Régression non-paramétrique

Philippe Real

16/09/2019

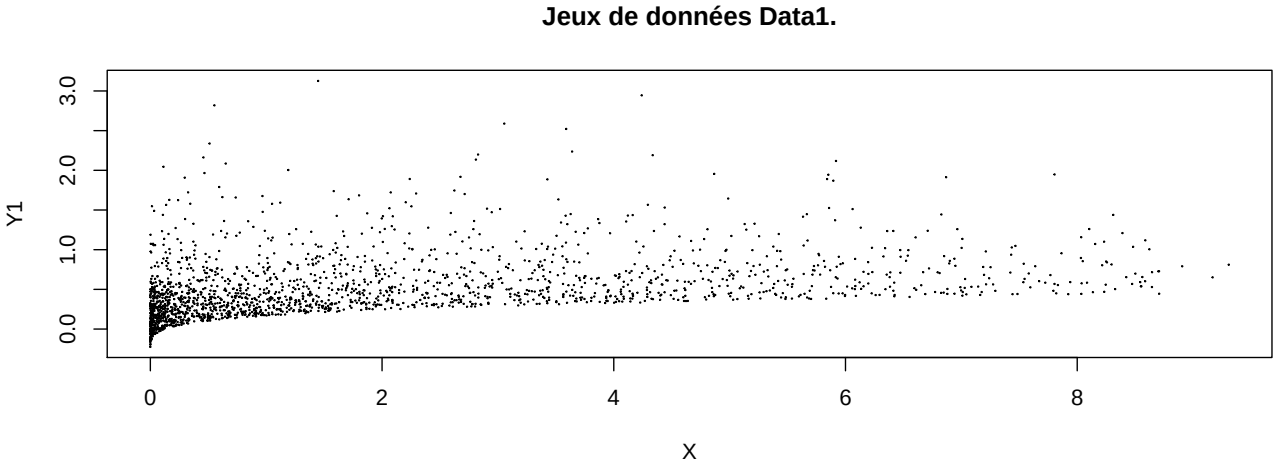
Contents

1 Etude de la densité g des X

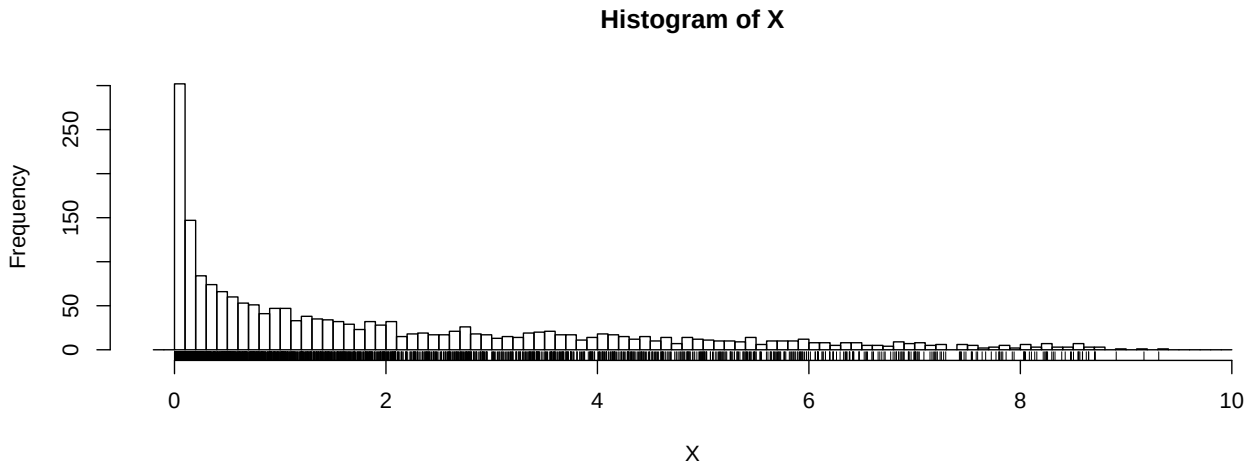
On utilise les données Data1. Jeux de données (X_i, Y_i) i=1,..,2000 Avec la représentation suivante $Y_i = r(X_i) + \sigma\xi_i(X_i)$ où (X_i, Y_i) sont i.i.d. et σ ne dépend pas de X (homoscédastique). - La loi de X_i admet une densité g(x) - Les ξ_i sont i.i.d avec $\mathbb{E}(\xi_i) = 0$ et $\mathbb{E}(\xi_i^2) = 1$ et ont une densité μ. - X_i et ξ_i sont indépendants.

