Réseaux de neurones : évaluation et sélection de modèle

Fabrice Rossi

http://apiacoa.org/contact.html.

Université Paris-IX Dauphine

Plan du cours “Évaluation et sélection de modèle”

1. Le problème
2. La validation
3. méthodes de ré-échantillonnage :
   (a) la validation croisée
   (b) le *Bootstrap*
4. méthodes “statistiques” :
   (a) contrôle de complexité
   (b) dimension de Vapnik-Chervonenkis

Évaluation et sélection de modèle

Comment évaluer les performances d'un modèle ? Problème :

- On connaît les performances du modèle sur l'ensemble de données utilisé pour le construire (l'ensemble d'apprentissage)
- On cherche les performances sur de nouvelles données :
  - pour prédire le comportement du modèle
  - pour pouvoir le comparer à d'autres modèles
  - en particulier pour pouvoir choisir les hyper-paramètres :
    - nombre de neurones
    - nature des neurones (par exemple position des centres des RBF)
    - taux de régularisation
    - etc.

Formalisation

On dispose

- d'un ensemble Θ d'hyper-paramètres
- d'un ensemble de classes de modèles indexés par Θ, les $(\mathcal{M}_\theta)_{\theta\in\Theta}$
- d'une mesure d'erreur, $\mathcal{E}$ qui à un modèle f et des données $\mathcal{D}$ associe $\mathcal{E}(f,\mathcal{D})$, l'erreur commise par f en tant que modèle de $\mathcal{D}$

Buts :

- Étant donné un modèle f (obtenu d'une façon à déterminer), construire un estimateur de $\mathcal{E}(f,\mathcal{D})$ pour des données $\mathcal{D}$ "semblables" à celles utilisées pour construire f : **évaluation**
- Trouver, à partir de $\mathcal{D}$, $\theta \in \Theta$ tel que le meilleur modèle dans $\mathcal{M}_\theta$ soit le meilleur modèle des données $\mathcal{D}$ dans l'ensemble des modèles indexés par Θ : **sélection de modèle**

Remarques

- par données “semblables” on entend données de même distribution que les données d’apprentissage
- le choix de f dans $\mathcal{M}_\theta$ (pour θ fixé) obéit à un algorithme fixe spécifique à $\mathcal{M}_\theta$. Par exemple :
  - si θ correspond au nombre de neurones pour un modèle pseudo-linéaire basé sur des B-splines, le choix de f se fait au sens des moindres carrés
  - si θ correspond au nombre de neurones et à un paramètre de régularisation, le choix de f se fait au sens des moindres carrés pénalisés par le terme de régularisation
  - etc.
- l’idée est donc de choisir θ de sorte que le choix naturel de f dans $\mathcal{M}_\theta$ donne de bonnes performances

Approches possibles

La **mauvaise idée** : estimer les performances grâce aux performances sur l'ensemble d'apprentissage.

- ça ne fonctionne pas : cf les exemples des cours précédents
- estimateur biaisé : les performances sur l'ensemble d'apprentissage sont **toujours** meilleures que les performances réelles

Quelques méthodes qui fonctionnent :

- Découpage des données (validation)
- Validation croisée (et *leave-one-out*)
- Ré-échantillonnage (*bootstrap*)
- Contrôle de complexité
- Dimension de Vapnik-Chervonenkis

Découpage des données (la validation)

Si on a beaucoup de données, on coupe l'ensemble en deux, **apprentissage** et **test** :

- on utilise les données de l'ensemble d'apprentissage pour estimer les paramètres du modèle
- on utilise les données de l'ensemble de test pour évaluer la qualité du modèle optimal (estimateur non biaisé + loi des grands nombres)

Pour donner un intervalle de confiance, on utilise des inégalités de concentration. Par exemple Hoeffding :

$$P\left(\left|\frac{1}{N}\sum_{i=1}^{N} U^i - E(U)\right| \geq \epsilon\right) \leq 2e^{-N\epsilon^2}$$

Application à la sélection de modèle

L'application est immédiate :

- pour chaque valeur de l'hyper-paramètre θ, on détermine f_θ, le meilleur modèle de $\mathcal{M}_\theta$ grâce aux données d'apprentissage
- on évalue les performances de f_θ grâce aux données de test
- le meilleur θ est celui dont le f_θ donne les meilleures performances

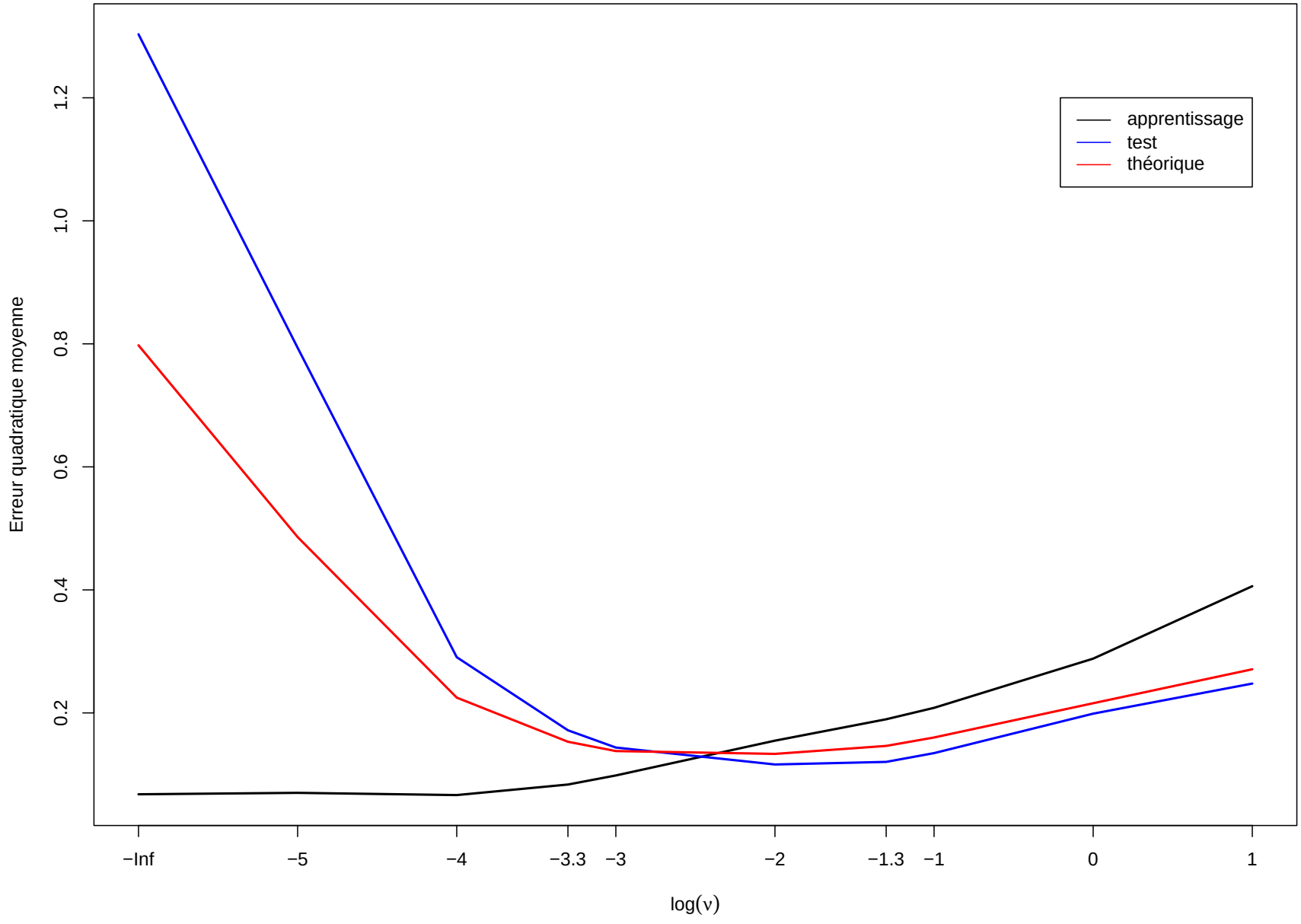
Attention : les performances de f_θ évaluées grâce aux données de test ne constituent pas un bon estimateur des performances de f_θ sur de nouvelles données !

Exemple

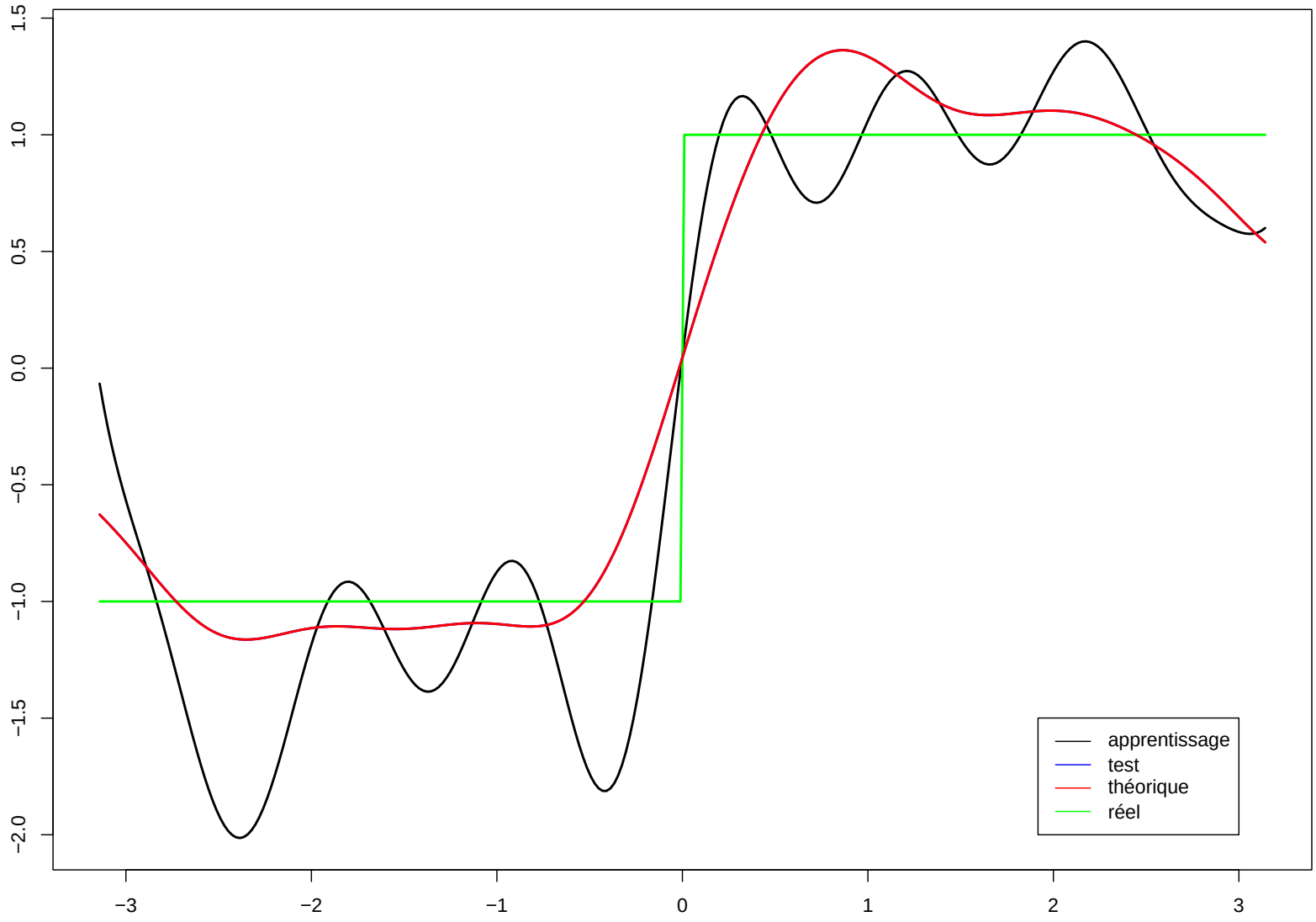
On reprend l'exemple du créneau :

- ensemble d'apprentissage : 40 exemples
- ensemble de test : 40 exemples
- on trace l'évolution des erreurs en fonction du paramètre de régularisation :
  - erreur sur l'ensemble d'apprentissage
  - erreur sur l'ensemble de test
  - erreur par rapport au modèle réel (à laquelle on ajoute la variance du bruit)

Erreur en fonction de ν



Sélection de modèle



Erreur quadratique moyenne réelle $\simeq 0.071$

Exemple d'intervalle de confiance

Pour le créneau, on a $N = 40$. Pour obtenir une confiance de 95% dans les valeurs observées, on doit accepter une erreur de

$$\epsilon \geq \sqrt{\frac{-\ln 0.025}{N}}$$

soit $\epsilon \geq 0.3$. Or, le créneau régularisé à 0.001 donne une erreur quadratique moyenne d'environ 0.15 !

Pour faire 10% d'erreur dans l'estimation de cette erreur, il faudrait 16400 exemples !

Il existe de meilleures bornes, mais rien d'extraordinaire.

Critique de la validation

Points positifs :

- facile à mettre en œuvre
- temps de calcul réduit

Points négatifs :

- nécessite beaucoup de données :
  - deux ensembles distincts pour l'évaluation d'un modèle
  - trois ensembles distincts pour la sélection et l'évaluation d'un modèle (apprentissage, sélection puis test)
- sensible au découpage
- réduit drastiquement les données disponibles pour la construction du modèle : mauvaise estimation des paramètres

Validation croisée

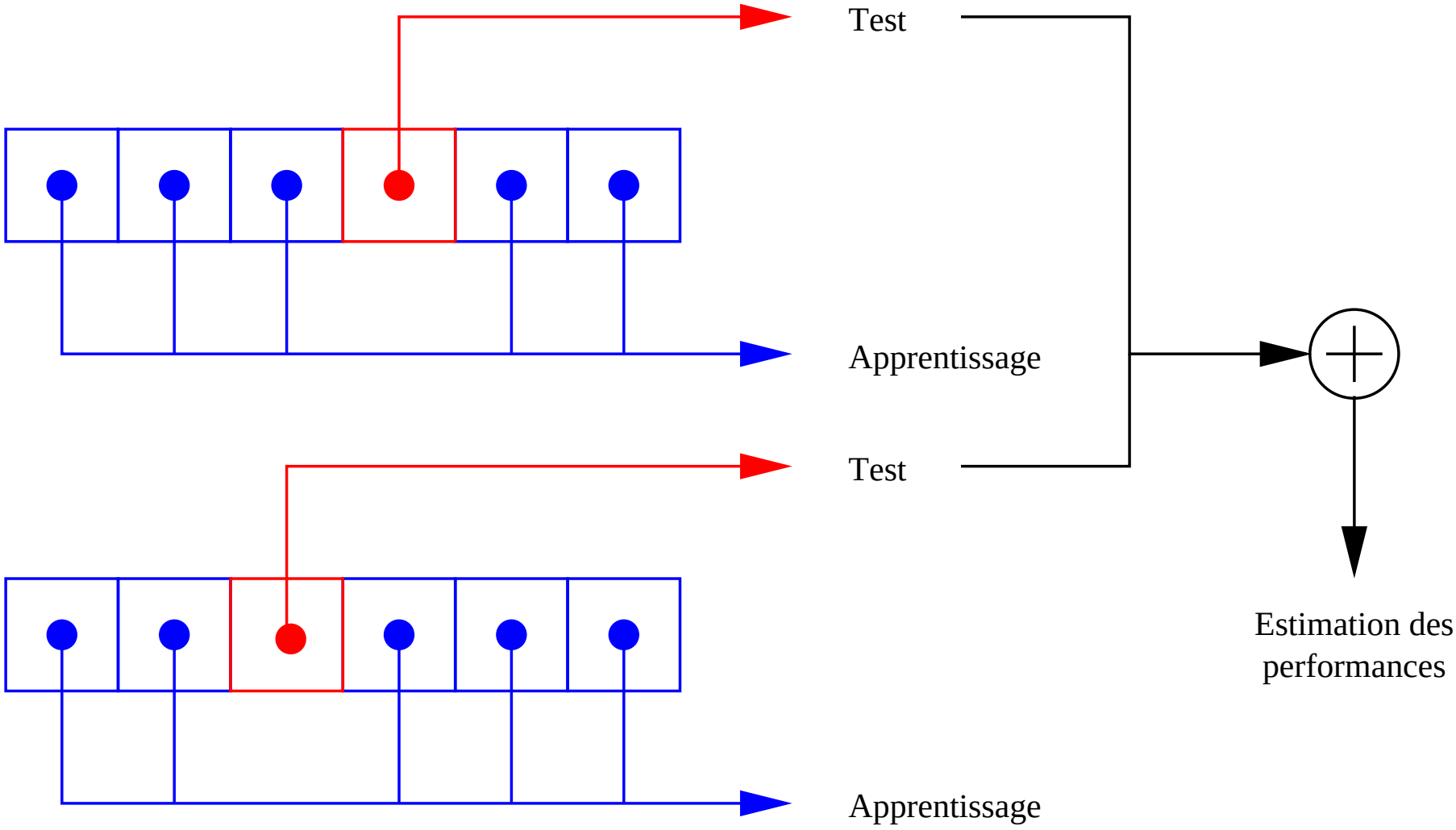
Quand on a peu de données, on ne peut pas découper l'ensemble. On introduit alors du hasard artificiellement en engendrant de nouveaux ensembles d'exemples à partir des données d'origine (ré-échantillonnage).

Validation croisée :

1. on coupe les données en n sous-ensembles $D_1, \ldots, D_n$
2. pour tout i :
   (a) on estime les paramètres du modèle sur l'union des D_j avec $j \neq i$
   (b) on évalue le modèle obtenu sur D_i
3. on somme les évaluations pour obtenir une évaluation globale

Dans le cas limite où $n = N$, on parle de *leave-one-out*.

Validation croisée (2)



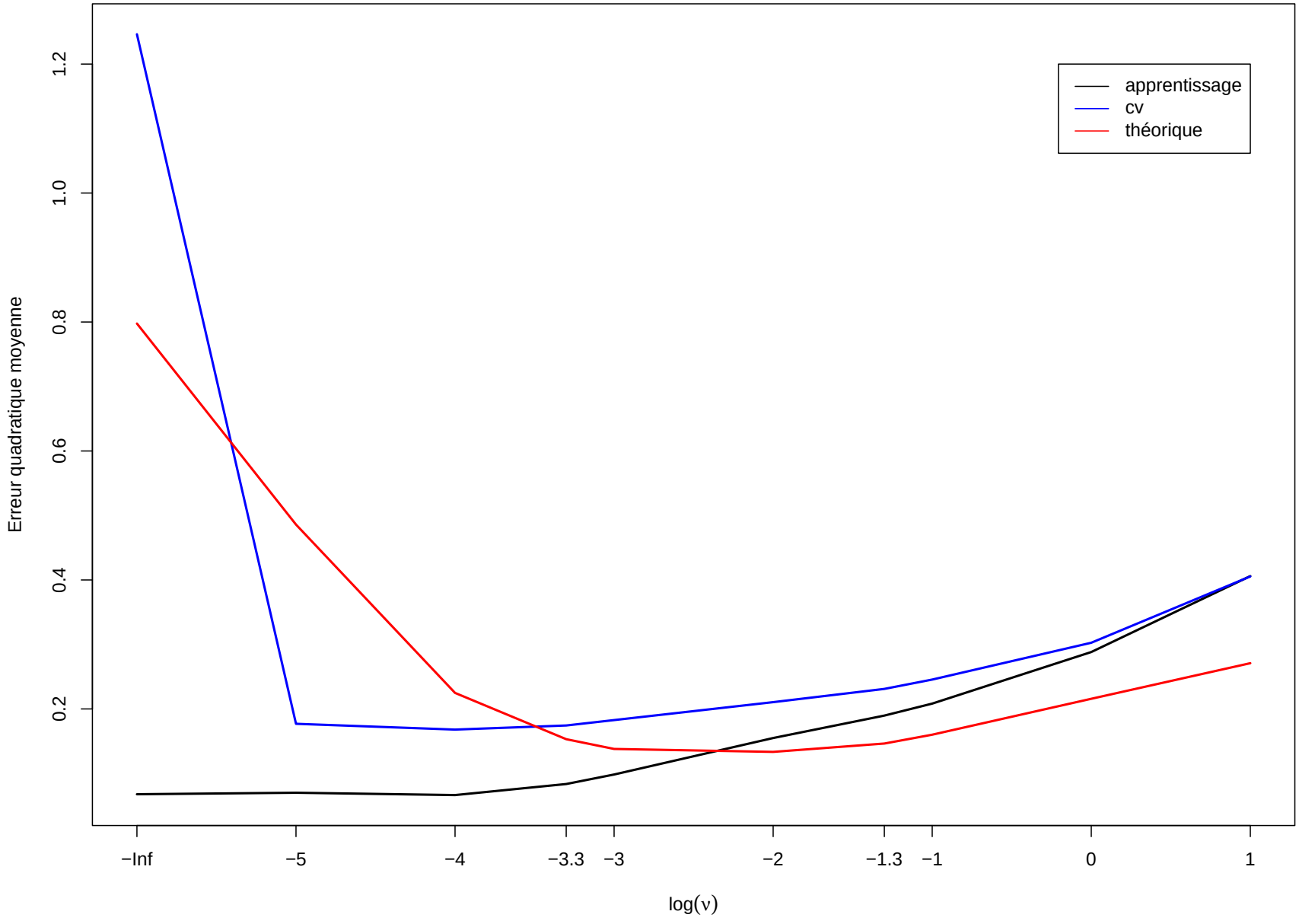
Application à la sélection de modèle

On procède de la façon suivante :

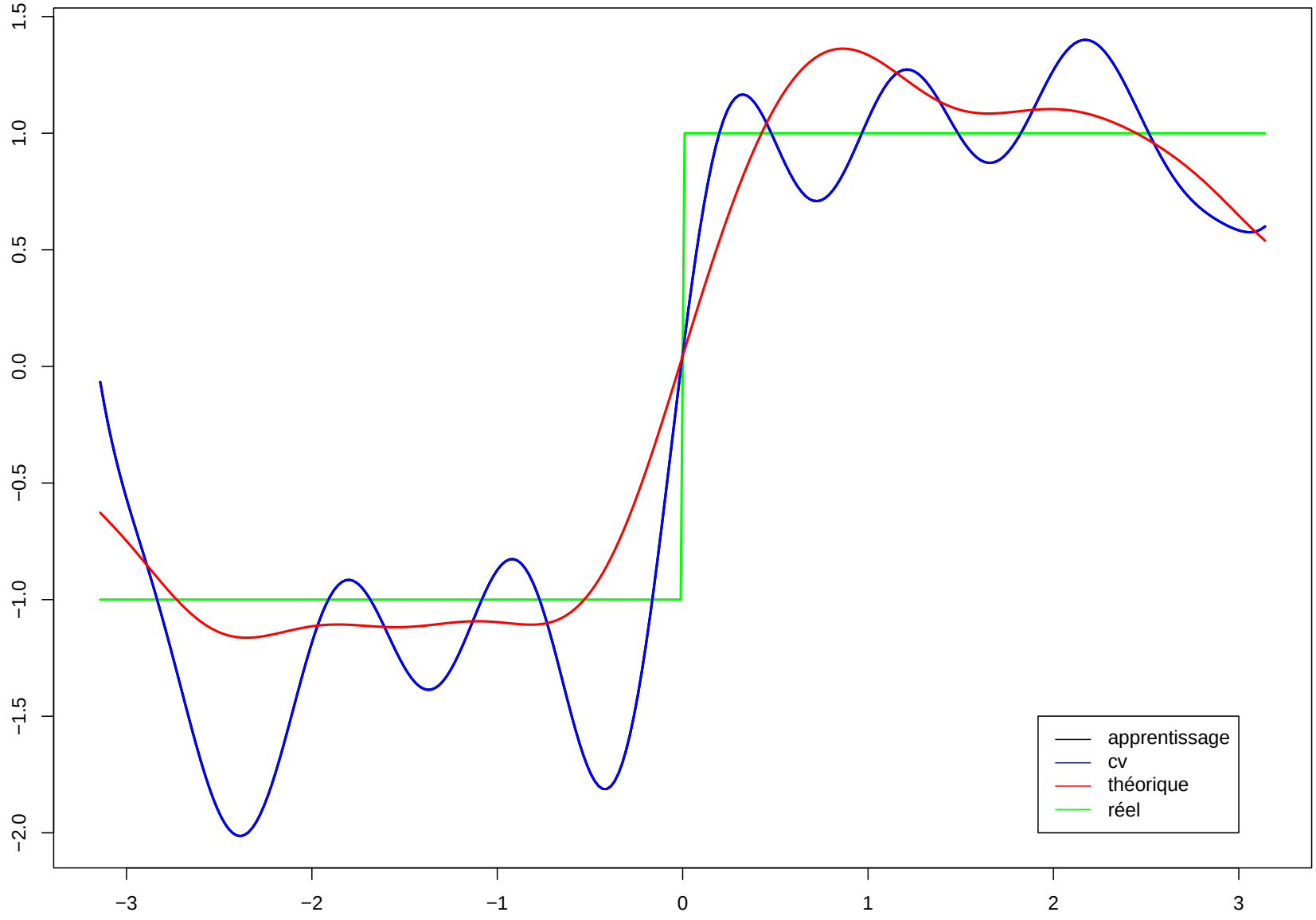
- pour chaque valeur de l'hyper-paramètre θ, on évalue les performances du meilleur f de $\mathcal{M}_\theta$ selon la procédure de validation croisée
- le meilleur θ est celui qui donne les meilleures performances estimées par VC

Attention : la VC ne donne pas de modèle, mais seulement des performances. Il faut ensuite estimer le meilleur f de $\mathcal{M}_\theta$, puis évaluer ses performances (avec un ensemble de test, par exemple).

