quand  $\mu_X = \mu_Y$  ainsi que quand  $\mu_X \neq \mu_Y$ , se trouvent dans la section A.2. La première chose à remarquer dans tous les cas est que le poids  $\hat{\lambda}_{CV}$  ne se comporte pas bien du tout, comme on a mentionné dans la section 2.2 quand on a présenté le terme de correction  $\delta(n, X, Y)$  de Wang et Zidek (2005b). Il est souhaitable que le poids soit entre 0 et 1, par contre le poids  $\hat{\lambda}_{CV}$  prend souvent des valeurs négatives et varie énormément. En se référant aux histogrammes de la section A.2.2 on voit que ce n'est seulement à partir de  $n = 300$  où ce poids ne prend plus de valeurs négatives dans le cas  $\mu_X \neq \mu_Y$ . On a vu en détail dans la section 2.2 une explication de ce phénomène, lors de la discussion de l'équation (2.2.9). Dans le cas de moyennes égales dans la section A.2.1, la situation est pire, et rendu à  $n = 1000$  il y a des valeurs négatives de  $\hat{\lambda}_{CV}$  de l'ordre -500. Ceci est à cause de l'espérance infinie de  $\hat{\lambda}_{CV}$  dans ce cas, qu'on a noté en discutant l'équation (2.2.7). On voit que "l'instabilité" de  $\hat{\lambda}_{CV}$  dans le cas de moyennes différentes dont parlent Wang et Zidek (2005a) est un problème très sérieux, et que la situation n'est pas mieux quand les moyennes sont égales. Leur terme de

correction  $\delta(n, X, Y)$  qui mène au poids  $\hat{\lambda}_{CVmod}$  apporte de gros changements à la détermination du poids. En fait, les histogrammes nous donnent l'impression que  $\hat{\lambda}_{CVmod}$  est le poids qui a la plus petite variance. On confirmera ceci en examinant les poids et les estimateurs résultants de plus près.

Les histogrammes des autres poids nous indiquent des distributions à peu près normales, sauf pour  $\hat{\lambda}_p^*$  avec les plus petites valeurs de  $p$  dans le cas  $\mu_X \neq \mu_Y$ , et pour le poids Welch, pour lequel les distributions sont plus asymétriques. Le poids Welch a une distribution bimodale pour les  $n$  de taille modérée, ce qui est prévu.

On trouve un petit nombre de valeurs de  $\hat{\lambda}_p^*$  négatives quand  $n = 10$ , pour tous les  $p$ , quand les moyennes sont égales ainsi que quand elles sont différentes. De plus on y trouve des valeurs plus grandes que 1. Mais ce phénomène est unique à  $n = 10$ , toutes les valeurs de  $\hat{\lambda}_p^*$  sont entre 0 et 1, comme désiré, pour  $n \geq 20$ .

Les résultats concernant les poids et leur écart type se trouvent dans les tableaux 2.6 et 2.7 pour les cas  $\mu_X = \mu_Y$  et  $\mu_X \neq \mu_Y$  respectivement. Cinq poids bootstrap sont affichés, soit avec  $p = 0; 0,05; 0,25; 0,5$  et  $0,75$ .

On remarque en premier les valeurs négatives et la grande variance du poids  $\hat{\lambda}_{CV}$ , ainsi que la petite variance de  $\hat{\lambda}_{CVmod}$ , déjà mentionnées. On note que dans le cas  $\mu_X = \mu_Y$  le poids bootstrap avec  $p = 0,75$  mène aux poids les plus près du poids optimal de  $\lambda_{opt} = 0,5$  ce qui concorde avec la théorie développée dans la section 2.3. Dans le cas  $\mu_X \neq \mu_Y$ , encore une fois comme prévu, c'est  $p = 0$  et  $p = 0,05$  qui mènent aux poids les plus près du poids optimal. Le poids Welch est aussi très similaire à celui avec  $p = 0,05$  en général, ce qui est bon quand les moyennes sont différentes mais moins bon quand elles sont égales. Jouer avec différents niveaux pour le test de Welch pourrait trouver un meilleur équilibre entre la performance du poids Welch sous les deux hypothèses.

Les tableaux 2.8 et 2.9 présentent les erreurs quadratiques moyennes ainsi que l'écart type du carré des différences entre l'estimateur et le paramètre, pour les estimateurs pondérés résultant des poids choisis. Dans le cas  $\mu_X = \mu_Y$  l'estimateur  $EVP_{0,75}^*$  a l'erreur quadratique moyenne la plus petite, et dans le cas  $\mu_X \neq \mu_Y$  ce sont les estimateurs  $EVP_{CVmod}$ ,  $EVP_0^*$  et  $EVP_{0,05}^*$ , suivis de  $EVP_W^*$  qui ont les

erreurs quadratiques moyennes les plus petites à partir d'environ  $n \geq 40$  tandis que  $p = 0,75$  et  $p = 0,5$  donnent les erreurs quadratiques moyennes les plus petites pour  $n = 10$  et  $20$  respectivement. Par contre, il faut noter que les écarts type de toutes les différences au carré entre l'estimateur et le paramètre sont relativement élevés.

L'erreur quadratique moyenne de chaque estimateur a été comparé à celle de l'estimateur de vraisemblance maximale. Les ratios de ces erreurs quadratiques moyennes sont présentés dans les tableaux 2.10 et 2.11 pour  $\mu_X = \mu_Y$  et  $\mu_X \neq \mu_Y$  respectivement. Dans le cas de moyennes différentes, des valeurs de  $n$  jusqu'à 10000 sont incluses afin d'illustrer que les poids n'ont pas encore commencé à converger vers le poids optimal, discussion qu'on revisite après avoir discuté du cas de moyennes égales qui est plus simple.

Dans le cas  $\mu_X = \mu_Y$ , on voit que le ratio est minimisé dans le cas du poids  $\hat{\lambda}_{0,75}^*$ , mais qu'à l'exception de  $\hat{\lambda}_{CV}$  tous les autres poids donnent un ratio inférieur à 1, indiquant une amélioration par rapport à l'estimateur de vraisemblance maximale. Le poids  $\hat{\lambda}_{CV}$  quant à lui prend des valeurs supérieures à 1 qui augmentent avec  $n$ . Aussi à noter, le ratio pour chaque poids bootstrap diminue lorsque  $n$  augmente, tandis que celui pour  $\hat{\lambda}_{CVmod}$  augmente avec  $n$  tout en restant inférieur à 1 (des simulations pour quelques valeurs de  $n$  plus élevés ont montré que lorsque  $n = 1000$ , ce ratio est de 0,96, et lorsque  $n = 5000$ , il est de 0,99). Toutes les tendances mentionnées continuent pour ces  $n$  plus élevés.

Dans le cas  $\mu_X \neq \mu_Y$  la situation est un peu plus compliquée. Pour la plupart des estimateurs pondérés le ratio augmente vers un maximum qui est supérieur à 1, avant de redescendre vers 1. Pour les valeurs plus élevées de  $p$ ,  $p = 0,5$  et  $0,75$ , par contre, les ratios pour les poids bootstrap deviennent énormes et empirent au fur et à mesure que  $n$  augmente pour les valeurs de  $n \leq 50000$  qu'on a considérées. Rappelons-nous que dans la section 2.1 on a vu que pour un  $\lambda_0$  fixe, le ratio en question tend vers l'infini lorsque  $n$  tend vers l'infini, indiquant que la pondération n'aide pas du tout sauf pour des  $n$  relativement petits. Bien qu'ici nous avons des poids adaptatifs et non fixes, l'explication pour le cas fixe nous donne une idée de ce qui arrive. On a vu dans la section 2.3 que le poids adaptatif par bootstrap

converge vers la bonne valeur pour tout  $0 < p < 1$  (résumé dans les tableaux 2.4 et 2.5), mais on voit maintenant que la convergence est trop lente pour que les valeurs de  $p \geq 1/2$  soient utiles en conjonction avec les valeurs de  $n$  (sauf  $n \leq 20$ ) qu'on est probable de rencontrer en pratique. Rappelons-nous aussi que la pénalité  $p$  a été introduite comme avantage dans le cas où  $\mu_X = \mu_Y$ . Il y avait une crainte que le terme du biais aurait une influence non désirée sur le choix de poids, menant à des poids peu performants. Par contre, dans le cas  $\mu_X \neq \mu_Y$  comme on a noté, on veut que le biais continue à dominer l'erreur quadratique moyenne. Le terme du biais est  $(1 - \hat{\lambda}_p^*)^2 n^{-p} (\bar{X} - \bar{Y})^2$ . On voit dans le tableau 2.7 que les poids  $\hat{\lambda}_p^*$  sont significativement plus petits que le poids optimal pour ces valeurs de  $p$  élevées, et alors le facteur  $(1 - \hat{\lambda}_p^*)^2$  est plus grand que dans le cas optimal. On peut donc déduire que le poids trop petit sert à enfler le biais et ainsi l'erreur quadratique moyenne.

On vient de voir que même sans pénalité, tous les poids bootstrap performant très bien dans le cas de moyennes égales, alors la pénalité semble ne pas valoir la peine, à moins qu'on soit certain que les moyennes de nos deux populations soient égales. Alors, bien que la théorie de la section 2.3 nous indique qu'il faut  $0 < p < 1$  pour que le poids converge en probabilité vers le poids optimal, les simulations nous montrent qu'en pratique  $p = 0$  est le meilleur choix, c'est-à-dire qu'il ne faut pas en pratique pénaliser le terme du biais de l'erreur quadratique moyenne. La raison pour laquelle  $p = 0$  fonctionne bien est que ça mène à un poids qui est encore raisonnable même s'il n'est pas optimal. Ce poids fera en sorte que les deux échantillons seront utilisés, ce qui est certain d'améliorer l'estimation en diminuant la variance de l'EVP puisque les moyennes des deux distributions sont égales. Ceci revient à ce qu'on a vu dans la section 2.1 lors de la discussion du cas de  $\lambda_0$  fixe, où on a vu que le ratio de l'erreur quadratique moyenne de l'EVP par rapport à celui de l'EVM est inférieur à 1 pour tout  $0 < \lambda_0 < 1$  fixe.

Retournons à l'analyse du cas  $\mu_X \neq \mu_Y$ . Il est à noter que pour les  $n$  très petits,  $n \leq 20$ , tous les estimateurs pondérés performant bien, même ceux avec les  $p$  plus élevés.

Pour les poids de  $p = 0; 0,05; 0,25$ , et le poids Welch, quand  $n \leq 40$  l'estimation est meilleure que l'estimateur de vraisemblance maximale. Ensuite les ratios deviennent supérieurs à 1 et ce n'est que lorsque  $n = 5000$  que les ratios pour  $\hat{\lambda}_{0,05}^*$  et  $\hat{\lambda}_W^*$  atteignent le niveau 1,005, c'est-à-dire presque la même performance que l'estimateur de vraisemblance maximale. Le poids  $\hat{\lambda}_{CVmod}$  donne les meilleurs résultats dans le cas  $\mu_X \neq \mu_Y$  : son ratio est inférieur à 1 pour  $n < 200$ , il atteint un maximum de 1,026 à  $n = 400$ , et est redescendu à 1,001 quand  $n = 5000$ .

Aussi à noter, le poids  $\hat{\lambda}_{CV}$  mène à des ratios élevés de façon inacceptable pour les  $n$  petits et moyens, mais pour  $n \geq 5000$  ce poids atteint lui aussi une performance comparable à l'estimateur de vraisemblance maximale.

La figure 2.1 offre une aide visuelle pour comprendre le comportement des poids  $\hat{\lambda}_p^*$ . Illustré pour le cas  $\mu_X \neq \mu_Y$  sont les ratios  $\frac{EQM_p^*}{EQM_{EVM}}$  pour les valeurs de  $p$  entre 0 et 1, pour des valeurs de  $n$  entre 10 et 60. On y voit que pour  $n = 10$  les valeurs élevées de  $p$  sont les meilleures mais que pour  $n = 20$  et 30 il y a un changement rapide, et rendu à  $n = 40$  les valeurs de  $p$  très près de 0 sont optimales pour cette classe d'estimateur. À mentionner est le fait que les graphiques continuent d'avoir l'air de ceux illustrés pour  $n = 50$  et 60, quand  $n$  est plus élevé, et que les graphiques équivalents dans le cas  $\mu_X = \mu_Y$  ont l'air du graphique illustré ici pour  $n = 10$ .

Les simulations ont aussi été faites pour d'autres valeurs de  $|\mu_X - \mu_Y|$ , en gardant  $\sigma_X^2 = 1 = \sigma_Y^2$ . Pour  $|\mu_X - \mu_Y| = 0,1$  les résultats bootstrap (pour toutes les valeurs de  $p$ ) sont meilleurs que ceux de l'estimateur de vraisemblance maximale pour  $n \leq 300$ , tandis que les ratios pour  $\hat{\lambda}_{CVmod}$  demeurent inférieurs à 1, tout en augmentant vers 1, pour les  $n$  entre 10 et 1000. Pour  $|\mu_X - \mu_Y| = 0,4$  c'est seulement avec  $n \leq 20$  que le bootstrap (pour toutes les valeurs de  $p$ ) apporte une amélioration sur l'estimateur de vraisemblance maximale, et avec  $n \leq 40$  que l'estimateur de validation croisée modifiée apporte une amélioration sur l'estimateur de vraisemblance maximale. Pour  $|\mu_X - \mu_Y| = 1$ , c'est-à-dire quand la différence entre les moyennes est égale à la variance des distributions, les résultats confirment notre intuition et les conclusions de Wang et Zidek (2005a) : l'estimateur de vraisemblance maximale est mieux que les estimateurs pondérés dans ce

cas, puisque la deuxième population, ayant une moyenne très différente, n'aide pas à estimer la moyenne de la première. Dans ce cas, l'estimateur  $\hat{\lambda}_{CVmod}$  est le meilleur des estimateurs pondérés, suivi de  $\hat{\lambda}_0^*$ , pour lesquels le ratio de leur erreur quadratique moyenne et de celui de l'estimateur de vraisemblance maximale diminue d'environ 1,10 jusqu'à 1,02 pour  $n$  entre 10 et 150. C'est environ lorsque  $|\mu_X - \mu_Y| \geq 0,4$  que les estimateurs pondérés  $\hat{\lambda}_p^*$  en particulier commencent à être systématiquement pires que l'estimateur de vraisemblance maximale pour  $n \geq 20$ .

En conclusion, on a vu que l'estimateur pondéré qui donne les meilleurs résultats en général quand les moyennes sont égales ainsi que quand elles ne le sont pas, et pour toutes valeurs de  $n$ , est celui basé sur  $\hat{\lambda}_{CVmod}$ . L'estimateur basé sur  $\hat{\lambda}_{CV}$  sans terme de correction, par contre, est complètement inacceptable.

Si on a  $\mu_X = \mu_Y$ , tous les poids pondérés sauf  $\hat{\lambda}_{CV}$  apportent une amélioration sur l'estimateur de vraisemblance maximale, et ce pour toutes les valeurs de  $n$ . On a vu que ce phénomène s'explique si on considère un  $0 < \lambda_0 < 1$  fixe à la fois, puisque le ratio des erreurs quadratiques moyennes de l'EVP et de l'EVM montre que l'EVP performe toujours mieux parmi les deux. Donc même les poids qui sont loin de l'optimal apportent une amélioration sur l'estimateur de vraisemblance maximale : celui-ci utilise seulement  $n$  observations, tandis que les estimateurs pondérés en utilisent  $2n$  et ont donc une variance plus petite, menant à une erreur quadratique moyenne inférieure.

Si les moyennes sont "près" de l'égalité des moyennes (par exemple si  $|\mu_X - \mu_Y| \approx \frac{1}{10}\sigma_X^2 = \frac{1}{10}\sigma_Y^2$ ), tous les poids sauf  $\hat{\lambda}_{CV}$  performent encore bien, mais il y a un seuil pour  $n$ , au-delà duquel il vaut mieux utiliser l'estimateur de vraisemblance maximale. Le seuil est de  $n = 300$  pour  $|\mu_X - \mu_Y| = 0,1$  par exemple. Par contre, quand  $\mu_X \neq \mu_Y$ , les poids de classe  $\hat{\lambda}_p^*$  avec  $p \geq 0,5$  se montrent inacceptables pour tout  $n \geq 30$ , et ceux avec  $p \leq 0,25$  donnent des estimateurs pires que l'estimateur de vraisemblance maximale pour  $50 \leq n \leq 5000$ .

Le poids Welch se comporte comme le poids  $\hat{\lambda}_{0,05}^*$ , alors on voit que le fait de faire ce test de Welch sur l'égalité des moyennes au niveau 50% n'améliore pas vraiment le choix du poids.



La performance des estimateurs pondérés dépend surtout du fait que  $\mu_X$  soit assez près de  $\mu_Y$ . Dans le cas considéré où les variances des populations sont égales, si on a raison de croire que  $|\mu_X - \mu_Y| \geq 0,4\sigma_X^2$  par exemple, il vaut mieux simplement utiliser l'estimateur de vraisemblance maximale, à moins qu'on ait  $n \leq 20$ . Mais, si  $\mu_X$  et  $\mu_Y$  sont similaires, les poids peuvent grandement améliorer l'estimation. Les poids les plus intéressants à considérer sont donc  $\hat{\lambda}_{CV_{mod}}$  et  $\hat{\lambda}_0^*$ .

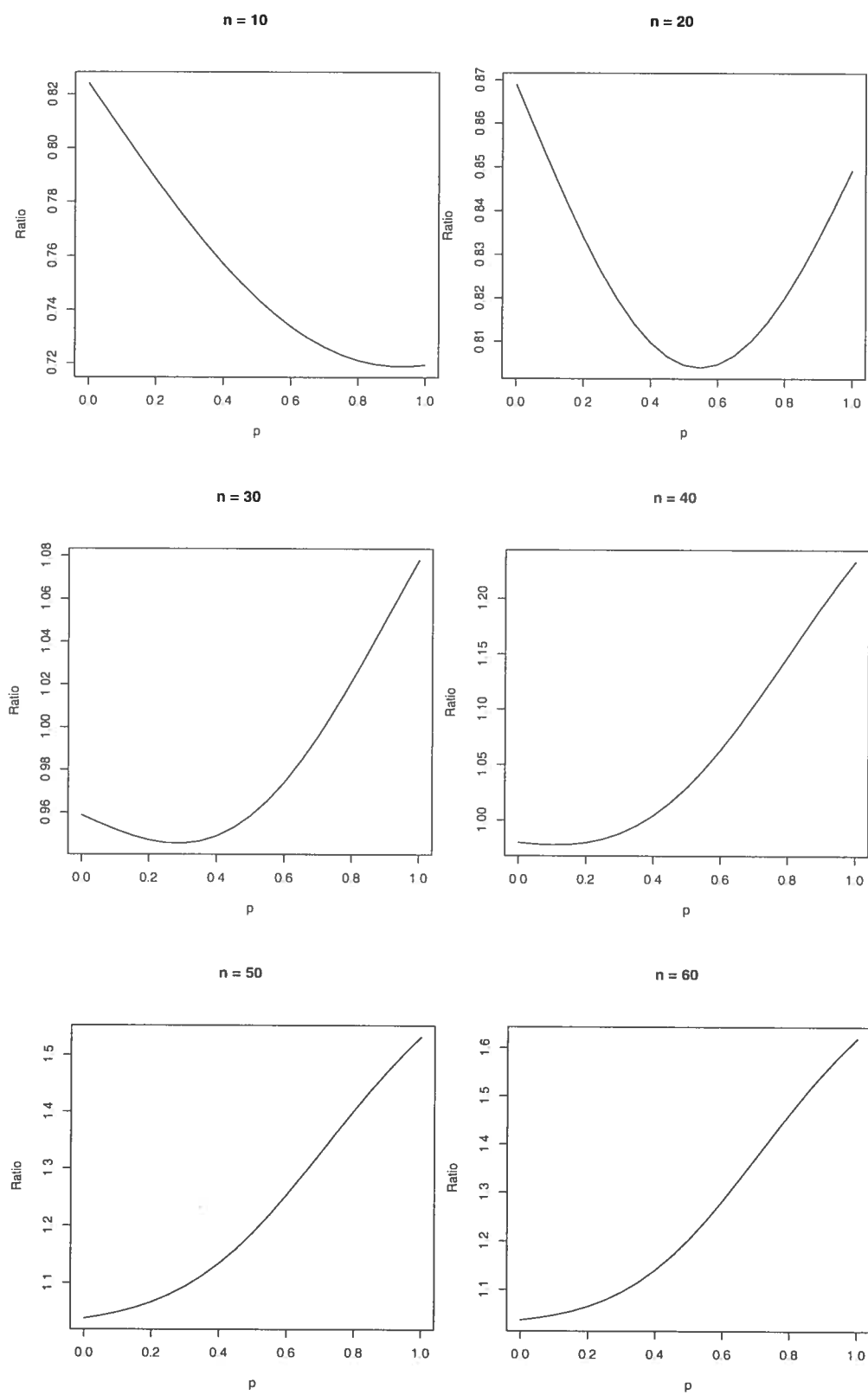


FIG. 2.1. Graphiques de  $\frac{EQM_p^*}{EQM_{EVM}}$  en fonction de  $p$  dans le cas de données normales,  $|\mu_X - \mu_Y| = 0,3$  et  $\sigma_X^2 = \sigma_Y^2 = 1$

$n$	$\lambda_{opt}$	$\hat{\lambda}_{CV}$ ( <i>ET</i> )	$\hat{\lambda}_{CVmod}$ ( <i>ET</i> )	$\hat{\lambda}_0^*$ ( <i>ET</i> )	$\hat{\lambda}_{0,05}^*$ ( <i>ET</i> )	$\hat{\lambda}_{0,25}^*$ ( <i>ET</i> )	$\hat{\lambda}_{0,5}^*$ ( <i>ET</i> )	$\hat{\lambda}_{0,75}^*$ ( <i>ET</i> )	$\hat{\lambda}_W^*$ ( <i>ET</i> )
10	0,5	-0,585 (1,748)	0,783 (0,094)	0,690 (0,188)	0,682 (0,188)	0,648 (0,185)	0,609 (0,180)	0,577 (0,177)	0,664 (0,202)
20	0,5	-1,323 (2,943)	0,837 (0,049)	0,680 (0,157)	0,668 (0,155)	0,623 (0,143)	0,577 (0,128)	0,543 (0,119)	0,648 (0,170)
30	0,5	-1,975 (4,118)	0,866 (0,034)	0,673 (0,153)	0,660 (0,149)	0,611 (0,132)	0,562 (0,111)	0,529 (0,098)	0,640 (0,164)
40	0,5	-2,479 (5,28)	0,885 (0,025)	0,679 (0,142)	0,664 (0,137)	0,612 (0,117)	0,561 (0,094)	0,529 (0,082)	0,644 (0,153)
50	0,5	-2,907 (6,184)	0,898 (0,019)	0,673 (0,140)	0,658 (0,135)	0,603 (0,111)	0,552 (0,085)	0,523 (0,073)	0,638 (0,149)
60	0,5	-3,167 (7,091)	0,909 (0,016)	0,679 (0,138)	0,663 (0,133)	0,604 (0,107)	0,551 (0,080)	0,522 (0,067)	0,643 (0,148)
70	0,5	-3,881 (8,317)	0,918 (0,015)	0,670 (0,141)	0,654 (0,135)	0,596 (0,108)	0,544 (0,079)	0,517 (0,064)	0,633 (0,150)
80	0,5	-4,087 (9,212)	0,924 (0,013)	0,672 (0,138)	0,656 (0,132)	0,596 (0,104)	0,544 (0,073)	0,517 (0,058)	0,637 (0,146)
90	0,5	-4,722 (10,045)	0,930 (0,012)	0,670 (0,141)	0,653 (0,134)	0,593 (0,103)	0,541 (0,070)	0,515 (0,055)	0,633 (0,148)
100	0,5	-4,365 (10,424)	0,935 (0,011)	0,679 (0,139)	0,661 (0,133)	0,598 (0,102)	0,543 (0,068)	0,516 (0,052)	0,641 (0,148)
120	0,5	-5,458 (13,398)	0,942 (0,010)	0,680 (0,137)	0,661 (0,130)	0,596 (0,096)	0,542 (0,060)	0,516 (0,047)	0,642 (0,144)
150	0,5	-6,357 (15,835)	0,950 (0,008)	0,675 (0,137)	0,656 (0,130)	0,590 (0,095)	0,536 (0,058)	0,513 (0,043)	0,636 (0,145)

TAB. 2.6. Espérance et écart type des poids pour des données normales,  $\mu_X = \mu_Y$  et  $\sigma_X^2 = \sigma_Y^2 = 1$