1.1 Lecture des données et premières analyses

```
##       X.1              X                Y1          
##  Min.   :   1.0   Min.   :0.000005   Min.   :-0.2244  
##  1st Qu.: 500.8   1st Qu.:0.258074   1st Qu.: 0.2619  
##  Median :1000.5   Median :1.192414   Median : 0.4303  
##  Mean   :1000.5   Mean   :2.029447   Mean   : 0.5112  
##  3rd Qu.:1500.2   3rd Qu.:3.318174   3rd Qu.: 0.6735  
##  Max.   :2000.0   Max.   :9.308684   Max.   : 3.1263
```



Pour avoir une idée de la densité g de X on peut tracer son histogramme qui est un estimateur de la densité. Obtenu par la série d'approximation suivante: $g(x_0) \approx_{h\to 0} \frac{\mathbb{P}(X \in [x_0 - h, x_0 + h])}{2h} \approx_{n\to\infty} \frac{1}{2nh} \sum_{i=1}^{n} 1_{X_i \in [x_0-h, x_0+h]} := \hat{g}_{n,h}(x_0)$



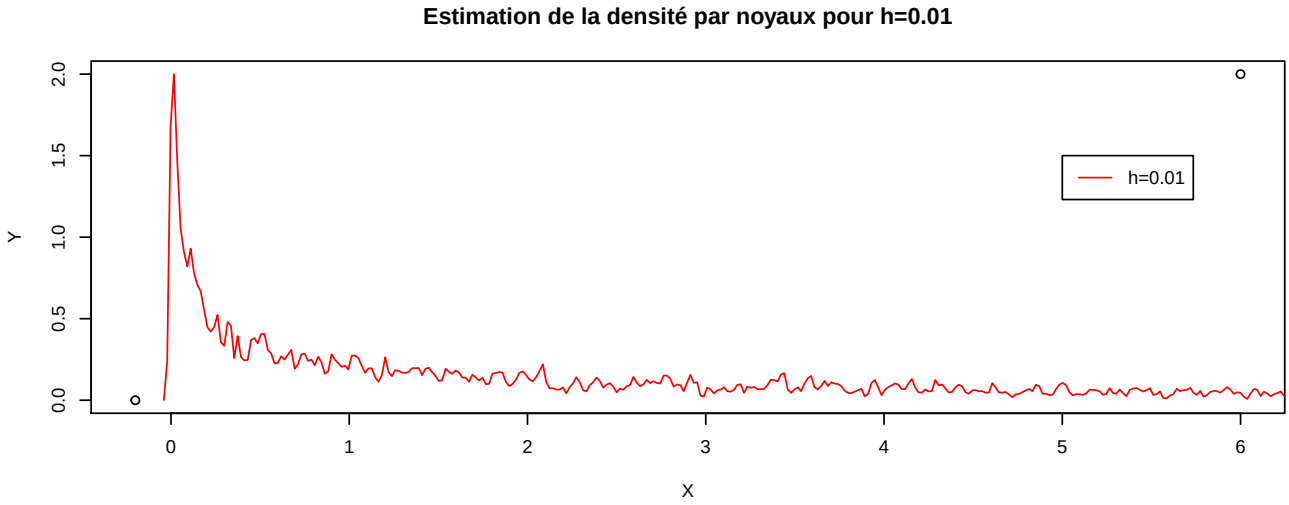
Cet estimateur a de bonnes propriétés statistiques mais a l'inconvénient de n'être jamais régulier.