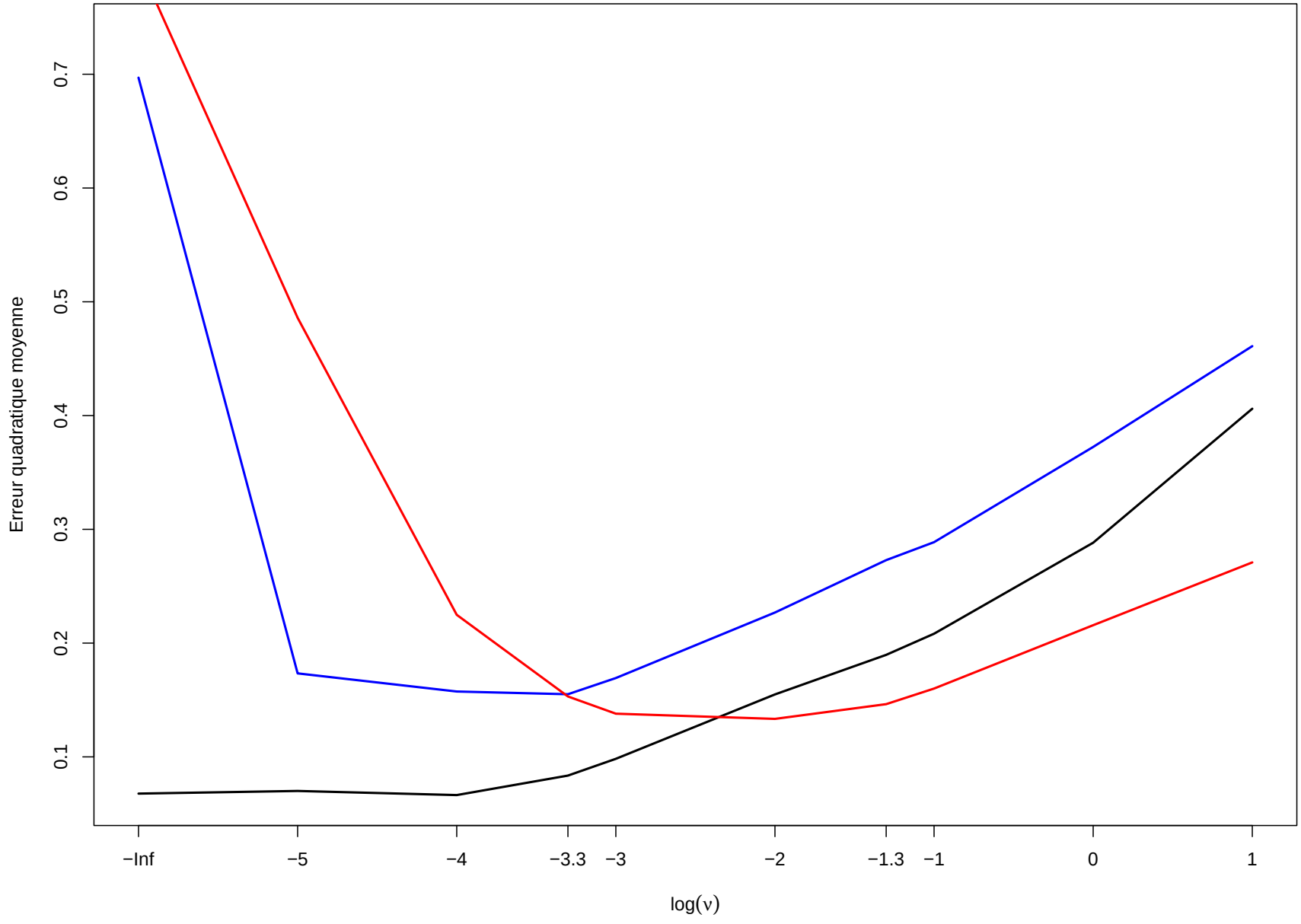
Créneau : erreur en fonction de ν



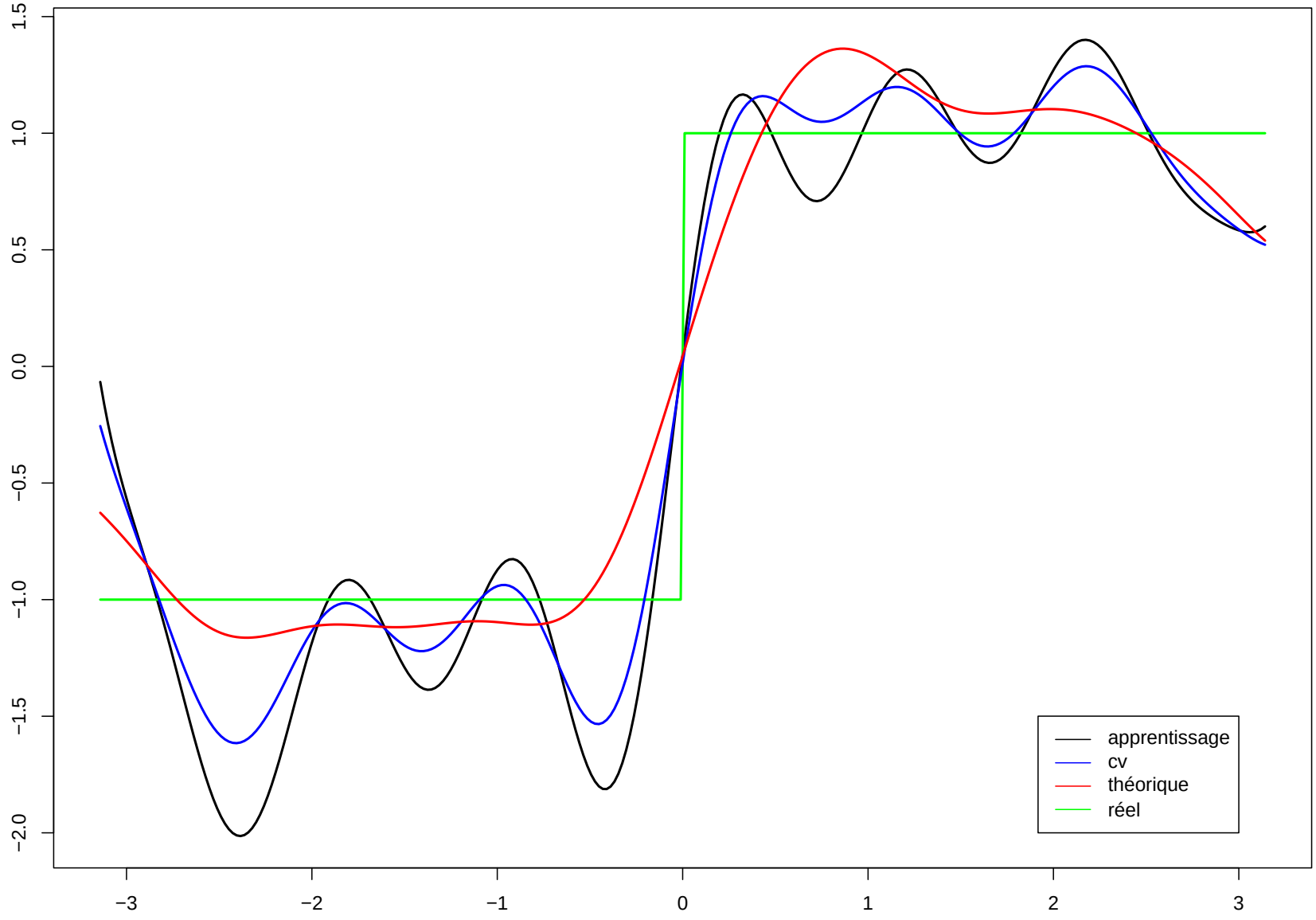
Sélection de modèle



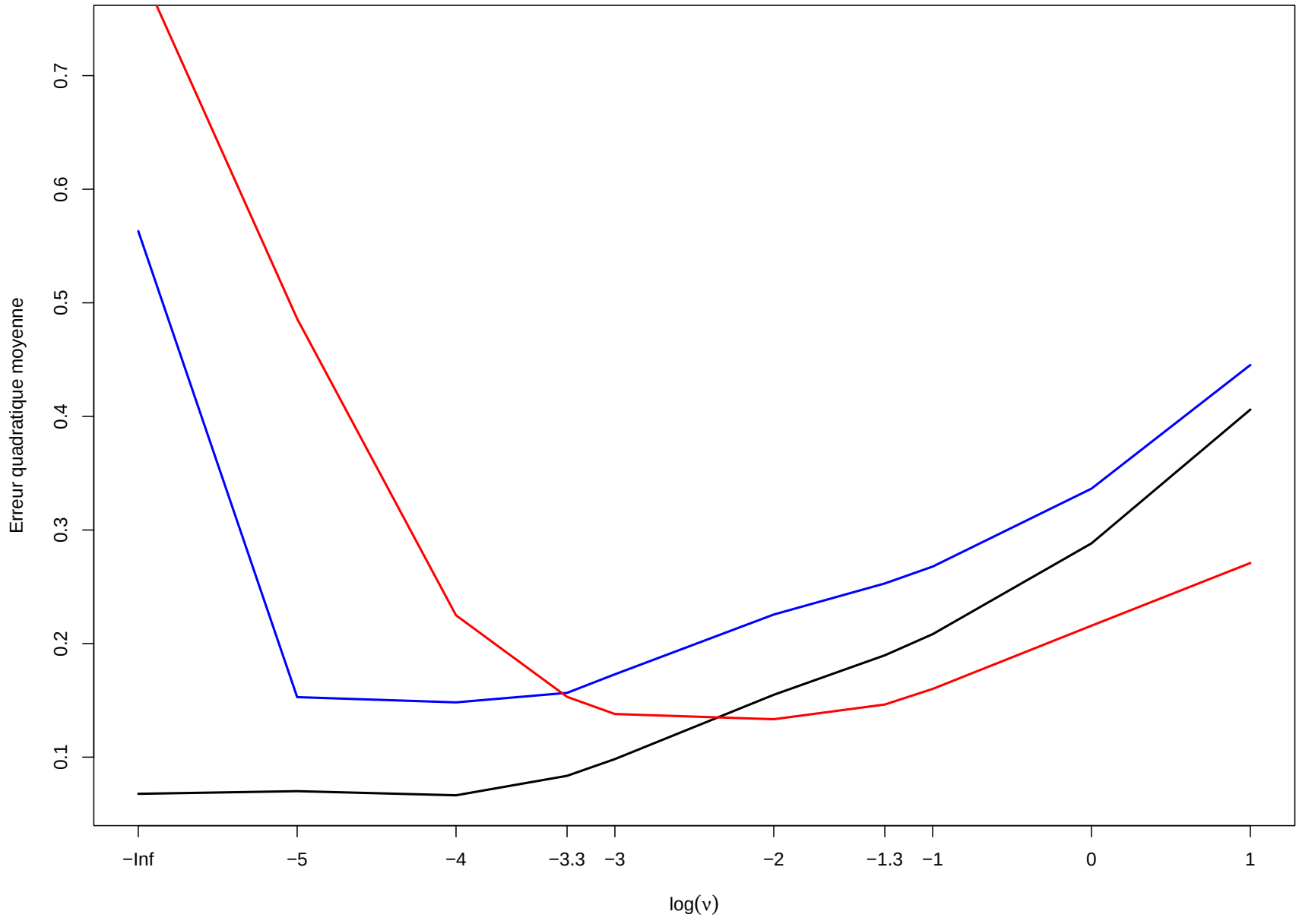
Sensible au choix des morceaux !



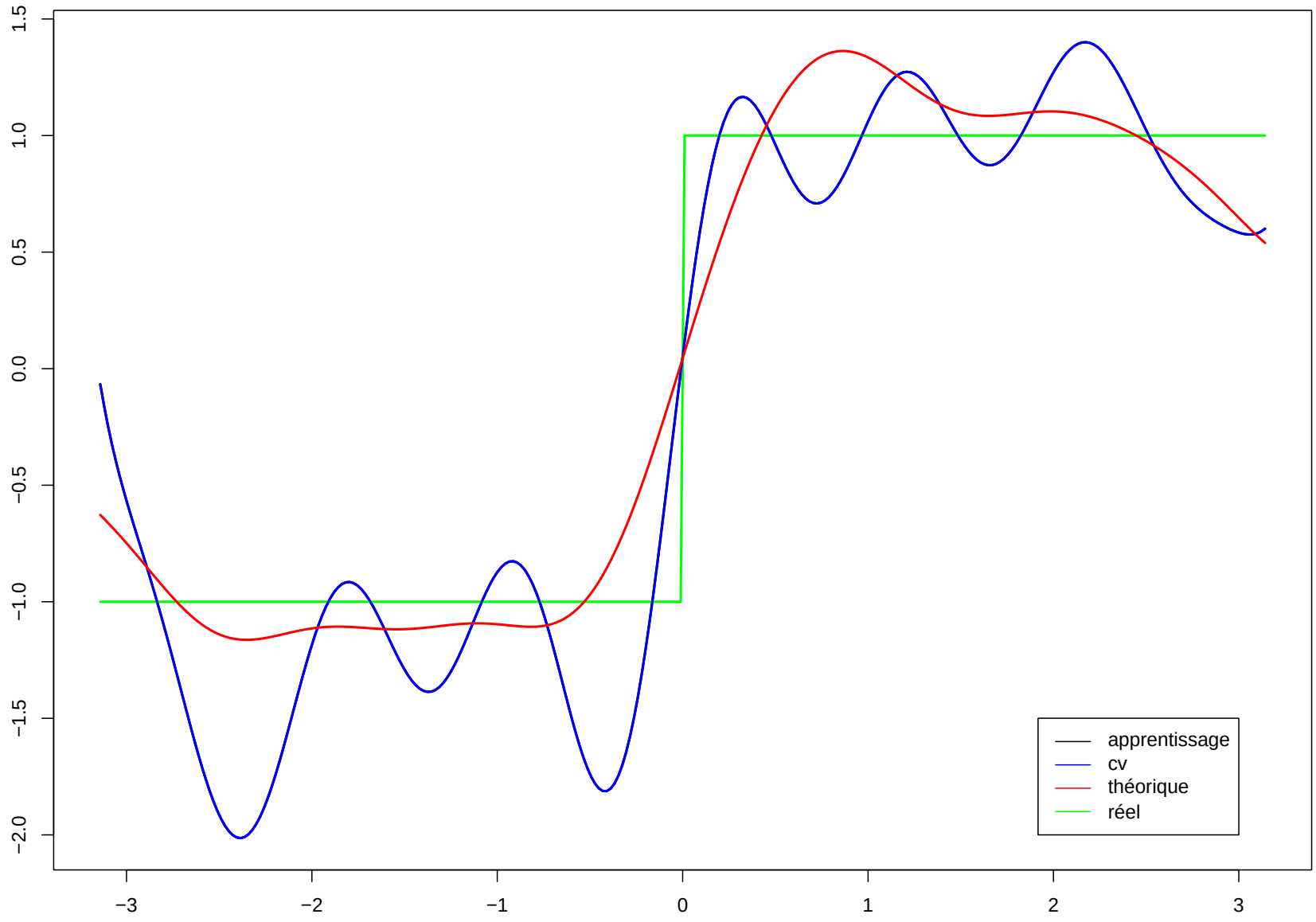
Sensible au choix des morceaux !



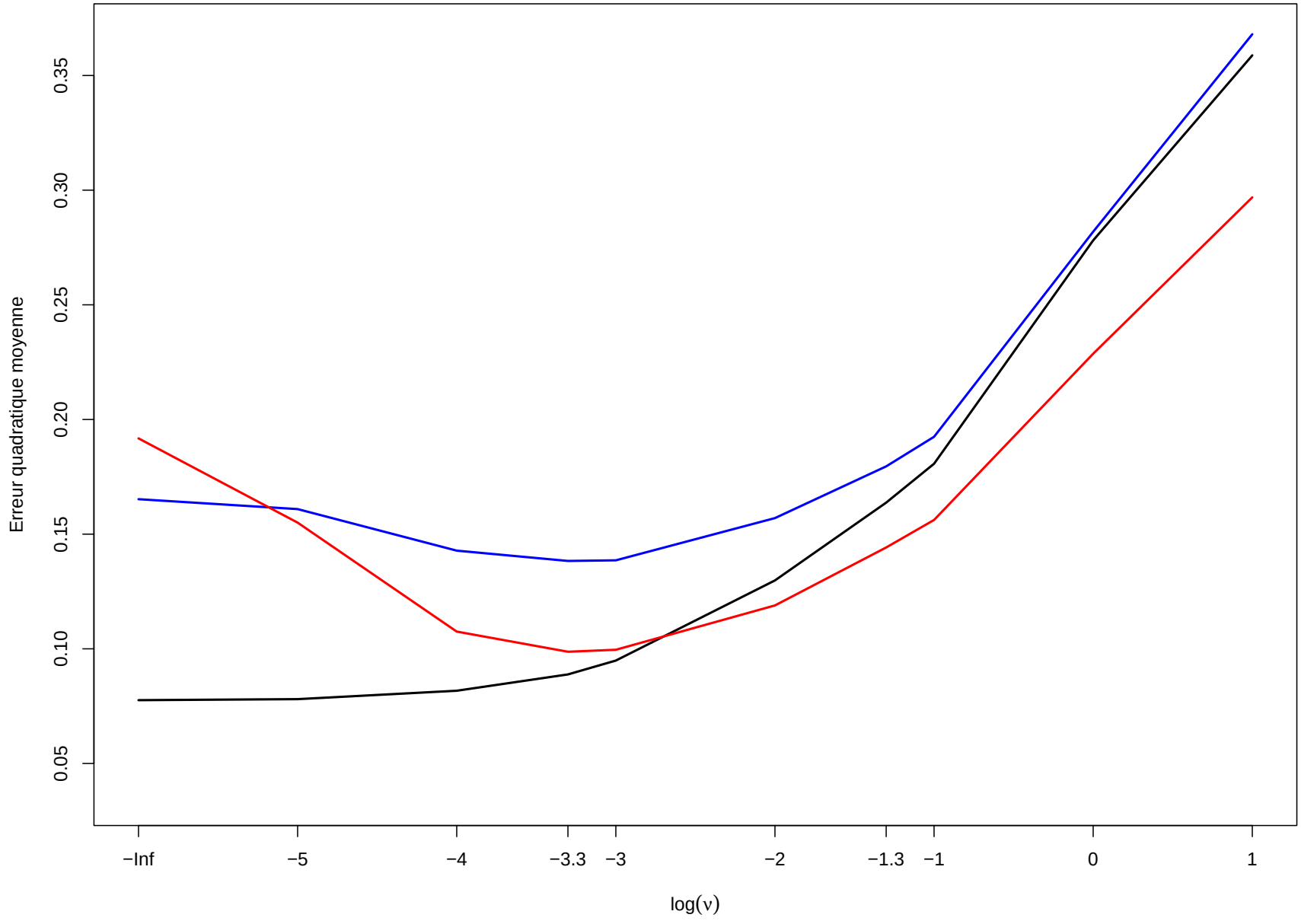
Sensible au nombre de morceaux (8)



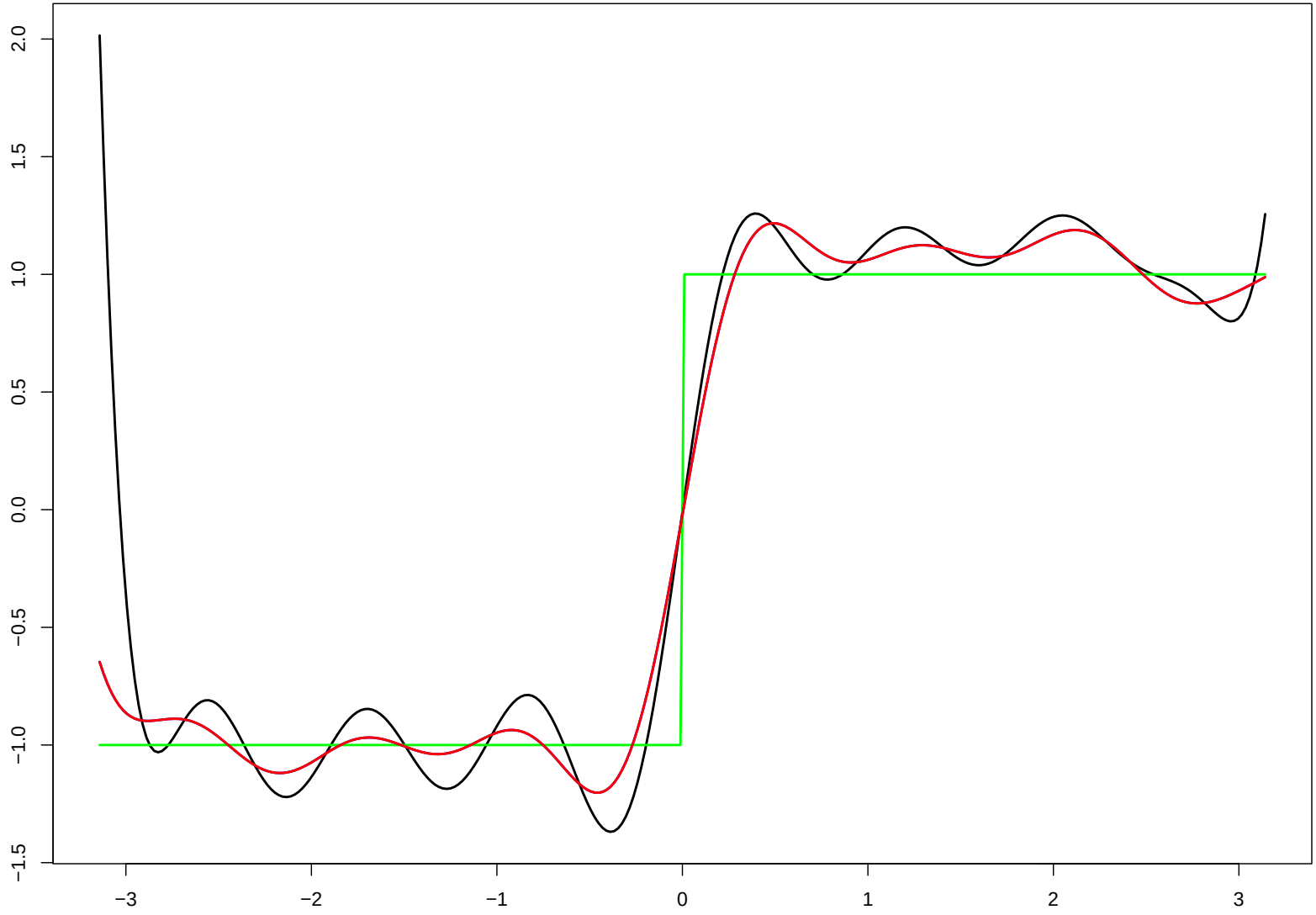
Sensible au nombre de morceaux (8)