$n$	$\lambda_{opt}$	$\hat{\lambda}_{CV}$ (ET)	$\hat{\lambda}_{CVmod}$ (ET)	$\hat{\lambda}_0^*$ (ET)	$\hat{\lambda}_{0,05}^*$ (ET)	$\hat{\lambda}_{0,25}^*$ (ET)	$\hat{\lambda}_{0,5}^*$ (ET)	$\hat{\lambda}_{0,75}^*$ (ET)	$\hat{\lambda}_W^*$ (ET)
10	0,655	-0,241 (1,525)	0,795 (0,097)	0,727 (0,180)	0,718 (0,180)	0,680 (0,180)	0,636 (0,178)	0,598 (0,176)	0,702 (0,196)
20	0,736	-0,558 (2,572)	0,855 (0,055)	0,748 (0,162)	0,736 (0,161)	0,686 (0,153)	0,626 (0,139)	0,577 (0,125)	0,722 (0,177)
30	0,787	-0,759 (3,477)	0,881 (0,040)	0,759 (0,159)	0,745 (0,158)	0,687 (0,148)	0,619 (0,128)	0,565 (0,108)	0,732 (0,173)
40	0,821	-0,802 (3,908)	0,899 (0,031)	0,776 (0,153)	0,761 (0,152)	0,698 (0,141)	0,624 (0,116)	0,567 (0,094)	0,751 (0,166)
50	0,846	-0,633 (4,415)	0,910 (0,026)	0,796 (0,142)	0,780 (0,142)	0,710 (0,133)	0,626 (0,108)	0,563 (0,083)	0,770 (0,156)
60	0,864	-0,535 (4,824)	0,921 (0,024)	0,819 (0,139)	0,802 (0,139)	0,730 (0,132)	0,639 (0,106)	0,569 (0,079)	0,795 (0,152)
70	0,879	0,0715 (3,315)	0,927 (0,021)	0,838 (0,123)	0,820 (0,124)	0,742 (0,122)	0,642 (0,099)	0,567 (0,073)	0,814 (0,137)
80	0,891	-0,206 (4,739)	0,933 (0,018)	0,841 (0,122)	0,823 (0,124)	0,743 (0,120)	0,640 (0,096)	0,564 (0,068)	0,818 (0,134)
90	0,901	0,184 (3,69)	0,938 (0,017)	0,863 (0,114)	0,846 (0,116)	0,764 (0,116)	0,653 (0,095)	0,569 (0,067)	0,841 (0,126)
100	0,909	0,0120 (4,969)	0,943 (0,016)	0,867 (0,113)	0,850 (0,115)	0,768 (0,117)	0,656 (0,096)	0,570 (0,066)	0,846 (0,125)
120	0,921	0,415 (3,246)	0,949 (0,014)	0,886 (0,101)	0,869 (0,104)	0,785 (0,110)	0,664 (0,091)	0,573 (0,060)	0,867 (0,110)
150	0,935	0,540 (3,419)	0,956 (0,012)	0,907 (0,086)	0,891 (0,090)	0,806 (0,099)	0,676 (0,085)	0,574 (0,055)	0,889 (0,096)

TAB. 2.7. Espérance et écart type des poids des données normales,

$$|\mu_X - \mu_Y| = 0,3 \text{ et } \sigma_X^2 = \sigma_Y^2 = 1$$

$n$	EQM EVM (ET)	EQM EVP <sub>CV</sub> (ET)	EQM EVP <sub>CVmod</sub> (ET)	EQM EVP <sub>0</sub> (ET)	EQM EVP <sub>0,05</sub> (ET)	EQM EVP <sub>0,25</sub> (ET)	EQM EVP <sub>0,5</sub> (ET)	EQM EVP <sub>0,75</sub> (ET)	EQM EVP <sub>W</sub> (ET)
10	0,094 (0,138)	0,101 (0,135)	0,071 (0,107)	0,072 (0,111)	0,071 (0,109)	0,066 (0,101)	0,061 (0,091)	0,058 (0,082)	0,071 (0,109)
20	0,051 (0,072)	0,066 (0,086)	0,040 (0,058)	0,038 (0,057)	0,037 (0,055)	0,034 (0,049)	0,031 (0,043)	0,028 (0,039)	0,037 (0,055)
30	0,033 (0,050)	0,055 (0,073)	0,027 (0,042)	0,025 (0,041)	0,025 (0,040)	0,022 (0,035)	0,019 (0,029)	0,018 (0,025)	0,025 (0,040)
40	0,025 (0,036)	0,045 (0,061)	0,020 (0,030)	0,018 (0,029)	0,018 (0,028)	0,016 (0,024)	0,014 (0,020)	0,013 (0,018)	0,018 (0,028)
50	0,019 (0,025)	0,040 (0,054)	0,016 (0,021)	0,014 (0,019)	0,013 (0,019)	0,012 (0,016)	0,010 (0,014)	0,010 (0,014)	0,013 (0,019)
60	0,017 (0,023)	0,037 (0,054)	0,014 (0,020)	0,012 (0,018)	0,012 (0,017)	0,010 (0,015)	0,009 (0,013)	0,008 (0,013)	0,011 (0,017)
70	0,013 (0,018)	0,035 (0,047)	0,011 (0,016)	0,010 (0,014)	0,009 (0,014)	0,008 (0,012)	0,007 (0,010)	0,006 (0,009)	0,009 (0,014)
80	0,012 (0,016)	0,032 (0,046)	0,010 (0,014)	0,008 (0,013)	0,008 (0,012)	0,007 (0,010)	0,006 (0,009)	0,006 (0,008)	0,008 (0,012)
90	0,011 (0,016)	0,034 (0,048)	0,009 (0,014)	0,007 (0,012)	0,007 (0,012)	0,006 (0,010)	0,005 (0,008)	0,005 (0,007)	0,007 (0,012)
100	0,009 (0,013)	0,028 (0,043)	0,008 (0,012)	0,007 (0,011)	0,006 (0,010)	0,005 (0,008)	0,005 (0,007)	0,004 (0,007)	0,006 (0,010)
120	0,008 (0,011)	0,025 (0,039)	0,007 (0,010)	0,006 (0,009)	0,005 (0,008)	0,004 (0,007)	0,004 (0,006)	0,004 (0,005)	0,005 (0,008)
150	0,006 (0,009)	0,022 (0,035)	0,005 (0,008)	0,004 (0,007)	0,004 (0,006)	0,003 (0,005)	0,003 (0,004)	0,003 (0,004)	0,004 (0,006)

TAB. 2.8. EQM des EVP et écart type des différences au carré entre l'EVP et  $\mu_X$ , pour des données normales,  $\mu_X = \mu_Y$  et  $\sigma_X^2 = \sigma_Y^2 = 1$

$n$	EQM EVM (ET)	EQM EVP <sub>CV</sub> (ET)	EQM EVP <sub>CVmod</sub> (ET)	EQM EVP <sub>0</sub> (ET)	EQM EVP <sub>0,05</sub> (ET)	EQM EVP <sub>0,25</sub> (ET)	EQM EVP <sub>0,5</sub> (ET)	EQM EVP <sub>0,75</sub> (ET)	EQM EVP <sub>W</sub> (ET)
10	0,094 (0,138)	0,114 (0,152)	0,076 (0,110)	0,077 (0,114)	0,076 (0,112)	0,073 (0,105)	0,070 (0,097)	0,068 (0,091)	0,077 (0,112)
20	0,051 (0,072)	0,073 (0,101)	0,044 (0,061)	0,045 (0,060)	0,044 (0,059)	0,042 (0,056)	0,041 (0,053)	0,042 (0,053)	0,044 (0,059)
30	0,033 (0,050)	0,060 (0,088)	0,031 (0,045)	0,032 (0,046)	0,032 (0,046)	0,032 (0,044)	0,032 (0,043)	0,034 (0,043)	0,032 (0,046)
40	0,025 (0,036)	0,050 (0,076)	0,023 (0,033)	0,025 (0,033)	0,024 (0,033)	0,025 (0,032)	0,026 (0,032)	0,028 (0,033)	0,025 (0,033)
50	0,019 (0,025)	0,038 (0,062)	0,018 (0,023)	0,020 (0,025)	0,020 (0,025)	0,020 (0,025)	0,023 (0,027)	0,026 (0,029)	0,020 (0,025)
60	0,017 (0,023)	0,031 (0,050)	0,016 (0,022)	0,017 (0,023)	0,017 (0,023)	0,018 (0,023)	0,020 (0,025)	0,024 (0,027)	0,017 (0,023)
70	0,013 (0,018)	0,025 (0,043)	0,013 (0,017)	0,014 (0,019)	0,014 (0,019)	0,015 (0,020)	0,018 (0,022)	0,022 (0,024)	0,014 (0,019)
80	0,012 (0,016)	0,021 (0,039)	0,011 (0,016)	0,013 (0,017)	0,013 (0,017)	0,014 (0,018)	0,017 (0,020)	0,022 (0,022)	0,013 (0,017)
90	0,011 (0,016)	0,019 (0,036)	0,010 (0,015)	0,011 (0,016)	0,012 (0,016)	0,013 (0,017)	0,016 (0,019)	0,021 (0,020)	0,012 (0,016)
100	0,009 (0,013)	0,016 (0,031)	0,009 (0,013)	0,010 (0,014)	0,010 (0,015)	0,012 (0,016)	0,015 (0,017)	0,020 (0,019)	0,010 (0,015)
120	0,008 (0,011)	0,013 (0,025)	0,008 (0,011)	0,009 (0,012)	0,009 (0,012)	0,010 (0,013)	0,014 (0,015)	0,019 (0,017)	0,009 (0,012)
150	0,006 (0,009)	0,009 (0,016)	0,006 (0,008)	0,007 (0,009)	0,007 (0,009)	0,008 (0,010)	0,012 (0,013)	0,018 (0,015)	0,007 (0,009)

TAB. 2.9. EQM des EVP et écart type des différences au carré entre l'EVP et  $\mu_X$ , pour des données normales,  $|\mu_X - \mu_Y| = 0,3$  et  $\sigma_X^2 = \sigma_Y^2 = 1$

$n$	$\frac{EQM_{CV}}{EQM_{EVM}}$	$\frac{EQM_{CVmod}}{EQM_{EVM}}$	$\frac{EQM_0^*}{EQM_{EVM}}$	$\frac{EQM_{0.05}^*}{EQM_{EVM}}$	$\frac{EQM_{0.25}^*}{EQM_{EVM}}$	$\frac{EQM_{0.5}^*}{EQM_{EVM}}$	$\frac{EQM_{0.75}^*}{EQM_{EVM}}$	$\frac{EQM_W^*}{EQM_{EVM}}$
10	1,078	0,754	0,767	0,756	0,709	0,656	0,616	0,756
20	1,276	0,783	0,747	0,730	0,664	0,596	0,553	0,729
30	1,636	0,811	0,755	0,736	0,662	0,586	0,541	0,738
40	1,769	0,821	0,744	0,722	0,640	0,562	0,523	0,724
50	2,082	0,834	0,735	0,713	0,632	0,563	0,534	0,713
60	2,182	0,844	0,731	0,707	0,619	0,549	0,520	0,704
70	2,578	0,851	0,734	0,709	0,615	0,536	0,508	0,708
80	2,709	0,860	0,740	0,715	0,620	0,540	0,509	0,714
90	3,084	0,862	0,720	0,692	0,584	0,495	0,461	0,689
100	2,996	0,870	0,736	0,708	0,604	0,522	0,495	0,706
120	3,064	0,883	0,731	0,702	0,596	0,520	0,498	0,699
150	3,514	0,893	0,730	0,700	0,587	0,506	0,485	0,697

TAB. 2.10. Ratios des EQM de l'EVP par rapport à l'EVM pour des données normales,  $\mu_X = \mu_Y$  et  $\sigma_X^2 = \sigma_Y^2 = 1$

$n$	$\frac{EQM_{CV}}{EQM_{EVM}}$	$\frac{EQM_{CVmod}}{EQM_{EVM}}$	$\frac{EQM_0^*}{EQM_{EVM}}$	$\frac{EQM_{0.05}^*}{EQM_{EVM}}$	$\frac{EQM_{0.25}^*}{EQM_{EVM}}$	$\frac{EQM_{0.5}^*}{EQM_{EVM}}$	$\frac{EQM_{0.75}^*}{EQM_{EVM}}$	$\frac{EQM_{IV}^*}{EQM_{EVM}}$
10	1,214	0,805	0,824	0,815	0,780	0,744	0,723	0,818
20	1,406	0,855	0,869	0,859	0,826	0,804	0,814	0,864
30	1,774	0,914	0,958	0,955	0,946	0,958	1,008	0,963
40	1,989	0,929	0,980	0,978	0,983	1,030	1,126	0,985
50	2,003	0,950	1,038	1,043	1,079	1,188	1,365	1,053
60	1,866	0,947	1,037	1,042	1,078	1,204	1,418	1,05
70	1,839	0,960	1,065	1,076	1,154	1,366	1,680	1,084
80	1,807	0,976	1,089	1,106	1,211	1,476	1,839	1,112
90	1,724	0,966	1,077	1,091	1,192	1,479	1,902	1,098
100	1,742	0,979	1,106	1,125	1,254	1,600	2,107	1,132
120	1,601	0,980	1,102	1,123	1,270	1,691	2,315	1,127
150	1,446	0,993	1,115	1,144	1,355	1,986	2,909	1,148
500	1,081	1,019	1,067	1,098	1,478	3,782	8,403	1,098
1000	1,037	1,016	1,034	1,054	1,382	4,899	15,054	1,054
5000	1,002	1,001	1,002	1,005	1,141	7,182	57,996	1,005
10000	1,003	1,003	1,003	1,006	1,125	8,313	103,351	1,006
50000	1,000	1,000	1,000	1,001	1,046	9,747	383,279	1,001

TAB. 2.11. Ratios des EQM de l'EVP par rapport à l'EVM pour des données normales,  $|\mu_X - \mu_Y| = 0,3$  et  $\sigma_X^2 = \sigma_Y^2 = 1$



## Chapitre 3

---

### DONNÉES LOGNORMALES

Considérons le cas de deux échantillons indépendants de données indépendantes et identiquement distribuées suivant des distributions lognormales :

$$X_i \sim \text{LN}(\mu_X, \sigma_X^2), \quad Y_i \sim \text{LN}(\mu_Y, \sigma_Y^2), \quad i = 1, \dots, n.$$

Rappelons-nous que dans ce cas on a

$$E[X_i] = e^{\mu_X + \frac{\sigma_X^2}{2}}, \quad \text{Var}(X_i) = e^{2(\mu_X + \sigma_X^2)} - e^{2\mu_X + \sigma_X^2}$$

et un résultat similaire pour  $Y_i$ .

On définit  $U_i = \log(X_i)$  et  $V_i = \log(Y_i)$ . On a donc

$$U_i \sim N(\mu_X, \sigma_X^2), \quad V_i \sim N(\mu_Y, \sigma_Y^2), \quad i = 1, \dots, n.$$

L'estimateur de vraisemblance maximale pour  $\mu_X$  est

$$\text{EVM} = \frac{1}{n} \sum_{i=1}^n \log(X_i),$$

et l'estimateur de vraisemblance pondérée

$$\begin{aligned} \text{EVP} &= \lambda \frac{1}{n} \sum_{i=1}^n \log(X_i) + (1 - \lambda) \frac{1}{n} \sum_{i=1}^n \log(Y_i) \\ &= \lambda \bar{U} + (1 - \lambda) \bar{V} \end{aligned}$$

comme on a vu aux équations (1.2.4) et (1.2.5) respectivement.

### 3.1. ERREUR QUADRATIQUE MOYENNE ET LA VALEUR OPTIMALE DE $\lambda$

L'erreur quadratique moyenne de l'estimateur pondéré est

$$\text{EQM}(\lambda) = \text{Var}(\lambda\bar{U} + (1-\lambda)\bar{V}) + \text{Biais}^2(\lambda\bar{U} + (1-\lambda)\bar{V}).$$

De cette manière le problème est réduit à celui traité dans la section 2.1 puisque les  $U_i$  et  $V_i$  suivent des distributions normales.

La valeur optimale de  $\lambda$  est donc la même que celle de l'équation (2.1.2) :

$$\lambda_{opt} = 1 - \frac{\frac{\sigma_X^2}{n}}{\frac{\sigma_X^2 + \sigma_Y^2}{n} + (\mu_X - \mu_Y)^2}.$$

### 3.2. CHOIX DE $\lambda$ PAR VALIDATION CROISÉE

Pour faire la validation croisée, on utilise encore une fois la mesure d'écart quadratique moyen  $D$  introduite dans l'équation (1.2.2). Rappelons-nous que cette mesure est basée sur la prédiction de  $X_i$  à partir de l'estimateur du maximum de vraisemblance pondérée sans la  $i^{\text{ième}}$  observation :

$$\phi(\tilde{\mu}_X^{(-i)}) = \exp\left\{\frac{\lambda}{n-1} \sum_{k \neq i} U_k + \frac{1-\lambda}{n-1} \sum_{k \neq i} V_k + \frac{\hat{\sigma}_X^{2(-i)}}{2}\right\} \quad (3.2.1)$$

où  $\phi(\mu_X) = \exp\{\mu_X + \frac{\sigma_X^2}{2}\}$  est la fonction décrite au Chapitre 1 qui transforme le paramètre en l'espérance d'une observation de la distribution, et où  $\hat{\sigma}_X^{2(-i)} = \frac{1}{n-1} \sum_{k \neq i} (X_k - \bar{X}^{(-i)})^2$ . Notons que ceci est plus général que le cas traité par Wang et Zidek (2005a) qui ont considéré  $\sigma_X^2 = \sigma_Y^2$  fixes et connus.

Ceci nous mène à notre mesure d'écart

$$\begin{aligned} D(\lambda) &= \frac{1}{n} \sum_{i=1}^n \left( X_i - \phi(\tilde{\mu}_X^{(-i)}) \right)^2 \\ &= \frac{1}{n} \sum_{i=1}^n \left( X_i - \exp\left\{ \frac{\lambda}{n-1} \sum_{k \neq i} \log(X_k) \right. \right. \\ &\quad \left. \left. + \frac{1-\lambda}{n-1} \sum_{k \neq i} \log(Y_k) + \frac{\hat{\sigma}_X^{2(-i)}}{2} \right\} \right)^2. \end{aligned} \quad (3.2.2)$$

On voudrait choisir le poids  $\hat{\lambda}_{CV}$  afin de minimiser l'écart  $D$ , ou de façon équivalente, on voudrait trouver les racines de sa dérivée par rapport à  $\lambda$  :

$$\begin{aligned} \frac{\partial D}{\partial \lambda} = & \frac{2}{n} \sum_{i=1}^n \left[ \left( X_i - \exp \left\{ \frac{\lambda}{n-1} \sum_{k \neq i} \log(X_k) + \frac{1-\lambda}{n-1} \sum_{k \neq i} \log(Y_k) + \frac{\hat{\sigma}_X^{2(-i)}}{2} \right\} \right) \right. \\ & \cdot \exp \left\{ \frac{\lambda}{n-1} \sum_{k \neq i} \log(X_k) + \frac{1-\lambda}{n-1} \sum_{k \neq i} \log(Y_k) + \frac{\hat{\sigma}_X^{2(-i)}}{2} \right\} \\ & \left. \cdot \sum_{k \neq i} \frac{(\log(X_k) - \log(Y_k))}{n-1} \right] \end{aligned} \quad (3.2.3)$$

Il n'y a pas de solution explicite à cette équation. Il faut alors procéder avec des méthodes numériques pour trouver les racines. Pour nos simulations on utilise la méthode de bisection ; notre démarche et nos résultats sont décrits dans la section 3.4.

Wang et Zidek (2005a, Corollary 3.2, p. 474) ont démontré que dans le cas  $\mu_X \neq \mu_Y$ ,

$$\hat{\lambda}_{CV} \xrightarrow{\text{Pr}} 1$$

lorsque  $n \rightarrow \infty$ , quand on a des variances connues  $\sigma_X^2 = \sigma_Y^2$ , et que  $\hat{\lambda}_{CV}$  minimise l'équation (3.2.2) où  $\hat{\sigma}_X^{2(-i)}$  a été remplacé par  $\sigma_X^2$ . On a voulu généraliser ce résultat à notre cas mais nous avons trouvé une erreur dans la démonstration, ainsi que d'autres difficultés, qu'on discute maintenant.

Leur démonstration repose à deux reprises sur leur affirmation que  $D$  est une fonction convexe. Ils affirment que puisque  $f(x) = (x - a)^2$  et  $g(x) = e^{bx+c}$  sont des fonctions convexes, alors leur composition  $f(g(x))$ , dont  $D$  prend la forme, est aussi convexe, tandis que ceci n'est pas vrai en général. Il faudrait que  $f$  soit une fonction croissante pour que ceci soit vrai, mais puisque  $a \neq 0$ ,  $f$  n'est pas croissante sur le domaine  $x \geq 0$  et la convexité de  $f(g(x))$  ne suit alors pas. On a trouvé plusieurs exemples lors des simulations où  $D$  n'était pas convexe, voir les graphiques 3.3 et 3.4 pour des exemples. Dans nos simulations on a estimé la variance  $\hat{\sigma}_X^{2(-i)}$  pour chaque  $i = 1, \dots, n$ , mais on a aussi simulé en posant  $\sigma_X^2 \equiv 1$  comme ont fait Wang et Zidek (2005a) et on a trouvé le même phénomène d'instances où  $D$  était non-convexe. Une des applications de la convexité de  $D$  dans la démonstration est afin de conclure que  $D$  a un minimum unique. Lors

des simulations on n'a jamais trouvé un cas où il y avait plus d'une solution à l'équation (3.2.3), mais la démonstration de l'unicité du minimum de  $D$  aurait besoin de prendre une approche qui ne repose pas sur la convexité.

À part la correction de cette erreur, il faut aussi généraliser la démonstration afin d'appliquer le résultat dans le cas de variances inconnues et estimées qu'on considère ici. La généralisation de plusieurs parties de la démonstration qui reposent sur le fait que  $\sigma_X^2$  est connue n'est pas directement évidente, d'autant plus que certains autres arguments de la démonstration sont incorrects bien qu'il ne soit pas évident si ceci affecte le résultat final.

### 3.3. CHOIX DE $\lambda$ PAR BOOTSTRAP

L'erreur quadratique moyenne de l'EVP pour laquelle on veut utiliser le bootstrap pour estimer chaque terme est

$$\begin{aligned} \text{EQM}(\lambda) &= \lambda^2 \text{Var}(\bar{U}) + (1 - \lambda)^2 \text{Var}(\bar{V}) + 2\lambda(1 - \lambda) \text{Cov}(\bar{U}, \bar{V}) \\ &\quad + (1 - \lambda)^2 (\mu_X - \mu_Y)^2 \end{aligned} \quad (3.3.1)$$

où  $U_i \stackrel{\text{i.i.d.}}{\sim} N(\mu_X, \sigma_X^2)$  indépendant de  $V_i \stackrel{\text{i.i.d.}}{\sim} N(\mu_Y, \sigma_Y^2)$ . Ceci est la même erreur quadratique moyenne que dans l'équation (1.3.1) pour le cas de données normales.

Le choix de  $\lambda$  par bootstrap se fait alors encore une fois par l'équation (2.3.1) :

$$\hat{\lambda}_p^* = 1 - \frac{\hat{\sigma}_X^2 - \widehat{\text{Cov}}}{\hat{\sigma}_X^2 + \hat{\sigma}_Y^2 - 2\widehat{\text{Cov}} + n^{1-p}(\bar{U} - \bar{V})^2}$$

Tous les résultats de la section 2.3 concernant les propriétés asymptotiques de ce poids s'appliquent donc au cas lognormal aussi. En particulier, rappelons-nous que la théorie nous dicte un choix de  $p$  tel que  $0 < p < 1$ .

Prenons cette opportunité pour noter des différences entre la validation croisée et le bootstrap pour le cas de données lognormales, différences qui sont beaucoup plus importantes que dans le cas de données normales. Comme on a noté dans la section 1.3, avec le bootstrap on peut estimer directement l'erreur quadratique moyenne d'un estimateur de  $\theta_X$  sans passer par la prévision de données  $X_i$ . Il n'est donc pas nécessaire de passer par l'erreur quadratique moyenne de prévision. L'erreur quadratique moyenne de l'estimateur pondéré, donné dans l'équation

(3.3.1), est alors essentiellement le critère utilisé pour déterminer le poids par le bootstrap ; à part l'ajout du facteur  $n^{-p}$  on remplace simplement chaque terme dans cette expression par son estimé bootstrap. Par contre, pour la validation croisée, on a vu dans la section 1.2 que dans le cas de données lognormales la fonction  $\phi$  n'est plus une fonction linéaire de  $\mu_X$  comme ce l'était dans le cas de données normales. Pour cette raison on ne travaille plus avec l'estimateur de  $\theta_X$  directement, on travaille plutôt avec un estimé de  $E[X_i] = \exp\{\mu_X + \frac{\sigma_X^2}{2}\}$ , c'est-à-dire l'accent est sur l'espérance d'une observation au lieu d'être sur le paramètre d'intérêt  $\mu_X$ . On confirmera dans la section suivante, concernant les simulations, que dans le cas de données lognormales le bootstrap sert mieux nos objectifs que la validation croisée.

### 3.4. SIMULATIONS

Des simulations similaires à celles pour le cas de données normales ont été effectuées afin d'examiner et de comparer la performance des estimateurs par validation croisée et bootstrap. Encore une fois les simulations ont été effectuées à l'aide du logiciel R. Les programmes se trouvent dans l'annexe B.

La démarche est similaire à celle prise dans le cas normal. Deux échantillons de taille  $n$  ont été tirés des distributions  $LN(\mu_X, \sigma_X^2)$  et  $LN(\mu_Y, \sigma_Y^2)$  respectivement. Les poids  $\hat{\lambda}_{CV}$  et  $\hat{\lambda}_p^*$  pour plusieurs valeurs de  $p$  ont été calculés, ainsi que les estimateurs de vraisemblance pondérée

$$EVP_{CV} = \hat{\lambda}_{CV}\bar{U} + (1 - \hat{\lambda}_{CV})\bar{V}$$

et

$$EVP_p^* = \hat{\lambda}_p^*\bar{U} + (1 - \hat{\lambda}_p^*)\bar{V},$$

et leur erreur quadratique moyenne. L'estimateur de vraisemblance maximale et son erreur quadratique moyenne ont aussi été calculés pour fin de comparaison. Ces étapes ont été répétées 1000 fois, et les moyennne et écart type de chaque quantité d'intérêt calculés.