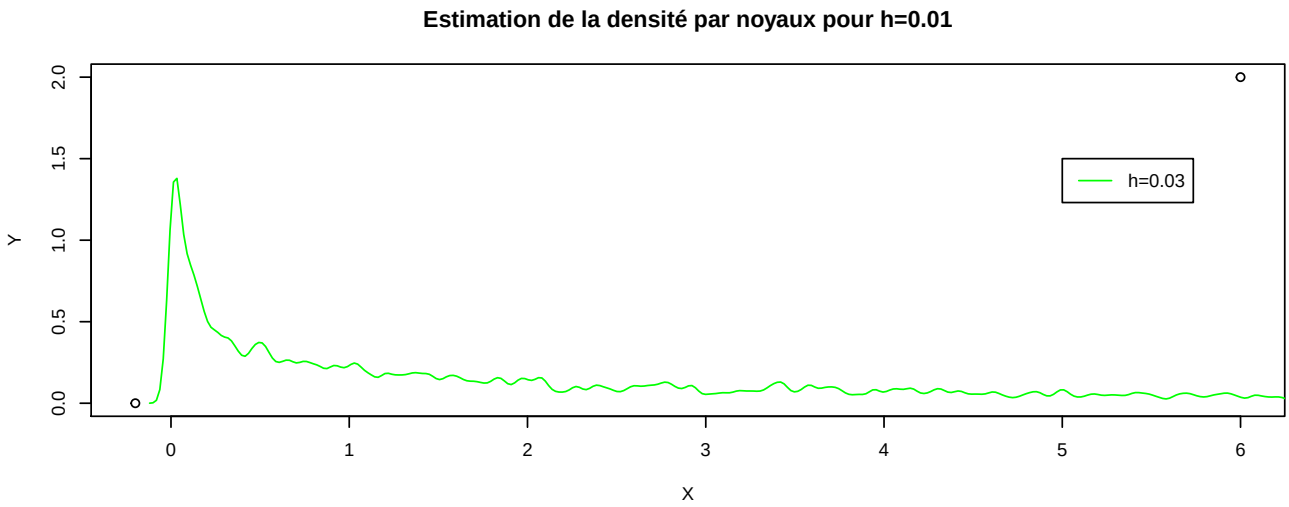
1.2 Estimateur par noyau de la densité g(x) de X

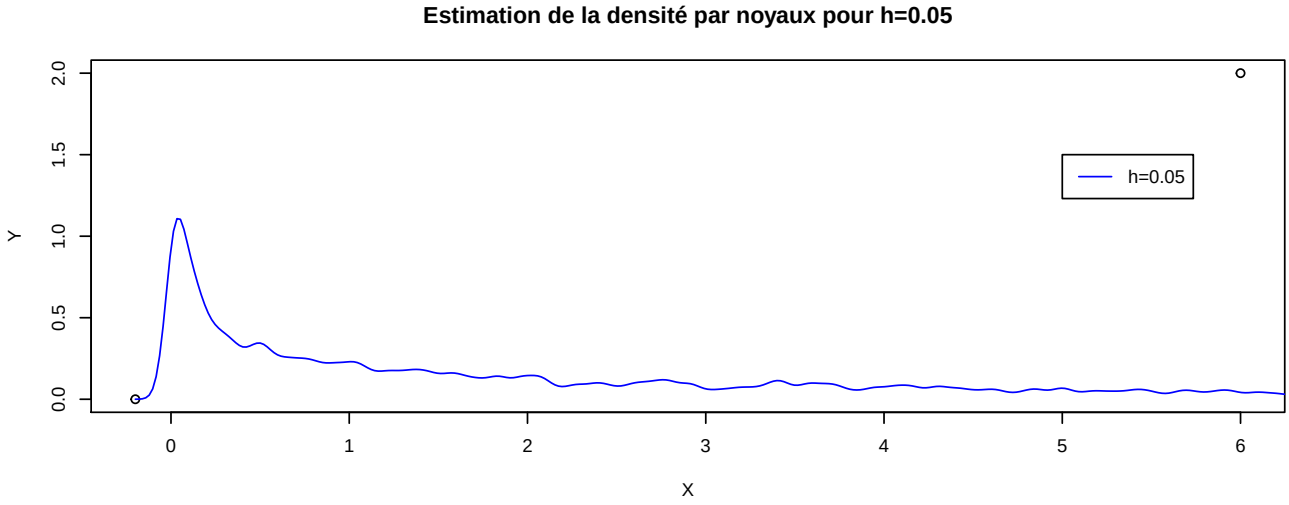
La méthode par noyau est une généralisation de la méthode d'estimation par histogramme. Elle permet de construire un estimateur avec de meilleur propriété de régularité notamment de continuité. On a un échantillon $(X_1, X_2, ...X_n)$ i.i.d, réalisation d'une variable aléatoire X dont on cherche à estimer la densité g. Un estimateur par noyaux est donné par $\hat{g}(x_0) = \frac{1}{nh}\sum_{i=1}^{n} K(\frac{x_0 - X_i}{h})$ K est le noyau régularisant que l'on prend gaussien $K(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2}$ L'estimateur est basé sur la série d'approximation suivante : $g(x_0) \approx_{h\to 0} E(\frac{1}{h}K(\frac{x_0 - X}{h})) \approx_{n\to\infty} \frac{1}{nh}\sum_{i=1}^{n} K(\frac{x_0 - X_i}{h}) := \hat{g}_{n,h}(x_0)$