Avec 80 exemples

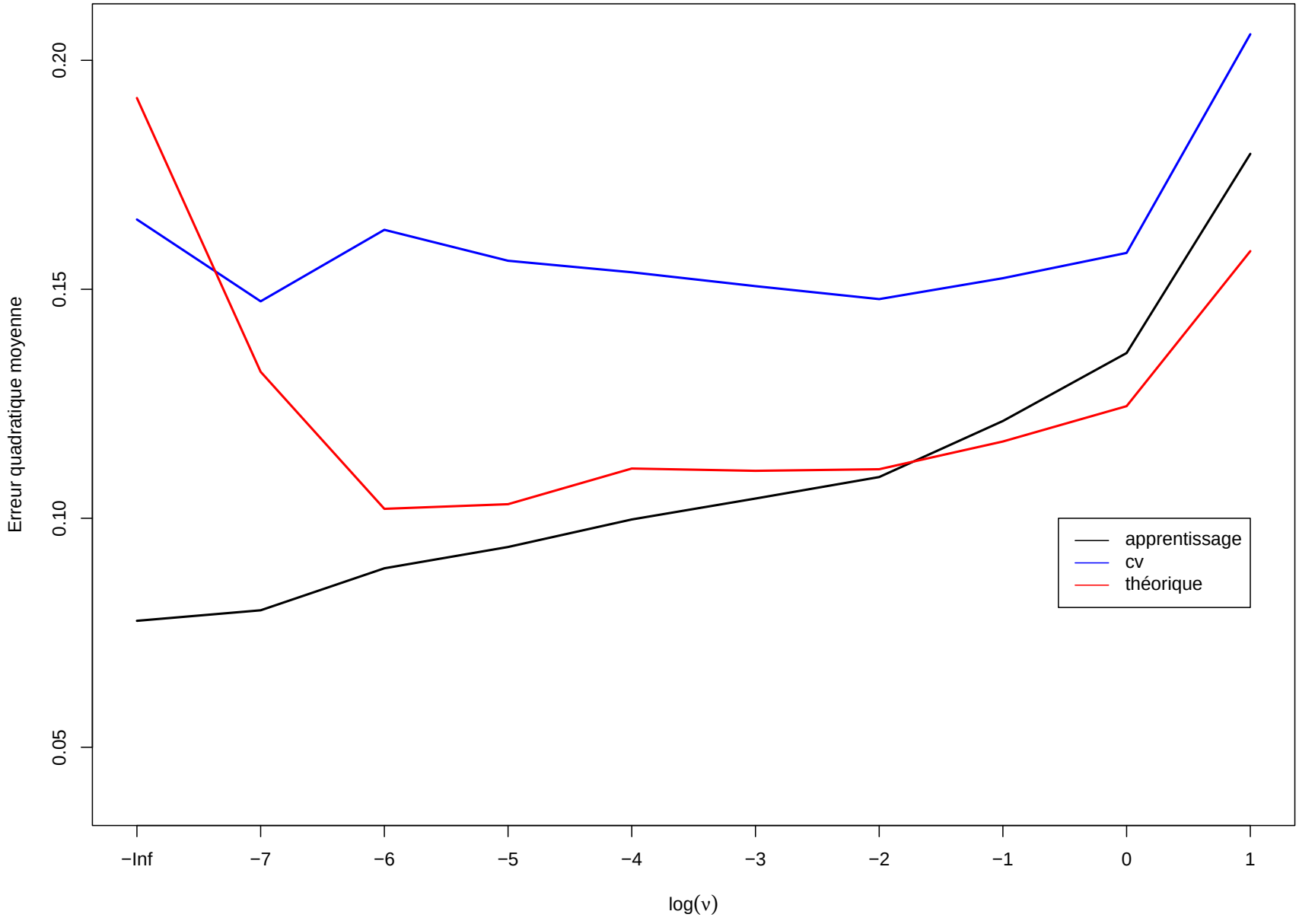


Avec 80 exemples

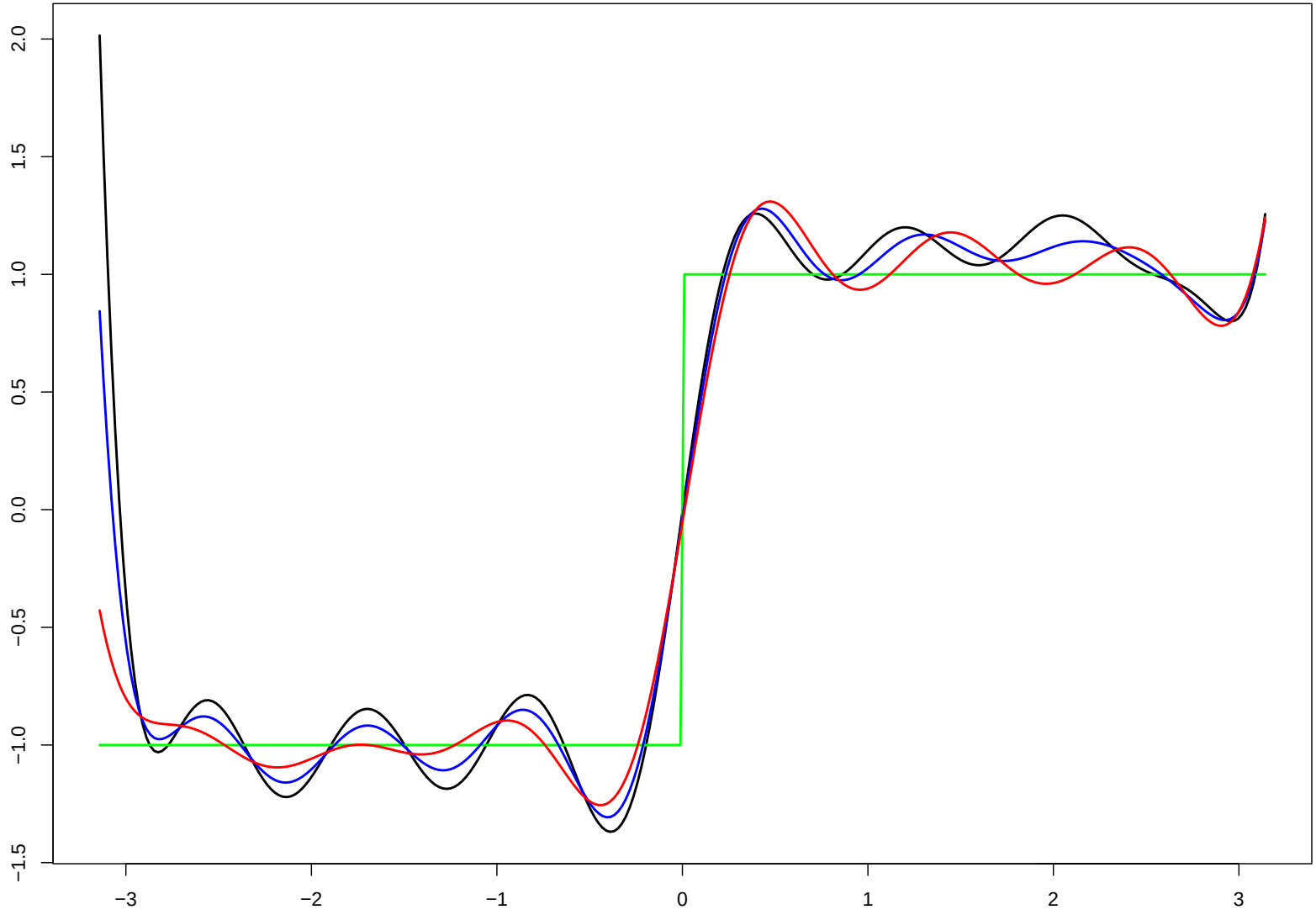


Erreur quadratique moyenne réelle $\simeq 0.036$

80 exemples “weight decay”

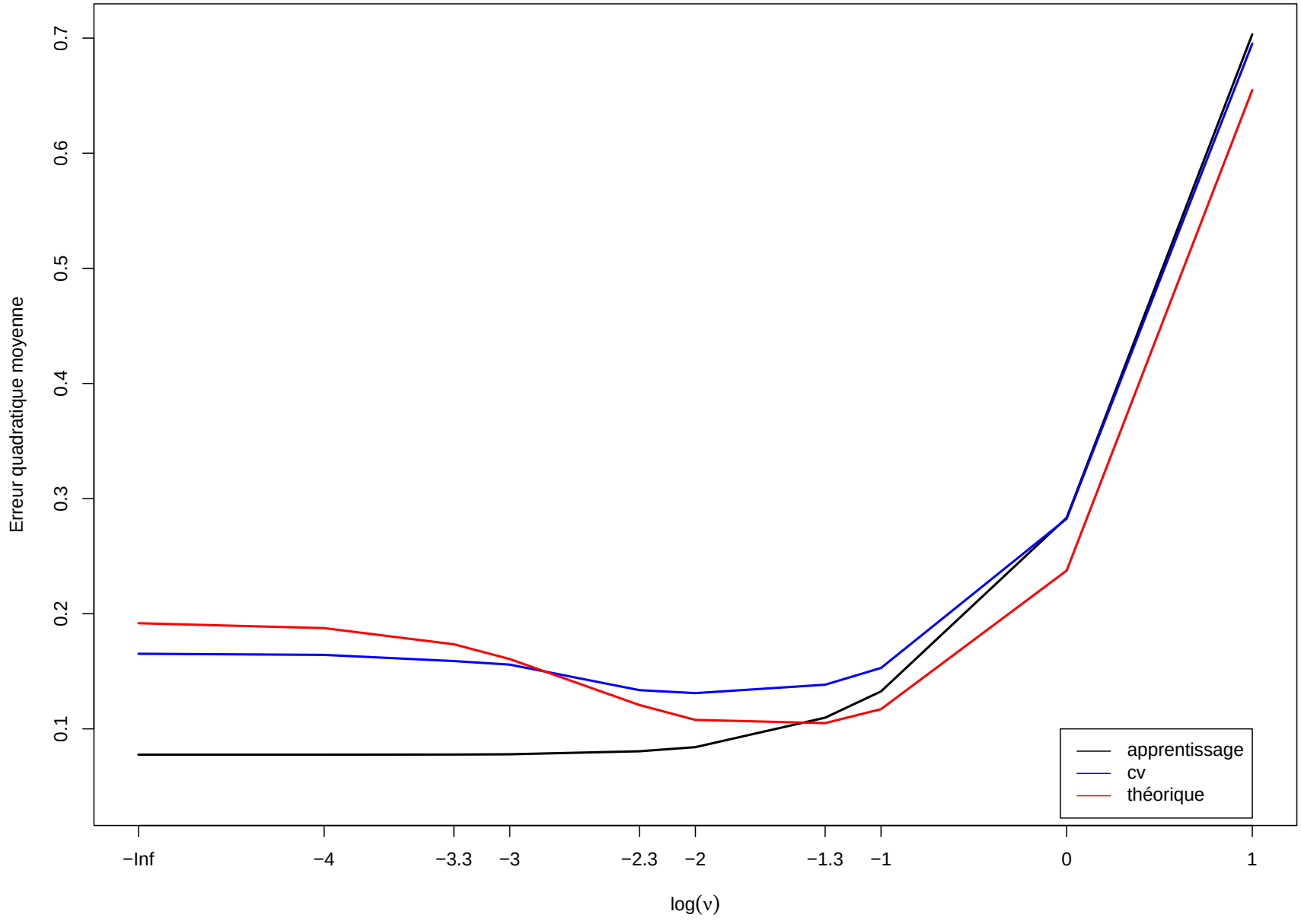


80 exemples “weight decay”

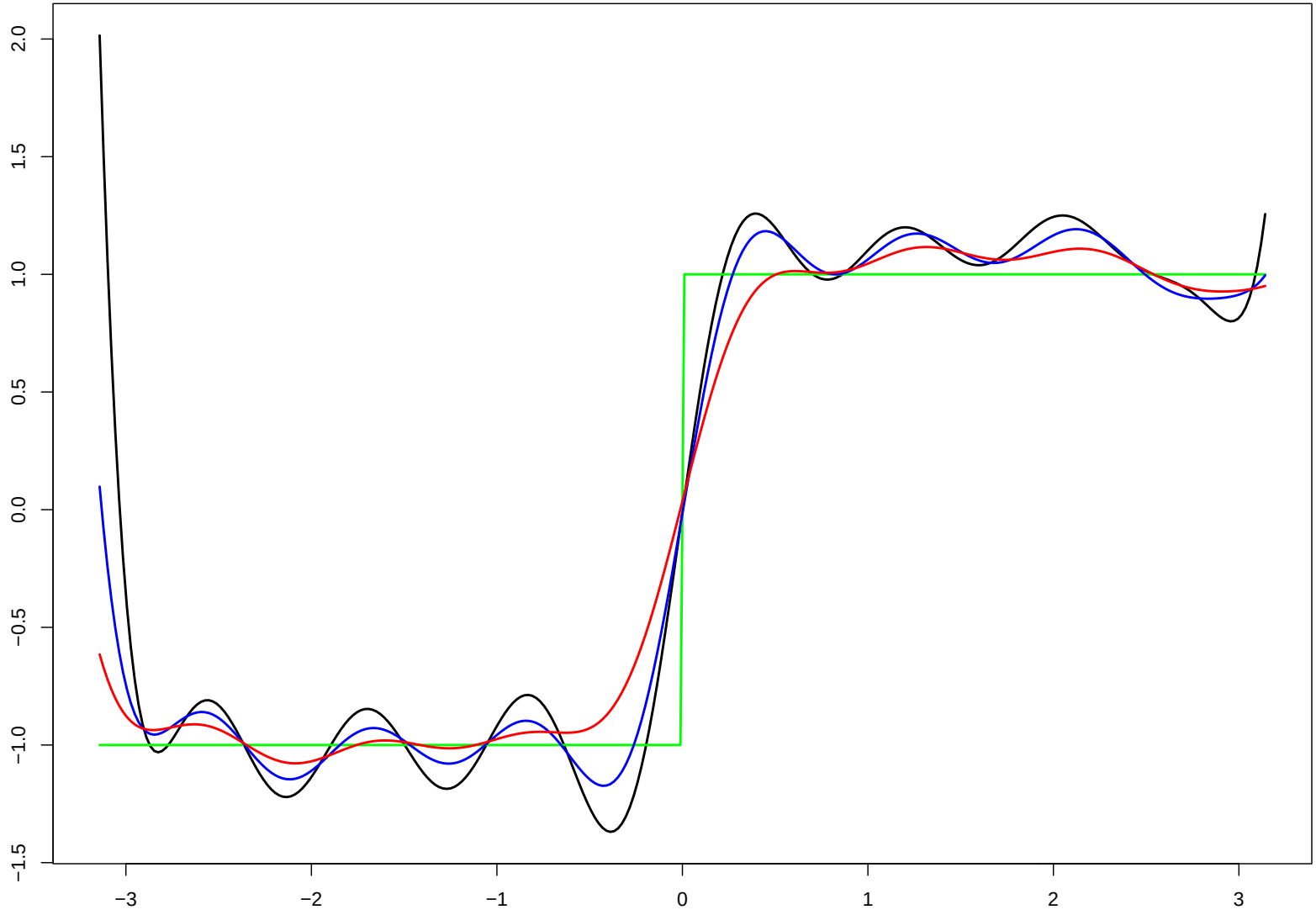


Erreur quadratique moyenne réelle $\simeq 0.069$

80 exemples dérivée première



80 exemples dérivée première



Erreur quadratique moyenne réelle $\simeq 0.045$

Critique de la Validation Croisée

Points positifs :

- facile à mettre en œuvre
- utilise toutes les données

Points négatifs :

- sensible au découpage :
  - choix du nombre de blocs
  - choix des blocs eux-mêmes
- temps de calcul élevé
- la VC ne donne pas de modèle

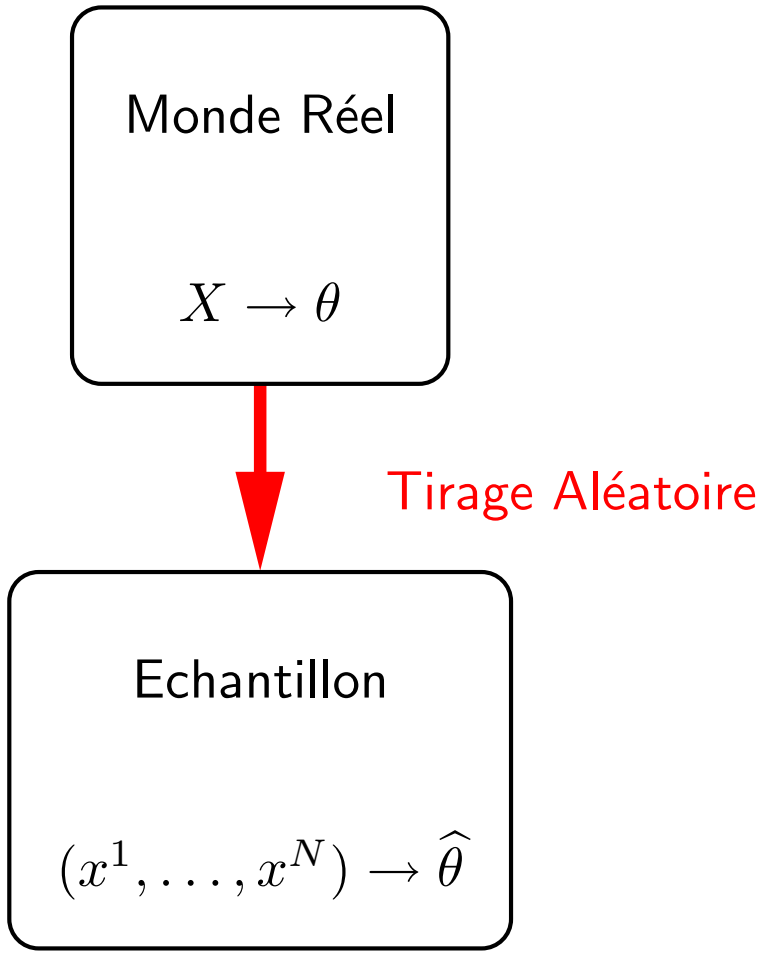
Le *Bootstrap*

Méthode générale d'estimation de la qualité d'un estimateur, basée sur un ré-échantillonnage :

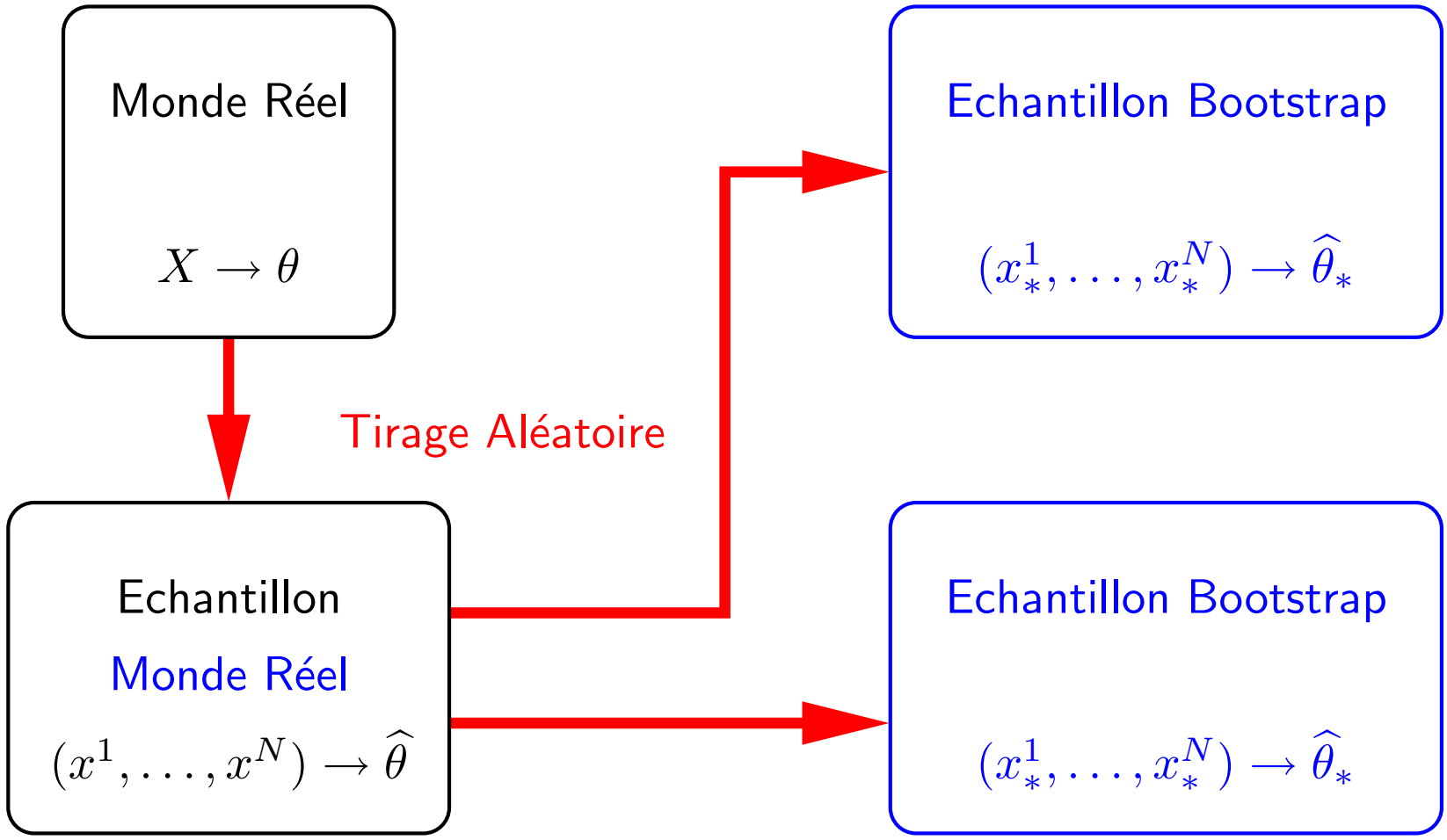
- on cherche à estimer θ, un paramètre associé à la loi d'un ensemble d'observations, les x^i
- on se donne $\widehat{\theta}(x^1, \ldots, x^N)$ un estimateur de θ
- on cherche à déterminer :
  - le biais de $\widehat{\theta}$
  - la variance de $\widehat{\theta}$

Le *bootstrap* permet d'estimer ces deux quantités grâce à des échantillons *bootstrap* : un échantillon *bootstrap* est un N-uplet, $(x^1_*, \ldots, x^N_*)$ obtenu par **tirage aléatoire uniforme avec remise** dans l'échantillon d'origine $(x^1, \ldots, x^N)$.

Principe


Monde Réel
$X \to \theta$
Tirage Aléatoire
Echantillon
$(x^1, \ldots, x^N) \to \widehat{\theta}$

Principe


Monde Réel
$X \to \theta$
Tirage Aléatoire
Echantillon
Monde Réel
$(x^1, \ldots, x^N) \to \widehat{\theta}$
Echantillon Bootstrap
$(x_*^1, \ldots, x_*^N) \to \widehat{\theta}_*$
Echantillon Bootstrap
$(x_*^1, \ldots, x_*^N) \to \widehat{\theta}_*$

Estimation du biais

Algorithme :

1. pour b allant de 1 à n
   (a) engendrer un échantillon bootstrap $(x^1_{*b}, \ldots, x^N_{*b})$
   (b) calculer $\widehat{\theta}_{*b} = \widehat{\theta}(x^1_{*b}, \ldots, x^N_{*b})$
2. l'estimation du biais est

$$\frac{1}{n}\sum_{b=1}^{n} \widehat{\theta}_{*b} - \widehat{\theta}(x^1, \ldots, x^N)$$

Idée, remplacer le monde réel par l'échantillon :

- le premier terme estime l'espérance de l'estimateur
- le second terme estime l'estimateur

Estimation de la variance

Algorithme :

1. pour b allant de 1 à n
   (a) engendrer un échantillon bootstrap $(x^1_{*b}, \ldots, x^N_{*b})$
   (b) calculer $\widehat{\theta}_{*b} = \widehat{\theta}(x^1_{*b}, \ldots, x^N_{*b})$
2. calculer

$$\widehat{\theta}_* = \frac{1}{b} \sum_{b=1}^{n} \widehat{\theta}_{*b}$$

3. l'estimation de la variance est

$$\frac{1}{n-1} \sum_{b=1}^{n} \left(\widehat{\theta}_{*b} - \widehat{\theta}_* \right)^2$$

Application à l'évaluation d'un modèle

Raisonnement :

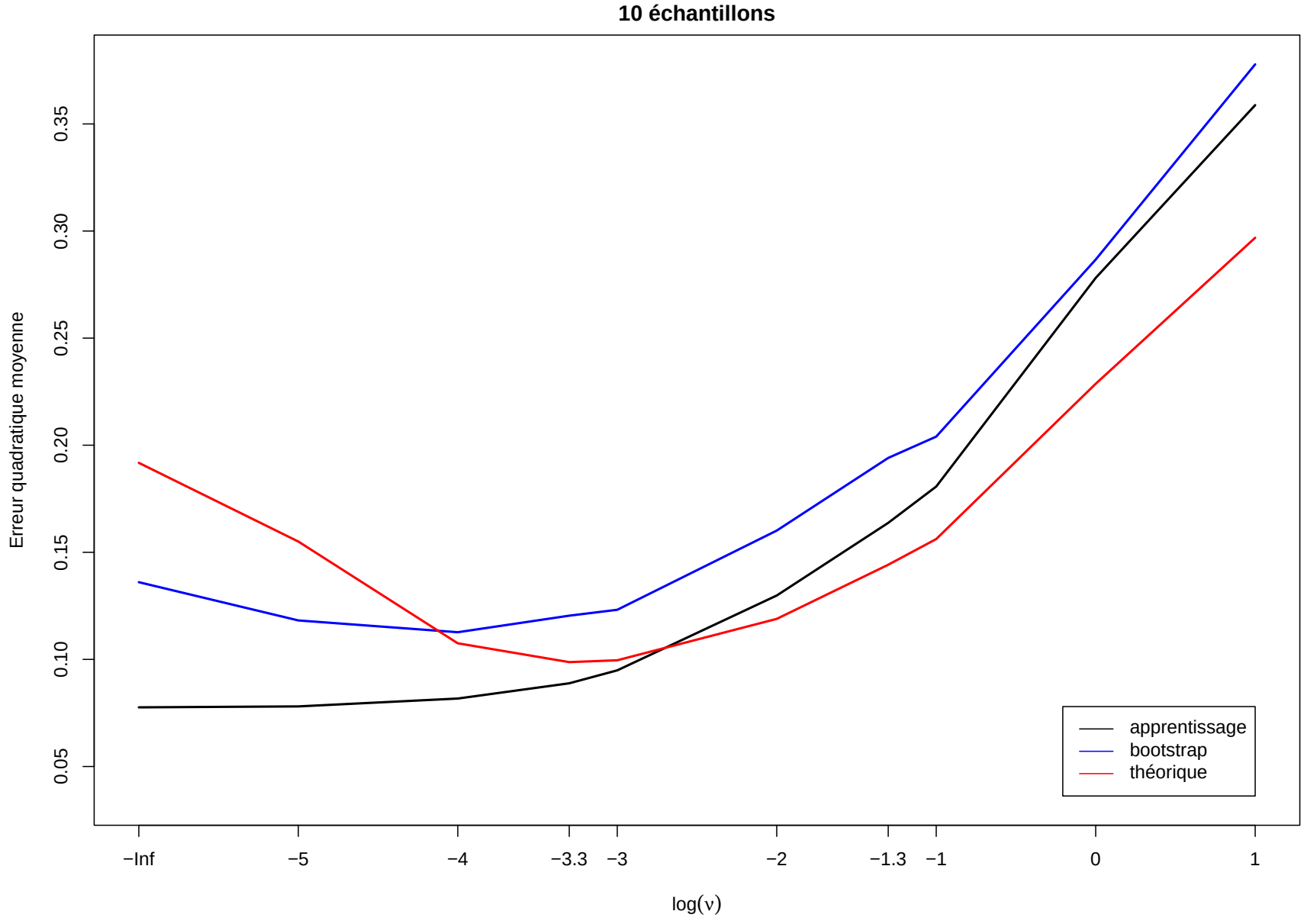
- l'évaluation d'un modèle consiste à estimer ses performances
- l'erreur résiduelle sur l'ensemble d'apprentissage sous-estime l'erreur réelle
- idée, estimer l'ampleur de la sous-estimation par *bootstrap* :
  - calculer la sous-estimation pour un échantillon *bootstrap*
  - moyenner les sous-estimations pour beaucoup d'échantillons *bootstrap*
  - corriger l'erreur résiduelle en ajoutant la moyenne