L'estimateur  $\hat{\lambda}_{W}^*$  a encore une fois été créé basé sur un test Welch de  $H_0 : \mu_X = \mu_Y$  versus  $H_1 : \mu_X \neq \mu_Y$ , où l'estimateur choisit  $p = 0,05$  si la valeur- $p$  du test Welch est inférieure à 50%, mais  $p = 0,5$  sinon.

Les simulations ont été faites pour  $\mu_X = \mu_Y$  et pour  $|\mu_X - \mu_Y| = 0,3$  avec plusieurs valeurs de  $n$  entre 10 et 1000. Les variances des distributions simulées étaient de  $\sigma_X^2 = 1 = \sigma_Y^2$ . À noter que Wang et Zidek (2005a) n'ont pas inclus de simulations lognormales dans leur article.

Il est important de noter que les résultats bootstrap sont les mêmes que dans le cas de données normales, tel qu'expliqué dans la section 3.3. Ainsi tous les commentaires de la section 2.4 concernant les poids bootstrap et leur erreur quadratique moyenne ne vont pas être répétés ici.

La principale différence dans la réalisation de ces simulations par rapport à celles du chapitre précédent se trouve dans le calcul du poids  $\hat{\lambda}_{CV}$ . Comme on a vu dans la section 3.2, le poids  $\hat{\lambda}_{CV}$  n'a pas d'expression explicite et il faut utiliser des méthodes numériques afin de le calculer. On utilise la méthode de bisection, décrite dans Conte et de Boor (1980, p. 75). La méthode de Newton, qui est plus rapide, a entraîné des difficultés particulières dans cette simulation, alors cette méthode n'a pas été poursuivie. Avant de résoudre l'équation des études préliminaires ont été faites pour vérifier le comportement de la fonction  $D$  et de sa dérivée. Il pourrait y avoir des problèmes si la fonction oscille rapidement près de 0, alors des graphiques ont été dessinés afin de s'assurer qu'elle était assez lisse. Plusieurs exemples typiques se trouvent dans les figures 3.1, 3.2, 3.3, et 3.4 (ceux-ci pour une valeur de  $n = 20$ ). On a jugé qu'il n'y aura pas de problème d'oscillation. Par contre, la méthode de Newton n'a pas fonctionné pour une raison différente. Rappelons-nous que la fonction  $D$  à minimiser n'est pas nécessairement convexe contrairement à ce qu'ont dit Wang et Zidek (2005a) dans la démonstration du résultat  $\hat{\lambda}_{CV} \xrightarrow{\text{Pr}} 1$ . Les deux paires de graphiques dans les figures 3.3 et 3.4 montrent deux exemples de cas où la fonction  $D$  peut des fois croître jusqu'à un point au-dessus de 0, puis ensuite redescendre asymptotiquement vers 0 lentement. C'est dans ces cas assez fréquents que la méthode de Newton ne fonctionnait pas; l'algorithme avait tendance à ne pas converger sur

le vrai zéro de la fonction, mais à chaque étape de l'algorithme se trouver avec un  $\lambda$  de plus en plus élevé, finissant avec un  $\lambda$  énorme pour lequel la fonction  $D$  était redevenue près de zéro bien que  $D$  ne croise jamais l'axe à ce point. On a donc utilisé la méthode de bisection, plus lente mais pour laquelle nous sommes certain de trouver un vrai zéro.

La méthode de bisection pour résoudre une équation  $f$  consiste à commencer avec deux bornes,  $a_0$  et  $b_0$  pour lesquelles  $\text{signe}(f(a_0)) \neq \text{signe}(f(b_0))$ . On sait alors que  $f$  a un zéro entre  $a_0$  et  $b_0$ . On calcule  $f(\frac{a_0+b_0}{2})$ ; si son signe est le même que celui de  $f(a_0)$  alors on pose  $a_1 := \frac{a_0+b_0}{2}$  et  $b_1 := b_0$ , sinon, on pose  $a_1 := a_0$  et  $b_1 := \frac{a_0+b_0}{2}$ . On continue à calculer  $a_i$  et  $b_i$  de cette manière jusqu'à ce que  $|a_i - b_i|$  soit plus petit qu'un seuil prédéterminé. Dans nos simulations, nous avons utilisé la valeur  $10^{-6}$  comme seuil.

On a commencé par vérifier la condition de signes différents pour  $a_0 = 1 - 1 = 0$ ,  $b_0 = 1 + 1 = 2$ . Si la condition n'était pas satisfaite alors on soustrayait 1 à  $a_0$  et ajoutait 1 à  $b_0$ , on vérifiait la condition, et on continuait à soustraire 1 à  $a_0$  et additionner 1 à  $b_0$  jusqu'à ce que la condition  $\text{signe}(f(a_0)) \neq \text{signe}(f(b_0))$  soit satisfaite. Ensuite on a procédé avec l'algorithme ci-haut.

De cette manière on a résolu l'équation (3.2.3),  $\partial D(\lambda)/\partial \lambda = 0$ , afin de trouver sa solution  $\hat{\lambda}_{CV}$  qui minimise la fonction  $D$ . On a trouvé que l'écart type de ce poids est très grand alors on a aussi calculé un poids  $\hat{\lambda}_{CV_{tronq}}$  en tronquant  $\hat{\lambda}_{CV}$  entre 0 et 1. Dans le cas de données normales, Wang et Zidek (2005a) ont introduit  $\hat{\lambda}_{CV_{mod}}$  pour pallier aux problèmes de  $\hat{\lambda}_{CV}$ , comme on a vu dans la section 2.2. Toutefois, la modification n'est pas facilement généralisable au cas des données lognormales.

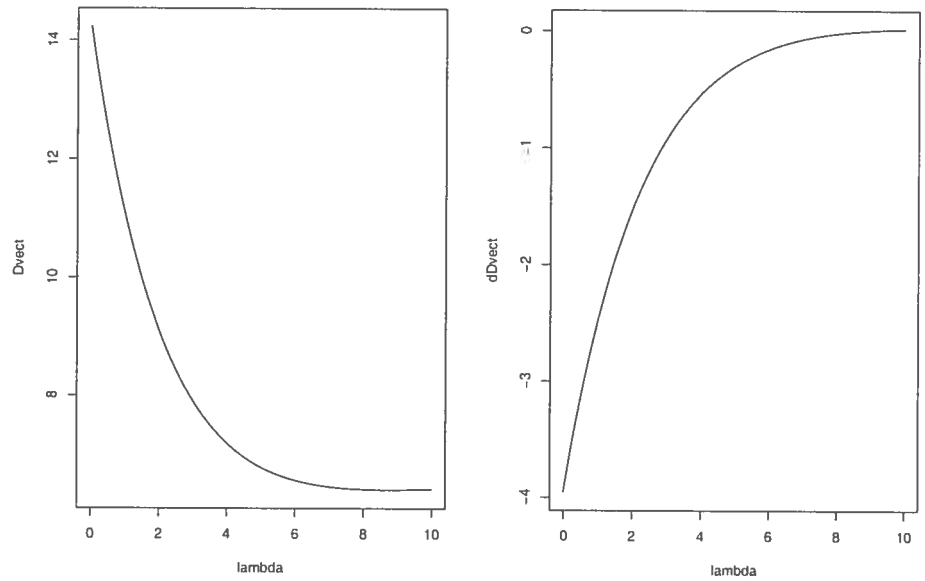


FIG. 3.1. Exemple de paire de graphiques du critère de la validation croisée  $D$  (“Dvect” à gauche) et de sa dérivée  $\partial D/\partial \lambda$  (“dDvect” à droite) en fonction de  $\lambda$ , lorsque  $n = 20$ ,  $\mu_X \neq \mu_Y$ , et  $\sigma_X^2 = \sigma_Y^2 = 1$ , pour des données lognormales

Une sélection d’histogrammes des poids se trouvent dans la section B.2. On voit d’abord la grande variance du poids  $\hat{\lambda}_{CV}$  et les valeurs négatives. Quand  $\mu_X = \mu_Y$ , les valeurs négatives deviennent de plus en plus grandes en valeur absolue, ce qui reste vrai au moins pour tous les  $n$  jusqu’à 1000 (bien que tous les graphiques ne soient pas inclus dans l’annexe). Quand  $\mu_X \neq \mu_Y$ , c’est lorsque  $n = 400$  que  $\hat{\lambda}_{CV}$  n’a plus de valeurs négatives.

Le poids  $\hat{\lambda}_{CV_{tronq}}$  prend la valeur 0 la plupart du temps quand les moyennes sont égales, ce qui n’est pas surprenant étant donné l’observation qu’on vient de faire sur  $\hat{\lambda}_{CV}$ . Quand les moyennes sont différentes, pour  $n \leq 50$ ,  $\hat{\lambda}_{CV_{tronq}}$  est



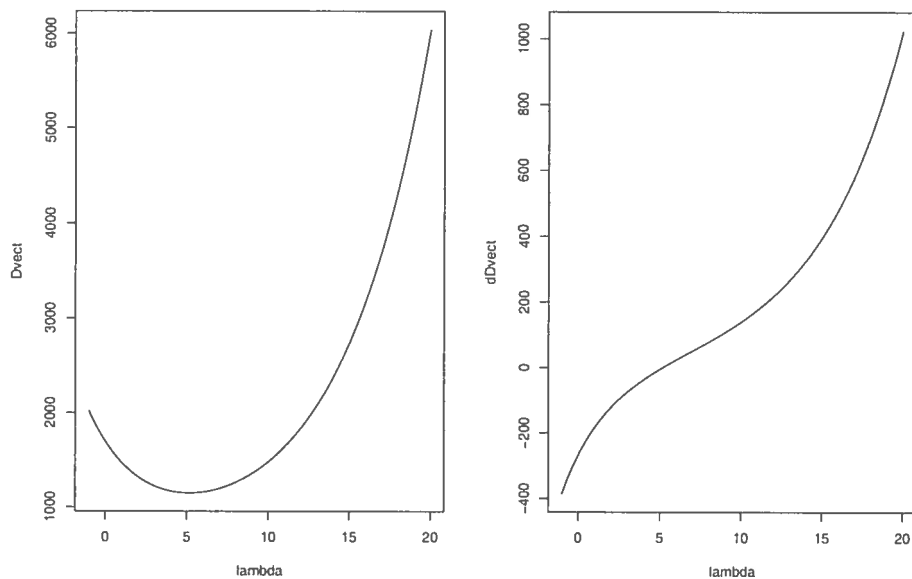


FIG. 3.2. Exemple de paire de graphiques du critère de la validation croisée  $D$  (“Dvect” à gauche) et de sa dérivée  $\partial D/\partial\lambda$  (“dDvect” à droite) en fonction de  $\lambda$ , lorsque  $n = 20$ ,  $\mu_X \neq \mu_Y$ , et  $\sigma_X^2 = \sigma_Y^2 = 1$ , pour des données lognormales

presque également distribué moitié 0, moitié 1. Au-delà de  $n = 100$ , ce poids prend la valeur 1 la plupart du temps comme prévu.

Les résultats concernant les poids et leur écart type se trouvent dans les tableaux 3.1 et 3.2 pour les cas de moyennes égales et différentes respectivement. Sous les deux hypothèses, on remarque encore une fois les valeurs négatives de  $\hat{\lambda}_{CV}$  ainsi que son très grand écart type. On note aussi que l’écart type de  $\hat{\lambda}_{CVtronq}$  est plus élevé que ceux des poids bootstrap. Le poids  $\hat{\lambda}_{CVtronq}$  a tendance à être inférieur au poids optimal, tandis que les poids bootstrap en général ont tendance à être supérieurs au poids optimal.

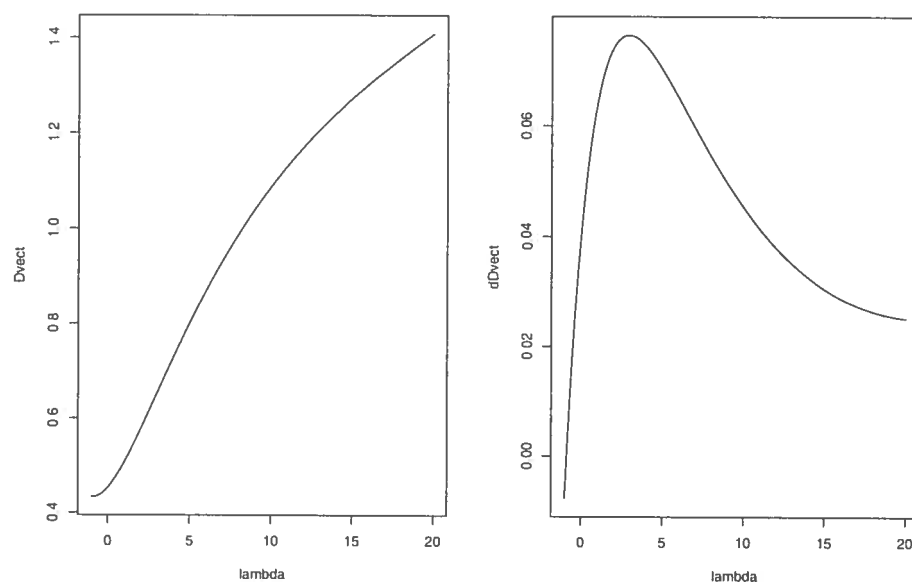


FIG. 3.3. Exemple de paire de graphiques du critère de la validation croisée  $D$  (“Dvect” à gauche) et de sa dérivée  $\partial D/\partial \lambda$  (“dDvect” à droite) en fonction de  $\lambda$ , lorsque  $n = 20$ ,  $\mu_X \neq \mu_Y$ , et  $\sigma_X^2 = \sigma_Y^2 = 1$ , pour des données lognormales

Les tableaux 3.3 et 3.4 présentent les erreurs quadratiques moyennes et l'écart type des différences au carré entre l'EVP et  $\mu_X$  pour les estimateurs pondérés résultant des poids. Quand  $\mu_X = \mu_Y$  on note que  $\hat{\lambda}_{CV}$  a une erreur quadratique moyenne très élevée comparé aux autres estimateurs, et  $\hat{\lambda}_{CVtronq}$  a une erreur quadratique moyenne comparable à celle de l'estimateur bootstrap avec  $p = 0$ . Les mêmes observations sont vraies quand  $\mu_X \neq \mu_Y$  sauf que l'erreur quadratique moyenne de  $\hat{\lambda}_{CVtronq}$  semble être un peu plus élevée que celle de l'estimateur bootstrap avec  $p = 0$ , bien que la différence ne soit pas importante. Notons que les erreurs quadratiques moyennes de tous les poids sauf  $\hat{\lambda}_{CV}$  ont des écarts type comparable. Celui de  $\hat{\lambda}_{CV}$  est par contre beaucoup plus élevé.

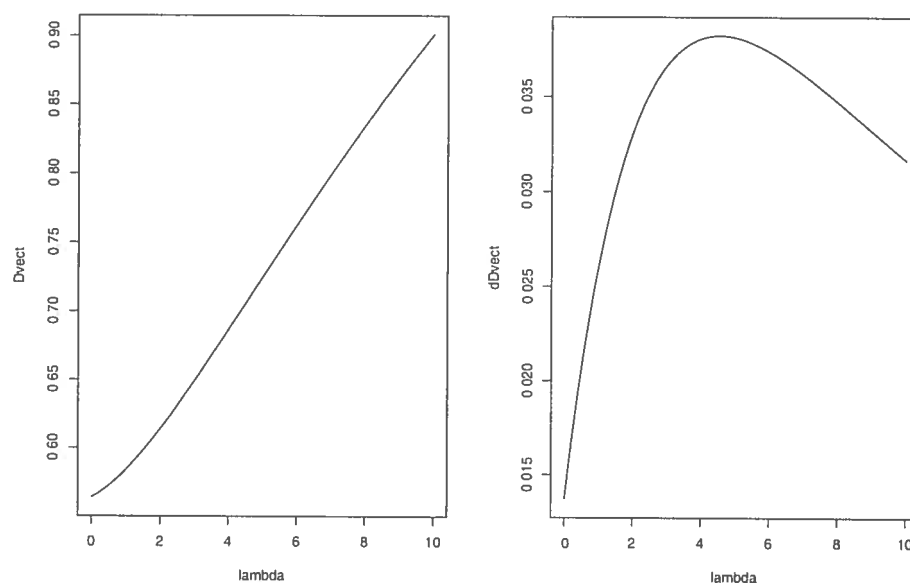


FIG. 3.4. Exemple de paire de graphiques du critère de la validation croisée  $D$  (“Dvect” à gauche) et de sa dérivée  $\partial D/\partial \lambda$  (“dDvect” à droite) en fonction de  $\lambda$ , lorsque  $n = 20$ ,  $\mu_X \neq \mu_Y$ , et  $\sigma_X^2 = \sigma_Y^2 = 1$ , pour des données lognormales

Les ratios des erreurs quadratiques moyennes de chaque estimateur et de celui de l’estimateur de vraisemblance maximale se trouvent dans les tableaux 3.5 et 3.6 pour  $\mu_X = \mu_Y$  et  $\mu_X \neq \mu_Y$  respectivement. Quand  $\mu_X = \mu_Y$  encore une fois le ratio est minimisé par le poids  $\hat{\lambda}_{0,75}^*$  mais tous les poids sauf  $\hat{\lambda}_{CV}$  mènent à une amélioration par rapport à l’estimateur de vraisemblance maximale. Le poids  $\hat{\lambda}_{CV}$  donne des ratios très élevés, tandis que les ratios de  $\hat{\lambda}_{CVtronq}$  sont juste un peu moins bons que ceux des poids bootstrap.

Quand  $\mu_X \neq \mu_Y$ , les ratios de  $\hat{\lambda}_{CV}$  sont encore très élevés, mais sont moins mauvais que dans le cas de moyennes égales. Pour  $n = 10$  et 20, le ratio de  $\hat{\lambda}_{CVtronq}$  est inférieur à 1, mais à partir de  $n = 30$  le ratio empire, atteignant un

maximum de 1,33 quand  $n = 100$ , et ensuite commence à redescendre lentement. Les ratios de  $\hat{\lambda}_{CV}$  atteignent aussi leur maximum d'environ 3,9, vers  $n = 60$ , et commencent à descendre lentement. Toutefois ils sont plus élevés que ceux du poids tronqué mais avec une différence qui perd de l'ampleur. Selon le théorème de Wang et Zidek (2005a) mentionné dans la section 3.2,  $\hat{\lambda}_{CV} \xrightarrow{\text{Pr}} 1$ , mais comme on a discuté, la démonstration a des trous même dans le cas de variances connues  $\sigma_X^2 = \sigma_Y^2$ . Ces simulations ne permettent pas de tirer une conclusion claire à cet égard; on peut seulement dire que s'il y a convergence vers 1, elle est très lente.

Rappelons-nous encore une fois que tous les résultats pour les poids bootstrap dans le cas de données normales s'appliquent aussi ici. Parmi les poids bootstrap,  $\hat{\lambda}_0^*$  performe le mieux en général, avec  $\hat{\lambda}_{0,05}^*$  et  $\hat{\lambda}_{WV}^*$  performant aussi très bien.

Les simulations ont aussi été faites pour d'autres valeurs de  $|\mu_X - \mu_Y|$ . Pour  $|\mu_X - \mu_Y| = 0,1$  les résultats bootstrap demeurent une amélioration par rapport à l'estimateur de vraisemblance maximale pour  $n < 300$ , tandis que le poids par validation croisée tronqué performe bien seulement pour  $n < 150$ . Pour  $|\mu_X - \mu_Y| = 0,4$ , les poids bootstrap performant bien seulement pour  $n \leq 20$  et le poids  $\hat{\lambda}_{CV\text{tronq}}$  seulement pour  $n = 10$ . Rendu à  $|\mu_X - \mu_Y| = 1$  les résultats pour tous les estimateurs pondérés sont pires que ceux de l'estimateur de vraisemblance maximale.

Puisque Wang et Zidek (2005a) ont seulement traité le cas de la variance connue dans leur théorie, on a refait les simulations en posant  $\sigma_X^2 = 1$  au lieu de l'estimer à partir des données. Les ratios d'erreur quadratique moyenne pour le poids  $\hat{\lambda}_{CV\text{tronq}}$  dans ce cas sont pires que ceux qu'on a obtenus ici. Ils suivent la même tendance mais les ratios sont plus élevés. Donc l'estimation de la variance  $\hat{\sigma}_X^{2(-i)}$  dans les calculs du poids  $\hat{\lambda}_{CV\text{tronq}}$  vaut la peine. Par contre, l'estimation de  $\hat{\sigma}_X^{2(-i)}$  ne semble pas améliorer  $\hat{\lambda}_{CV}$  qui n'est pas acceptable, variance connue ou non.

En conclusion, l'estimateur pondéré qui donne les meilleurs résultats en général sous les deux hypothèses et pour toutes valeurs de  $n$  est l'estimateur bootstrap avec une valeur de  $p = 0$ . L'estimateur basé sur le poids  $\hat{\lambda}_{CV}$ , encore une fois ne performe pas bien. Tronquer ce poids entre 0 et 1 améliore sa performance, mais

les estimateurs bootstrap avec  $p$  près de zéro, ou Welch, demeurent mieux, puisque le poids tronqué a encore une plus grande variance que les poids bootstrap.

Si les moyennes des deux populations sont égales ou presque, on trouve encore que tous les estimateurs sauf  $\hat{\lambda}_{CV}$  performant mieux que l'estimateur de vraisemblance maximale, mais que si les moyennes sont différentes,  $\hat{\lambda}_{CVtronq}$  est pire que l'estimateur de vraisemblance maximale à partir de  $n = 30$  (et les estimateurs bootstrap avec de petits  $p$  sont meilleurs que l'estimateur de vraisemblance maximale pour  $n \leq 40$  seulement).

Encore une fois comme dans le cas de données normales, la performance des estimateurs pondérés dépend du fait que  $\mu_X$  soit près de  $\mu_Y$ . Comme dans l'autre cas, si on a  $\sigma_X^2 = \sigma_Y^2$  et si on a raison de croire que  $|\mu_X - \mu_Y| \geq 0,4\sigma_X^2$  il vaut mieux simplement utiliser l'estimateur de vraisemblance maximale, à moins qu'on ait  $n \leq 20$ . Dans ce cas d'un tel petit  $n$ , un poids bootstrap avec n'importe quel  $p$  entre 0 et 1 serait un bon choix. Mais, si  $\mu_X$  et  $\mu_Y$  sont similaires, les poids peuvent aussi grandement améliorer l'estimation pour n'importe quelle taille échantillonnale. Le poids le plus intéressant à considérer en général dans le cas de données lognormales est le poids bootstrap  $\hat{\lambda}_p^*$  pour  $p = 0$ . Ces résultats contrastent avec le cas des données normales, où les modifications apportées au poids de validation croisée donnaient un poids au moins aussi intéressant, même plus intéressant, que  $\hat{\lambda}_0^*$ . Dans le cas de données lognormales par contre, où l'espérance d'une observation n'est plus une fonction linéaire du paramètre  $\mu_X$ , les poids bootstrap,  $\hat{\lambda}_0^*$  en particulier, sont définitivement meilleurs que le poids par validation croisée.