Pour implémenter ces méthodes, dans le cas de l'estimation de la densité on va utiliser la fonction bkde du package KernSmooth. On prendra pour noyau le noyau normal.Ce choix peut sembler arbitraire, mais on a vu que ce n'est pas le choix du noyau qui est le plus important dans l'estimation de la densité. On calcul cette estimateur de la densité pour différentes largeur de fenêtre: h (bandwidth) Et on va déterminer de manière empirique une valeur de h qui semble adapté.

1.2.1 Graphique pour de petites valeurs de la fenêtre h: 0.01 / 0.03 / 0.05

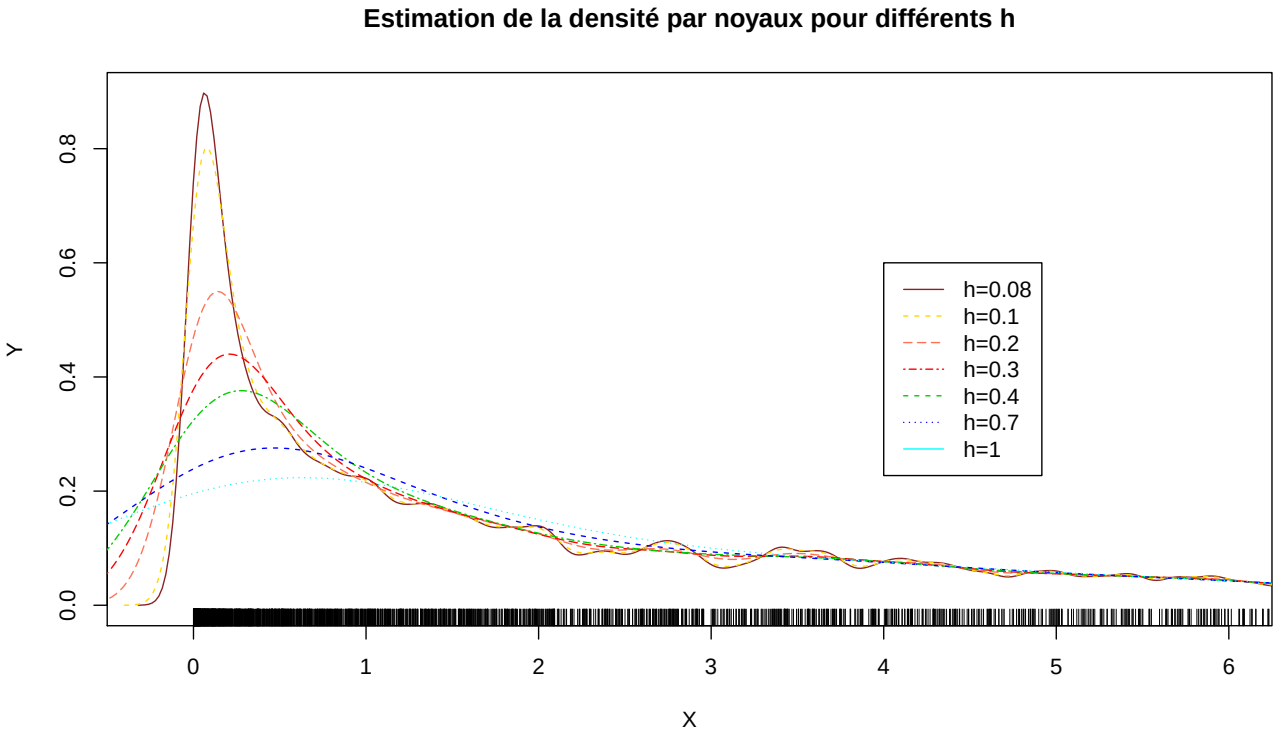






On remarque de fortes oscilations pour des h entre 0.01 et 0.05.

1.2.2 Estimateurs pour h variant de 0.08 à 1



A partir de h = 0.2 l'approximation est plus régulière.

1.2.3 Raison pour laquelle ce choix est important et ce qui se produit si h est mal choisi

- Si h est trop grand (courbes bleues) noyau trop régularisant, estimateur régulier mais biaisé.
- Si h est trop petit (courbes marron du graphique ci-dessus) l'estimateur est très oscillant, la variance est importante mais le biais est faible. Il faut donc trouver un h intermédiare, un compromis qui minimise la variance sans entrainer trop de biais. Un h compris entre 0.8 et 3 semble correct.

1.3 Estimation par noyau gaussien de la densité g(x) de X avec un h optimal

On commence par déterminer un paramètre de lissage ou fenêtre h optimale. Dans le sens qu'elle minimise une fonction de côut, ou critère d'erreur. Du type MSE (Erreur quadratique moyenne), MISE (Erreur Quadratique Intégrée Moyenne) ou AMISE (Erreur Quadratique Intégrée Moyenne Asymptotique). Plus pécisémmment on introduit l'estimateur du risque par validation croisé $\hat{J}(h) = \int_{\mathbb{R}} \hat{f}_{n,h}(x)^2\, dx - \frac{1}{2n}\sum_{i=1}^{n} \hat{f}_{(-i),n,h}(X_i)$ Avec $\hat{f}_{(-i),n,h}$ obtenu en calculant l'estimateur sur l'échantillon privé de la donnée X_i. On choisi $\hat{h}$ qui minimise ce risque $\hat{J}(\hat{h}) = min_h \hat{J}(h)$ On obtient cette fenêtre optimale par la fonction bw.ucv de la librairie stats qui implémente la validation croisée, mais aussi en utilisant d'autres méthodes.

1.3.1 Fenêtre h optimale par validation croisée

1.3.1.1 Utilisation de la fonction bw.ucv library stats

Cette fonction est adaptèe à la validation croisée en densité.

```
## [1] 0.05675136
```