Évaluation d'un modèle

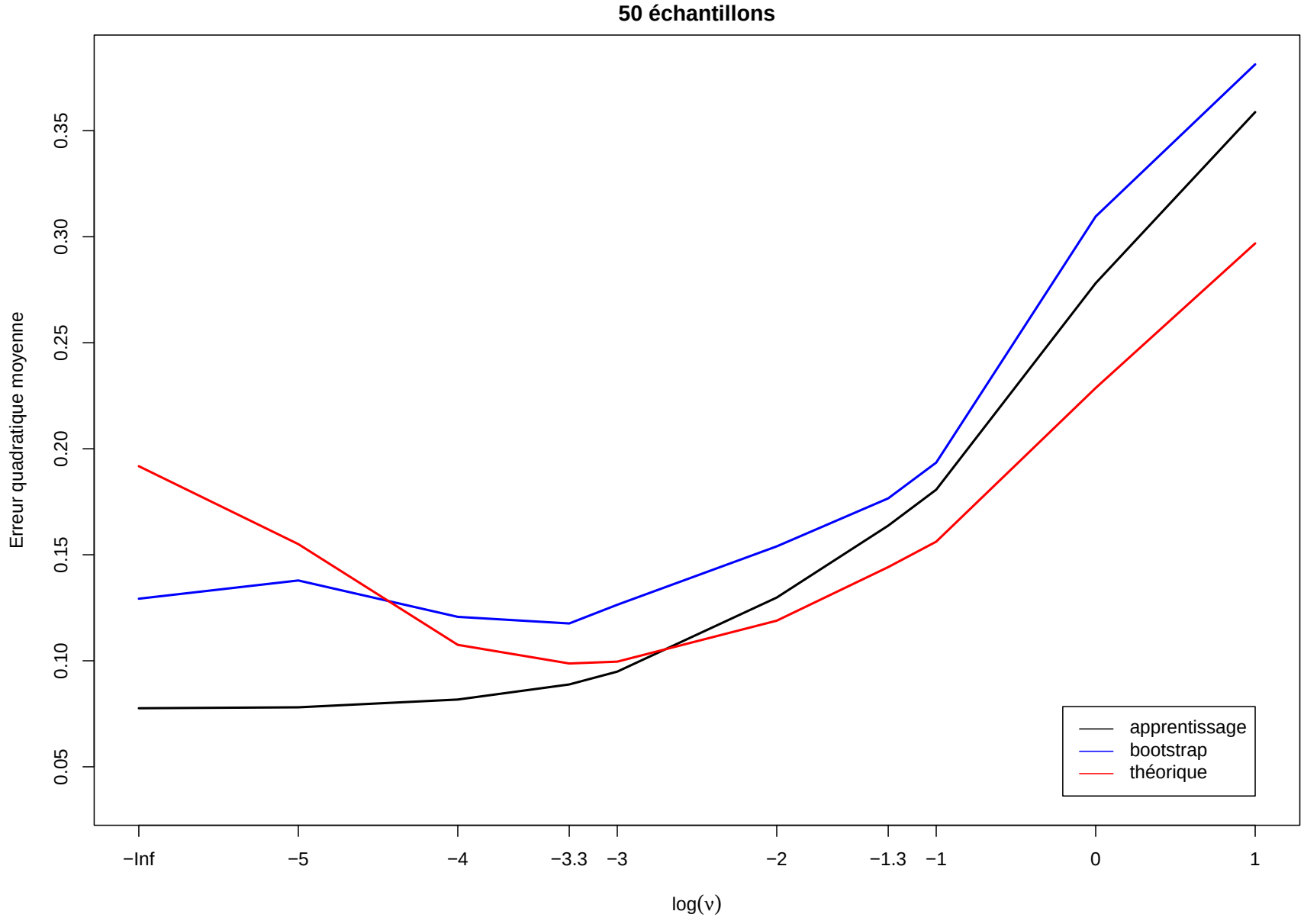
Algorithme :

1. pour b allant de 1 à n
   (a) engendrer un échantillon bootstrap $(x_{*b}^1, \ldots, x_{*b}^N)$ (à partir de l'ensemble d'apprentissage)
   (b) estimer le modèle optimal pour l'échantillon **bootstrap**
   (c) calculer $\widehat{\mathcal{B}}_{*b}$ comme la différence entre l'erreur résiduelle du modèle sur l'échantillon d'**apprentissage** et l'erreur résiduelle du modèle sur l'échantillon **bootstrap**
2. estimer l'erreur résiduelle $\widehat{\mathcal{E}}$ du modèle optimal sur l'ensemble d'apprentissage
3. corriger cette erreur en lui ajoutant $\frac{1}{n}\sum_{b=1}^{n}\widehat{\mathcal{B}}_{*b}$