$n$	$\lambda_{opt}$	$\hat{\lambda}_{CV}$ (ET)	$\hat{\lambda}_{CVtronq}$ (ET)	$\hat{\lambda}_0^*$ (ET)	$\hat{\lambda}_{0,05}^*$ (ET)	$\hat{\lambda}_{0,25}^*$ (ET)	$\hat{\lambda}_{0,5}^*$ (ET)	$\hat{\lambda}_{0,75}^*$ (ET)	$\hat{\lambda}_W^*$ (ET)
10	0,5	-0,744 (3,423)	0,392 (0,399)	0,690 (0,188)	0,682 (0,188)	0,648 (0,185)	0,609 (0,180)	0,577 (0,177)	0,664 (0,202)
20	0,5	-1,500 (4,601)	0,351 (0,385)	0,680 (0,157)	0,668 (0,155)	0,623 (0,143)	0,577 (0,128)	0,543 (0,119)	0,648 (0,170)
30	0,5	-2,640 (7,599)	0,315 (0,377)	0,673 (0,153)	0,660 (0,149)	0,611 (0,132)	0,562 (0,111)	0,529 (0,098)	0,640 (0,164)
40	0,5	-2,878 (8,081)	0,318 (0,374)	0,679 (0,142)	0,664 (0,137)	0,611 (0,117)	0,561 (0,094)	0,529 (0,082)	0,644 (0,153)
50	0,5	-3,715 (9,592)	0,307 (0,369)	0,673 (0,140)	0,658 (0,135)	0,603 (0,111)	0,552 (0,085)	0,523 (0,073)	0,638 (0,149)
60	0,5	-4,143 (12,854)	0,331 (0,376)	0,679 (0,138)	0,663 (0,132)	0,604 (0,107)	0,551 (0,080)	0,522 (0,067)	0,643 (0,148)
70	0,5	-5,175 (13,721)	0,310 (0,378)	0,670 (0,141)	0,654 (0,135)	0,596 (0,108)	0,544 (0,079)	0,517 (0,064)	0,633 (0,150)
80	0,5	-5,964 (17,834)	0,289 (0,363)	0,672 (0,138)	0,656 (0,132)	0,595 (0,104)	0,544 (0,073)	0,517 (0,058)	0,637 (0,146)
90	0,5	-6,071 (15,926)	0,301 (0,367)	0,670 (0,141)	0,653 (0,134)	0,593 (0,103)	0,541 (0,070)	0,515 (0,055)	0,633 (0,148)
100	0,5	-5,738 (17,549)	0,321 (0,375)	0,679 (0,139)	0,661 (0,133)	0,598 (0,102)	0,543 (0,068)	0,516 (0,052)	0,641 (0,148)
120	0,5	-6,613 (18,818)	0,320 (0,371)	0,680 (0,137)	0,661 (0,130)	0,596 (0,096)	0,541 (0,060)	0,516 (0,047)	0,642 (0,144)
150	0,5	-7,953 (22,953)	0,303 (0,371)	0,675 (0,137)	0,656 (0,130)	0,590 (0,095)	0,536 (0,058)	0,513 (0,043)	0,636 (0,144)

TAB. 3.1. Espérance et écart type des poids pour des données log-normales,  $\mu_X = \mu_Y = 0$  et  $\sigma_X^2 = \sigma_Y^2 = 1$

$n$	$\lambda_{opt}$	$\hat{\lambda}_{CV}$ (ET)	$\hat{\lambda}_{CVtrouq}$ (ET)	$\hat{\lambda}_0^*$ (ET)	$\hat{\lambda}_{0,05}^*$ (ET)	$\hat{\lambda}_{0,25}^*$ (ET)	$\hat{\lambda}_{0,5}^*$ (ET)	$\hat{\lambda}_{0,75}^*$ (ET)	$\hat{\lambda}_W^*$ (ET)
10	0,655	-0,206 (3,141)	0,549 (0,432)	0,727 (0,180)	0,717 (0,180)	0,680 (0,180)	0,636 (0,178)	0,598 (0,176)	0,702 (0,196)
20	0,736	-0,706 (5,148)	0,596 (0,426)	0,748 (0,162)	0,736 (0,161)	0,686 (0,153)	0,626 (0,139)	0,577 (0,125)	0,722 (0,177)
30	0,787	-1,143 (7,142)	0,614 (0,415)	0,759 (0,159)	0,745 (0,158)	0,687 (0,148)	0,619 (0,128)	0,565 (0,108)	0,732 (0,173)
40	0,821	-0,745 (5,228)	0,642 (0,404)	0,776 (0,153)	0,761 (0,152)	0,698 (0,141)	0,624 (0,116)	0,566 (0,094)	0,751 (0,166)
50	0,846	-0,519 (4,708)	0,678 (0,388)	0,796 (0,142)	0,780 (0,142)	0,710 (0,132)	0,626 (0,108)	0,563 (0,083)	0,770 (0,156)
60	0,864	-0,634 (7,183)	0,717 (0,369)	0,819 (0,139)	0,802 (0,139)	0,730 (0,132)	0,639 (0,106)	0,569 (0,079)	0,794 (0,152)
70	0,879	0,0839 (4,505)	0,763 (0,339)	0,838 (0,123)	0,820 (0,124)	0,742 (0,122)	0,642 (0,099)	0,567 (0,073)	0,814 (0,137)
80	0,891	-0,175 (5,491)	0,757 (0,336)	0,841 (0,122)	0,823 (0,124)	0,743 (0,120)	0,640 (0,096)	0,564 (0,068)	0,818 (0,134)
90	0,901	0,162 (5,637)	0,804 (0,299)	0,863 (0,113)	0,846 (0,116)	0,764 (0,116)	0,653 (0,095)	0,569 (0,067)	0,841 (0,126)
100	0,909	-0,059 (6,426)	0,815 (0,296)	0,867 (0,113)	0,850 (0,115)	0,768 (0,117)	0,655 (0,096)	0,570 (0,066)	0,846 (0,124)
120	0,921	0,369 (4,856)	0,847 (0,256)	0,886 (0,101)	0,869 (0,104)	0,784 (0,110)	0,664 (0,091)	0,573 (0,060)	0,867 (0,110)
150	0,935	0,430 (6,362)	0,878 (0,222)	0,907 (0,086)	0,891 (0,090)	0,806 (0,099)	0,676 (0,085)	0,574 (0,055)	0,889 (0,096)

TAB. 3.2. Espérance et écart type des poids pour des données lognormales,  $\mu_X = 0$ ,  $\mu_Y = 0,3$  et  $\sigma_X^2 = \sigma_Y^2 = 1$

$n$	EQM EVM (ET)	EQM EVP <sub>CV</sub> (ET)	EQM EVP <sub>CVtrouq</sub> (ET)	EQM EVP <sub>0</sub> (ET)	EQM EVP <sub>0,05</sub> (ET)	EQM EVP <sub>0,25</sub> (ET)	EQM EVP <sub>0,5</sub> (ET)	EQM EVP <sub>0,75</sub> (ET)	EQM EVP <sub>W</sub> (ET)
10	0,094 (0,138)	1,130 (24,843)	0,076 (0,111)	0,072 (0,111)	0,071 (0,109)	0,066 (0,101)	0,061 (0,091)	0,058 (0,082)	0,071 (0,109)
20	0,051 (0,072)	0,216 (1,567)	0,040 (0,058)	0,038 (0,057)	0,037 (0,055)	0,034 (0,049)	0,030 (0,043)	0,028 (0,039)	0,037 (0,055)
30	0,033 (0,050)	0,286 (2,610)	0,027 (0,043)	0,025 (0,041)	0,025 (0,040)	0,022 (0,035)	0,019 (0,029)	0,018 (0,025)	0,025 (0,040)
40	0,025 (0,036)	0,151 (1,435)	0,019 (0,031)	0,018 (0,029)	0,018 (0,028)	0,016 (0,024)	0,014 (0,020)	0,013 (0,018)	0,018 (0,028)
50	0,019 (0,025)	0,125 (0,743)	0,014 (0,021)	0,014 (0,019)	0,013 (0,019)	0,012 (0,016)	0,010 (0,014)	0,010 (0,014)	0,013 (0,019)
60	0,017 (0,023)	0,215 (3,046)	0,013 (0,018)	0,012 (0,018)	0,012 (0,017)	0,010 (0,015)	0,009 (0,013)	0,008 (0,013)	0,011 (0,017)
70	0,013 (0,018)	0,161 (1,095)	0,010 (0,015)	0,010 (0,014)	0,009 (0,014)	0,008 (0,012)	0,007 (0,010)	0,006 (0,009)	0,009 (0,014)
80	0,012 (0,016)	0,252 (2,619)	0,009 (0,013)	0,008 (0,013)	0,008 (0,012)	0,007 (0,010)	0,006 (0,009)	0,006 (0,008)	0,008 (0,012)
90	0,011 (0,016)	0,106 (0,632)	0,007 (0,013)	0,007 (0,012)	0,007 (0,012)	0,006 (0,010)	0,005 (0,008)	0,005 (0,007)	0,007 (0,012)
100	0,009 (0,013)	0,120 (1,561)	0,007 (0,011)	0,007 (0,011)	0,006 (0,010)	0,005 (0,008)	0,005 (0,007)	0,004 (0,007)	0,006 (0,010)
120	0,008 (0,011)	0,070 (0,455)	0,006 (0,009)	0,006 (0,009)	0,005 (0,008)	0,004 (0,007)	0,004 (0,006)	0,004 (0,005)	0,005 (0,008)
150	0,006 (0,009)	0,065 (0,481)	0,004 (0,007)	0,004 (0,007)	0,004 (0,006)	0,003 (0,005)	0,003 (0,004)	0,003 (0,004)	0,004 (0,006)

TAB. 3.3. EQM des EVP et écart type des différences au carré entre l'EVP et  $\mu_X$ , pour des données lognormales,  $\mu_X = \mu_Y = 0$  et  $\sigma_X^2 = \sigma_Y^2 = 1$



$n$	EQM EVM (ET)	EQM EVP <sub>CV</sub> (ET)	EQM EVP <sub>CV tronq</sub> (ET)	EQM EVP <sub>0</sub> (ET)	EQM EVP <sub>0,05</sub> (ET)	EQM EVP <sub>0,25</sub> (ET)	EQM EVP <sub>0,5</sub> (ET)	EQM EVP <sub>0,75</sub> (ET)	EQM EVP <sub>IV</sub> (ET)
10	0,094 (0,138)	0,353 (3,350)	0,084 (0,117)	0,077 (0,113)	0,076 (0,112)	0,073 (0,105)	0,070 (0,097)	0,068 (0,091)	0,077 (0,112)
20	0,051 (0,072)	0,217 (2,560)	0,051 (0,066)	0,045 (0,060)	0,044 (0,059)	0,042 (0,056)	0,041 (0,053)	0,042 (0,053)	0,044 (0,059)
30	0,033 (0,050)	0,235 (2,168)	0,038 (0,053)	0,032 (0,046)	0,032 (0,046)	0,032 (0,044)	0,032 (0,043)	0,034 (0,043)	0,032 (0,046)
40	0,025 (0,036)	0,087 (0,801)	0,029 (0,038)	0,025 (0,033)	0,024 (0,033)	0,025 (0,032)	0,026 (0,032)	0,028 (0,033)	0,025 (0,033)
50	0,019 (0,025)	0,044 (0,086)	0,024 (0,032)	0,020 (0,025)	0,020 (0,025)	0,020 (0,025)	0,023 (0,027)	0,026 (0,029)	0,020 (0,025)
60	0,017 (0,023)	0,066 (0,721)	0,021 (0,027)	0,017 (0,023)	0,017 (0,023)	0,018 (0,023)	0,020 (0,025)	0,024 (0,027)	0,017 (0,023)
70	0,013 (0,018)	0,030 (0,085)	0,017 (0,023)	0,014 (0,019)	0,014 (0,019)	0,015 (0,020)	0,018 (0,022)	0,022 (0,024)	0,014 (0,019)
80	0,012 (0,016)	0,024 (0,066)	0,015 (0,021)	0,013 (0,017)	0,013 (0,017)	0,014 (0,018)	0,017 (0,020)	0,022 (0,022)	0,013 (0,017)
90	0,011 (0,016)	0,027 (0,229)	0,014 (0,020)	0,011 (0,016)	0,012 (0,016)	0,013 (0,017)	0,016 (0,019)	0,021 (0,020)	0,012 (0,016)
100	0,009 (0,013)	0,022 (0,109)	0,012 (0,018)	0,010 (0,014)	0,010 (0,015)	0,012 (0,016)	0,015 (0,017)	0,020 (0,019)	0,010 (0,015)
120	0,008 (0,011)	0,015 (0,042)	0,010 (0,014)	0,009 (0,012)	0,009 (0,012)	0,010 (0,013)	0,014 (0,015)	0,019 (0,017)	0,009 (0,012)
150	0,006 (0,009)	0,015 (0,173)	0,008 (0,011)	0,007 (0,009)	0,007 (0,009)	0,008 (0,010)	0,012 (0,013)	0,018 (0,015)	0,007 (0,009)

TAB. 3.4. EQM des EVP et écart type des différences au carré entre l'EVP et  $\mu_X$ , pour des données lognormales,  $\mu_X = 0$ ,  $\mu_Y = 0,3$  et  $\sigma_X^2 = \sigma_Y^2 = 1$

$n$	$\frac{EQM_{CV}}{EQM_{EVM}}$	$\frac{EQM_{CV\text{trong}}}{EQM_{EVM}}$	$\frac{EQM_0^*}{EQM_{EVM}}$	$\frac{EQM_{0,05}^*}{EQM_{EVM}}$	$\frac{EQM_{0,25}^*}{EQM_{EVM}}$	$\frac{EQM_{0,5}^*}{EQM_{EVM}}$	$\frac{EQM_{0,75}^*}{EQM_{EVM}}$	$\frac{EQM_{IV}^*}{EQM_{EVM}}$
10	11,990	0,815	0,767	0,755	0,709	0,656	0,616	0,756
20	4,172	0,784	0,747	0,730	0,664	0,596	0,553	0,729
30	8,449	0,810	0,755	0,736	0,662	0,586	0,541	0,738
40	5,942	0,780	0,744	0,722	0,640	0,562	0,523	0,723
50	6,443	0,769	0,735	0,713	0,632	0,563	0,533	0,713
60	12,685	0,766	0,731	0,707	0,619	0,549	0,520	0,704
70	11,871	0,796	0,734	0,709	0,615	0,536	0,508	0,708
80	20,863	0,766	0,740	0,715	0,620	0,540	0,509	0,714
90	9,644	0,715	0,720	0,692	0,584	0,495	0,461	0,689
100	12,410	0,782	0,736	0,708	0,604	0,522	0,495	0,706
120	8,439	0,734	0,730	0,701	0,596	0,520	0,498	0,699
150	10,286	0,754	0,730	0,700	0,587	0,506	0,485	0,697
200	9,672	0,739	0,727	0,695	0,578	0,500	0,481	0,694
300	11,712	0,782	0,732	0,699	0,583	0,515	0,501	0,698
400	16,079	0,748	0,715	0,677	0,548	0,484	0,478	0,675
500	10,966	0,806	0,734	0,698	0,576	0,519	0,513	0,697
1000	15,057	0,753	0,734	0,693	0,563	0,515	0,513	0,691

TAB. 3.5. Ratios des EQM de l'EVP par rapport à l'EVM pour des données lognormales,  $\mu_X = \mu_Y = 0$  et  $\sigma_X^2 = \sigma_Y^2 = 1$

$n$	$\frac{EQM_{CV}}{EQM_{EVM}}$	$\frac{EQM_{CV\text{trong}}}{EQM_{EVM}}$	$\frac{EQM_0^*}{EQM_{EVM}}$	$\frac{EQM_{0,05}^*}{EQM_{EVM}}$	$\frac{EQM_{0,25}^*}{EQM_{EVM}}$	$\frac{EQM_{0,5}^*}{EQM_{EVM}}$	$\frac{EQM_{0,75}^*}{EQM_{EVM}}$	$\frac{EQM_W^*}{EQM_{EVM}}$
10	3,748	0,893	0,824	0,815	0,780	0,744	0,723	0,818
20	4,180	0,984	0,868	0,859	0,826	0,804	0,814	0,864
30	6,935	1,131	0,958	0,955	0,946	0,958	1,008	0,963
40	3,428	1,140	0,980	0,978	0,983	1,029	1,125	0,985
50	2,270	1,285	1,037	1,042	1,078	1,187	1,364	1,052
60	3,916	1,284	1,037	1,041	1,078	1,203	1,417	1,050
70	2,205	1,293	1,064	1,076	1,153	1,365	1,680	1,083
80	2,063	1,313	1,089	1,105	1,211	1,475	1,839	1,111
90	2,444	1,310	1,076	1,091	1,192	1,478	1,901	1,098
100	2,372	1,336	1,105	1,125	1,253	1,599	2,107	1,132
120	1,903	1,290	1,101	1,123	1,270	1,691	2,314	1,126
150	2,434	1,289	1,115	1,143	1,354	1,985	2,909	1,147
200	1,450	1,265	1,126	1,160	1,435	2,345	3,712	1,162
300	1,347	1,227	1,093	1,128	1,466	2,888	5,229	1,128
400	1,374	1,282	1,098	1,137	1,552	3,554	7,090	1,137
500	1,347	1,237	1,067	1,098	1,477	3,782	8,402	1,098
1000	1,250	1,176	1,034	1,053	1,382	4,898	15,054	1,053

TAB. 3.6. Ratios des EQM de l'EVP par rapport à l'EVM pour des données lognormales,  $\mu_X = 0$ ,  $\mu_Y = 0,3$  et  $\sigma_X^2 = \sigma_Y^2 = 1$

# Chapitre 4

---

## CONCLUSION

Quand on a d'autres échantillons reliés à celui qui nous intéresse, il est utile d'incorporer l'information contenue dans les autres échantillons afin d'améliorer notre estimation. À ces fins, la vraisemblance pondérée a été développée. En mettant un poids sur l'information de chaque échantillon, on peut réduire l'erreur quadratique moyenne d'un estimateur, puisque si les poids sont bien choisis, la variance de l'estimateur va diminuer même si le biais augmente. Le choix des poids peut se faire de plusieurs manières (voir Wang et Zidek, 2005a, p. 484 pour un survol de d'autres manières que celles discutées ici). Wang et Zidek ont montré que les poids peuvent être choisis de manière adaptative, en fonction des données, et ont développé le choix des poids par la validation croisée. On a repris le choix de poids par la méthode de validation croisée *leave-one-out*, et on a développé le choix de poids par le bootstrap comme approche alternative, dans le cas de deux échantillons indépendants i.i.d. de taille égale. En particulier on a examiné les cas de données normales et lognormales.

Dans le cas de données normales, on a vu au Chapitre 2 que Wang et Zidek (2005a) ont démontré que  $\hat{\lambda}_{CV} \xrightarrow{\text{Pr}} 1$  quand  $\mu_X \neq \mu_Y$ . Par contre, si  $\mu_X = \mu_Y$ , il n'y a pas de convergence. La suggestion de Wang et Zidek dans ce cas, d'ajouter une constante  $\delta$  au dénominateur de  $1 - \hat{\lambda}_{CV}$ , mène à la convergence mais vers la mauvaise valeur, 0 au lieu de  $1 - \frac{\sigma_X^2}{\sigma_X^2 + \sigma_Y^2}$ . Nos simulations, faites pour  $\mu_X = \mu_Y = 0$  et  $|\mu_X - \mu_Y| = 0,3$ , avec  $\sigma_X^2 = \sigma_Y^2 = 1$  et pour des valeurs de  $n$  entre 10 et 50000, ont montré que  $\hat{\lambda}_{CV}$  ne performe pas bien du tout, même quand les moyennes sont différentes, sauf si  $n$  est très grand. Par contre, la modification  $\hat{\lambda}_{CV_{mod}}$  de

Wang et Zidek (2005b) avait la meilleure performance en général parmi tous les poids qu'on a simulés. Toutefois la modification a été développée pour des données normales et ne peut pas être facilement généralisée à d'autres distributions.

Pour ce qui est du bootstrap, on l'a utilisé afin d'estimer chaque terme de l'erreur quadratique moyenne de l'estimateur de vraisemblance pondérée. On a introduit le facteur  $n^{-p}$  au terme du biais afin de tenter de réduire un effet potentiel non désiré dans le cas de moyennes égales. La théorie qu'on a développée indique qu'il faut  $0 < p < 1$  pour que le poids converge à la valeur optimale dans les deux cas  $\mu_X = \mu_Y$  et  $\mu_X \neq \mu_Y$ . Par contre, nos simulations ont démontré qu'en pratique,  $p = 0$  fonctionne le mieux en général, c'est-à-dire le facteur  $n^{-p}$  n'est pas nécessaire.

Les conclusions générales des simulations pour données normales étaient que les poids de validation croisée modifiée et bootstrap apportent une amélioration par rapport à la vraisemblance maximale quand  $|\mu_X - \mu_Y| \leq 0,4\sigma_X^2 = 0,4\sigma_Y^2$ . Les poids qui performant le mieux parmi ceux qu'on a vus sont  $\hat{\lambda}_{CVmod}$  et  $\hat{\lambda}_0^*$ .

Dans le cas des données lognormales, on a vu au Chapitre 3 que le poids par validation croisée ne peut pas être exprimé analytiquement ; il faut le calculer numériquement. Wang et Zidek (2005a) ont démontré que  $\hat{\lambda}_{CV} \xrightarrow{\text{Pr}} 1$  quand  $\mu_X \neq \mu_Y$  et quand les variances sont connues, mais il y a des trous dans la démonstration. On a fait des simulations pour  $\mu_X = \mu_Y = 0$  et pour  $\mu_X = 0$  et  $\mu_Y = 0,3$ , avec  $\sigma_X^2 = \sigma_Y^2 = 1$  et pour des valeurs de  $n$  entre 10 et 1000, en calculant le poids par validation croisée à l'aide de la méthode numérique de bisection. On a vu que le poids  $\hat{\lambda}_{CV}$  encore une fois ne performe pas bien. Quant à la théorie de Wang et Zidek, on peut seulement dire que s'il y a convergence, elle est très lente.

Le poids de validation croisée performe mieux si on le tronque entre 0 et 1 afin d'obtenir  $\hat{\lambda}_{CVtronq}$ . On a aussi vu que ce poids est plus efficace quand on estime la variance lors des calculs, même si elle est connue. Par contre, les poids bootstrap, en particulier  $\hat{\lambda}_0^*$ , performant mieux, puisque le bootstrap permet de travailler avec un estimé de l'erreur quadratique moyenne de l'estimateur du paramètre d'intérêt tandis qu'avec la validation croisée il faut passer par la prévision. On a

vu que l'application et les résultats du bootstrap sont les mêmes que dans le cas normal puisque l'erreur quadratique prend la même forme.

Les conclusions générales des simulations lognormales étaient que les poids de validation croisée tronquée et bootstrap apportent une amélioration par rapport à la vraisemblance maximale quand  $|\mu_X - \mu_Y| \leq 0,4\sigma_X^2 = 0,4\sigma_Y^2$ , mais cette fois c'était clair que le poids  $\hat{\lambda}_0^*$  est le meilleur parmi ceux qu'on a étudiés, contrairement au cas de données normales où on avait la modification  $\hat{\lambda}_{CVmod}$  qui performe encore mieux que  $\hat{\lambda}_0^*$ .

Il y a plusieurs volets de travail futur à entreprendre sur ce problème. Une chose à faire, bien sûr, est la correction de la démonstration du résultat asymptotique de Wang et Zidek (2005a, Corollary 3.2) pour le cas lognormal, ainsi que sa généralisation au cas de variances inconnues et estimées. Il serait aussi utile d'explorer la modification  $\delta(n, X, Y)$  qui mène au poids  $\hat{\lambda}_{CVmod}$  dans le cas de données normales, afin de trouver un développement théorique d'une telle modification, ainsi que de généraliser ce type de modification à d'autres cas comme la lognormale. De plus, Wang et Zidek (2005a, p. 470) ont aussi considéré la situation plus générale où les échantillons n'ont pas nécessairement la même taille. Le choix de poids par bootstrap devrait être généralisé à cette situation aussi.

Ici, on a travaillé avec la distribution normale, pour laquelle la fonction  $\phi$  est linéaire, et la distribution lognormale pour laquelle  $\phi$  n'est plus linéaire et donc la validation croisée est plus compliquée, mais pour laquelle le bootstrap est fait exactement de la même manière que dans le cas normal puisque la forme de l'erreur quadratique moyenne est la même. Wang et Zidek (2005a, section 4) ont aussi considéré la distribution de Poisson lors de leurs simulations; les résultats étant similaires à celles du cas normal puisque  $\phi$  est aussi linéaire pour la distribution Poisson. Il faudrait maintenant se tourner vers d'autres distributions pour lesquelles l'application du bootstrap en plus que l'application de la validation croisée est différente, afin de comparer ces deux méthodes de sélection de poids dans une plus grande variété de cas.

## Bibliographie

---

- Samuel D. CONTE et Carl DE BOOR : *Elementary Numerical Analysis*. McGraw Hill, New York, 3<sup>e</sup> édition, 1980.
- Bradley EFRON : Bootstrap methods : Another look at the jackknife. *Ann. Stat.*, 7:1–26, 1979.
- Bradley EFRON : Estimating the error rate of a prediction rule : Improvement on cross-validation. *J. Amer. Statist. Assoc.*, 78:316–331, 1983.
- Seymour GEISSER : The predictive sample reuse method with applications. *J. Amer. Statist. Assoc.*, 70:320–328, 1975.
- Feifang HU et James V. ZIDEK : The weighted likelihood. *Canad. J. Statistics*, 30:347–371, 2002.
- Robert J. SERFLING : *Approximation Theorems of Mathematical Statistics*. John Wiley & Sons, Inc., New York, 1980.
- Jun SHAO et Dongsheng TU : *The Jackknife and Bootstrap*. Springer, New York, 1995.
- M. STONE : Cross-validatory choice and assessment of statistical predictions (with discussion). *J. Roy. Statist. Soc. Ser. B*, 36:111–147, 1974.
- M. STONE : Asymptotics for and against cross-validation. *Biometrika*, 64:29–35, 1977.
- Xiaogang WANG et James V. ZIDEK : Selecting likelihood weights by cross-validation. *Ann. Stat.*, 33:463–500, 2005a.
- Xiaogang WANG et James V. ZIDEK : Computational aspects on cross-validation. Unpublished, 18 pp, 2005b.