1.3.2 Fenêtre h optimale par la règle de Silverman

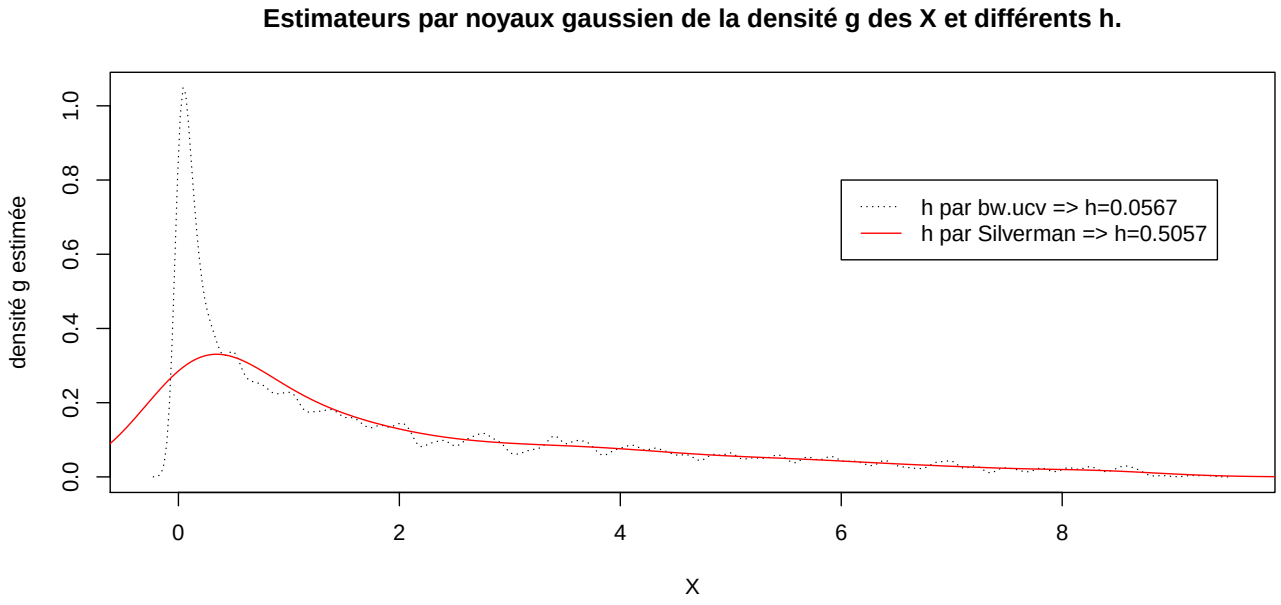
En appliquant la règle de Silverman, on obtient le h suivant:

```
## [1] 0.505695
```

On obtient le même résultat avec la fonction: bw.nrd

1.3.2.1 Fenêtre h optimale par avec la fonction bw.nrd

```
## [1] 0.505695
```



1.3.3 Fenêtre h optimale - Méthodes alternatives

1.3.3.1 Fonction density

On peut regrarder le résultat de la fonction density du package RSmooth qui teste différents noyaux et renvoie un h optimal

```
## [1] 0.4293637
```