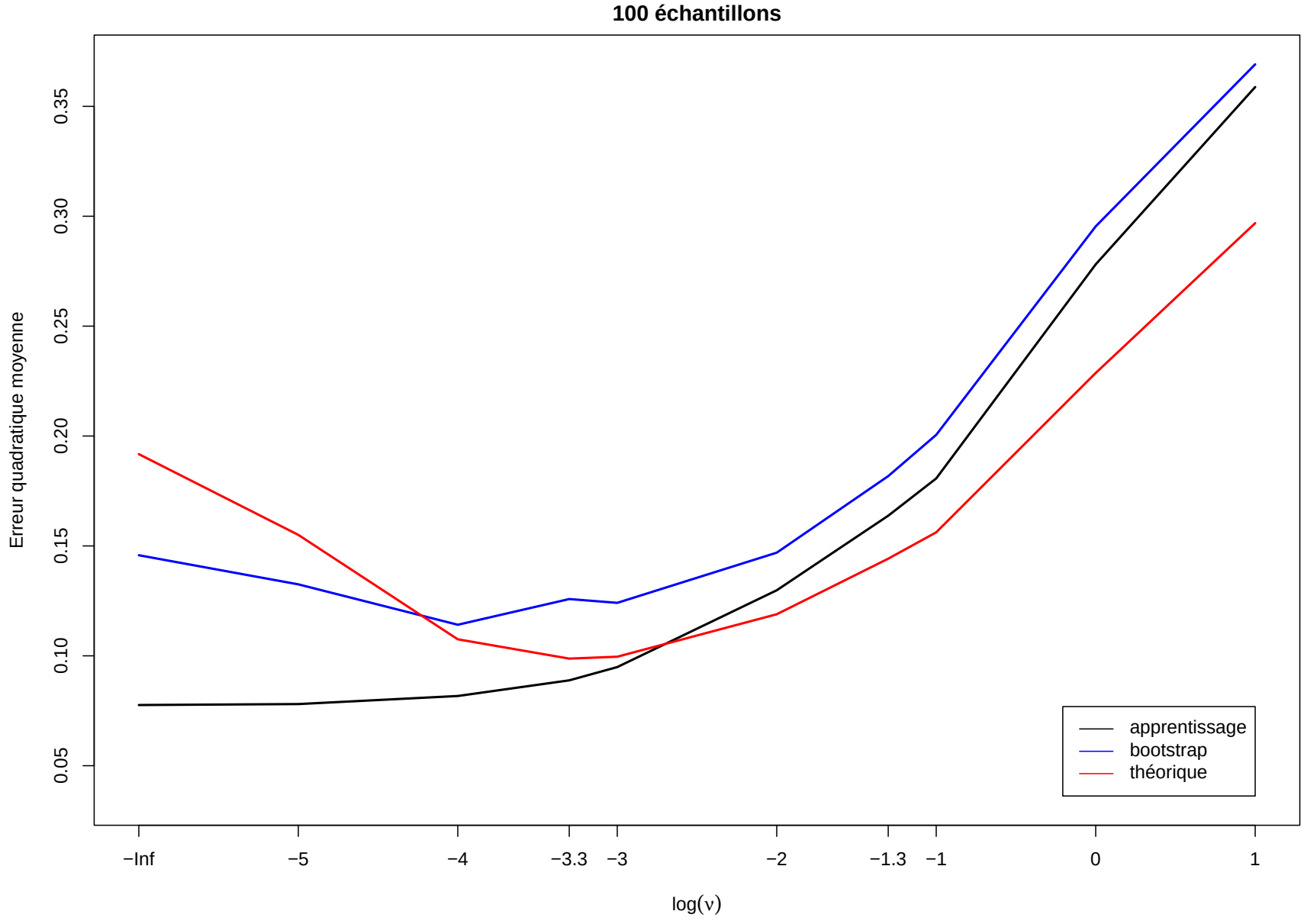
Créneau : erreur en fonction de ν



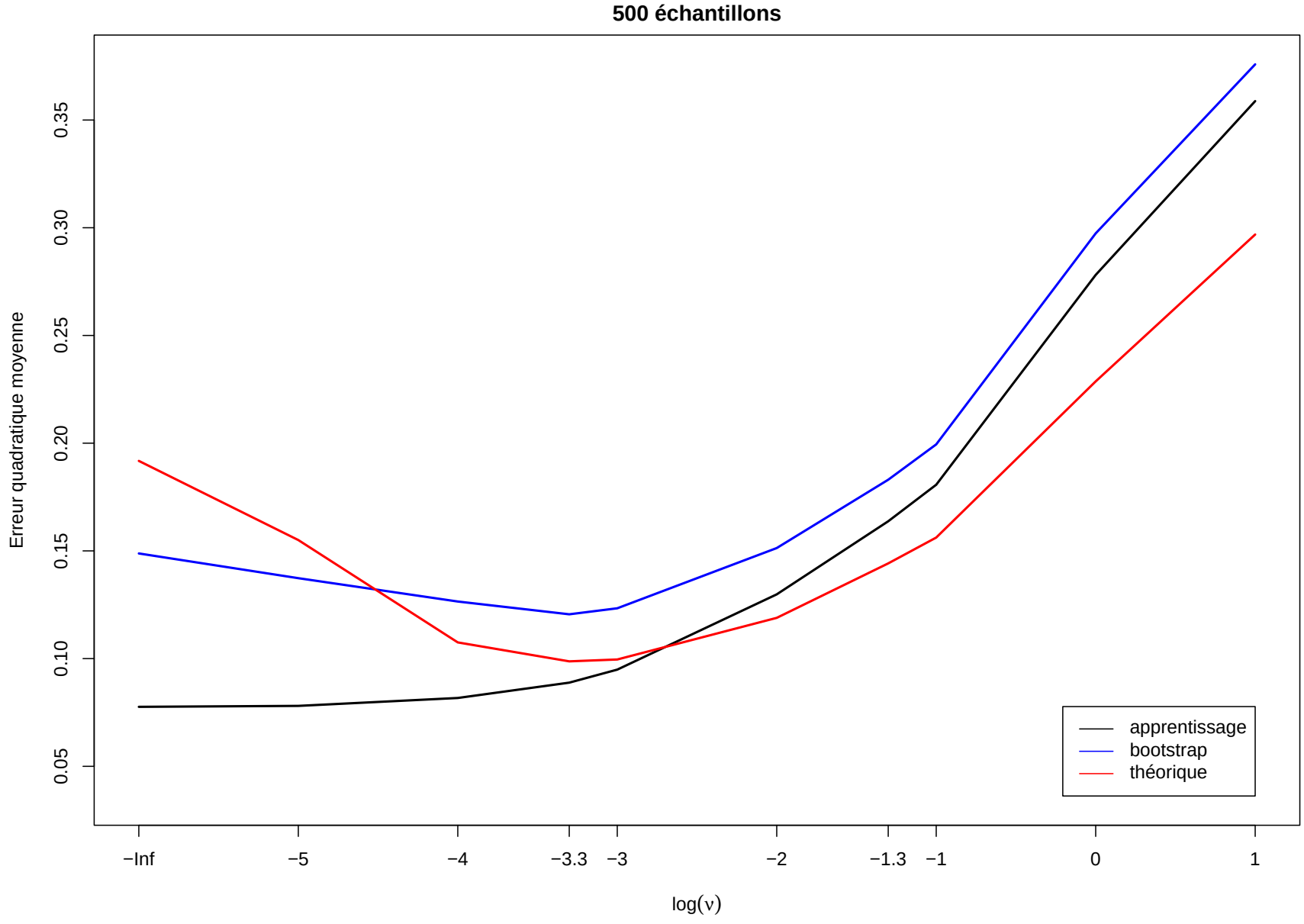
Créneau : erreur en fonction de ν



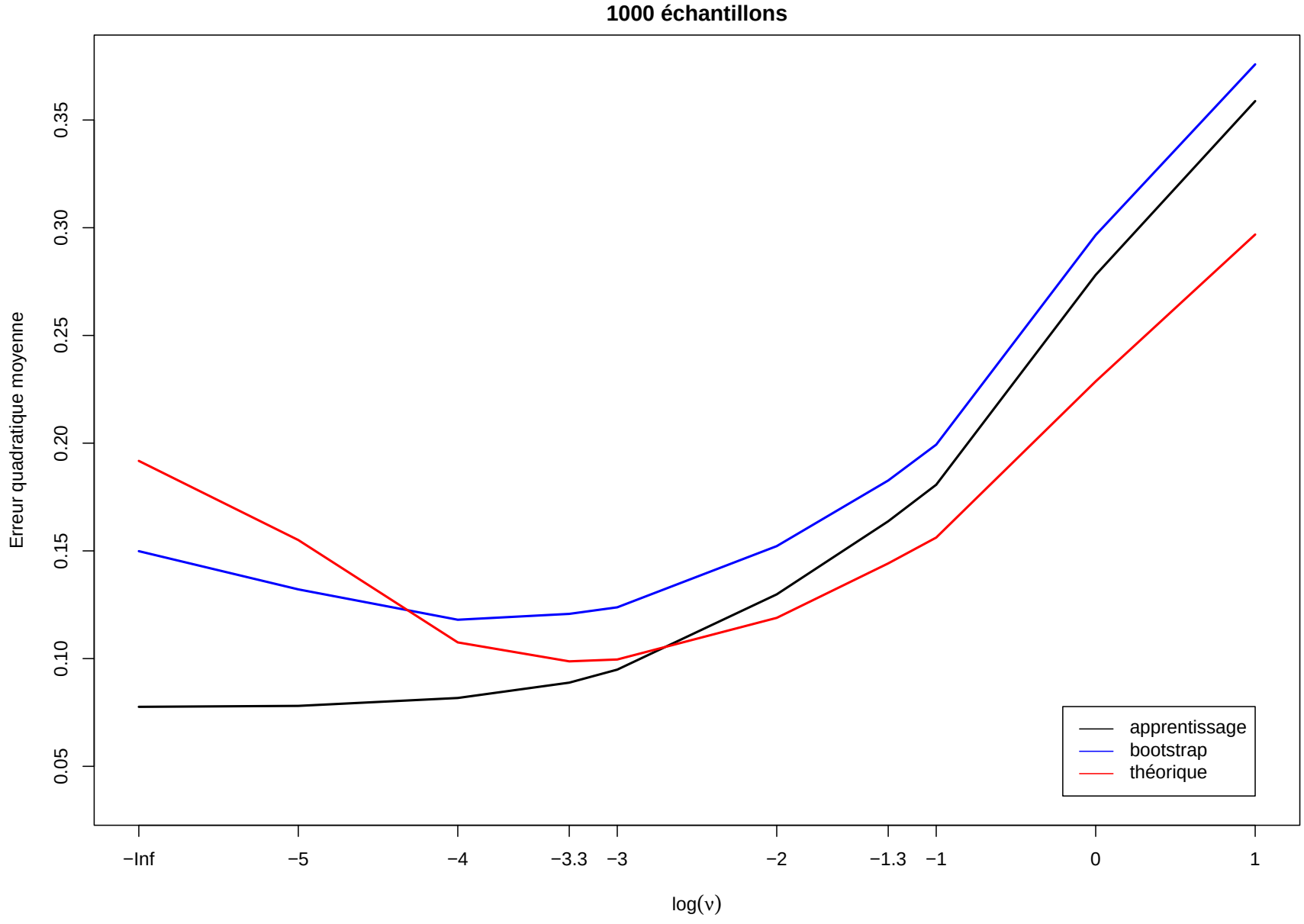
Créneau : erreur en fonction de ν



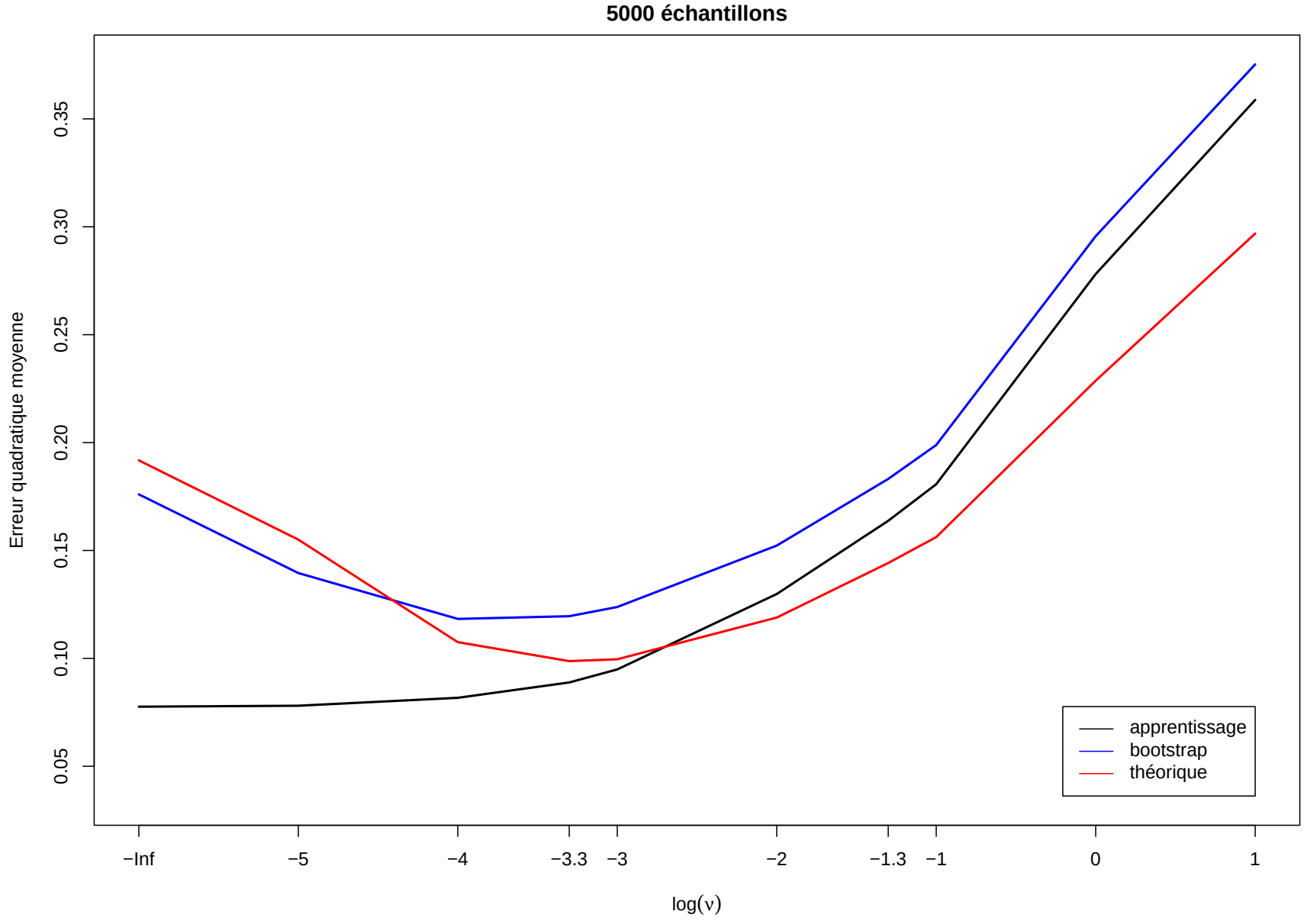
Créneau : erreur en fonction de ν



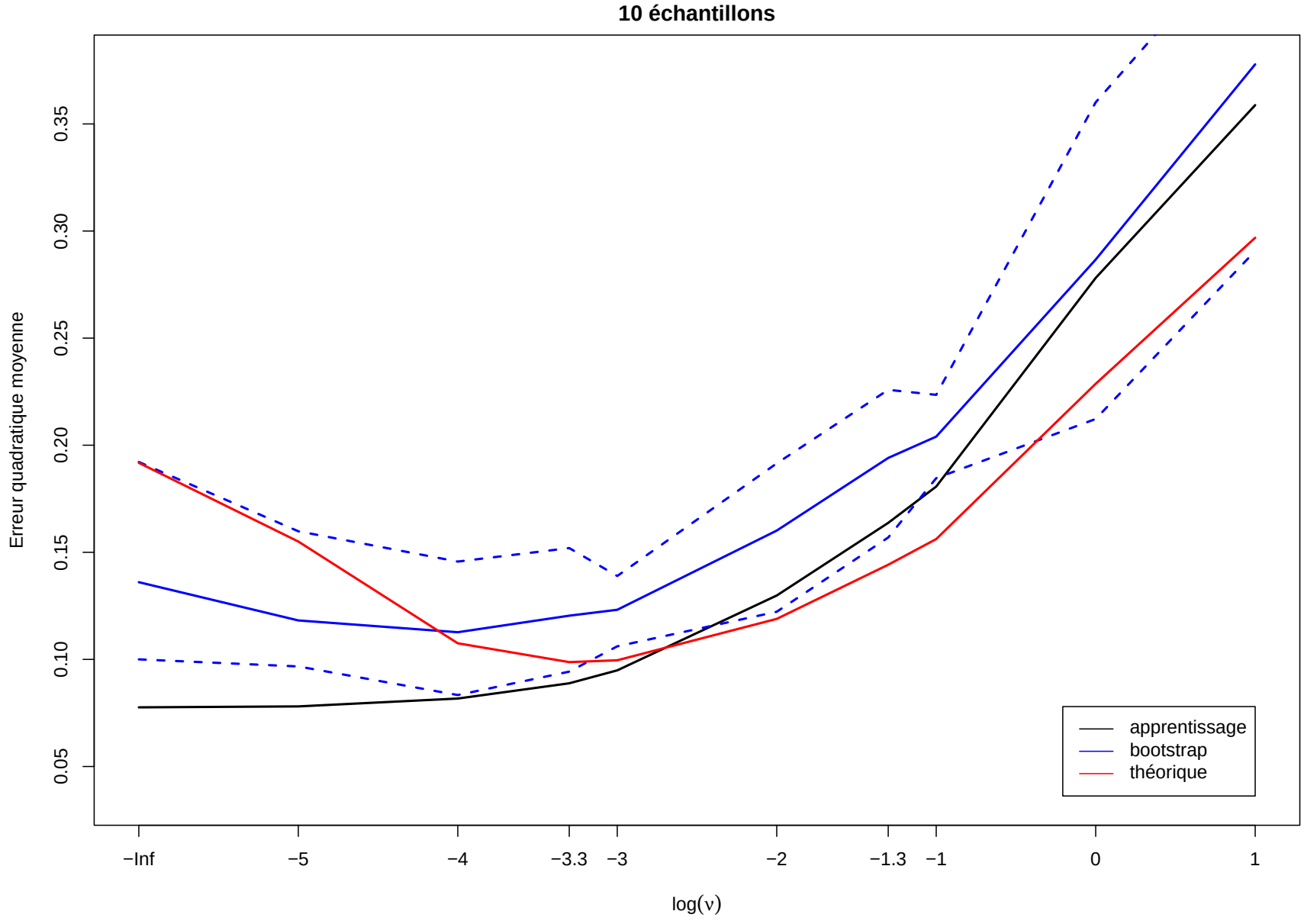
Créneau : erreur en fonction de ν



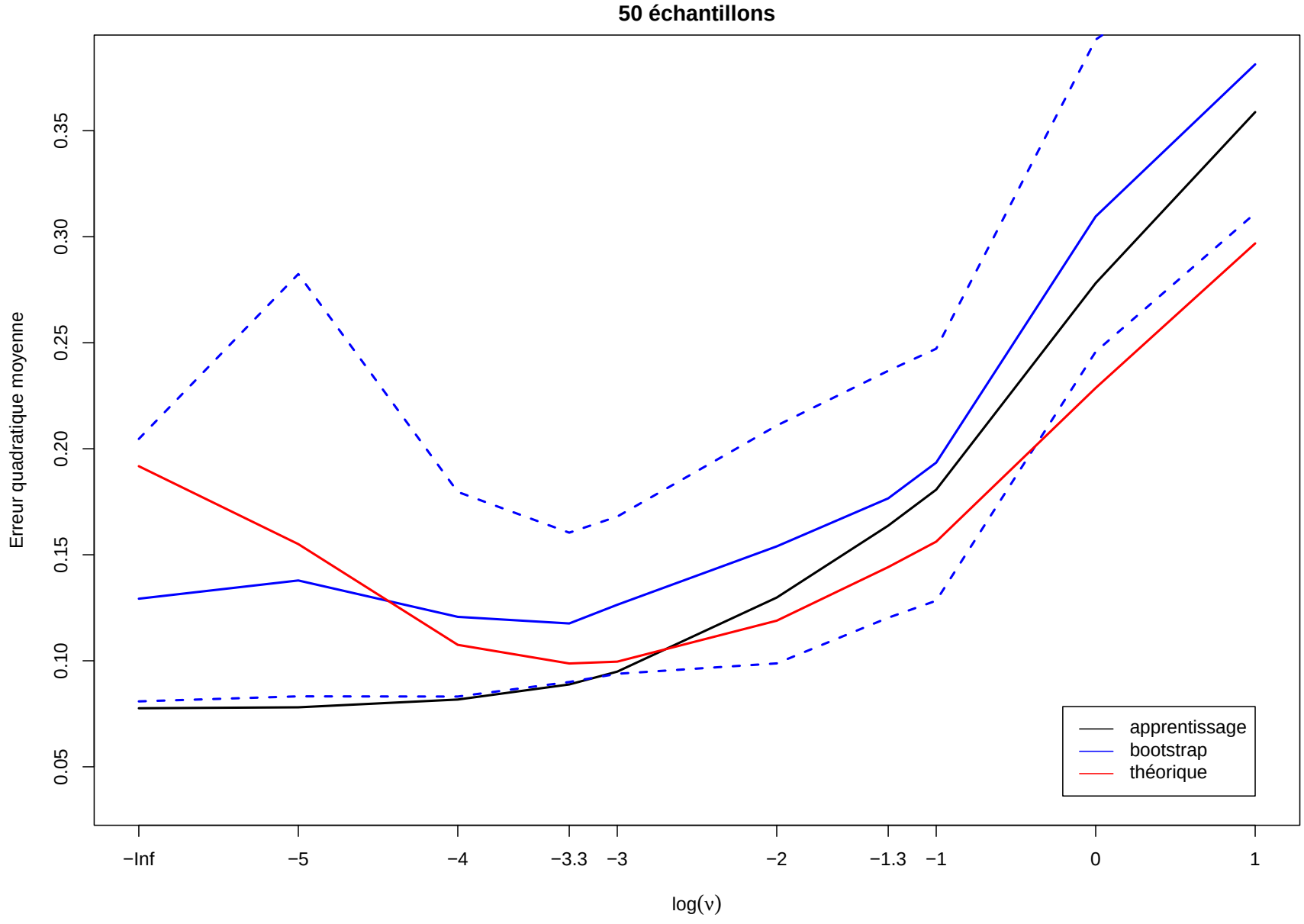
Créneau : erreur en fonction de ν



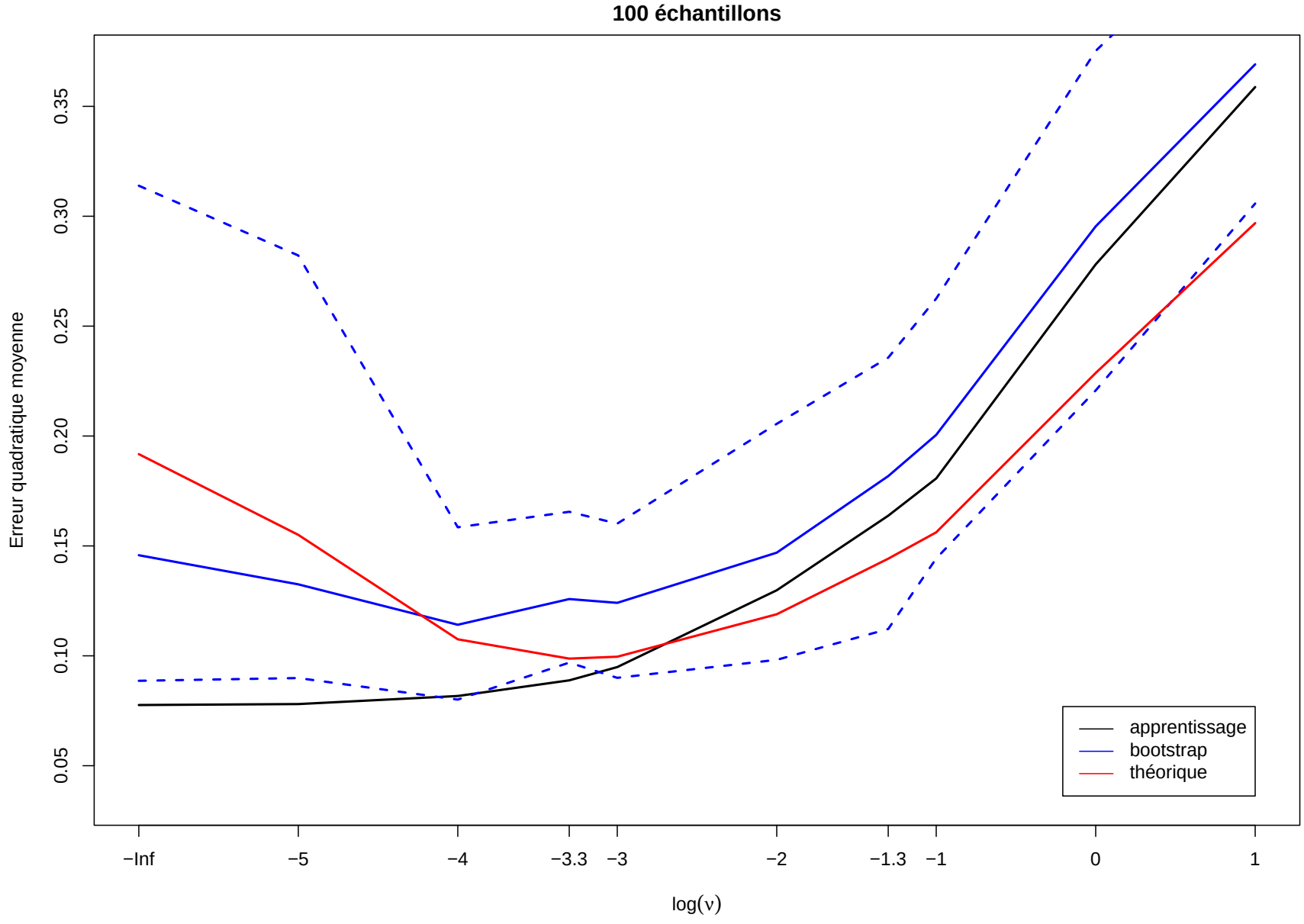
Intervalle de confiance



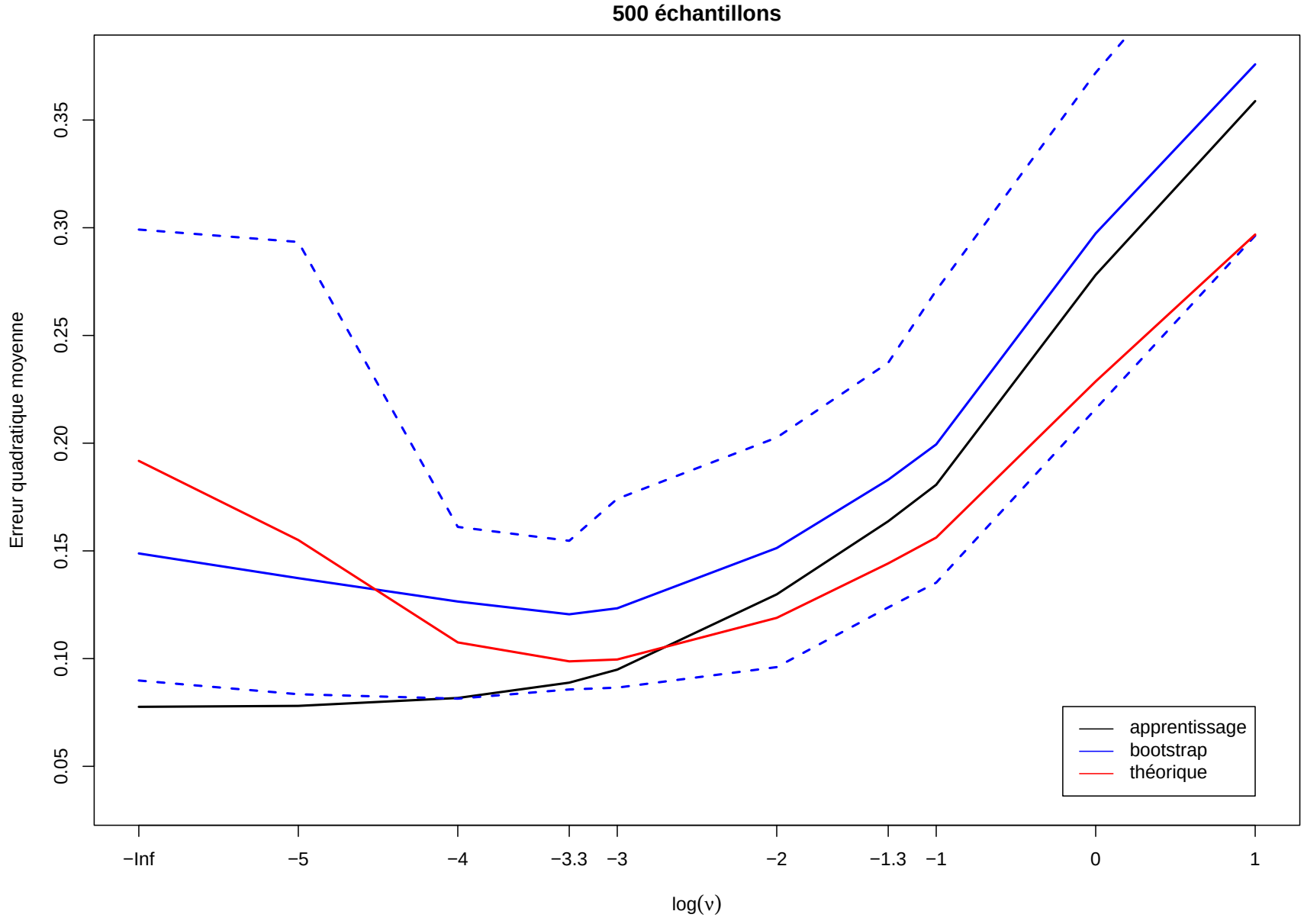
Intervalle de confiance



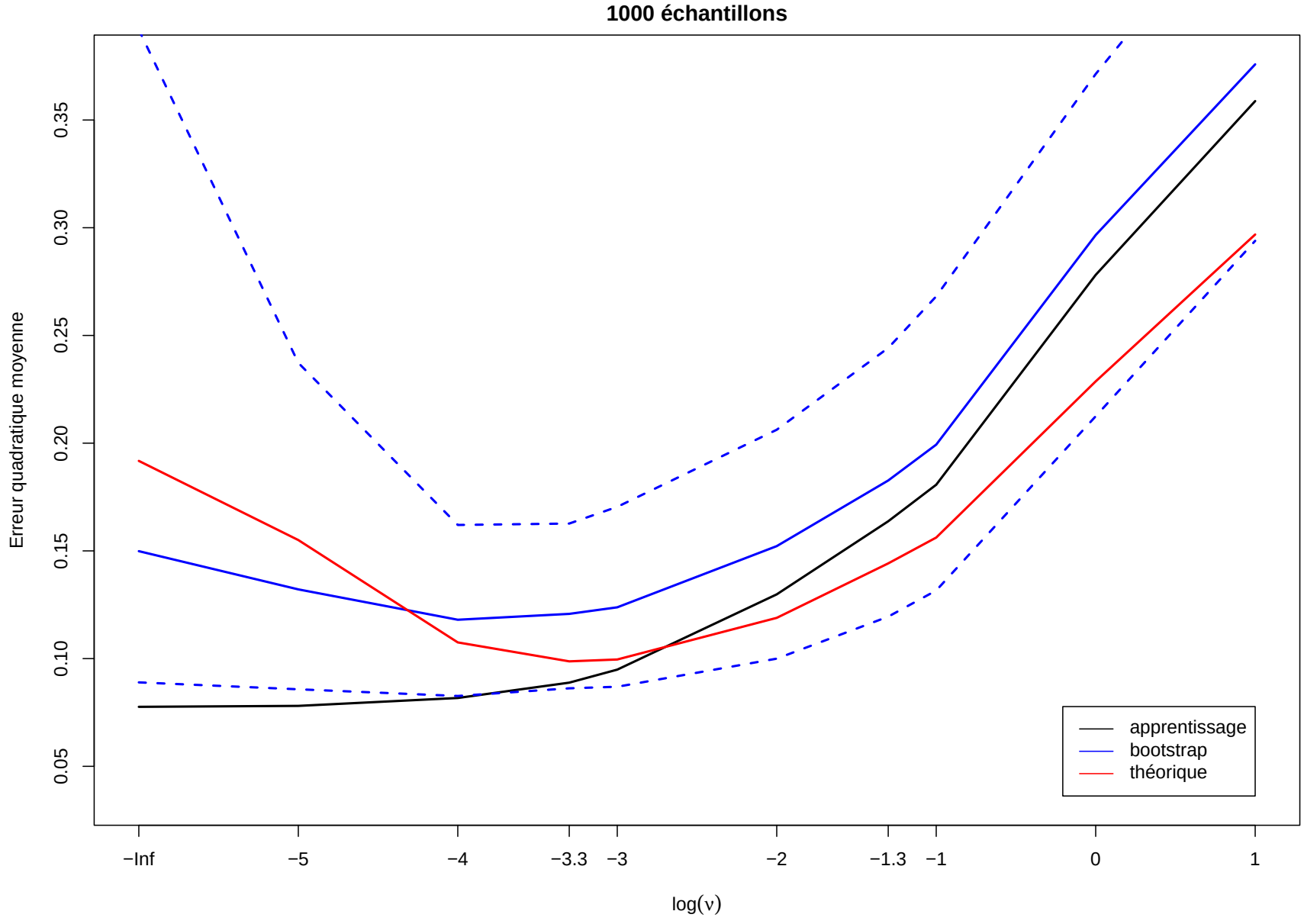
Intervalle de confiance



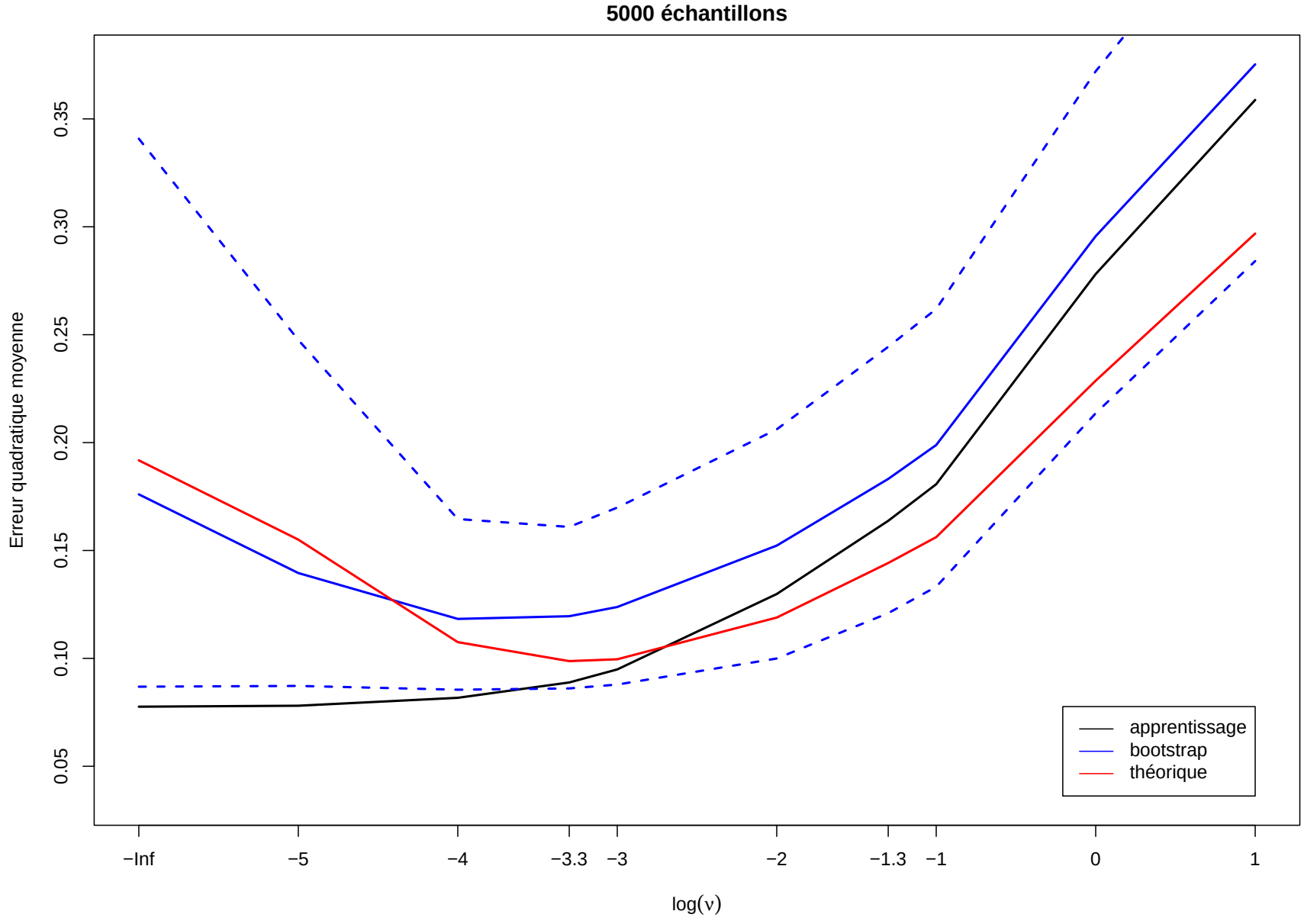
Intervalle de confiance



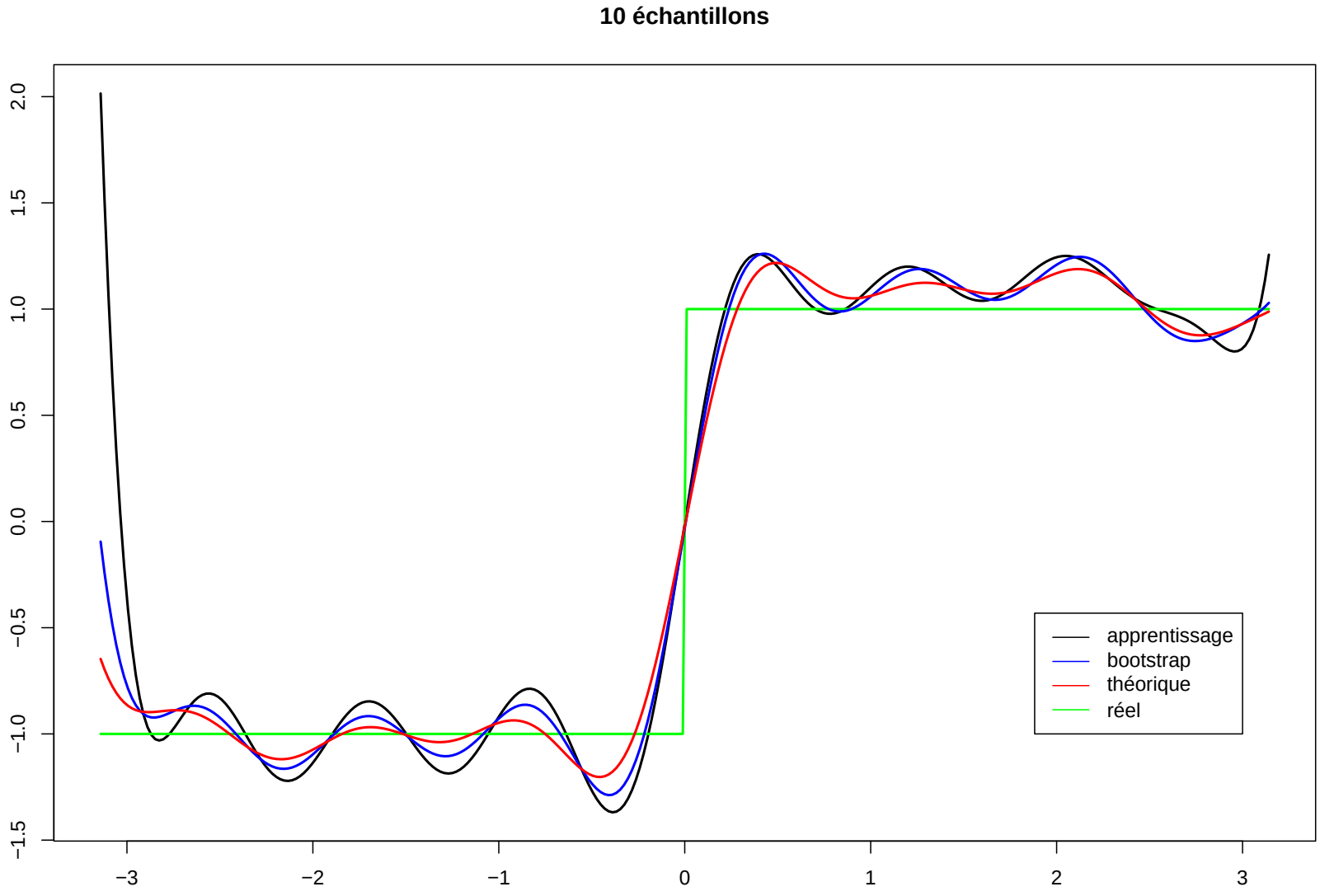
Intervalle de confiance



Intervalle de confiance

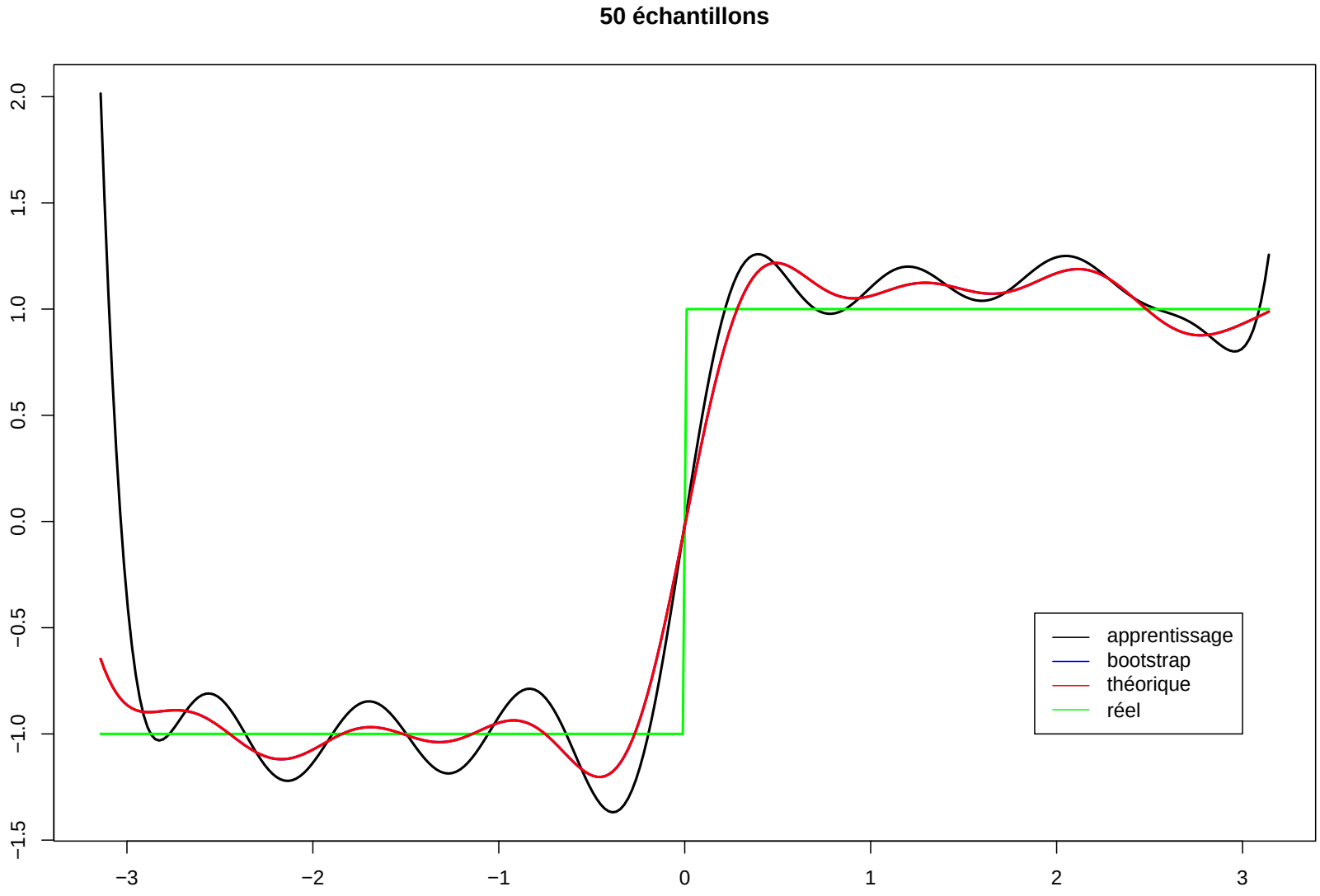


Sélection de modèle



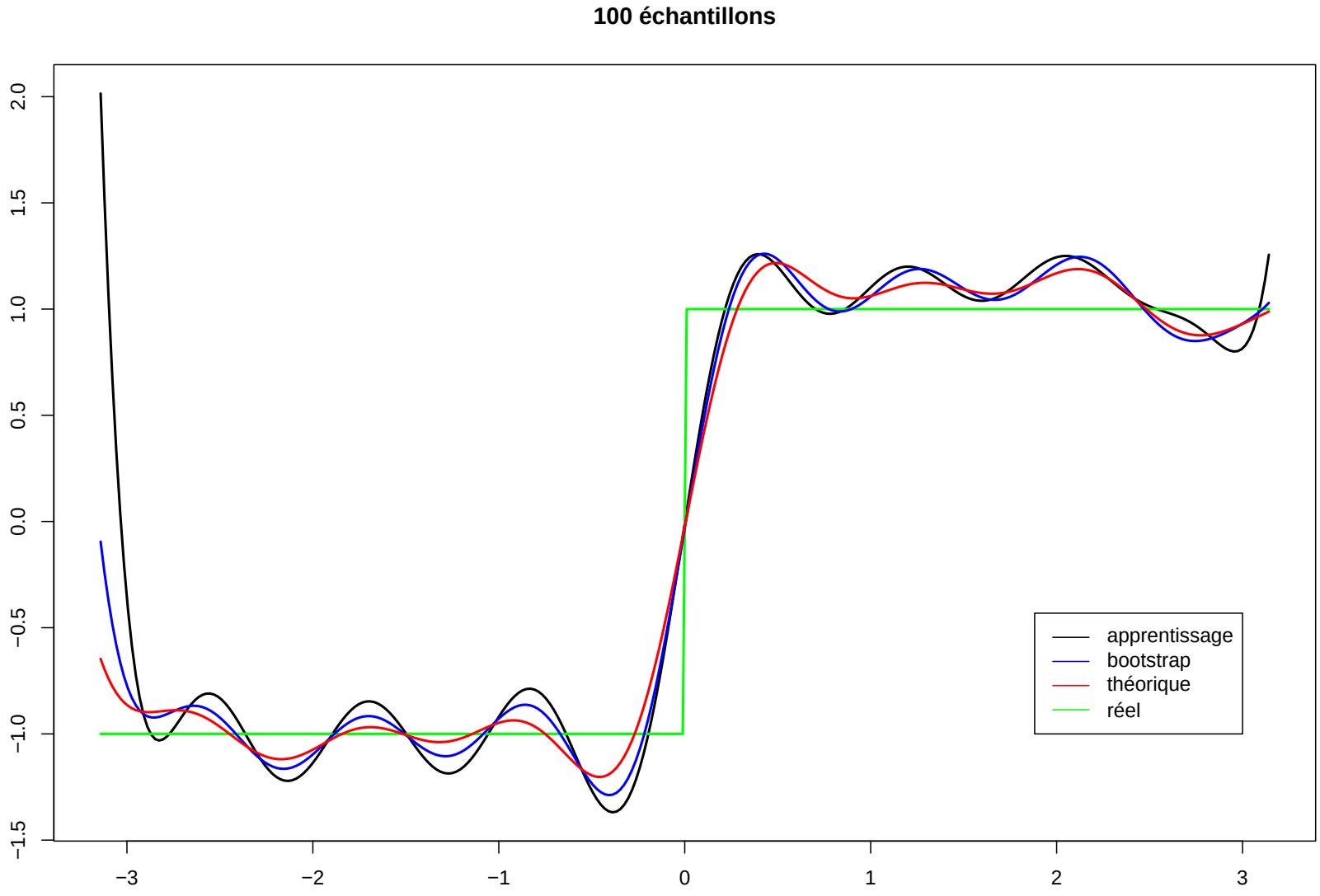
Erreur quadratique moyenne réelle $\simeq 0.045$