# Annexe A

---

## SIMULATIONS NORMALES

### A.1. PROGRAMMES

```
#vecteur des valeurs de p#
pvector<-matrix(seq(0,1,by=0.05),byrow=FALSE)

#simulation de base#
#n=taille des deux echantillons X et Y#
#nsimnombre de tels echantillons tires afin de calculer
la moyenne des estimateurs etc#
simulate<-function(n, mx, sx, my, sy, nsim)
{
  lambda.opt<-1-(sx^2)/(n*(mx-my)^2 + sx^2 + sy^2)

  #initialisation des vecteurs#

  #matrice lambda.star - lignes pour nsim repetitions,#
  #colonnes pour valeurs de p#
  lambda.star<-matrix(nrowsim,ncol=length(pvector),0)

  lambda.cv<-matrix(nrowsim,ncol=1,0)
  lambda.cv.mod<-matrix(nrowsim,ncol=1,0)
  lambda.welch<-matrix(nrowsim,ncol=2,0)
```



```
p.welch<-matrix(nrowsim,ncol=2,0)
j.welch<-matrix(nrow=1,ncol=2,0)
wle.welch<-matrix(nrowsim,ncol=2,0)
mse.wle.welch<-matrix(nrowsim,ncol=2,0)

mle<-matrix(nrowsim,ncol=1,0)
mse.mle<-matrix(nrowsim,ncol=1,0)

wle.cv<-matrix(nrowsim,ncol=1,0)
mse.wle.cv<-matrix(nrowsim,ncol=1,0)
wle.cv.mod<-matrix(nrowsim,ncol=1,0)
mse.wle.cv.mod<-matrix(nrowsim,ncol=1,0)

wle.star<-matrix(nrowsim,ncol=length(pvector),0)
mse.wle.star<-matrix(nrowsim,ncol=length(pvector),0)

p.value<-matrix(nrowsim,ncol=1,0)

#pour pouvoir reproduire les memes resultats#
set.seed(3)

for (i in 1:nrowsim)
{
  #tirer les deux echantillons#
  X<-rnorm(n, mean=mx, sd=sx)
  Y<-rnorm(n, mean=my, sd=sy)

  #test de welch#
  t<-sqrt(n)*(mean(X)-mean(Y))/sqrt(var(X) + var(Y))
  nu<-(n-1)*(var(X)+var(Y))^2/(var(X)^2+var(Y)^2)
```

```

if (t>0) p.value[i]<-2*(1-pt(t, nu)) else p.value[i]<-2*pt(t,nu)

#maximum de vraisemblance: estimateur et eqm#
mle[i]<-mean(X)
mse.mle[i]<-(mle[i]-mx)^2

#calcul des poids#
# lambda bootstrap (exact) pour chaque p#
j<-1
for (p in pvector)
{
  lambda.star[i,j]<-1-(((var(X)-(cov(X,Y)))*((n-1)/n))/(((n-1)/n)*
    (var(X)+var(Y)-2*cov(X,Y))+n^(1-p)*(mean(X)-mean(Y))^2)
  j<-j+1
}

# validation croisee#
lambda.cv[i]<-1-(((var(X)-cov(X,Y))/(n-1))/(n*(n-2)*(mean(X)-mean(Y))^2/(n-1)^2
  + sum((X-Y)^2)/(n*(n-1)^2))

# validation croisee modifiee#
lambda.cv.mod[i]<-1-(((var(X)-cov(X,Y))/(n-1))/(n*(n-2)*
  (mean(X)-mean(Y))^2/(n-1)^2 + sum((X-Y)^2)/(n*(n-1)^2)
  + (0.5*var(X))/n
  + ((1/(1+abs(mean(X)-mean(Y)))/sqrt(var(X)/n+var(Y)/n))*var(X))/log(n))

#estimateurs et eqm resultants#
for (j in 1:length(pvector))
{

```

```

wle.star[i,j]<-lambda.star[i,j]*mle[i]+(1-lambda.star[i,j])*mean(Y)
mse.wle.star[i,j]<-(wle.star[i,j]-mx)^2
}

wle.cv[i]<-lambda.cv[i]*mle[i]+(1-lambda.cv[i])*mean(Y)
wle.cv.mod[i]<-lambda.cv.mod[i]*mle[i]+(1-lambda.cv.mod[i])*mean(Y)

mse.wle.cv[i]<-(wle.cv[i]-mx)^2
mse.wle.cv.mod[i]<-(wle.cv.mod[i]-mx)^2

#estimateur par test de Welch
if (p.value[i]<=0.5) {p.welch[i,]<-c(0.05,0.05); j.welch<-c(2,2)}
else {p.welch[i,]<-c(0.25,0.5); j.welch<-c(6,11)}
lambda.welch[i,]<-c(lambda.star[i,j.welch[1]],lambda.star[i,j.welch[2]])
wle.welch[i,]<-c(wle.star[i,j.welch[1]],wle.star[i,j.welch[2]])
mse.wle.welch[i,]<-c(mse.wle.star[i,j.welch[1]],mse.wle.star[i,j.welch[2]])
}

#initialiser#
lambda.star.bar1<-matrix(0, nrow=1, ncol=length(pvector))
lambda.star.sd1<-matrix(0, nrow=1, ncol=length(pvector))
wle.star.bar1<-matrix(0, nrow=1, ncol=length(pvector))
wle.star.sd1<-matrix(0, nrow=1, ncol=length(pvector))
mse.wle.star.bar1<-matrix(0, nrow=1, ncol=length(pvector))
mse.wle.star.sd1<-matrix(0, nrow=1, ncol=length(pvector))
ratio.star.mle1<-matrix(0, nrow=1, ncol=length(pvector))

#calculer les moyennes des colonnes des matrices de differentes valeurs de p#
for (j in 1:length(pvector))
{
lambda.star.bar1[j]<-mean(lambda.star[,j])

```

```
lambda.star.sd1[j]<-sd(lambda.star[,j])
wle.star.bar1[j]<-mean(wle.star[,j])
wle.star.sd1[j]<-sd(wle.star[,j])
mse.wle.star.bar1[j]<-mean(mse.wle.star[,j])
mse.wle.star.sd1[j]<-sd(mse.wle.star[,j])
ratio.star.mle1[j]<-mse.wle.star.bar1[j]/mean(mse.mle)
}

#garder les modeles#
model<-c(n,mx,sx,my,sy)
names(model)<-c("n","mx","sx","my","sy")

#garder tous les nsim valeurs des estimateurs#
details<-list(lambda.star=lambda.star,
lambda.cv=lambda.cv, lambda.cv.mod=lambda.cv.mod, mle=mle,
mse.mle=mse.mle, wle.cv=wle.cv,mse.wle.cv=mse.wle.cv,
wle.cv.mod=wle.cv.mod, mse.wle.cv.mod=mse.wle.cv.mod,
wle.star=wle.star, mse.wle.star=mse.wle.star, p.value=p.value,
p.welch=p.welch, lambda.welch=lambda.welch, wle.welch=wle.welch,
mse.wle.welch=mse.wle.welch)

#les resultats pour tous les p#
lambda.star.all.results<-list(model=model, lambda.opt=lambda.opt,
lambda.star.bar=lambda.star.bar1, lambda.star.sd=lambda.star.sd1,
wle.star.bar=wle.star.bar1, wle.star.sd=wle.star.sd1,
mse.wle.star.bar=mse.wle.star.bar1, mse.wle.star.sd=mse.wle.star.sd1,
ratio.star.mle=ratio.star.mle1)

#les resultats pour cv, star.0, star.05, star.25, star.5, star.75#
lambda.star.bar<-matrix(c(lambda.star.bar1[1],lambda.star.bar1[2],
lambda.star.bar1[6],lambda.star.bar1[11],lambda.star.bar1[16]),
```

```

nrow=1,ncol=5,byrow=FALSE)
lambda.star.sd<-matrix(c(lambda.star.sd1[1],lambda.star.sd1[2],
  lambda.star.sd1[6],lambda.star.sd1[11],lambda.star.sd1[16]),nrow=1,
  ncol=5,byrow=FALSE)
wle.star.bar<-matrix(c(wle.star.bar1[1],wle.star.bar1[2],wle.star.bar1[6],
  wle.star.bar1[11],wle.star.bar1[16]),nrow=1,ncol=5,byrow=FALSE)
wle.star.sd<-matrix(c(wle.star.sd1[1],wle.star.sd1[2],wle.star.sd1[6],
  wle.star.sd1[41],wle.star.sd1[16]),nrow=1,ncol=5,byrow=FALSE)
mse.wle.star.bar<-matrix(c(mse.wle.star.bar1[1],mse.wle.star.bar1[2],
  mse.wle.star.bar1[6],mse.wle.star.bar1[11],mse.wle.star.bar1[16]),
  nrow=1,ncol=5,byrow=FALSE)
mse.wle.star.sd<-matrix(c(mse.wle.star.sd1[1],mse.wle.star.sd1[2],
  mse.wle.star.sd1[6],mse.wle.star.sd1[11],mse.wle.star.sd1[16]),
  nrow=1,ncol=5,byrow=FALSE)
ratio.star.mle<-matrix(c(ratio.star.mle1[1],ratio.star.mle1[2],
  ratio.star.mle1[6],ratio.star.mle1[11],ratio.star.mle1[16]),
  nrow=1,ncol=5,byrow=FALSE)
colnames(lambda.star.bar)<-c("p=0","p=.05","p=.25","p=.5","p=.75")
colnames(lambda.star.sd)<-c("p=0","p=.05","p=.25","p=.5","p=.75")
colnames(wle.star.bar)<-c("p=0","p=.05","p=.25","p=.5","p=.75")
colnames(wle.star.sd)<-c("p=0","p=.05","p=.25","p=.5","p=.75")
colnames(mse.wle.star.bar)<-c("p=.0","p=.05","p=.25","p=.5","p=.75")
colnames(mse.wle.star.sd)<-c("p=0","p=.05","p=.25","p=.5","p=.75")
colnames(ratio.star.mle)<-c("p=0","p=.05","p=.25","p=.5","p=.75")

lambda.welch.bar<-matrix(c(mean(lambda.welch[,1]),mean(lambda.welch[,2])),
  nrow=1,ncol=2)
lambda.welch.sd<-matrix(c(sd(lambda.welch[,1]),sd(lambda.welch[,2])),
  nrow=1,ncol=2)
wle.welch.bar<-matrix(c(mean(wle.welch[,1]),mean(wle.welch[,2])),
  nrow=1,ncol=2)

```

```

wle.welch.sd<-matrix(c(sd(wle.welch[,1]),sd(wle.welch[,2])),
  nrow=1,ncol=2)
mse.wle.welch.bar<-matrix(c(mean(mse.wle.welch[,1]),
  mean(mse.wle.welch[,2])), nrow=1,ncol=2)
mse.wle.welch.sd<-matrix(c(sd(mse.wle.welch[,1]),
  sd(mse.wle.welch[,2])), nrow=1,ncol=2)
ratio.welch.mle<-matrix(c(mean(mse.wle.welch[,1])/mean(mse.mle),
  mean(mse.wle.welch[,2])/mean(mse.mle)),nrow=1,ncol=2)
colnames(lambda.welch.bar)<-c("p.alt=0.25","p.alt=0.5")
colnames(lambda.welch.sd)<-c("p.alt=0.25","p.alt=0.5")
colnames(wle.welch.bar)<-c("p.alt=0.25","p.alt=0.5")
colnames(wle.welch.sd)<-c("p.alt=0.25","p.alt=0.5")
colnames(mse.wle.welch.bar)<-c("p.alt=0.25","p.alt=0.5")
colnames(mse.wle.welch.sd)<-c("p.alt=0.25","p.alt=0.5")
colnames(ratio.welch.mle)<-c("p.alt=0.25","p.alt=0.5")

results<-list(model=model, lambda.opt=lambda.opt,
  ic.width0=2*sqrt(var(lambda.star[,trunc(length(pvector)/2)])/nsim),
  lambda.star.bar=lambda.star.bar, lambda.star.sd=lambda.star.sd,
  lambda.cv.bar=mean(lambda.cv), lambda.cv.sd=sd(lambda.cv),
  lambda.cv.mod.bar=mean(lambda.cv.mod),
  lambda.cv.mod.sd=sd(lambda.cv.mod), mle.bar=mean(mle),
  mle.sd=sd(mle), mse.mle.bar=mean(mse.mle),
  mse.mle.sd=sd(mse.mle), wle.cv.bar=mean(wle.cv), wle.cv.sd=sd(wle.cv),
  mse.wle.cv.bar=mean(mse.wle.cv), mse.wle.cv.sd=sd(mse.wle.cv),
  wle.cv.mod.bar=mean(wle.cv.mod), wle.cv.mod.sd=sd(wle.cv.mod),
  mse.wle.cv.mod.bar=mean(mse.wle.cv.mod),
  mse.wle.cv.mod.sd=sd(mse.wle.cv.mod), wle.star.bar=wle.star.bar,
  wle.star.sd=wle.star.sd, mse.wle.star.bar=mse.wle.star.bar,
  mse.wle.star.sd=mse.wle.star.sd,
  ratio.cv.mle=(mean(mse.wle.cv)/mean(mse.mle)),

```

```

ratio.cv.mod.mle=(mean(mse.wle.cv.mod)/mean(mse.mle)),
ratio.star.mle=ratio.star.mle,welch.p.value=mean(p.value),
sd.welch=sd(p.value), lambda.welch.bar=lambda.welch.bar,
lambda.welch.sd=lambda.welch.sd, wle.welch.bar=wle.welch.bar,
wle.welch.sd=wle.welch.sd, mse.wle.welch.bar=mse.wle.welch.bar,
mse.wle.welch.sd=mse.wle.welch.sd, ratio.welch.mle=ratio.welch.mle)

res<-list(results=results,lambda.star.all=lambda.star.all.results,
          details=details)
res
}

#fonction qui applique la simulation de base a plusieurs valeurs de n#
#nvec est un vecteur contenant length(nvec) valeurs de n,#
#p.e. nvec=c(10,20,30)#
sim1<-function(mx, sx, my, sy, nsim, nvec, resuname)
{
  res<-vector("list",length(nvec))
  resu<-vector("list",4)
  names(resu)<-c("resultats","tous.p","details","meilleur.p")
  resu[[4]]<-matrix(0,nrow=length(nvec),ncol=3)

  #"a" est un indice qui va de 1 a length(nvec)#
  a<-1
  for (n in nvec)
  {
    #faire la simulation de base#
    res[[a]]<-simulate(n, mx, sx, my, sy, nsim)

#mettre les resultats dans une liste output qui a la structure desiree#
#resu[[1]] liste des resultats de chaque a#

```

```

#resu[[3]] liste des details de chaque a#
  resu[[1]][[a]]<-res[[a]][[1]]
  names(resu[[1]])[a]<-paste("iter",a,sep="")
  resu[[2]][[a]]<-res[[a]][[2]]
  names(resu[[2]])[a]<-paste("iter",a,sep="")
  resu[[3]][[a]]<-res[[a]][[3]]
  names(resu[[3]])[a]<-paste("iter",a,sep="")

#faire et sauvegarder le graphique des ratio.star.mle#
par(mfrow=c(1,1))
mod<-res[[a]][[1]]$model
plot(as.vector(res[[a]][[2]]$ratio.star.mle),pvector,
     xlab=paste("iter",a,"$ratio.star.mle",sep=""))
title( paste("\nModele : n = ", mod[1], ", mx = ",mod[2],", sx = ",mod[3],",
  my = ",mod[4],", sy = ",mod[5], sep=""), outer=TRUE, )
savePlot(filename=paste("C:/Documents and Settings/Joanne/Desktop/
  thesis-simulations/",resuname,".pvector2.",a,sep=""), type=c("ps"))
#garder la meilleure valeur de p#
resu[[4]][a,<-c(n,min(as.vector(res[[a]][[2]]$ratio.star.mle)),
  pvector[which.min(as.vector(res[[a]][[2]]$ratio.star.mle))])
colnames(resu[[4]])<-c("n","min.ratio.star","min.p")

  a<-a+1
}
resu
}

#fonction pour creer des histogrammes des poids#
#resu=resultat de la fonction precedente,
#et resuname=son nom entre guillemets#
histofunct<-function(resu, resuname)

```



```
{  
  #mettre les histogrammes pour 8 poids (d'un meme modele)  
  #sur une meme page#  
  
  #faire des histogrammes pour chaque n dans nvec#  
  ngraphs<-length(resu[[2]])  
  for(a in 1:ngraphs)  
  {  
    par(mfrow=c(4,2))  
    mod<-resu$resultats[[a]]$model  
    hist(resu$details[[a]]$lambda.cv, xlab="lambda.cv", main=NULL)  
    hist(resu$details[[a]]$lambda.cv.mod, xlab="lambda.cv.mod",  
          main=NULL)  
    hist(as.matrix(resu$details[[a]]$lambda.star)[,1],  
          xlab="lambda.star(0)", main=NULL)  
    hist(as.matrix(resu$details[[a]]$lambda.star)[,2],  
          xlab="lambda.star(.05)", main=NULL)  
    hist(as.matrix(resu$details[[a]]$lambda.star)[,6],  
          xlab="lambda.star(.25)", main=NULL)  
    hist(as.matrix(resu$details[[a]]$lambda.star)[,11],  
          xlab="lambda.star(.5)", main=NULL)  
    hist(as.matrix(resu$details[[a]]$lambda.star)[,16],  
          xlab="lambda.star(.75)", main=NULL)  
    hist(resu$details[[a]]$lambda.welch[,2], xlab="lambda.welch(0.5)",  
          main=NULL)  
  
    title( paste("\nModele : n = ", mod[1], ", mx = ",mod[2],", sx = ",mod[3],",  
                my = ",mod[4],", sy = ",mod[5], sep=""), outer=TRUE, )  
  
    #sauvegarder chaque page de 8 histogrammes lorsqu'elle est complete#  
    savePlot(filename=paste("C:/Documents and Settings/Joanne/Desktop/
```

```
thesis-simulations/",resuname,".histfinal.",a,sep=""), type=c("ps"))
```

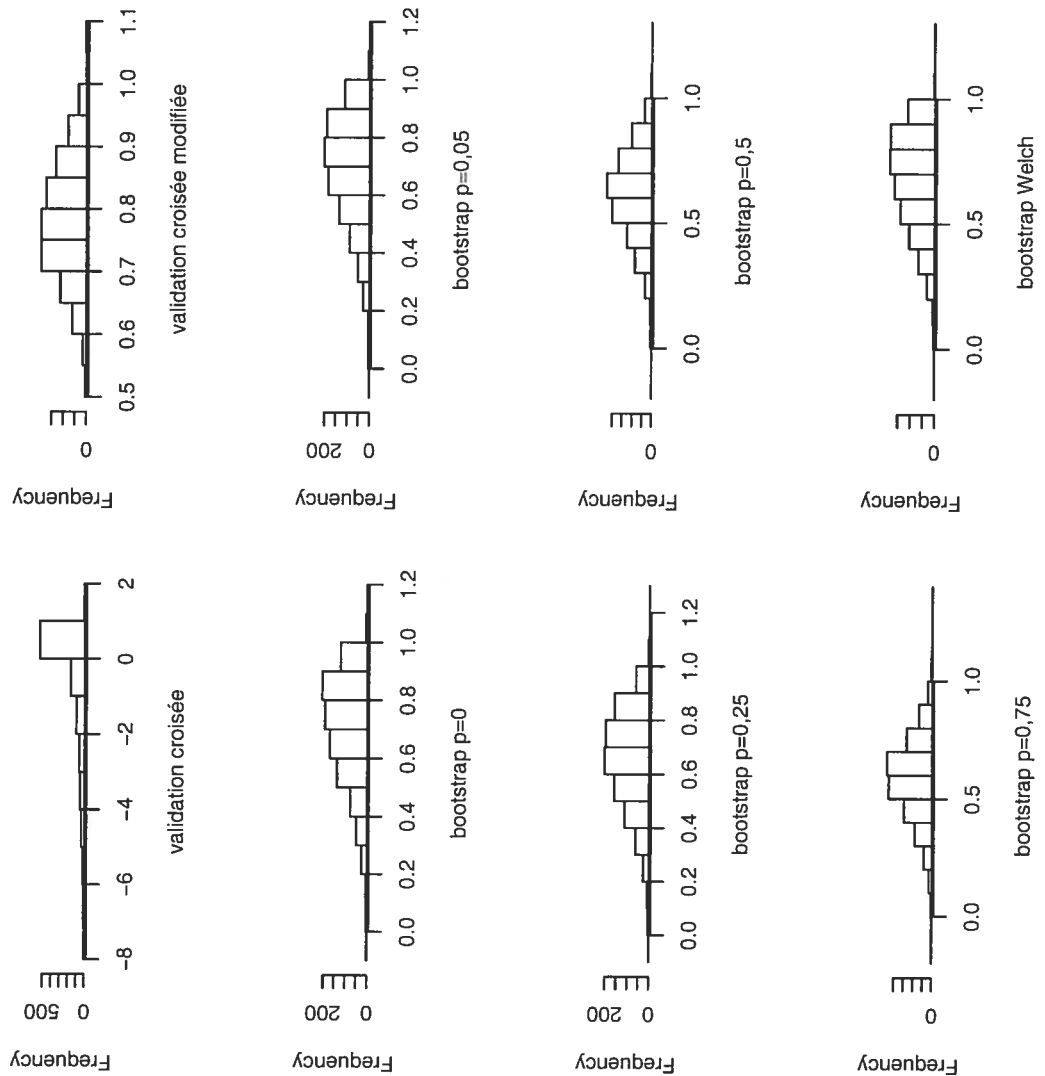
```
}
```

```
}
```

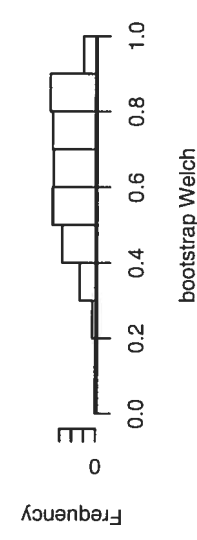
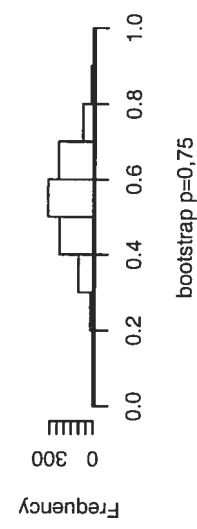
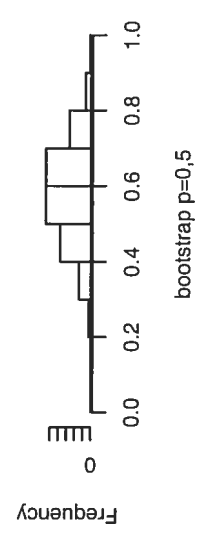
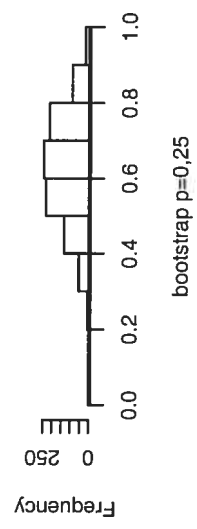
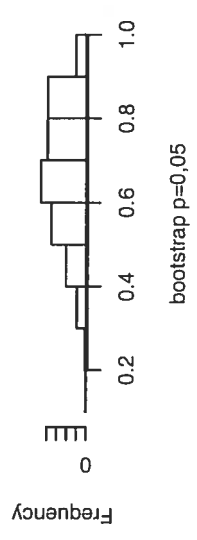
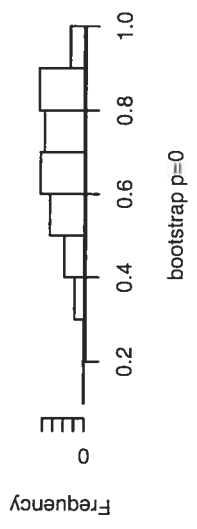
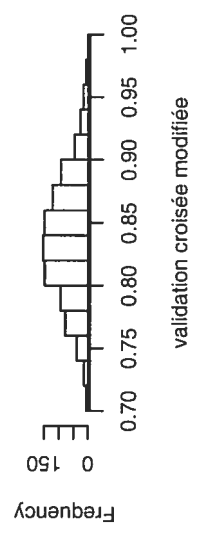
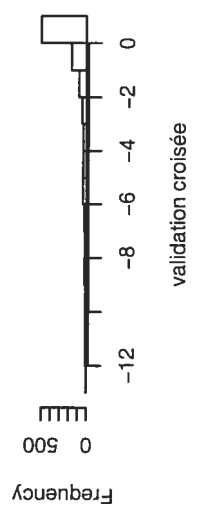
A.2. HISTOGRAMMES DES HUIT POIDS  $\hat{\lambda}$  EXAMINÉS, POUR PLUSIEURS VALEURS DE  $n$  ET DANS LES DEUX CAS  $\mu_X = \mu_Y$  ET  $|\mu_X - \mu_Y| = 0,3$ , POUR DONNÉES NORMALES

A.2.1.  $\mu_X = \mu_Y$

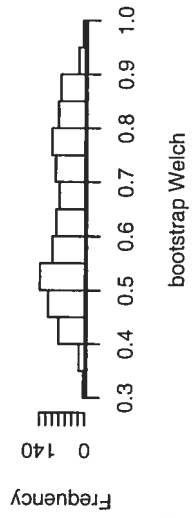
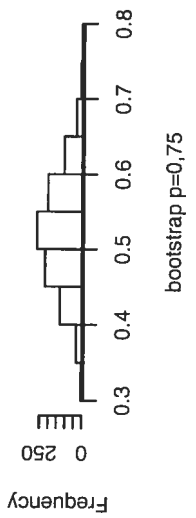
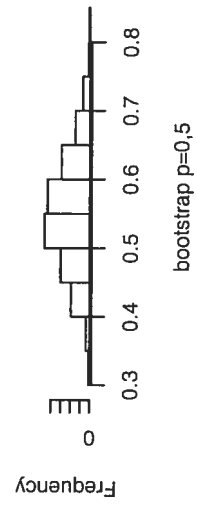
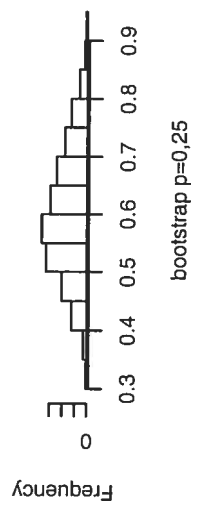
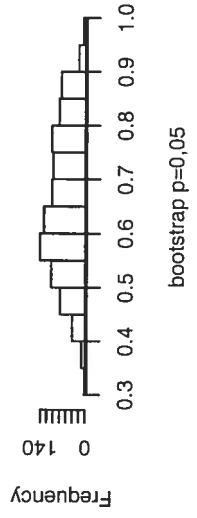
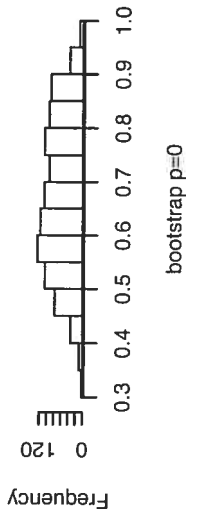
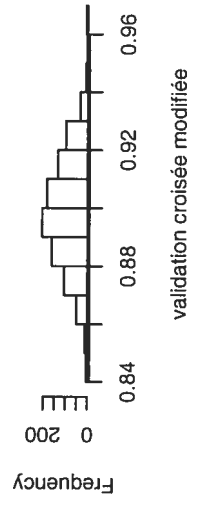
$n = 10$