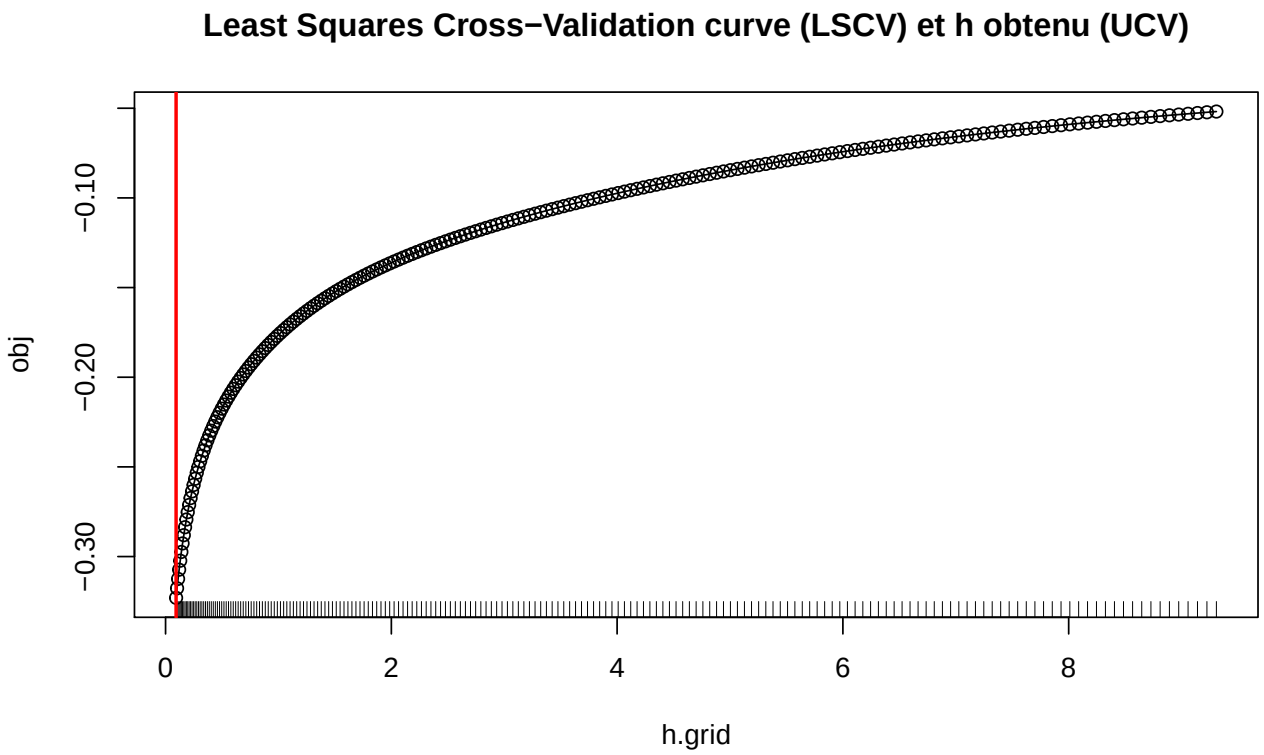
1.3.3.2 Fonction dpik

Méthode du package Kersmooth: pour la selection d'une fenêtre optimale.

```
## [1] 0.1439489
```

1.3.3.3 Fonction ucv recodée

La fenêtre h est obtenu par (UCV) de Least Squares Cross-Validation curve (LSCV) et h obtenu (UCV)