Sélection de modèle



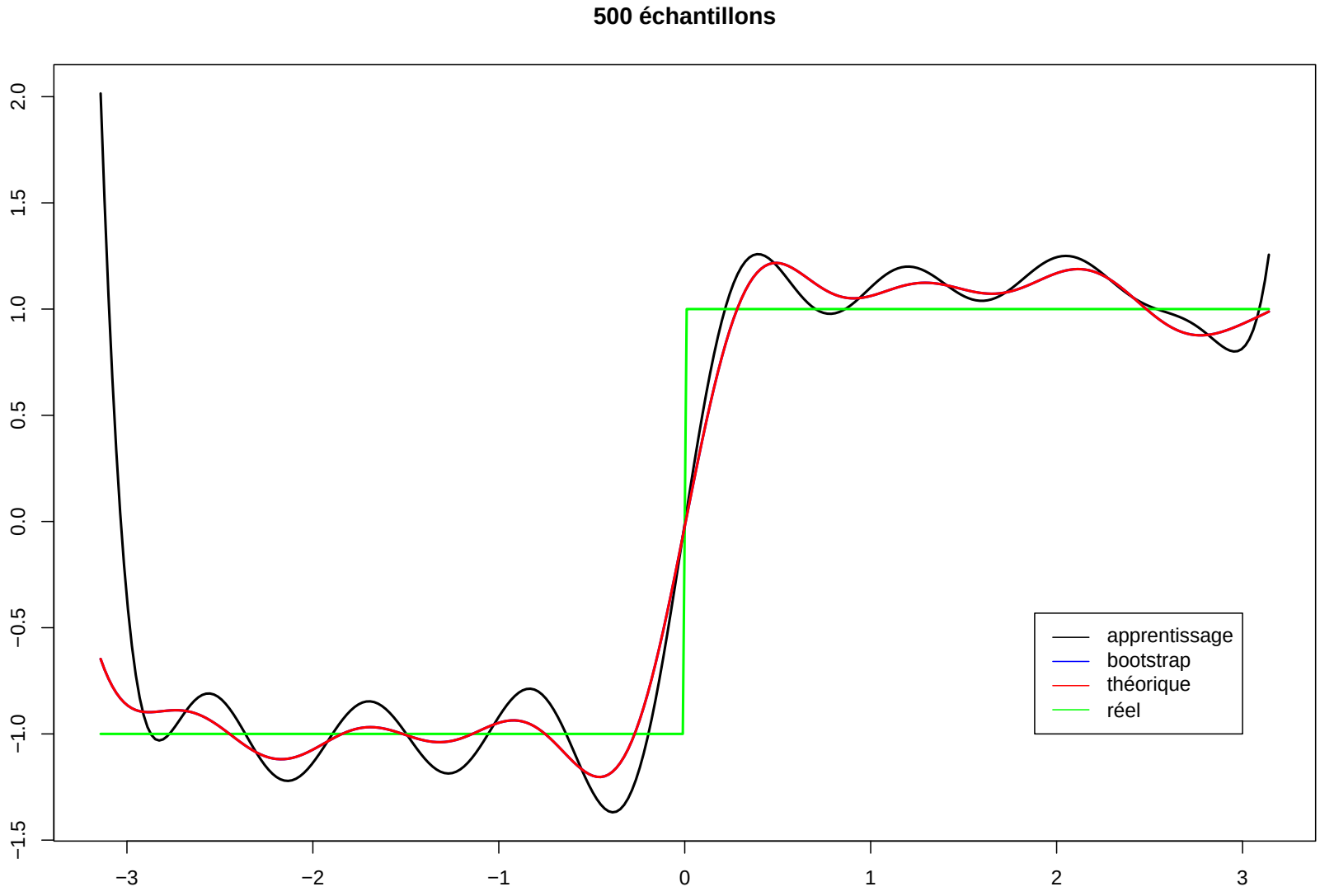
Erreur quadratique moyenne réelle $\simeq 0.036$

Sélection de modèle



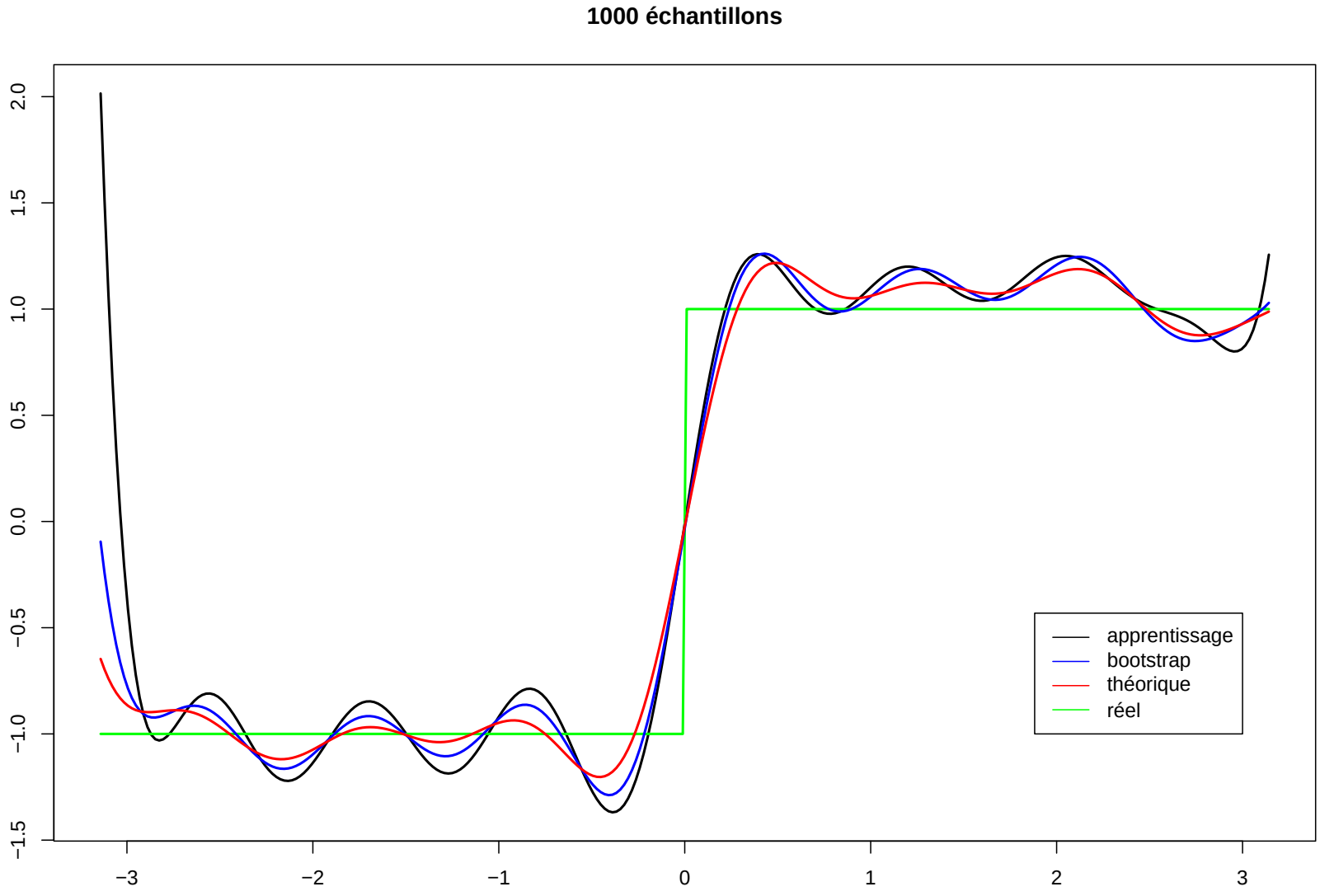
Erreur quadratique moyenne réelle $\simeq 0.045$

Sélection de modèle



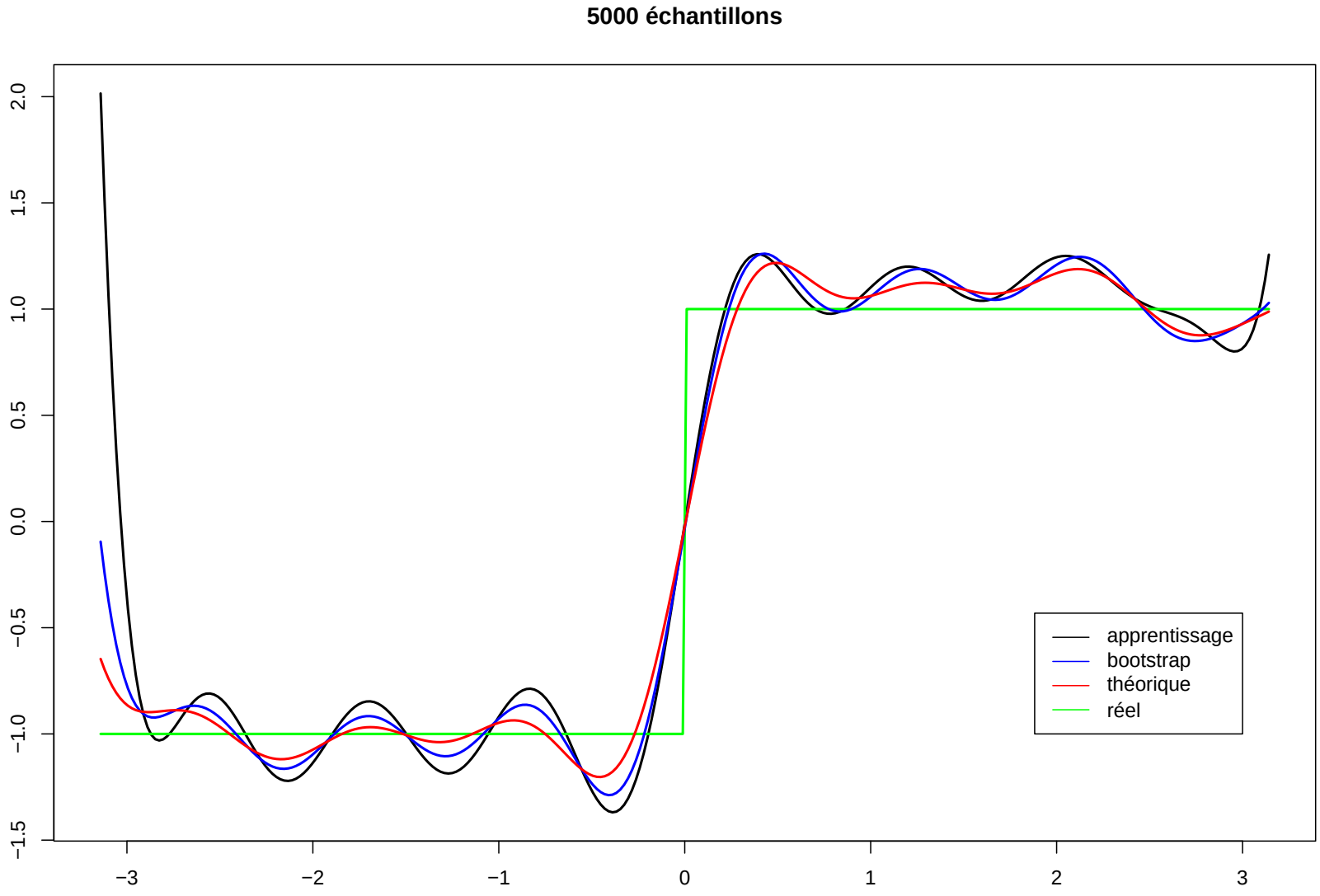
Erreur quadratique moyenne réelle $\simeq 0.036$

Sélection de modèle



Erreur quadratique moyenne réelle $\simeq 0.045$

Sélection de modèle



Erreur quadratique moyenne réelle $\simeq 0.045$

Variantes

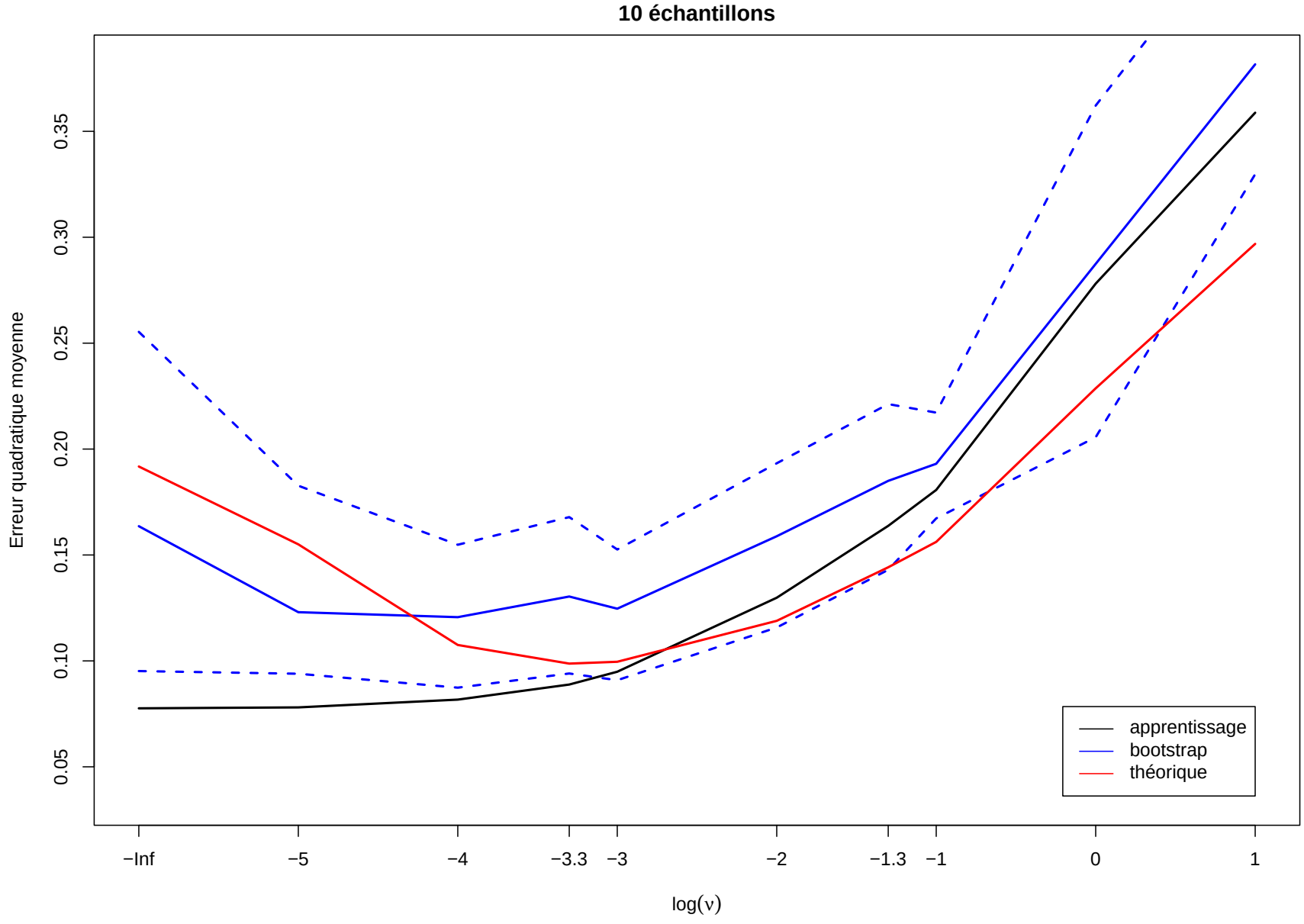
Estimation **directe** de l'erreur du modèle optimal

- moyenne empirique de l'erreur commise sur l'ensemble d'apprentissage par le modèle construit sur l'échantillon *bootstrap* ($\widehat{\mathcal{E}}_B$)
- moyenne empirique de l'erreur commise sur le complémentaire de l'échantillon *bootstrap* par le modèle construit sur l'échantillon (*bootstrap out-of-bag*, $\widehat{\mathcal{E}}_{oob}$)
- *bootstrap 632* : combinaison de l'estimation *out-of-bag* et de l'estimation naïve (sur l'ensemble d'apprentissage)

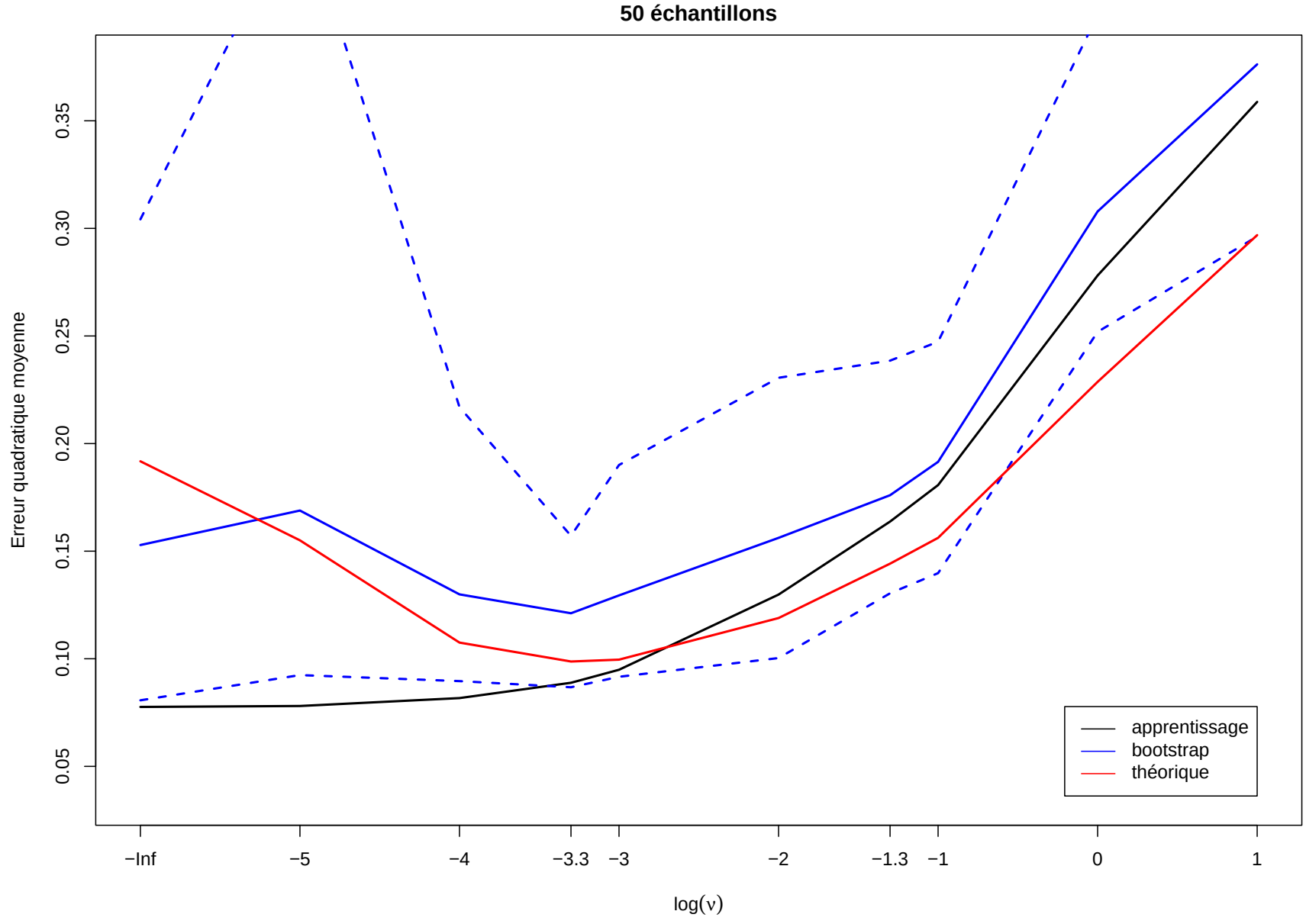
$$\widehat{\mathcal{E}}_{632} = 0.632\,\widehat{\mathcal{E}}_{oob} + 0.368\,\widehat{\mathcal{E}}$$

Probabilité qu'une observation de l'ensemble d'apprentissage soit dans un échantillon *bootstrap* : 0.632

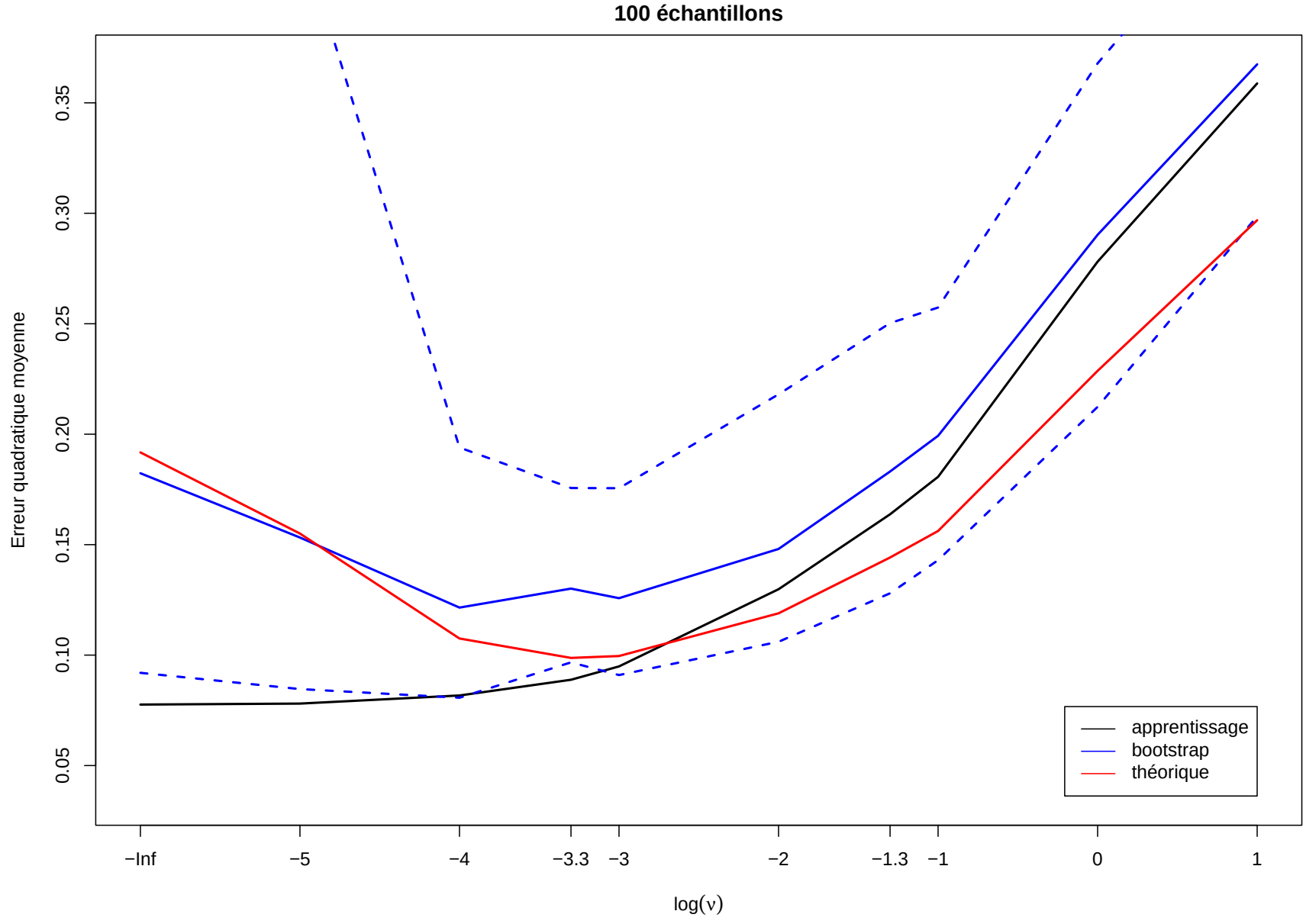
Intervalle de confiance (*Bootstrap* 632)