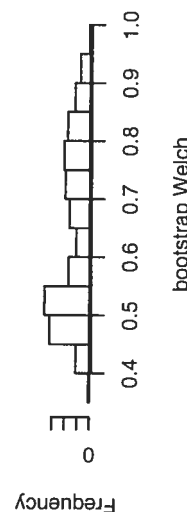
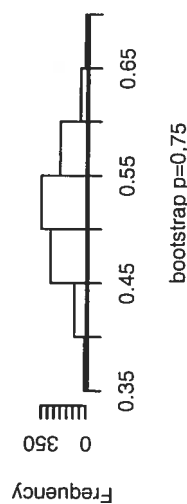
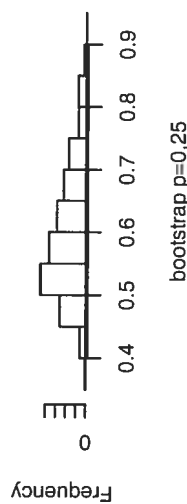
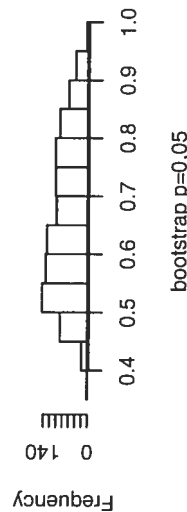
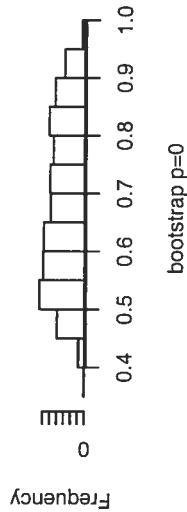
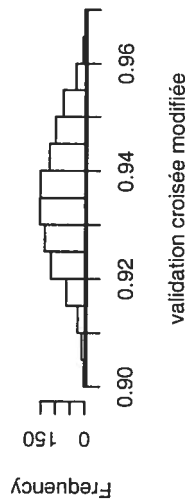
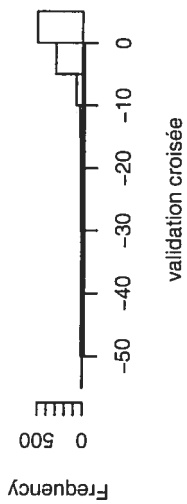
n = 20



n = 50

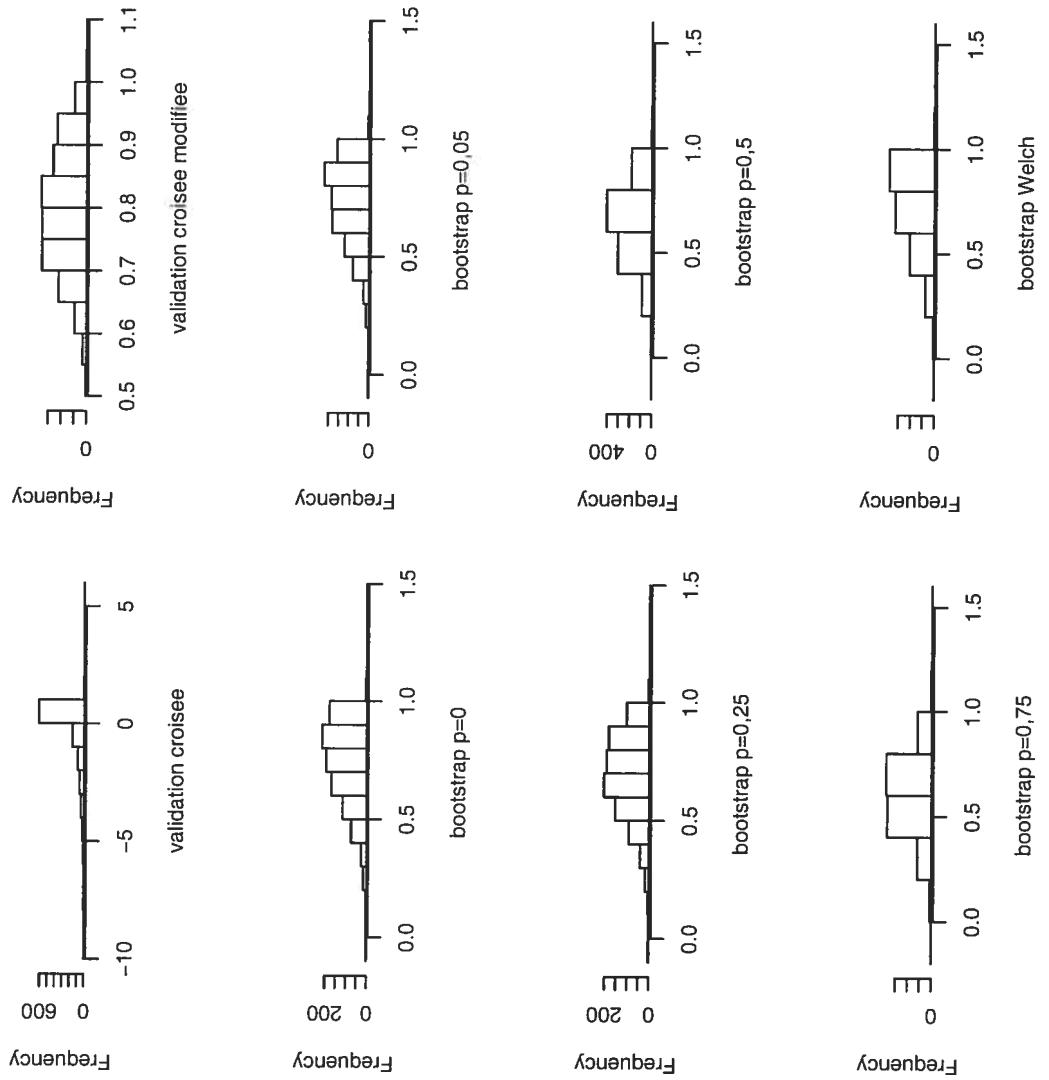


n = 100

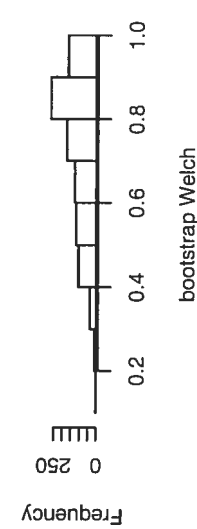
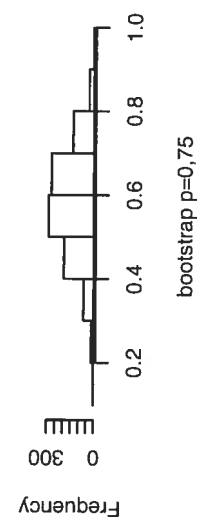
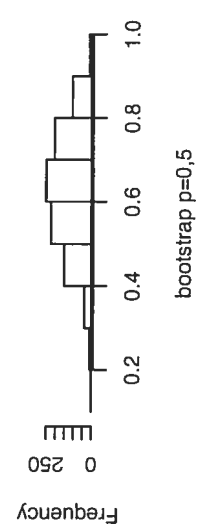
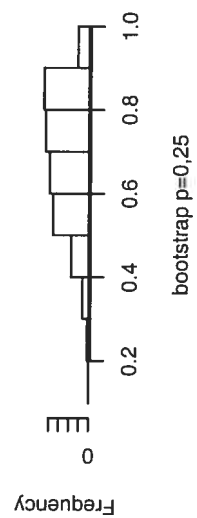
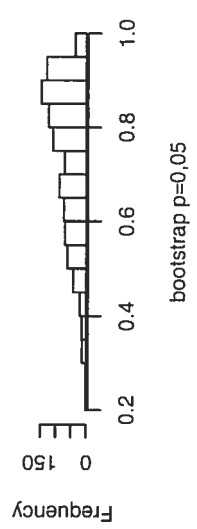
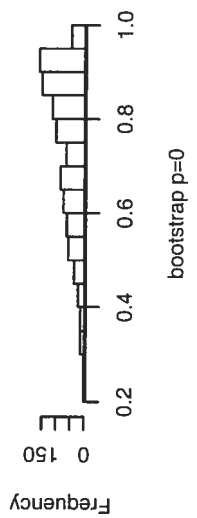
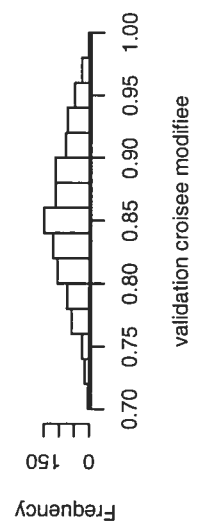
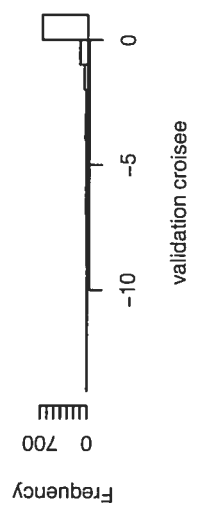


A.2.2.  $|\mu_X - \mu_Y| = 0,3$

n = 10

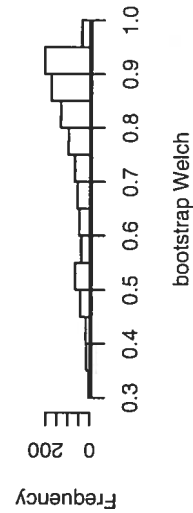
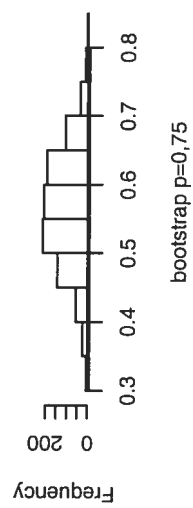
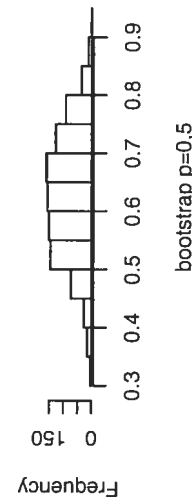
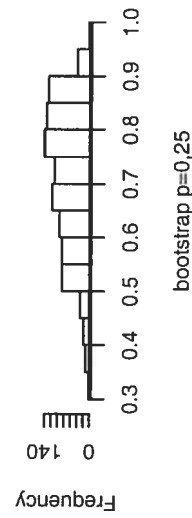
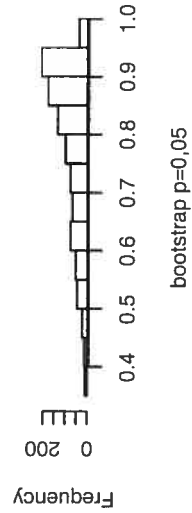
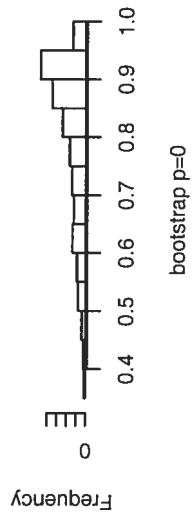
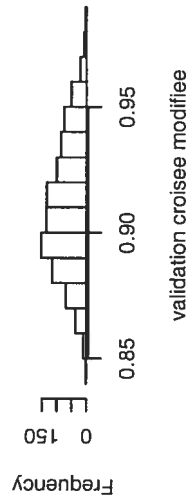
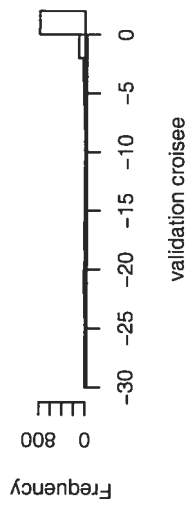


n = 20

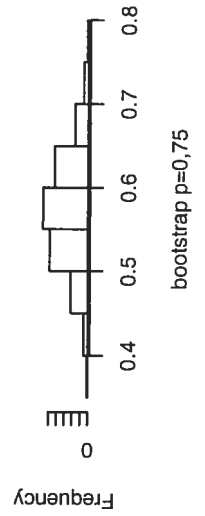
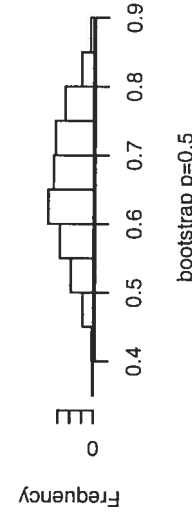
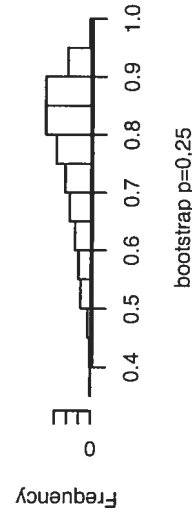
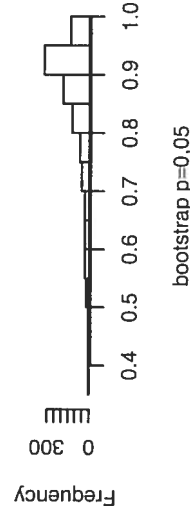
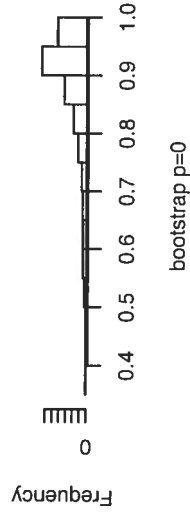
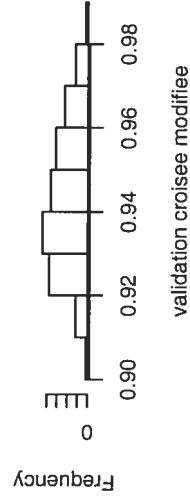
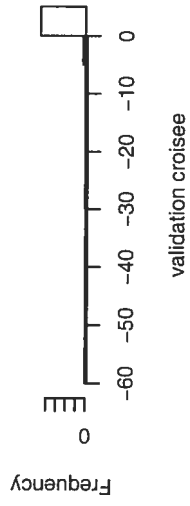




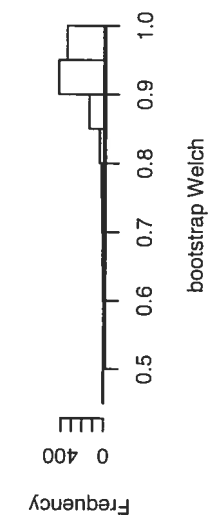
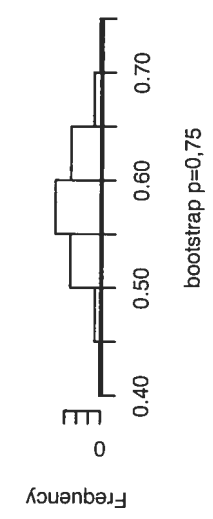
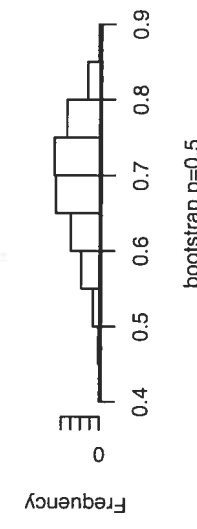
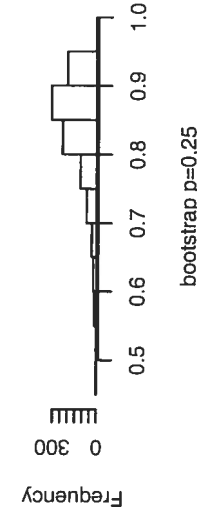
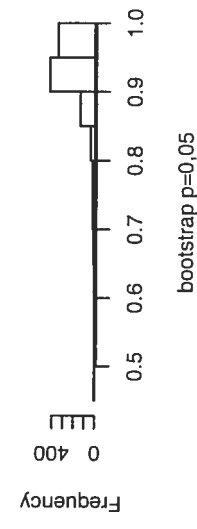
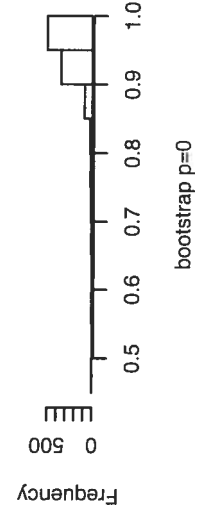
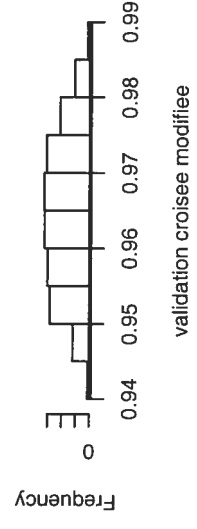
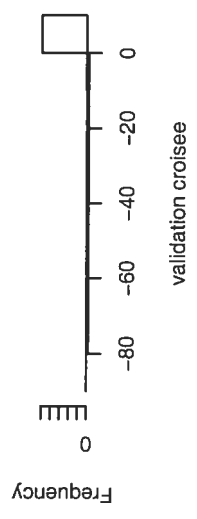
n = 50



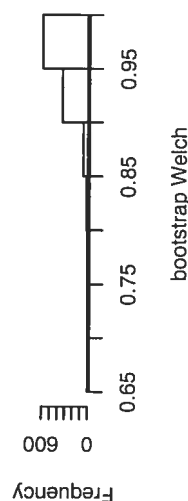
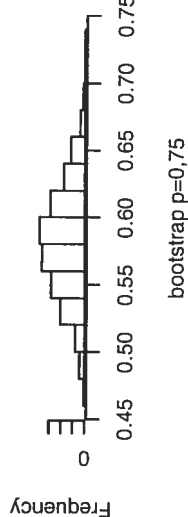
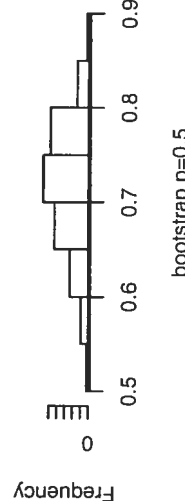
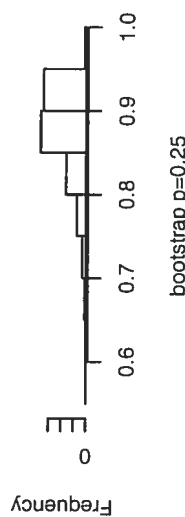
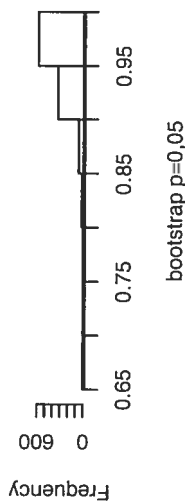
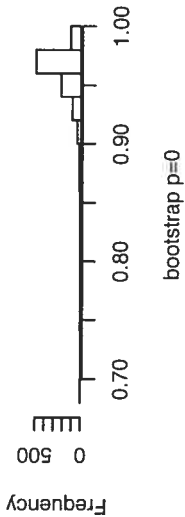
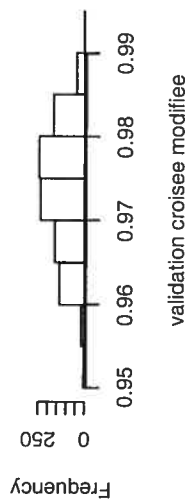
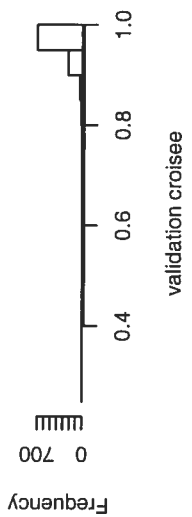
n = 100



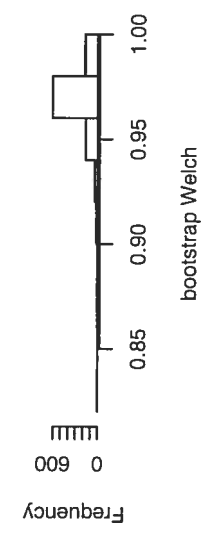
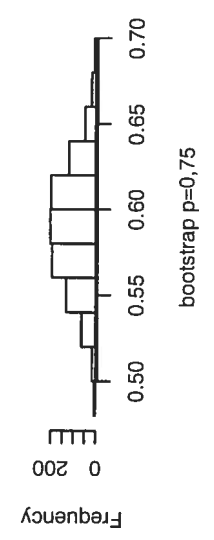
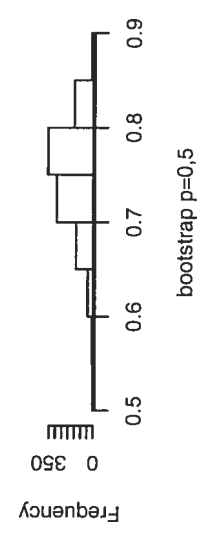
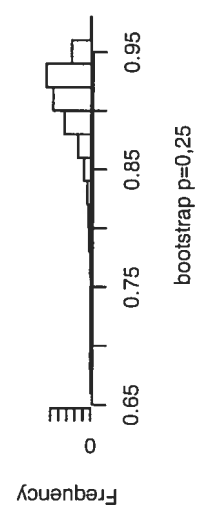
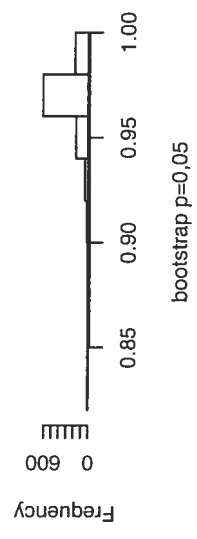
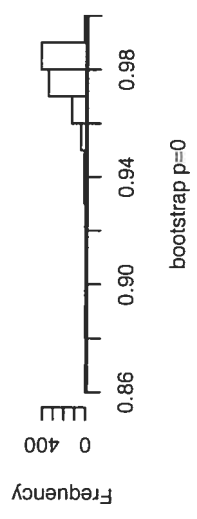
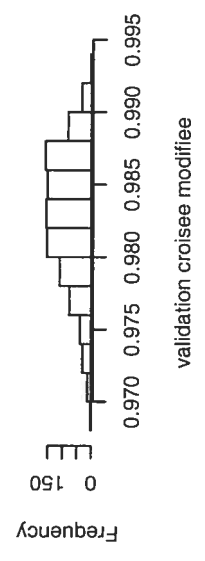
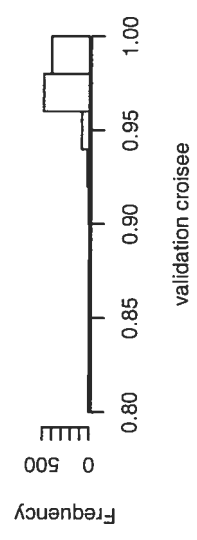
n = 200