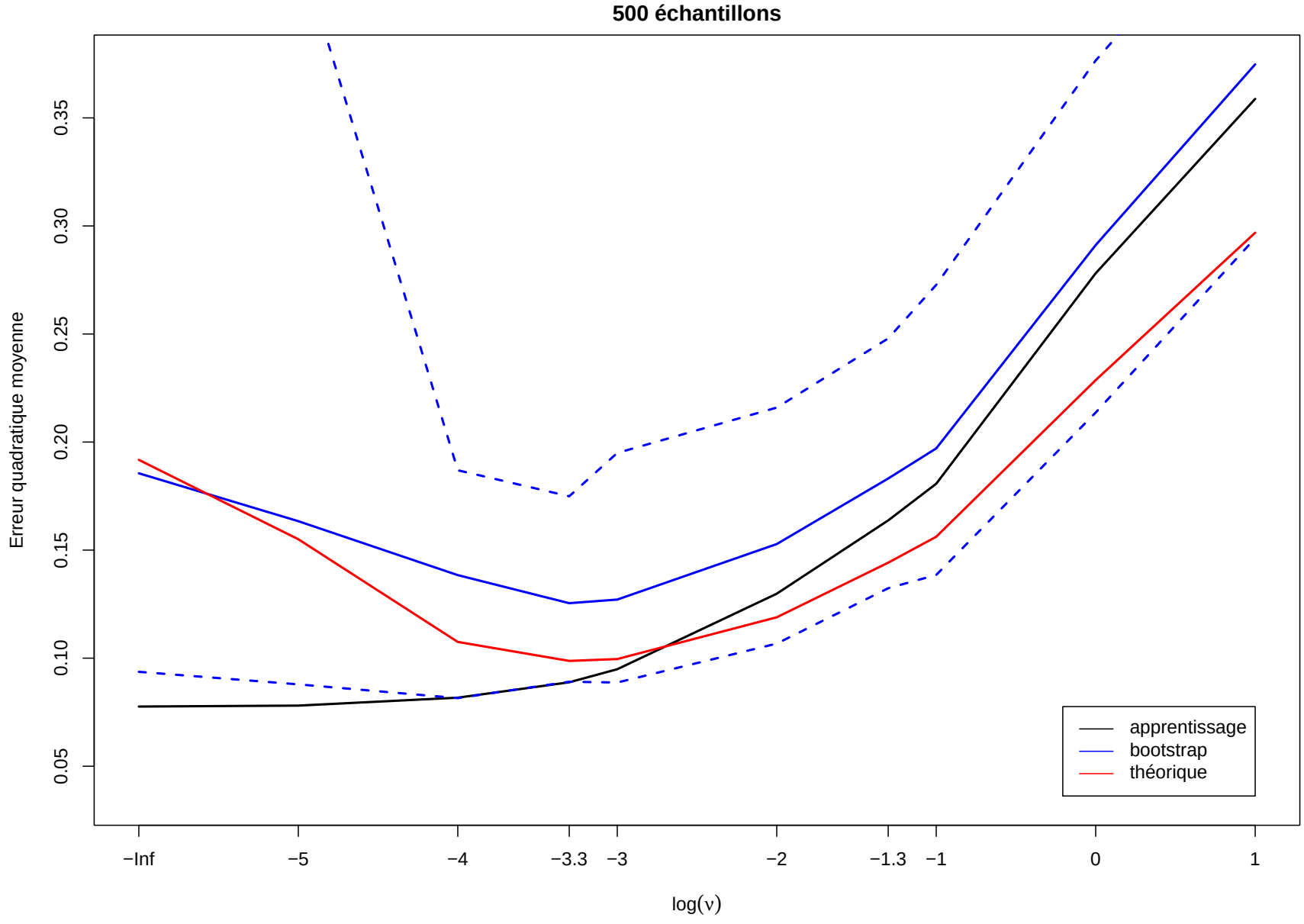
Intervalle de confiance (*Bootstrap* 632)



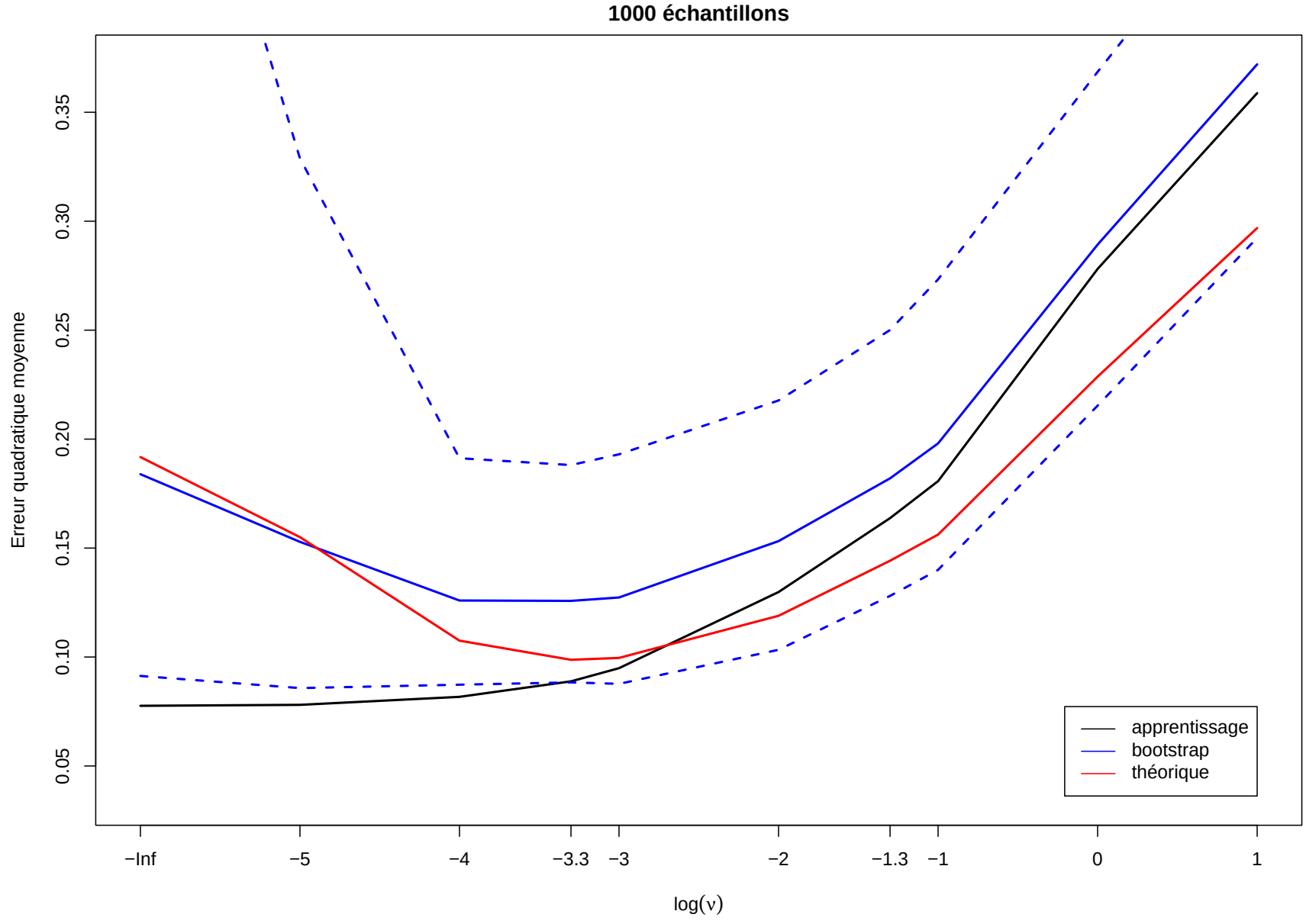
Intervalle de confiance (*Bootstrap* 632)



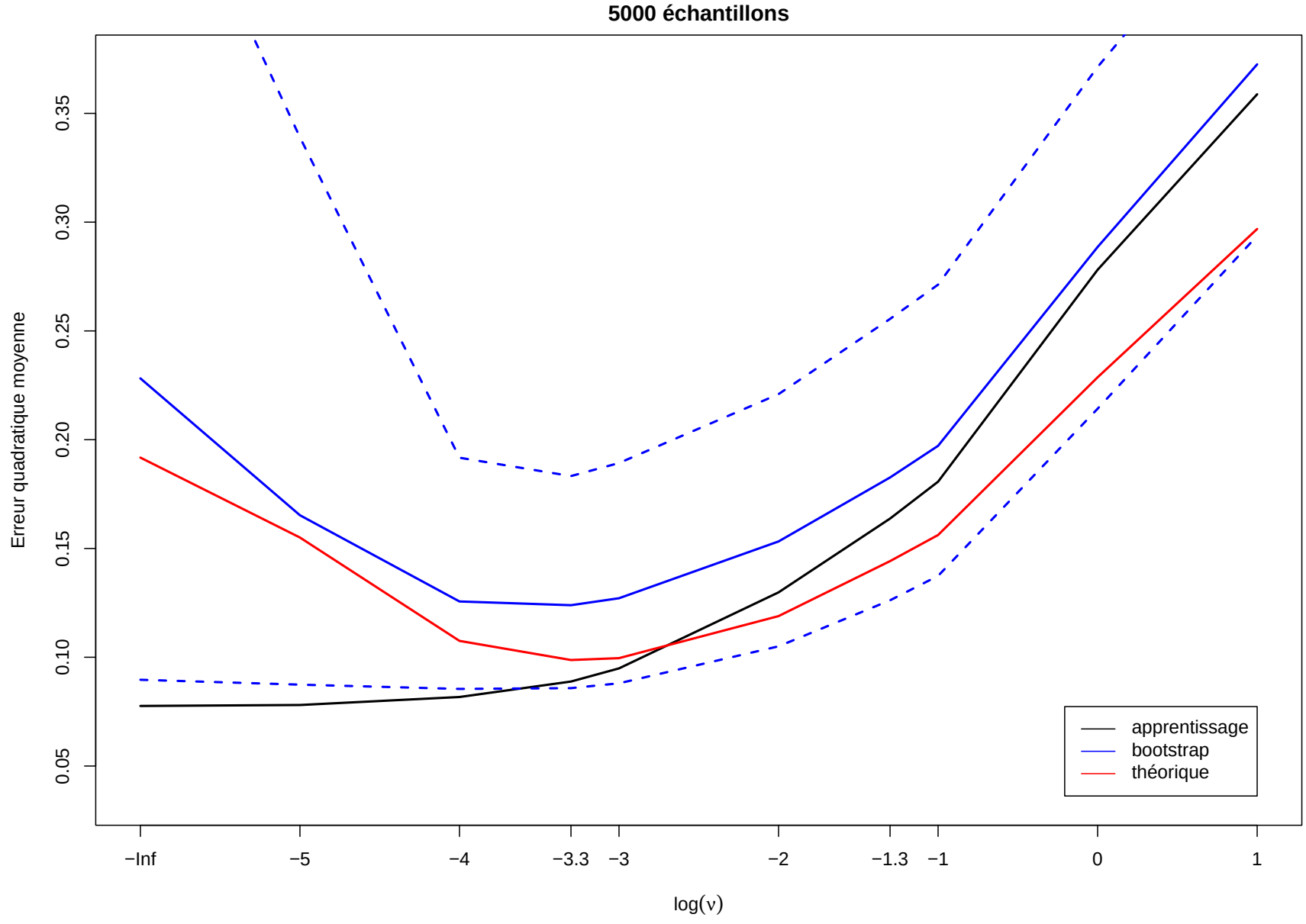
Intervalle de confiance (*Bootstrap* 632)



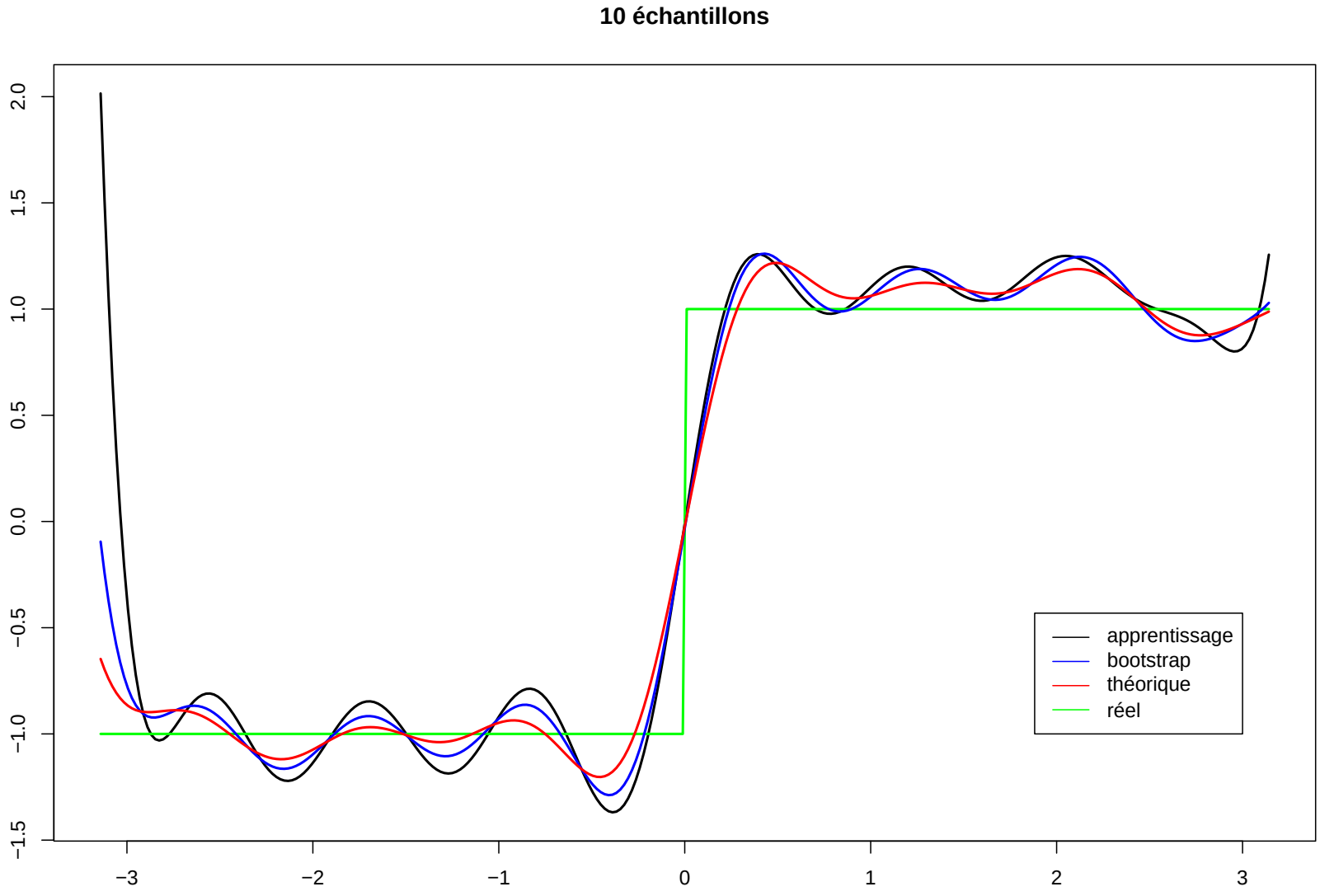
Intervalle de confiance (*Bootstrap* 632)



Intervalle de confiance (*Bootstrap* 632)

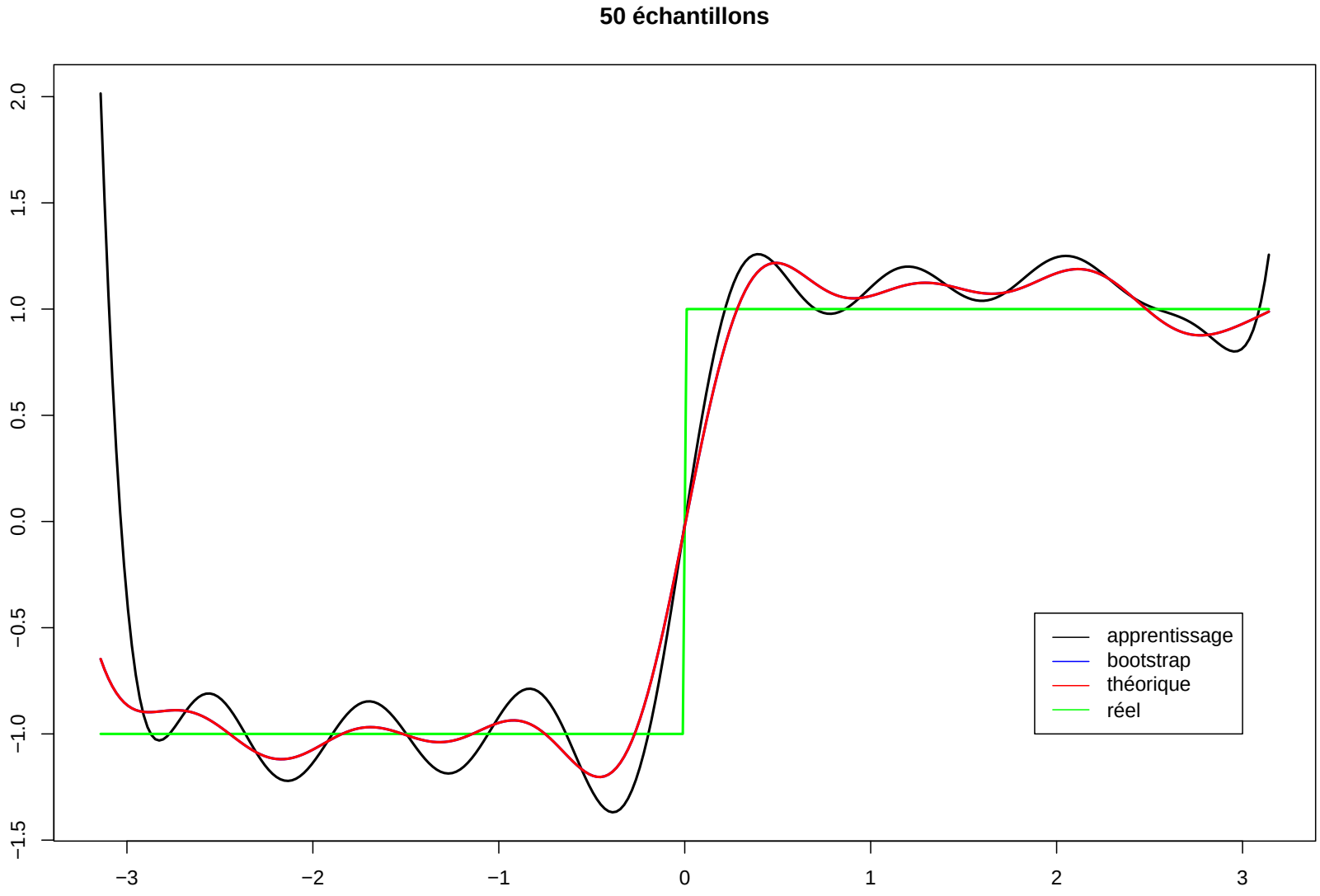


Sélection de modèle (*Bootstrap* 632)



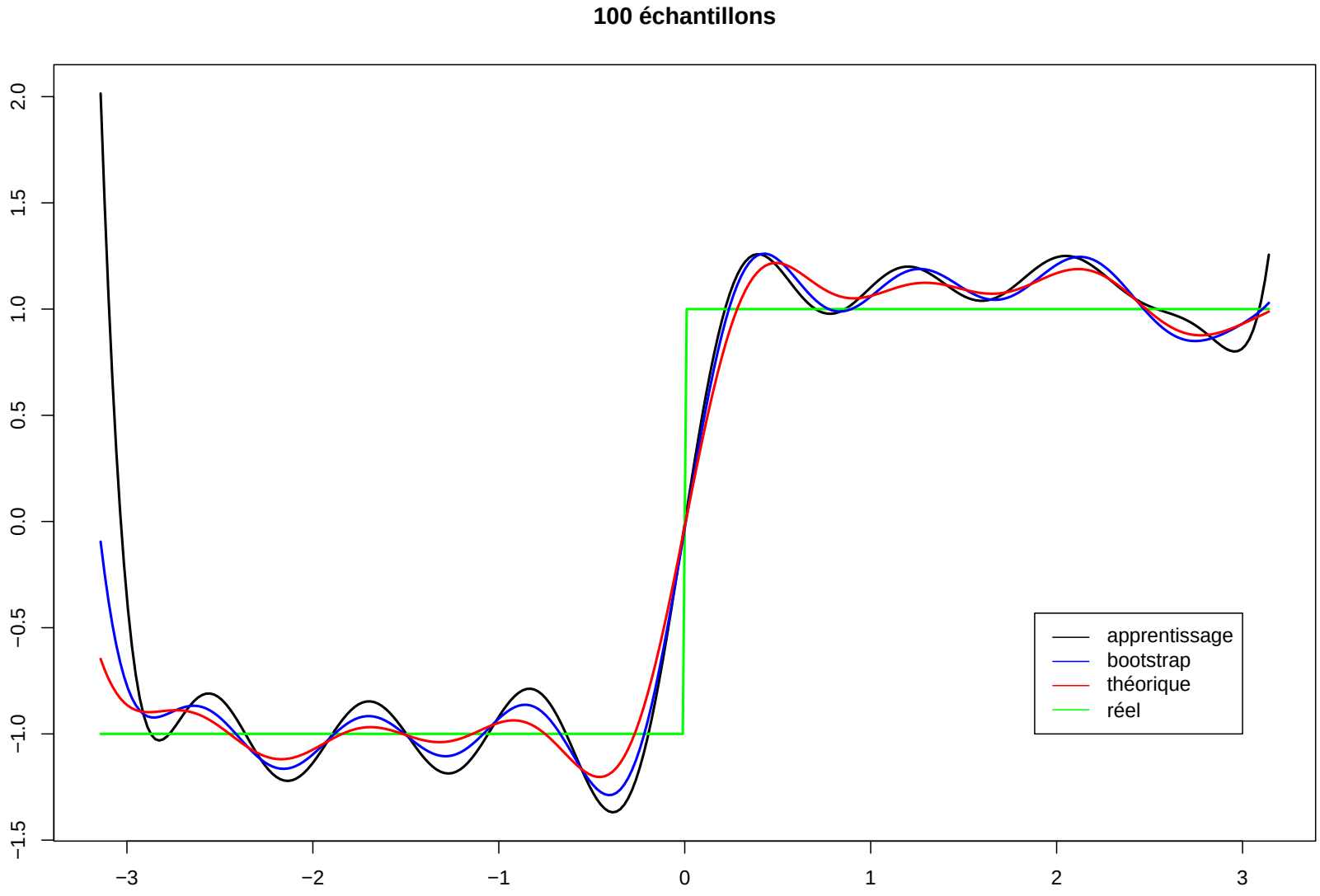
Erreur quadratique moyenne réelle $\simeq 0.045$

Sélection de modèle (*Bootstrap* 632)



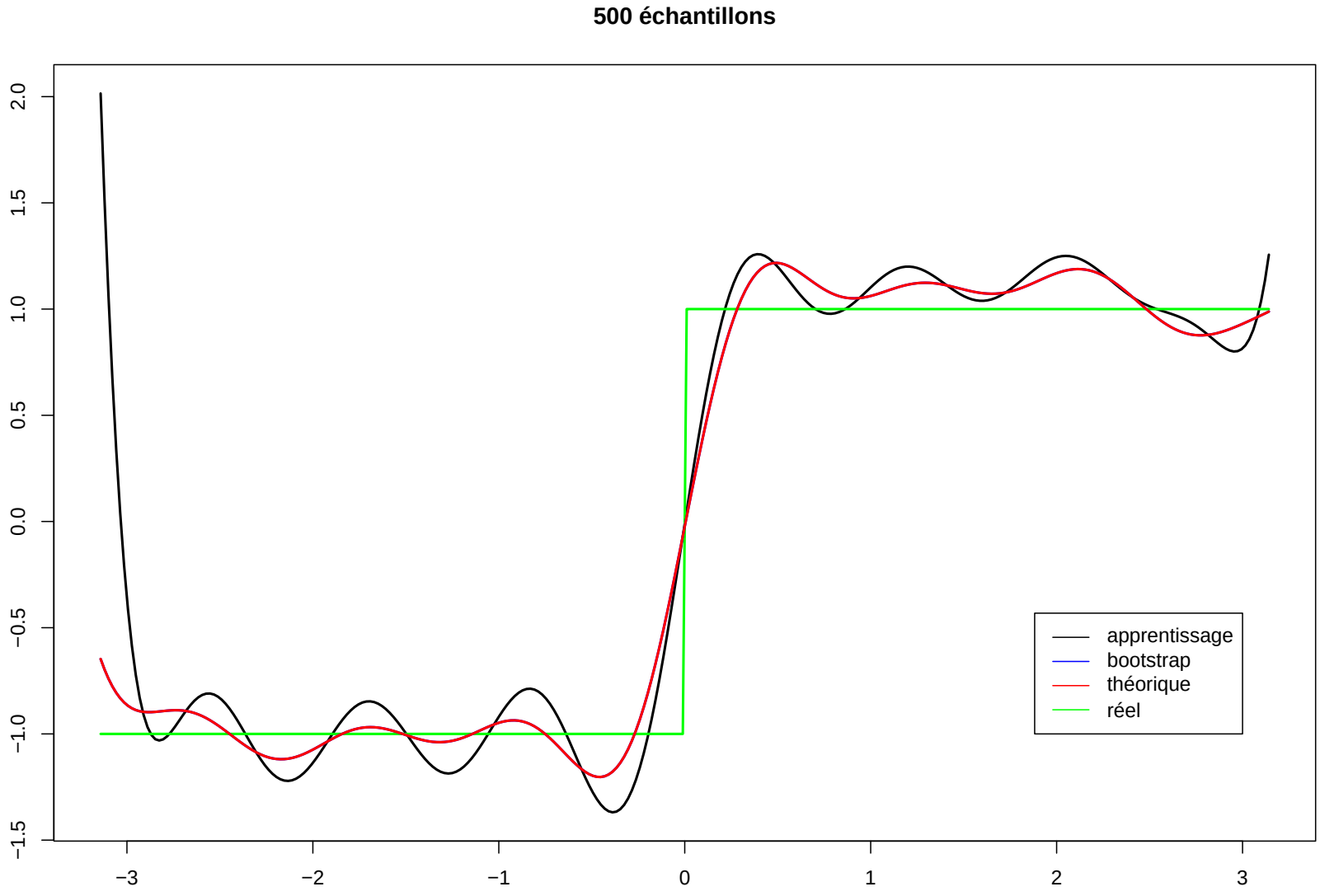
Erreur quadratique moyenne réelle $\simeq 0.036$

Sélection de modèle (*Bootstrap* 632)



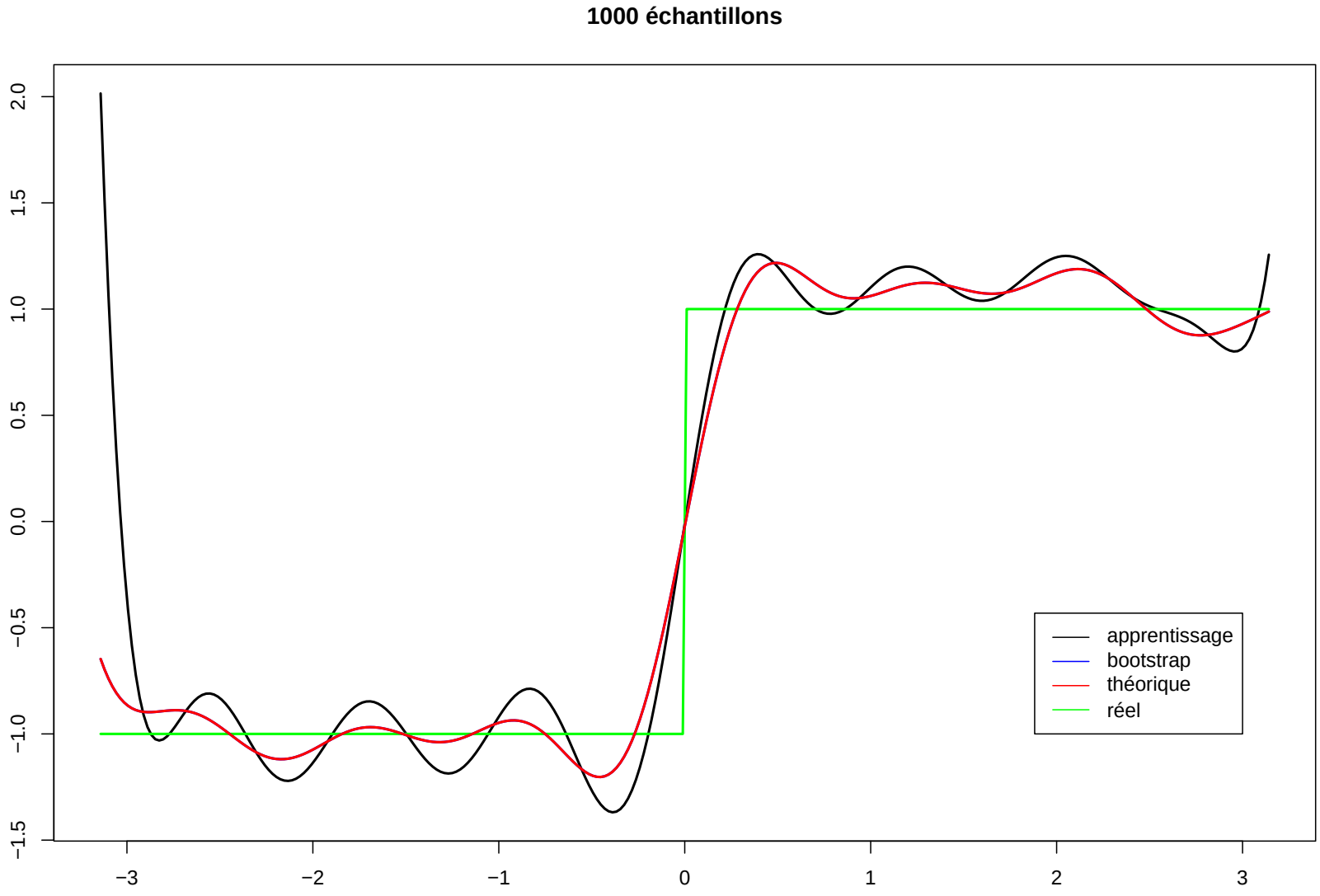
Erreur quadratique moyenne réelle $\simeq 0.045$

Sélection de modèle (*Bootstrap* 632)



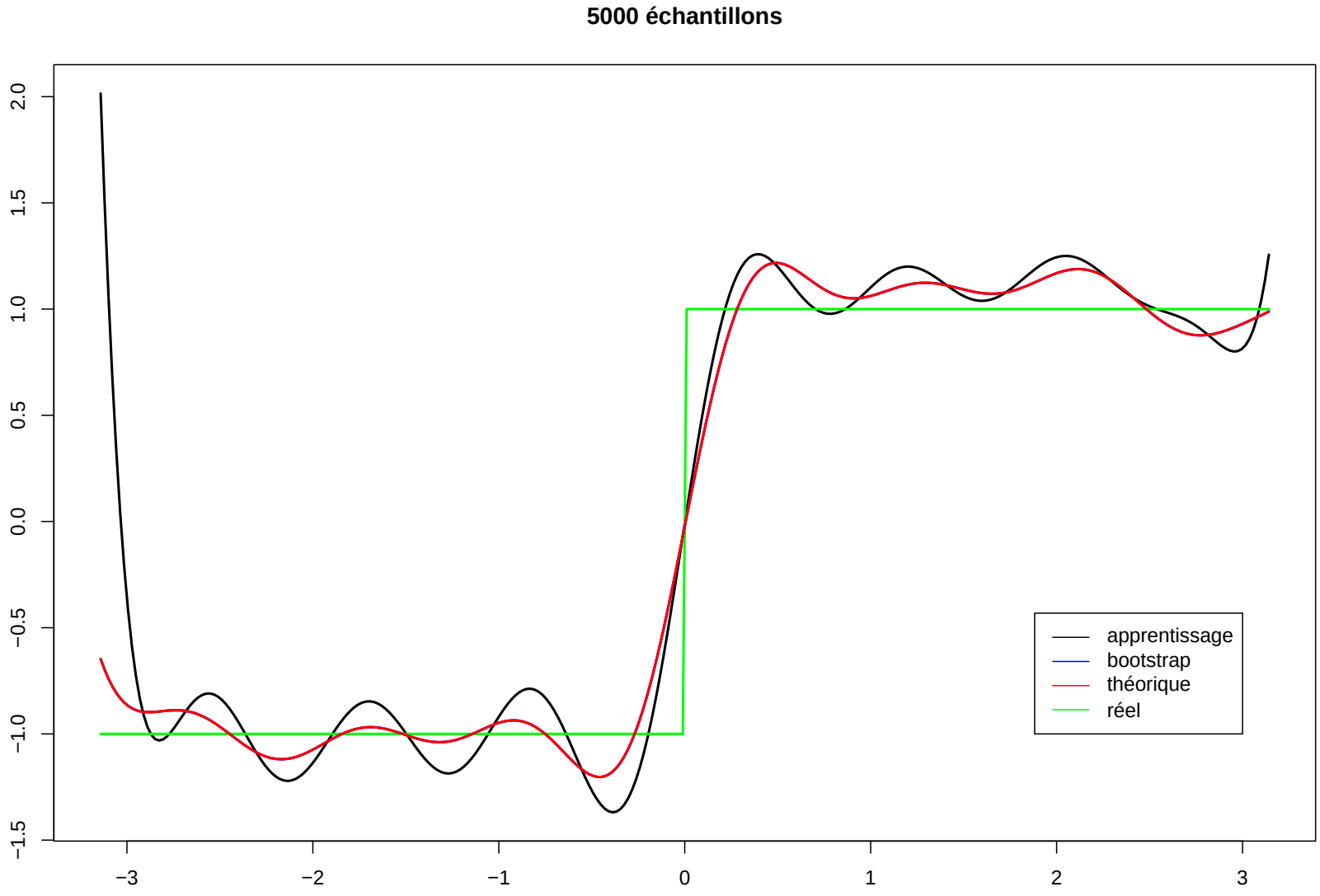
Erreur quadratique moyenne réelle $\simeq 0.036$

Sélection de modèle (*Bootstrap* 632)



Erreur quadratique moyenne réelle $\simeq 0.045$

Sélection de modèle (*Bootstrap* 632)



Erreur quadratique moyenne réelle $\simeq 0.036$

Critique du *Bootstrap*

Points positifs :

- facile à mettre en œuvre
- utilise toutes les données
- donne des intervalles de confiance

Points négatifs :

- temps de calcul très élevé
- nombreuses variantes

Remarques importantes :

- résultats théoriques de convergence
- asymptotiquement, pas de différence avec la validation croisée

Conclusion sur le ré-échantillonnage

- en pratique la validation croisée fonctionne de façon satisfaisante
- le *bootstrap* donne en plus des intervalles de confiance (meilleurs que Hoeffding par exemple) mais très coûteux
- une règle d'or : comparer ce qui est comparable !
  - ne pas comparer une estimation *bootstrap* avec une estimation par validation croisée
  - utiliser toujours le même découpage pour la validation croisée
  - utiliser les mêmes échantillons *bootstrap*

Contrôle de complexité

L'idée de base est d'étudier une combinaison :

$$\mathcal{E} + \mathcal{C}$$

et de prendre le modèle qui minimise cette combinaison. $\mathcal{E}$ désigne l'erreur de modélisation obtenue, alors que $\mathcal{C}$ mesure la complexité effective du modèle. Par exemple le critère de Mallows est donné par :

$$E + 2\frac{W}{N}\sigma^2$$

où E désigne l'erreur quadratique moyenne sur l'ensemble d'apprentissage, W le nombre de paramètres du modèle linéaire, N le nombre de données et σ^2 une estimation de la variance du bruit.

Critères de Mallows et AIC

Difficultés :

- n'est justifié (théoriquement) que pour le cas linéaire et l'erreur quadratique
- ne s'applique donc que pour la sélection de variables (choix des variables explicatives importantes)
- demande une estimation correcte de σ^2 : il faut donc utiliser un modèle avec un faible biais $\Rightarrow$ contradictoire avec le cas linéaire

Version plus générale, le critère d'information d'Akaïke (AIC) :

$$-2\mathcal{L} + 2W$$

où $\mathcal{L}$ désigne la log-vraisemblance. Coïncide avec le critère de Mallows quand on utilise un modèle d'erreur gaussienne.

Critère BIC

- BIC : Bayesian Information Criterion
- Même esprit que AIC, on compare les modèles selon :

$$-2\mathcal{L} + 2W \log(N)$$

- asymptotiquement exact : sélectionne ne bon modèle quand N tend vers l'infini
- pénalisation lourde des modèles complexes : sélectionne des modèles simples
- en cas de bruit gaussien, équivalent à

$$E + \log(N)\frac{W}{N}\sigma^2$$

Cas non linéaire

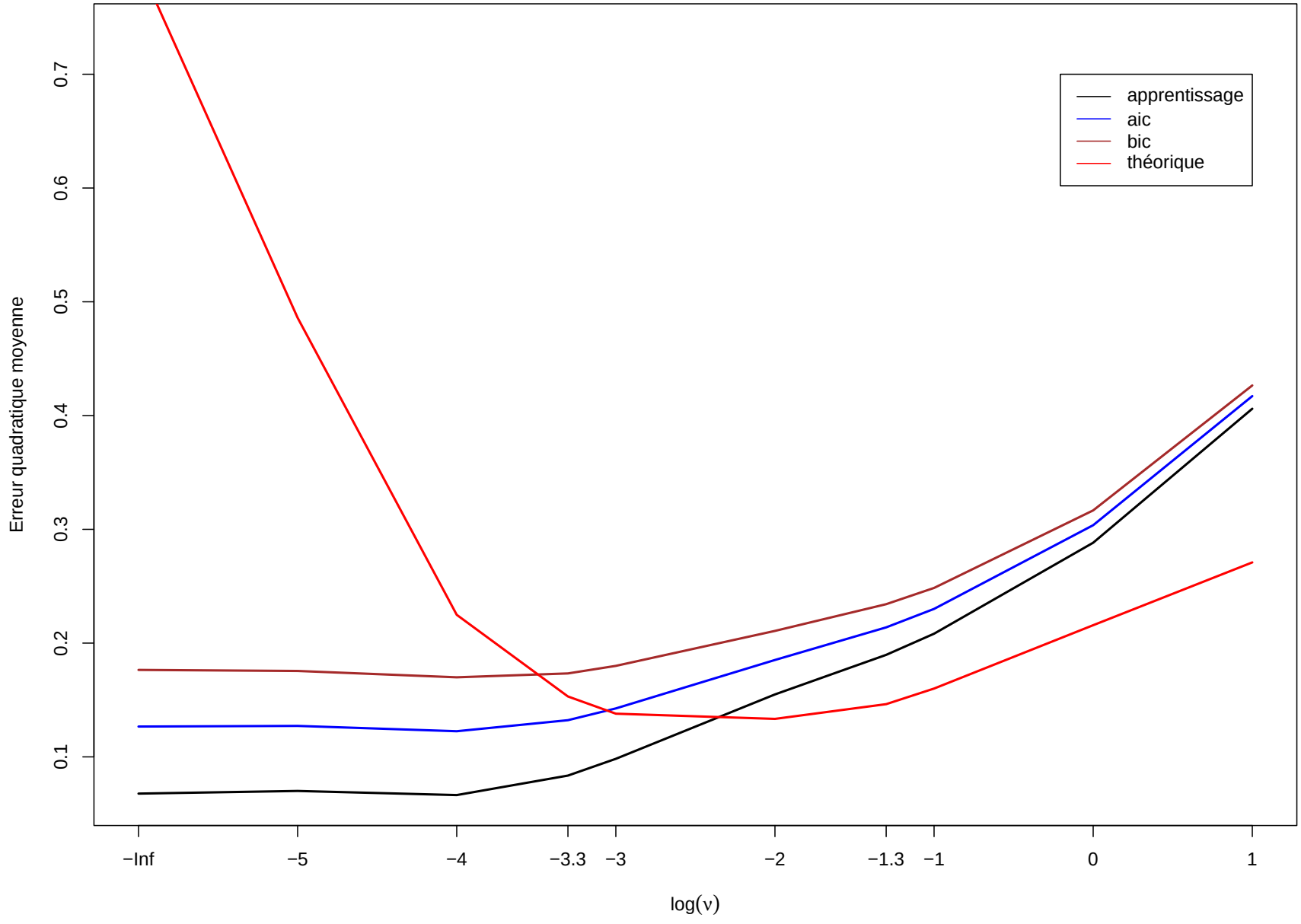
Il faut remplacer W par une mesure de complexité du modèle :

- modèle linéaire généralisé :
  - base de ϕ_i
  - W : nombre de ϕ_i linéairement indépendantes utilisées (en tenant compte du terme constant)
- prise en compte de la régularisation :
  - quand on régularise, les prédictions associées aux observations s'écrivent :

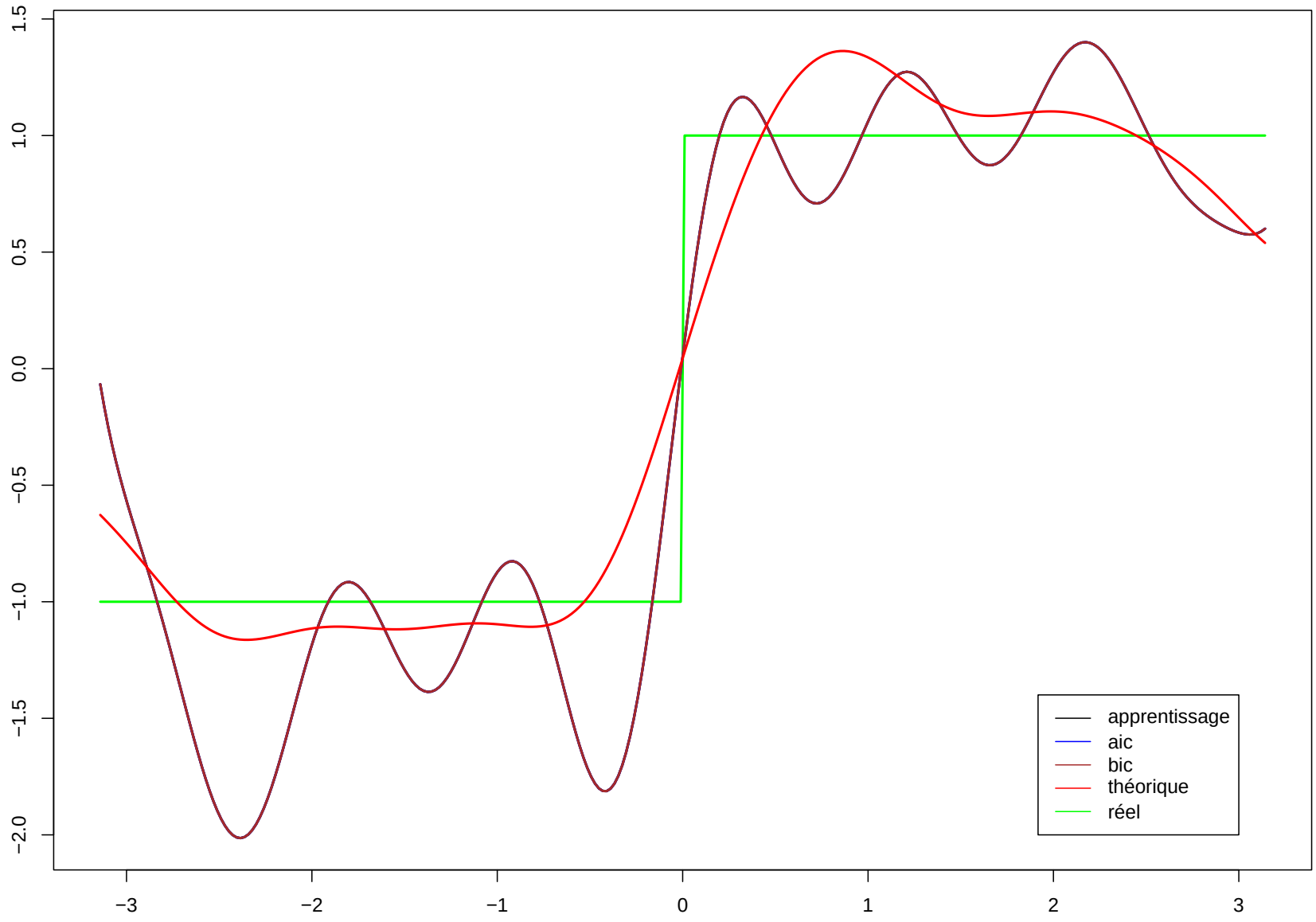
$$Y Z^T (Z Z^T + \nu P)^{-1^T} Z$$

  - W est alors obtenu comme la trace de $Z^T (Z Z^T + \nu P)^{-1} Z$

Exemple (Créneau) : erreur en fonction de ν



Exemple (Créneau) : sélection de modèle



Critique du contrôle de complexité

Points positifs :

- relativement facile à mettre en œuvre
- utilise toutes les données
- temps de calcul additionnel négligeable
- le BIC sélectionne asymptotiquement le meilleur modèle

Points négatifs :

- AIC sélectionne des modèles trop complexes avec N grand
- BIC sélectionne des modèles trop simples avec N petit
- comportement parfois décevant à distance finie (i.e., quand N est “raisonnable”)
- il faut estimer le bruit

Théorie de Vapnik-Chervonenkis

- cadre : la **discrimination**
- données : entrée $x \in X$, sortie $y \in \{0, 1\}$ (problème à deux classes)
- description statistique : P une probabilité sur $X \times \{0, 1\}$
- H : ensemble des modèles considérés, des fonctions de X dans $\{0, 1\}$
- erreur commise par $h \in H$: $E(h) = P(\{(x, y) \mid h(x) \neq y\})$
- on cherche $h \in H$ qui minimise $E(h)$ (!)
- échantillon : $z = \big((x^1, y^1), \ldots, (x^k, y^k)\big)$
- erreur sur un échantillon : $\widehat{E}(h, z) = \frac{1}{k} \left|\{i \mid y_i \neq h(x_i)\}\right|$
- but de la théorie de VC, majorer

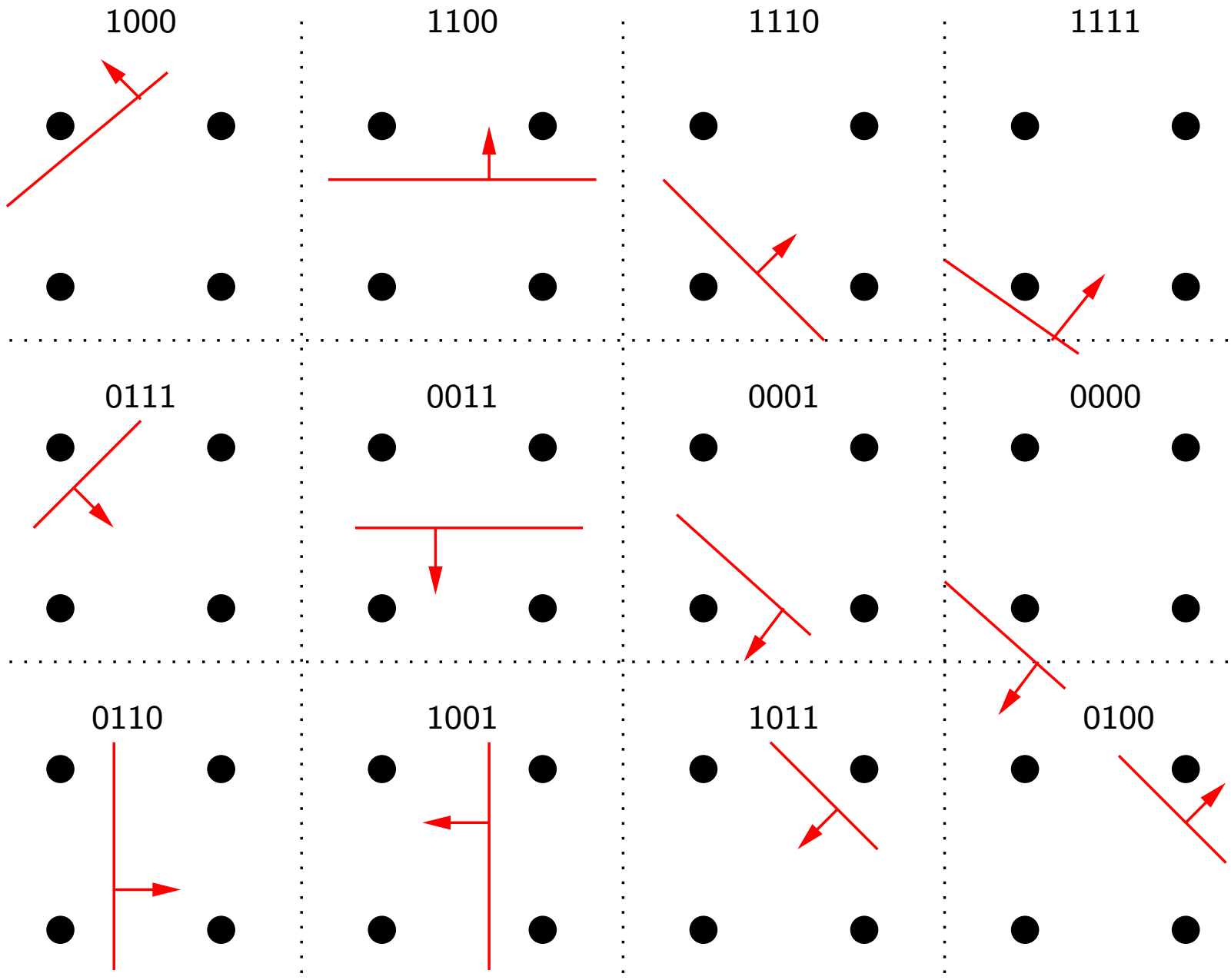
$$P(z,\ \sup_{h \in H} |\widehat{E}(h, z) - E(h)| > \epsilon)$$

Dichotomie

On considère un ensemble $S = \{x^1, \ldots, x^k\} \subset X$:

- une dichotomie de S est une fonction de S dans $\{0, 1\}$
- une dichotomie coupe S en deux classes
- $H_{|S} = \{h(x^1), \ldots, h(x^k) \mid h \in H\} \subset \{0, 1\}^k$
- si $\left|H_{|S}\right| = 2^k$, H réalise toutes les dichotomies de S, i.e. pour toute partition $S = S_0 \cup S_1$, il existe $h \in H$ tel que $h(x) = 1 \Leftrightarrow x \in S_1$
- $G_H(k) = \max\left\{\left|H_{|S}\right| \mid S \subset X,\ |S| = k\right\}$: la fonction de croissance de H (*growth function*)
- Exemple, H : les modèles linéaires sur $\mathbb{R}^2$
  - pour $k \in 1, 2, 3$, on peut réaliser toutes les dichotomies
  - à partir de $k = 4$, ça ne marche plus ! (exemple du XOR)
  - on montre que pour $k > 3$, $G_H(k) = 4(k - 1)$

Exemple



Dimension de Vapnik-Chervonenkis

Une façon de résumer la fonction de croissance, définie par :

$$\dim_{VC}(H) = \max\{k \mid G_H(k) = 2^k\}$$

C'est une mesure de la **capacité** de H :

- quand $G_H(k) = 2^k$:
  - H peut séparer arbitrairement tous les ensembles de taille k : apprentissage par cœur
  - si $|z| = k$, $\min_{h \in H} \widehat{E}(h, z) = 0$
  - les données n'apportent pas grand chose pour choisir h
- quand $G_H(k) < 2^k$, H est saturé, on ne peut plus apprendre par cœur
- exemple : modèle linéaire dans $\mathbb{R}^n$, $\dim_{VC} = n + 1$.
- on peut avoir $\dim_{VC} = \infty$

Théorèmes principaux

- si $\dim_{VC}(H) = d$ et $k > d$

$$G_H(k) \leq \left(\frac{ek}{d}\right)^d$$

- on a

$$P(z,\ |z| = k,\ \sup_{h \in H} |\widehat{E}(h,z) - E(h)| > \epsilon) \leq 4 G_H(2k) e^{-\frac{\epsilon^2 k}{8}}$$

- et donc quand $\dim_{VC}(H) = d$ et $k > d$

$$P\left(z,\ |z| = k,\ \sup_{h \in H} |\widehat{E}(h,z) - E(h)| < \sqrt{\frac{8}{k}\left(d \ln \frac{2ke}{d} + \ln \frac{4}{\eta}\right)}\right) \geq 1 - \eta$$

$\Rightarrow$ Intervalle de confiance

Critique de la théorie VC

Points positifs :

- l'une des théories les plus avancées pour l'apprentissage
- donne un intervalle de confiance dans le cas le pire
- aucune hypothèse sur la distribution des données

Points négatifs :

- dimension VC très difficile à calculer
- bornes très pessimistes (à cause de l'absence d'hypothèses sur les données)
- utilisation pratique difficile

Résumé et conclusion

Pas de solution miracle !

Méthode	Inconvénient
Découpage	Beaucoup de données
Ré-échantillonnage	Temps de calcul
Complexité	Fortes hypothèses
Vapnik	Pessimiste

En pratique :

- si le temps de calcul est acceptable : ré-échantillonnage
- sinon contrôle de complexité
- sinon découpage

Ne **jamais** se passer d'une méthode connue pour l'évaluation

et/ou la sélection de modèle