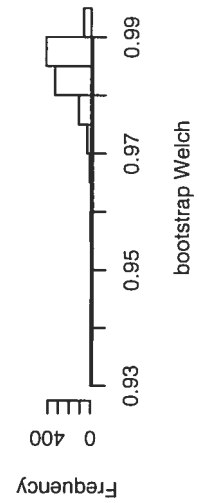
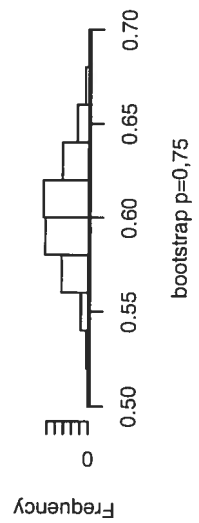
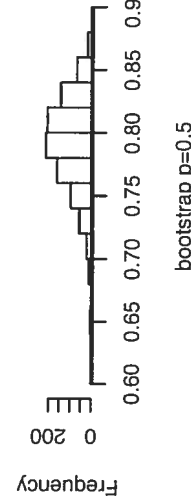
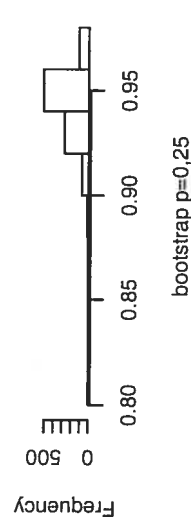
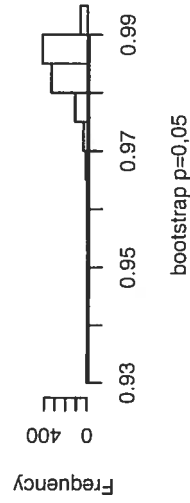
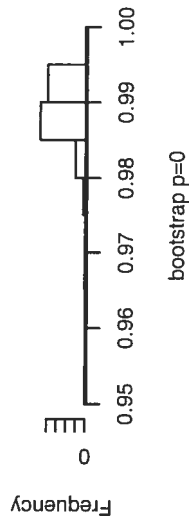
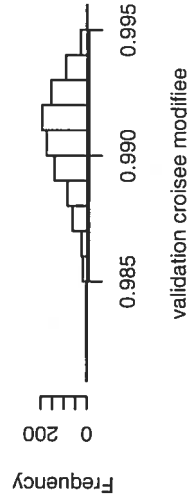
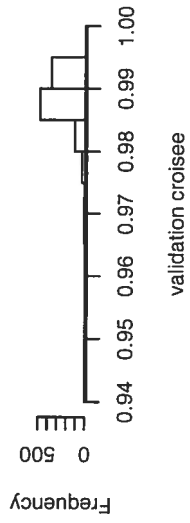
n = 300



n = 500



n = 1000



# Annexe B

---

## SIMULATIONS LOGNORMALES

### B.1. PROGRAMMES

```
#mesure d'ecart moyen, ainsi que sa derivee et deuxieme derivee#
D<-function(lamb,x,y,lx,ly,sx.moins.j,n)
{
D<-mean((x - exp(lamb*(lx-ly)+ly+sx.moins.j/2))^2)
D
}

dD<-function(lamb,x,y,lx,ly,sx.moins.j,n)
{
dD<--2*mean((x - exp(lamb*(lx-ly)+ly+sx.moins.j/2))*exp(lamb*(lx-ly)
+ly+sx.moins.j/2)*(lx-ly))
dD
}

#bisection#
bisection<-function(gauche,droite,x,y,lx,ly,sx.moins.j,n)
{
print("bisection starting")
print(paste("gauche=",gauche,", droite=",droite, sep=""))
while(abs(gauche-droite)>0.000001)
```

```

{
  milieu<-(gauche+droite)/2
  if(dD(gauche,x,y,lx,ly,sx.moins.j,n)
      *dD(milieu,x,y,lx,ly,sx.moins.j,n)<=0)
    droite<-milieu
  else gauche<-milieu
}
lambda<-(gauche+droite)/2
print(c("f at lambda", dD(lambda,x,y,lx,ly,sx.moins.j,n)))
lambda
}

#fonction pour trouver lambda.cv par la resolution de l'equation dD=0#
solution.dD<-function(x,y,lx,ly,sx.moins.j,lambinit,n)
{
  if (abs(dD(lambinit,x,y,lx,ly,sx.moins.j,n)) <= 0.000001)
  {
    val.fn<-dD(lambinit,x,y,lx,ly,sx.moins.j,n)
    lambda.cv<-lambinit
    return<-c("val.fn"=val.fn,"lambda.cv"=lambda.cv)
  }
  else
  {
    diff<-1
    while(sign(dD(lambinit-diff,x,y,lx,ly,sx.moins.j,n))
        ==sign(dD(lambinit+diff,x,y,lx,ly,sx.moins.j,n)))
    {
      diff<-diff+1
    }
    print(c(diff,dD(lambinit-diff,x,y,lx,ly,sx.moins.j,n),
              dD(lambinit+diff,x,y,lx,ly,sx.moins.j,n)))
  }
}

```



```

lambda.cv<-bisection(lambinit-diff,lambinit+diff,x,y,
                    lx,ly,sx.moins.j,n)
val.fn<-dD(lambda.cv,x,y,lx,ly,sx.moins.j,n)
return<-c("val.fn"=val.fn,"lambda.cv"=lambda.cv)
}
return
}
#simulation de base#
pvector<-matrix(seq(0,1,by=0.05),byrow=FALSE)

simln<-function(n, mx, sx, my, sy, nsim)
{
lambda.opt<-1-(sx^2)/(n*(mx-my)^2 + sx^2 + sy^2)

#initialisation des vecteurs#

#matrice lambda.star - lignes pour nsim repetitions,
#colonnes pour valeurs de p#
lambda.star<-matrix(nrowsim,ncol=length(pvector),0)

lambda.cv<-matrix(nrowsim,ncol=1,0)
lambda.cv.tronq<-matrix(nrowsim,ncol=1,0)
lambda.welch<-matrix(nrowsim,ncol=2,0)

p.welch<-matrix(nrowsim,ncol=2,0)
j.welch<-matrix(nrow=1,ncol=2,0)
wle.welch<-matrix(nrowsim,ncol=2,0)
mse.wle.welch<-matrix(nrowsim,ncol=2,0)

mle<-matrix(nrowsim,ncol=1,0)
mse.mle<-matrix(nrowsim,ncol=1,0)

```

```
wle.cv<-matrix(nrowsim,ncol=1,0)
mse.wle.cv<-matrix(nrowsim,ncol=1,0)
wle.cv.tronq<-matrix(nrowsim,ncol=1,0)
mse.wle.cv.tronq<-matrix(nrowsim,ncol=1,0)

wle.star<-matrix(nrowsim,ncol=length(pvector),0)
mse.wle.star<-matrix(nrowsim,ncol=length(pvector),0)

p.value<-matrix(nrowsim,ncol=1,0)

#pour pouvoir reproduire les memes resultats#
set.seed(3)
for (i in 1:nrowsim)
{
  #tirer les deux echantillons#
  X<-rlnorm(n, meanlog=mx, sdlog=sx)
  Y<-rlnorm(n, meanlog=my, sdlog=sy)
  lX<-rep(0,n)
  lY<-rep(0,n)
  sx.moins.j<-rep(0,n)
  for(j in 1:n)
  {
    lX[j]<-mean(log(X[-j]))
    lY[j]<-mean(log(Y[-j]))
    sx.moins.j[j]<-var(log(X[-j]))*(n-1)/n
  }
  logX<-log(X)
  logY<-log(Y)

  #test de welch
```

```

t<-sqrt(n)*(mean(logX)-mean(logY))/sqrt(var(logX) + var(logY))
nu<-(n-1)*(var(logX)+var(logY))^2/(var(logX)^2+var(logY)^2)

if (t>0) p.value[i]<-2*(1-pt(t, nu)) else p.value[i]<-2*pt(t,nu)

#maximum de vraisemblance: estimateur et eqm#
mle[i]<-mean(logX)
mse.mle[i]<-(mle[i]-mx)^2

#calcul des poids#
# lambda bootstrap (exact) pour chaque p#
j<-1
for (p in pvector)
{
  lambda.star[i,j]<-1-((var(logX)-(cov(logX,logY)))
                      *((n-1)/n))/(((n-1)/n)*
                      (var(logX)+var(logY)-2*cov(logX,logY))
                      +n^(1-p)*(mean(logX)-mean(logY))^2)
  j<-j+1
}

# validation croisee#
lambda.cv[i]<-solution.dD(X,Y,lX,lY,sx.moins.j,1,n)[2]

# validation croisee tronquee#
lambda.cv.tronq[i]<-lambda.cv[i]
if (lambda.cv.tronq[i]<0) lambda.cv.tronq[i]<-0
if (lambda.cv.tronq[i]>1) lambda.cv.tronq[i]<-1

#estimateurs et eqm resultants#
for (j in 1:length(pvector))

```

```

{
  wle.star[i,j]<-lambda.star[i,j]*mle[i]
    +(1-lambda.star[i,j])*mean(logY)
  mse.wle.star[i,j]<-(wle.star[i,j]-mx)^2
}

wle.cv[i]<-lambda.cv[i]*mle[i]+(1-lambda.cv[i])*mean(logY)
wle.cv.tronq[i]<-lambda.cv.tronq[i]*mle[i]
  +(1-lambda.cv.tronq[i])*mean(logY)

mse.wle.cv[i]<-(wle.cv[i]-mx)^2
mse.wle.cv.tronq[i]<-(wle.cv.tronq[i]-mx)^2

#estimateur par test de Welch
if (p.value[i]<=0.5) {p.welch[i,]<-c(0.05,0.05); j.welch<-c(2,2)}
else {p.welch[i,]<-c(0.25,0.5); j.welch<-c(6,11)}
lambda.welch[i,]<-c(lambda.star[i,j.welch[1]],
  lambda.star[i,j.welch[2]])
wle.welch[i,]<-c(wle.star[i,j.welch[1]],wle.star[i,j.welch[2]])
mse.wle.welch[i,]<-c(mse.wle.star[i,j.welch[1]],
  mse.wle.star[i,j.welch[2]])
}

#initialiser#
lambda.star.bar1<-matrix(0, nrow=1, ncol=length(pvector))
lambda.star.sd1<-matrix(0, nrow=1, ncol=length(pvector))
wle.star.bar1<-matrix(0, nrow=1, ncol=length(pvector))
wle.star.sd1<-matrix(0, nrow=1, ncol=length(pvector))
mse.wle.star.bar1<-matrix(0, nrow=1, ncol=length(pvector))
mse.wle.star.sd1<-matrix(0, nrow=1, ncol=length(pvector))
ratio.star.mle1<-matrix(0, nrow=1, ncol=length(pvector))

```

```
#calculer les moyennes des colonnes des matrices
#de differentes valeurs de p#
for (j in 1:length(pvector))
{
  lambda.star.bar1[j]<-mean(lambda.star[,j])
  lambda.star.sd1[j]<-sd(lambda.star[,j])
  wle.star.bar1[j]<-mean(wle.star[,j])
  wle.star.sd1[j]<-sd(wle.star[,j])
  mse.wle.star.bar1[j]<-mean(mse.wle.star[,j])
  mse.wle.star.sd1[j]<-sd(mse.wle.star[,j])
  ratio.star.mle1[j]<-mse.wle.star.bar1[j]/mean(mse.mle)
}

#garder les modeles#
model<-c(n,mx,sx,my,sy)
names(model)<-c("n","mx","sx","my","sy")

#garder tous les nsim valeurs des estimateurs#
details<-list(lambda.star=lambda.star,
  lambda.cv=lambda.cv, lambda.cv.tronq=lambda.cv.tronq, mle=mle,
  mse.mle=mse.mle, wle.cv=wle.cv, mse.wle.cv=mse.wle.cv,
  wle.cv.tronq=wle.cv.tronq, mse.wle.cv.tronq=mse.wle.cv.tronq,
  wle.star=wle.star, mse.wle.star=mse.wle.star,
  p.value=p.value,p.welch=p.welch, lambda.welch=lambda.welch,
  wle.welch=wle.welch, mse.wle.welch=mse.wle.welch)

#les resultats pour tous les p#
lambda.star.all.results<-list(model=model, lambda.opt=lambda.opt,
  lambda.star.bar=lambda.star.bar1, lambda.star.sd=lambda.star.sd1,
  wle.star.bar=wle.star.bar1, wle.star.sd=wle.star.sd1,
```

```

mse.wle.star.bar=mse.wle.star.bar1, mse.wle.star.sd=mse.wle.star.sd1,
ratio.star.mle=ratio.star.mle1)

#les resultats pour cv, star.0, star.05, star.25, star.5, star.75#
lambda.star.bar<-matrix(c(lambda.star.bar1[1],lambda.star.bar1[2],
  lambda.star.bar1[6],lambda.star.bar1[11],lambda.star.bar1[16]),
  nrow=1,ncol=5,byrow=FALSE)
lambda.star.sd<-matrix(c(lambda.star.sd1[1],lambda.star.sd1[2],
  lambda.star.sd1[6],lambda.star.sd1[11],lambda.star.sd1[16]),
  nrow=1,ncol=5,byrow=FALSE)
wle.star.bar<-matrix(c(wle.star.bar1[1],wle.star.bar1[2],wle.star.bar1[6],
  wle.star.bar1[11],wle.star.bar1[16]),nrow=1,ncol=5,byrow=FALSE)
wle.star.sd<-matrix(c(wle.star.sd1[1],wle.star.sd1[2],wle.star.sd1[6],
  wle.star.sd1[11],wle.star.sd1[16]),nrow=1,ncol=5,byrow=FALSE)
mse.wle.star.bar<-matrix(c(mse.wle.star.bar1[1],mse.wle.star.bar1[2],
  mse.wle.star.bar1[6],mse.wle.star.bar1[11],mse.wle.star.bar1[16]),
  nrow=1,ncol=5,byrow=FALSE)
mse.wle.star.sd<-matrix(c(mse.wle.star.sd1[1],mse.wle.star.sd1[2],
  mse.wle.star.sd1[6],mse.wle.star.sd1[11],mse.wle.star.sd1[16]),
  nrow=1,ncol=5,byrow=FALSE)
ratio.star.mle<-matrix(c(ratio.star.mle1[1],ratio.star.mle1[2],
  ratio.star.mle1[6],ratio.star.mle1[11],ratio.star.mle1[16]),
  nrow=1,ncol=5,byrow=FALSE)

colnames(lambda.star.bar)<-c("p=0", "p=.05", "p=.25", "p=.5", "p=.75")
colnames(lambda.star.sd)<-c("p=0", "p=.05", "p=.25", "p=.5", "p=.75")
colnames(wle.star.bar)<-c("p=0", "p=.05", "p=.25", "p=.5", "p=.75")
colnames(wle.star.sd)<-c("p=0", "p=.05", "p=.25", "p=.5", "p=.75")
colnames(mse.wle.star.bar)<-c("p=.0", "p=.05", "p=.25", "p=.5", "p=.75")
colnames(mse.wle.star.sd)<-c("p=0", "p=.05", "p=.25", "p=.5", "p=.75")
colnames(ratio.star.mle)<-c("p=0", "p=.05", "p=.25", "p=.5", "p=.75")

```

```

lambda.welch.bar<-matrix(c(mean(lambda.welch[,1]),
  mean(lambda.welch[,2])), nrow=1,ncol=2)
lambda.welch.sd<-matrix(c(sd(lambda.welch[,1]),sd(lambda.welch[,2])),
  nrow=1, ncol=2)
wle.welch.bar<-matrix(c(mean(wle.welch[,1]),mean(wle.welch[,2])),
  nrow=1,ncol=2)
wle.welch.sd<-matrix(c(sd(wle.welch[,1]),sd(wle.welch[,2])),nrow=1,ncol=2)
mse.wle.welch.bar<-matrix(c(mean(mse.wle.welch[,1]),
  mean(mse.wle.welch[,2])), nrow=1,ncol=2)
mse.wle.welch.sd<-matrix(c(sd(mse.wle.welch[,1]),sd(mse.wle.welch[,2])),
  nrow=1, ncol=2)
ratio.welch.mle<-matrix(c(mean(mse.wle.welch[,1])/mean(mse.mle),
  mean(mse.wle.welch[,2])/mean(mse.mle)),nrow=1,ncol=2)
colnames(lambda.welch.bar)<-c("p.alt=0.25","p.alt=0.5")
colnames(lambda.welch.sd)<-c("p.alt=0.25","p.alt=0.5")
colnames(wle.welch.bar)<-c("p.alt=0.25","p.alt=0.5")
colnames(wle.welch.sd)<-c("p.alt=0.25","p.alt=0.5")
colnames(mse.wle.welch.bar)<-c("p.alt=0.25","p.alt=0.5")
colnames(mse.wle.welch.sd)<-c("p.alt=0.25","p.alt=0.5")
colnames(ratio.welch.mle)<-c("p.alt=0.25","p.alt=0.5")

results<-list(model=model, lambda.opt=lambda.opt,
  ic.width0=2*sqrt(var(lambda.star[,trunc(length(pvector)/2)])/nsim),
  lambda.star.bar=lambda.star.bar, lambda.star.sd=lambda.star.sd,
  lambda.cv.bar=mean(lambda.cv), lambda.cv.sd=sd(lambda.cv),
  lambda.cv.tronq.bar=mean(lambda.cv.tronq),
  lambda.cv.tronq.sd=sd(lambda.cv.tronq),
  mle.bar=mean(mle), mle.sd=sd(mle), mse.mle.bar=mean(mse.mle),
  mse.mle.sd=sd(mse.mle),wle.cv.bar=mean(wle.cv), wle.cv.sd=sd(wle.cv),
  mse.wle.cv.bar=mean(mse.wle.cv), mse.wle.cv.sd=sd(mse.wle.cv),
  wle.cv.tronq.bar=mean(wle.cv.tronq), wle.cv.tronq.sd=sd(wle.cv.tronq),

```

```

mse.wle.cv.tronq.bar=mean(mse.wle.cv.tronq),
mse.wle.cv.tronq.sd=sd(mse.wle.cv.tronq),
wle.star.bar=wle.star.bar,wle.star.sd=wle.star.sd,
mse.wle.star.bar=mse.wle.star.bar,mse.wle.star.sd=mse.wle.star.sd,
ratio.cv.mle=(mean(mse.wle.cv)/mean(mse.mle)),
ratio.cv.tronq.mle=(mean(mse.wle.cv.tronq)/mean(mse.mle)),
ratio.star.mle=ratio.star.mle,welch.p.value=mean(p.value),
sd.welch=sd(p.value), lambda.welch.bar=lambda.welch.bar,
lambda.welch.sd=lambda.welch.sd, wle.welch.bar=wle.welch.bar,
wle.welch.sd=wle.welch.sd, mse.wle.welch.bar=mse.wle.welch.bar,
mse.wle.welch.sd=mse.wle.welch.sd, ratio.welch.mle=ratio.welch.mle)

res<-list(results=results,lambda.star.all=lambda.star.all.results,
  details=details)
res
}

sim1<-function(mx, sx, my, sy, nsim, nvec, resuname)
{
  res<-vector("list",length(nvec))
  resu<-vector("list",4)
  names(resu)<-c("resultats","tous.p","details","meilleur.p")
  resu[[4]]<-matrix(0,nrow=length(nvec),ncol=3)

  #"a" est un indice qui va de 1 a length(nvec)#
  a<-1
  for (n in nvec)
  {
    #faire la simulation de base#
    res[[a]]<-simln(n, mx, sx, my, sy, nsim)
  }
}

```



```

#mettre les resultats dans une liste output
#qui a la structure desiree#
#resu[[1]] liste des resultats de chaque a,
#resu[[3]] liste des details de chaque a#
resu[[1]][[a]]<-res[[a]][[1]]
names(resu[[1]])[a]<-paste("iter",a,sep="")
resu[[2]][[a]]<-res[[a]][[2]]
names(resu[[2]])[a]<-paste("iter",a,sep="")
resu[[3]][[a]]<-res[[a]][[3]]
names(resu[[3]])[a]<-paste("iter",a,sep="")

#faire et sauvegarder le graphique des ratio.star.mle#
par(mfrow=c(1,1))
mod<-res[[a]][[1]]$model
postscript(paste("/home/charlebj/memoire/final/pvector.n=",n,".",
  resuname,".ps",sep=""))
plot(as.vector(res[[a]][[2]]$ratio.star.mle),pvector,
  xlab=paste("iter",a,"$ratio.star.mle",sep=""))
title( paste("\nModele: n = ", mod[1], ", mx = ",mod[2],", sx = ",mod[3],",
  my = ",mod[4],", sy = ",mod[5], sep=""), outer=TRUE, )
dev.off()

#garder la meilleure valeur de p#
resu[[4]][a,]<-c(n,min(as.vector(res[[a]][[2]]$ratio.star.mle)),
  pvector[which.min(as.vector(res[[a]][[2]]$ratio.star.mle))])
colnames(resu[[4]])<-c("n","min.ratio.star","min.p")

a<-a+1
}
resu
}

```

```

#fonction pour creer des histogrammes des poids#
#resu=resultat de la fonction precedente,
#et resuname=son nom entre guillemets#
histofunct<-function(resu, resuname)
{
  #faire des histogrammes pour chaque n dans nvec#
  ngraphs<-length(resu[[2]])
  for(a in 1:ngraphs)
  {
    #mettre les histogrammes pour 8 poids (d'un meme modele)
    #sur une meme page#
    postscript(paste("/home/charlebj/memoire/final/histo.",resuname,
      ".",a,".ps",sep=""))
    par(mfrow=c(4,2))
    mod<-resu$resultats[[a]]$model
    hist(resu$details[[a]]$lambda.cv, xlab="lambda.cv", main=NULL)
    hist(resu$details[[a]]$lambda.cv.tronq,
      breaks=c(0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1),
      xlab="lambda.cv.tronq", main=NULL)
    hist(as.matrix(resu$details[[a]]$lambda.star)[,1],
      xlab="lambda.star(0)", main=NULL)
    hist(as.matrix(resu$details[[a]]$lambda.star)[,2],
      xlab="lambda.star(.05)", main=NULL)
    hist(as.matrix(resu$details[[a]]$lambda.star)[,6],
      xlab="lambda.star(.25)", main=NULL)
    hist(as.matrix(resu$details[[a]]$lambda.star)[,11],
      xlab="lambda.star(.5)", main=NULL)
    hist(as.matrix(resu$details[[a]]$lambda.star)[,16],
      xlab="lambda.star(.75)", main=NULL)
    hist(resu$details[[a]]$lambda.welch[,2],

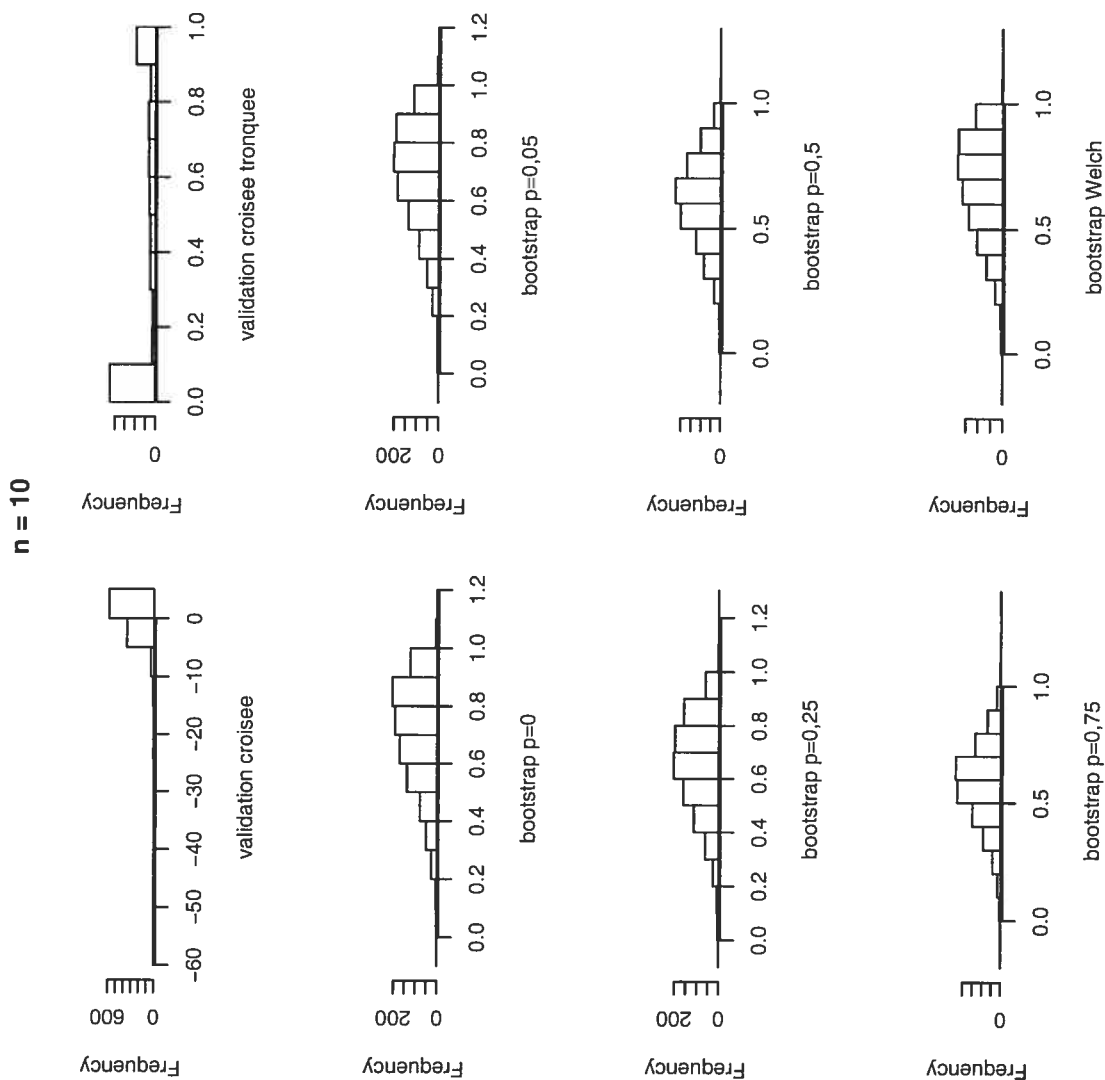
```

```
      xlab="lambda.welch(0.5)", main=NULL)

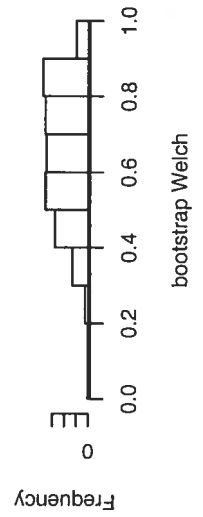
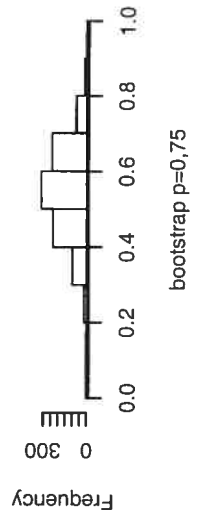
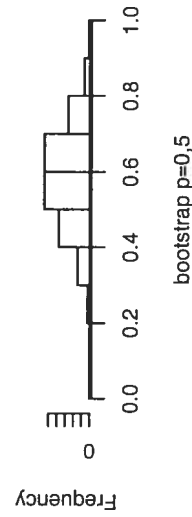
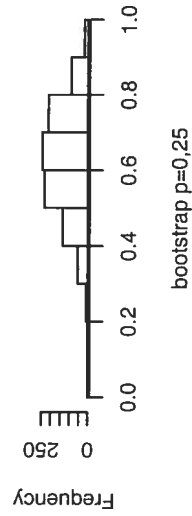
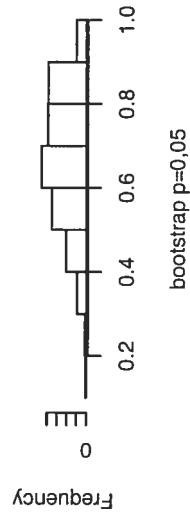
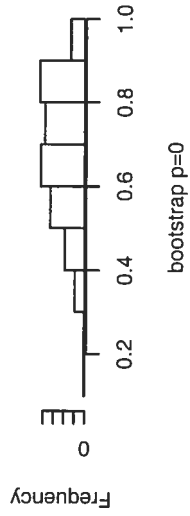
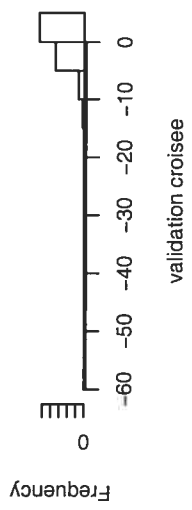
      title(paste("\nModele: n = ", mod[1], ", mx = ",mod[2],", sx = ",mod[3],",
        my = ",mod[4],", sy = ",mod[5], sep=""), outer=TRUE, )
      #sauvegarder chaque page de 8 histogrammes
      #lorsqu'elle est complete#
      dev.off()
    }
  }
```

B.2. HISTOGRAMMES DES HUIT POIDS  $\hat{\lambda}$  EXAMINÉS, POUR PLUSIEURS VALEURS DE  $n$  ET DANS LES DEUX CAS  $\mu_X = \mu_Y$  ET  $|\mu_X - \mu_Y| = 0,3$ , POUR DONNÉES LOGNORMALES

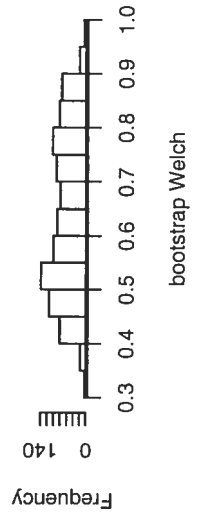
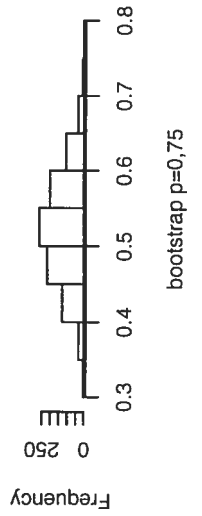
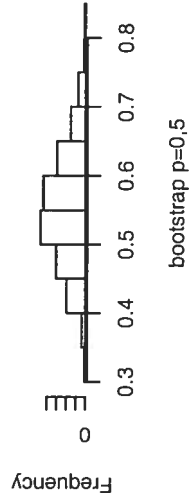
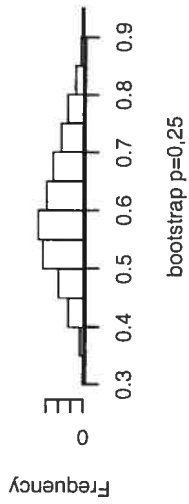
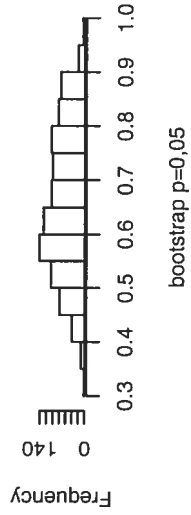
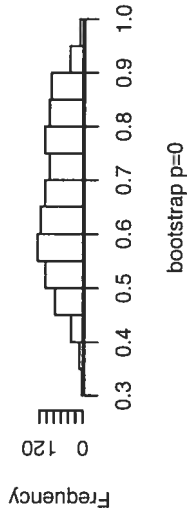
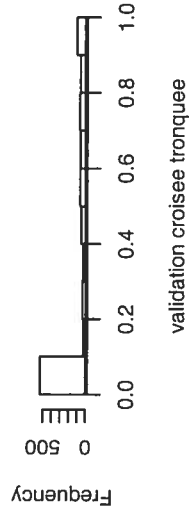
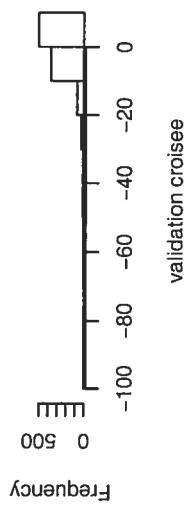
B.2.1.  $\mu_X = \mu_Y$



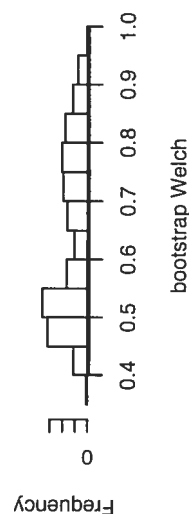
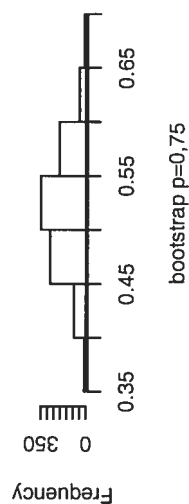
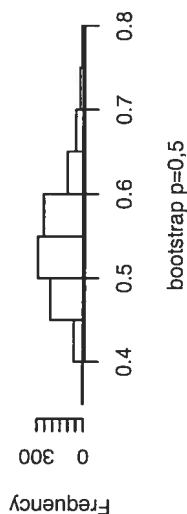
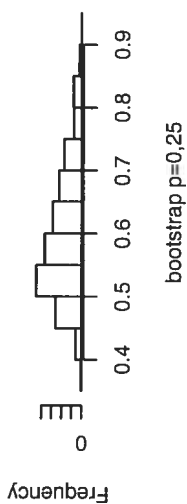
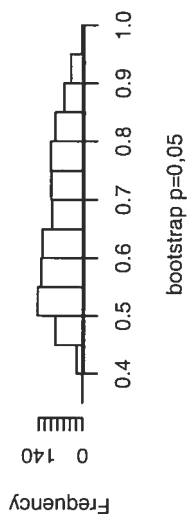
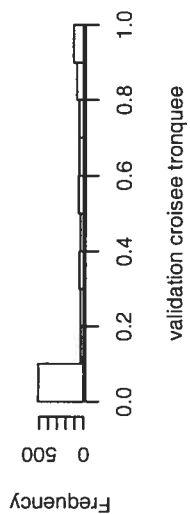
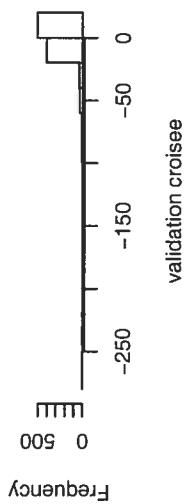
n = 20



n = 50

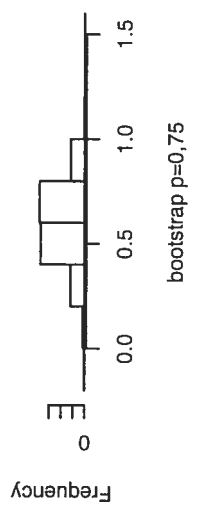
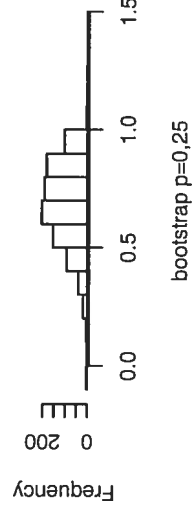
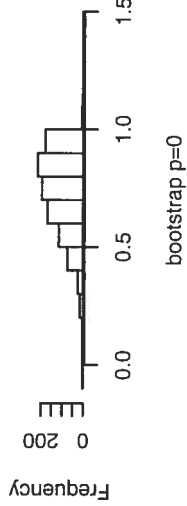
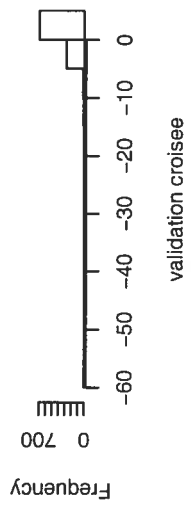
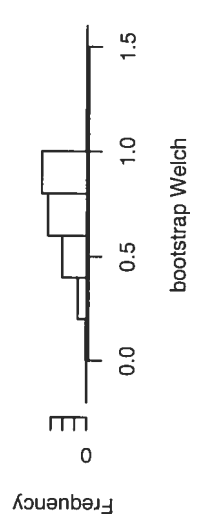
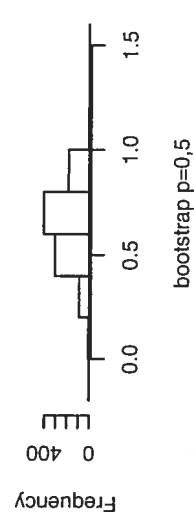
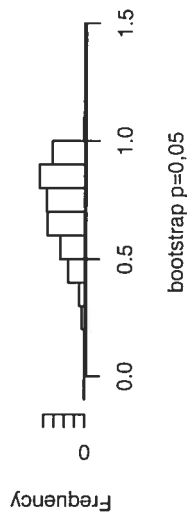
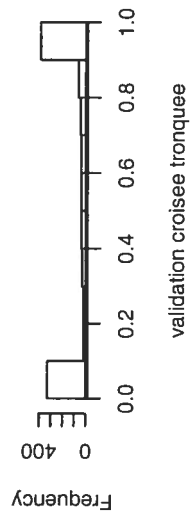


n = 100



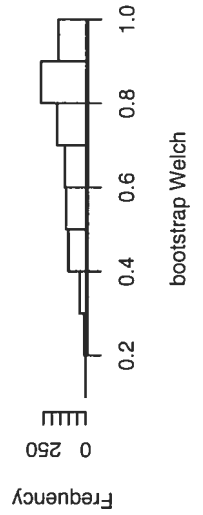
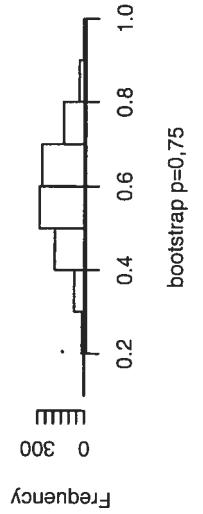
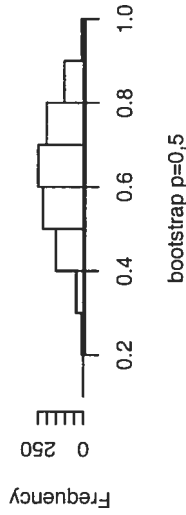
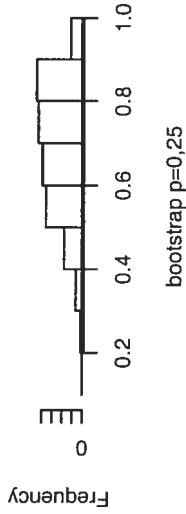
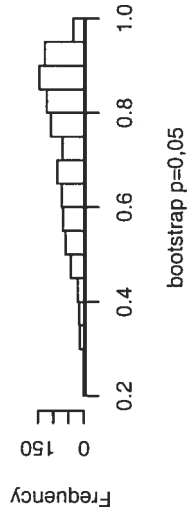
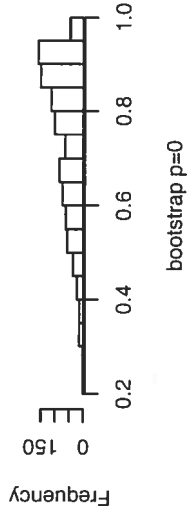
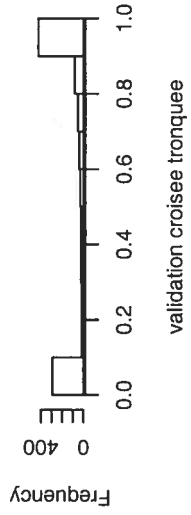
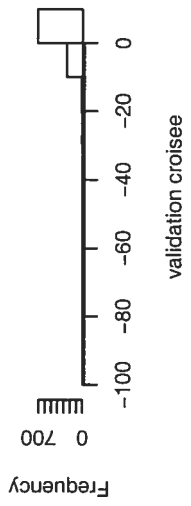
B.2.2.  $|\mu_X - \mu_Y| = 0,3$

n = 10

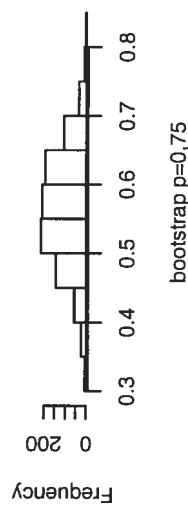
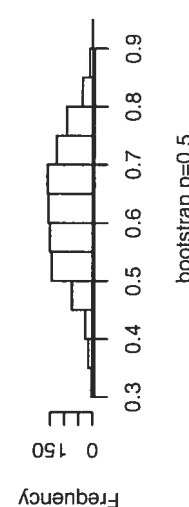
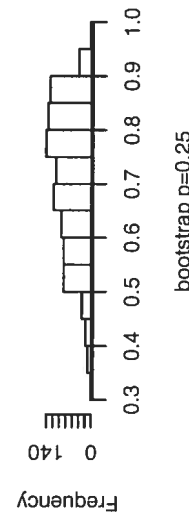
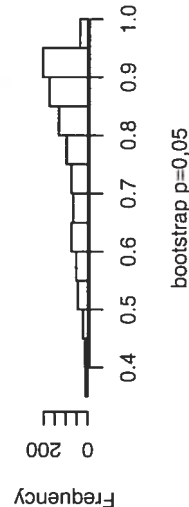
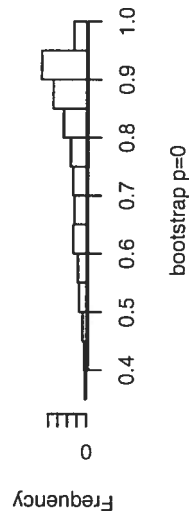
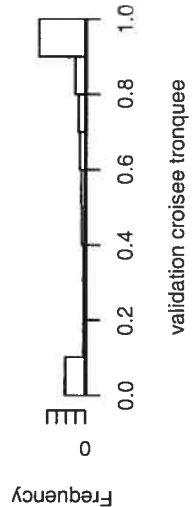
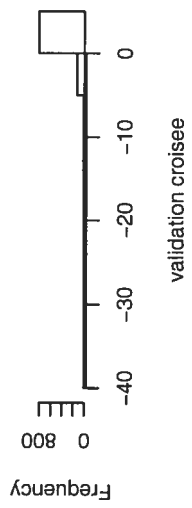




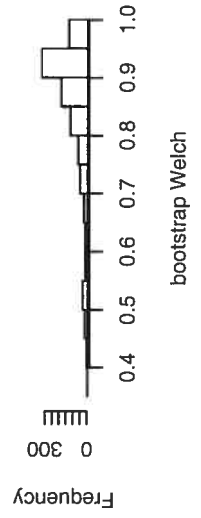
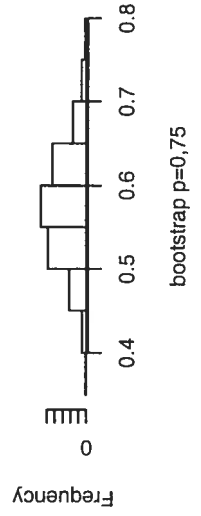
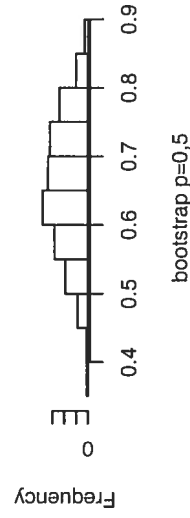
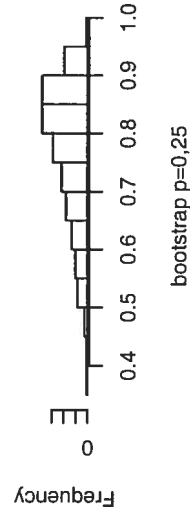
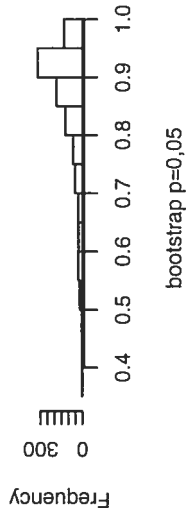
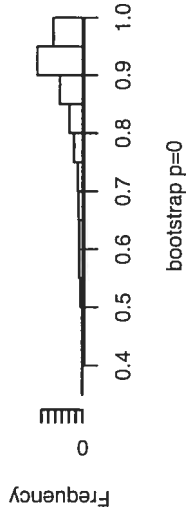
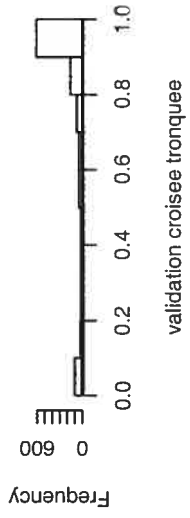
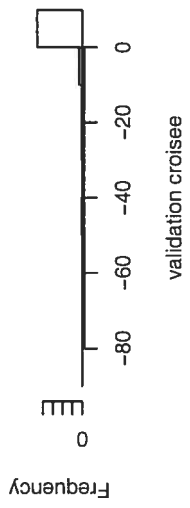
n = 20



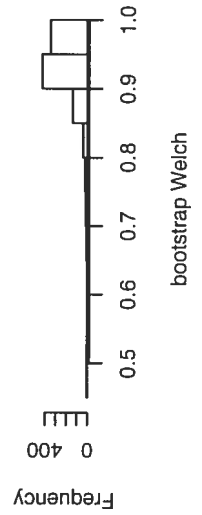
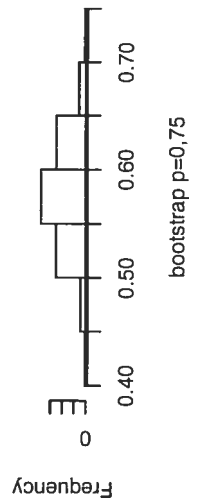
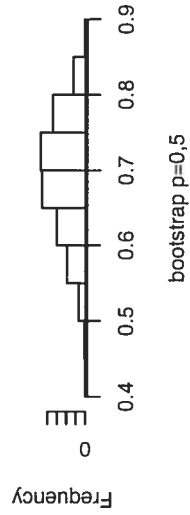
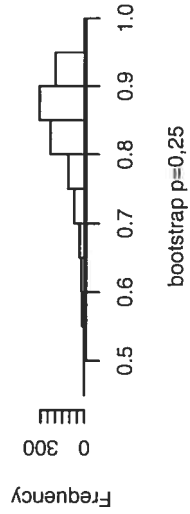
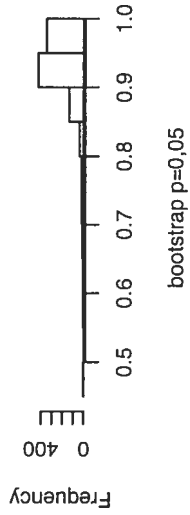
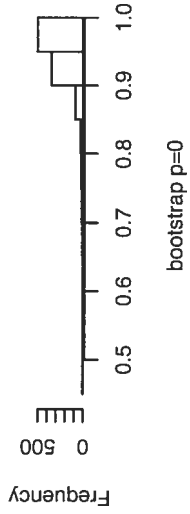
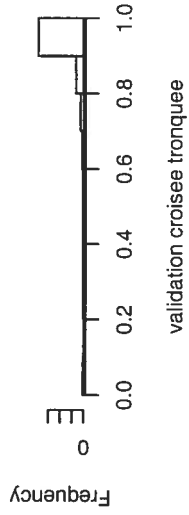
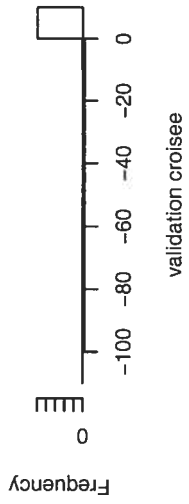
n = 50



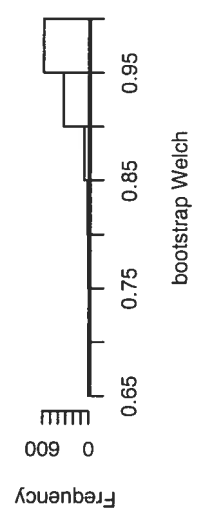
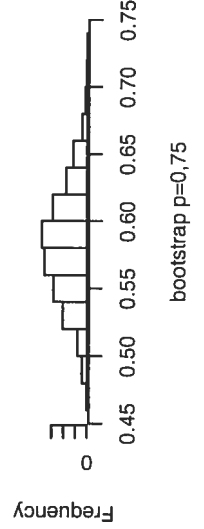
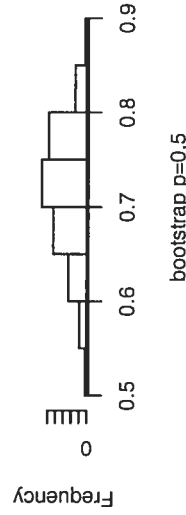
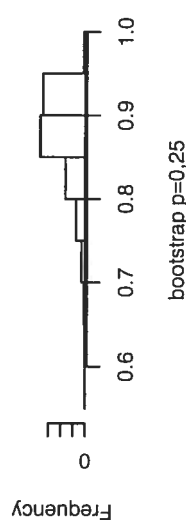
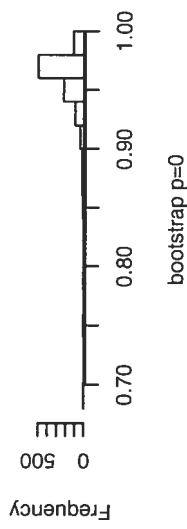
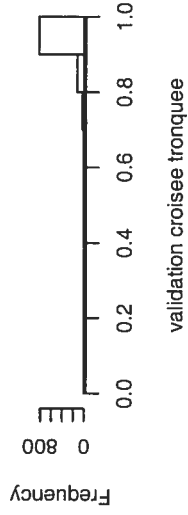
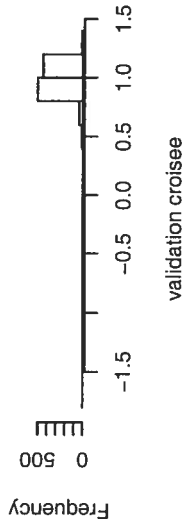
n = 100



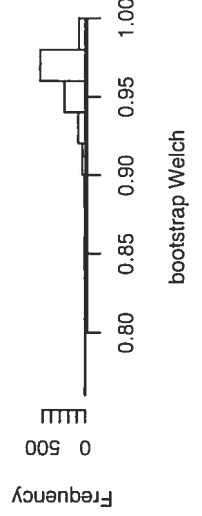
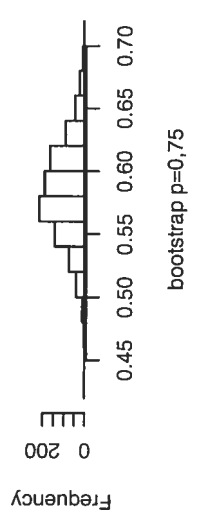
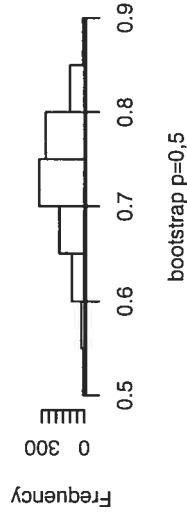
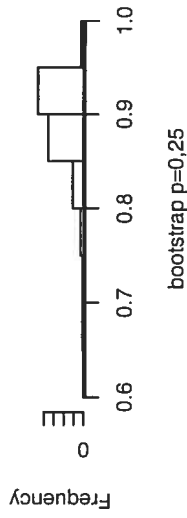
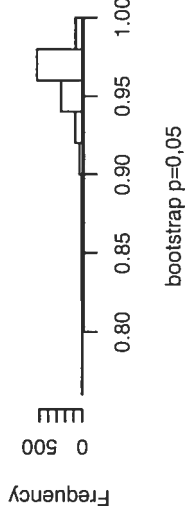
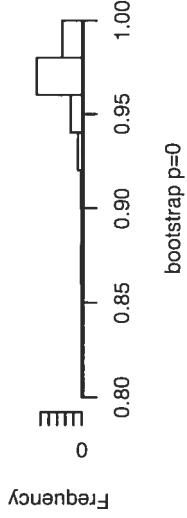
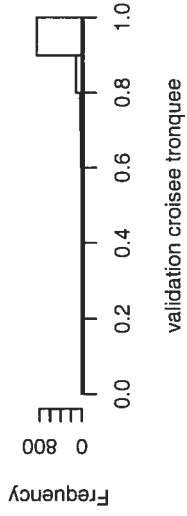
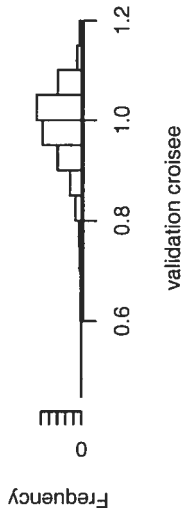
n = 200



n = 300



n = 400



n = 500