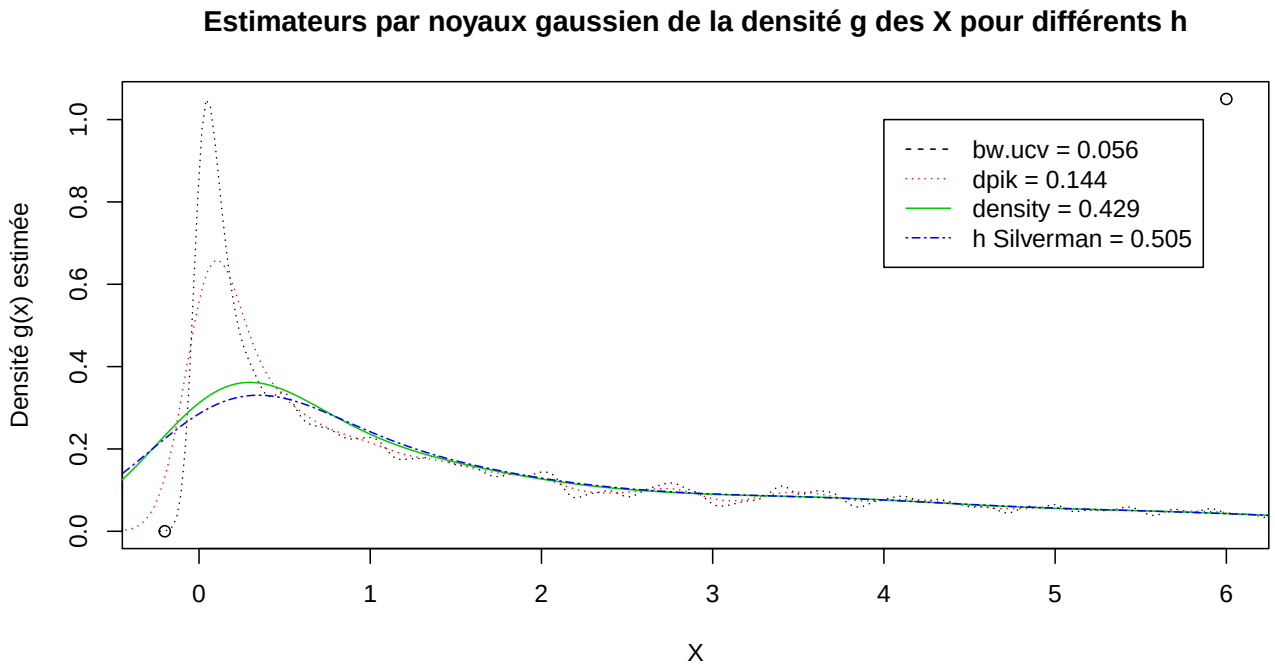



```
## [1] 0.09308678
```

1.3.4 Résumé des résultats

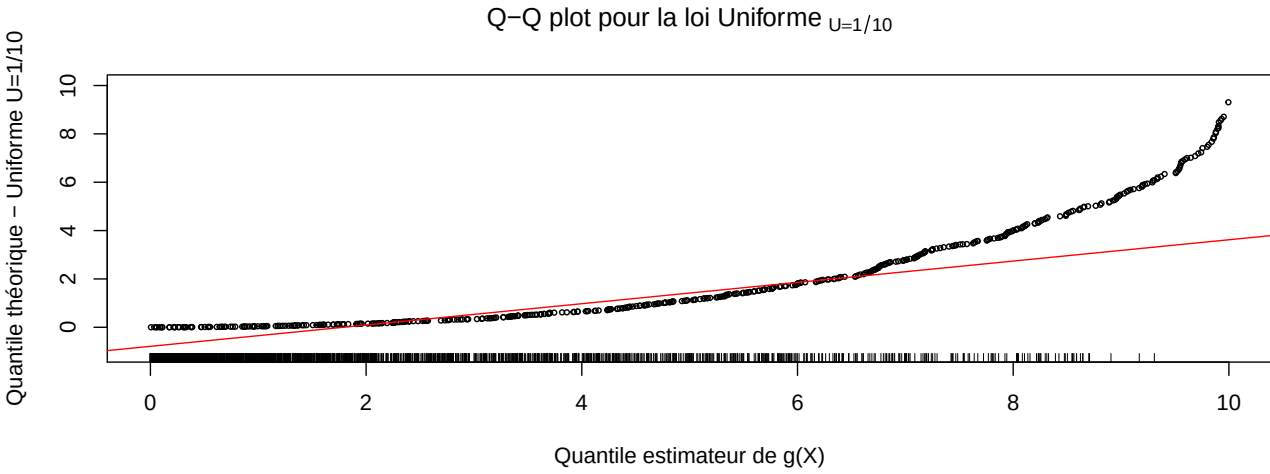
Méthode	valeur de h
bw.ucv	0.0567514
ucv recodée	0.0930868
dpik	0.1439489
density	0.4293637
Regle Silverman	0.505695

A partir de ces paramètres h on va implémenter les estimateurs à noyau de la densité à partir de la fonction bkde de R.



1.4 QQPlot - g(x) densité Uniforme

Implementation d'un QQ-plot pour vérifier empiriquement l'hypothèse g(x) suit (ou pas) une densité Uniforme



U=1/10 sur [0,10]
A la vue du graphique QQPlot par rapport à la loi uniforme (U=1/10) l'hypothèse selon laquelle g est uniforme n'est pas valide. En particulier dans la zone où X est entre 0 et 2 et où X est au delà de 6. La densité des individus est faible sur la dernière zone (X>6) mais à contrario très importante pour (X<2). On ne peut donc accépter l'hypothèse.

1.5 Zone de l'espace où l'estimation de r sera plus précise

On aura plus de précision où la densité des données est importante. Le quantile à 90% se situe au point x=5.461349. On a une répartition de 90% des données sur la 1ère moitié de l'intervalle: [0+dela , 5.46]. La variance ne dépend

pas de X et semble assez constante. Donc plus de précision dans l'intervalle [0+dela , 5.46]. La précision diminue ensuite (sur l'axe des abscisses à droite).

2 Reconstruction de r(x)

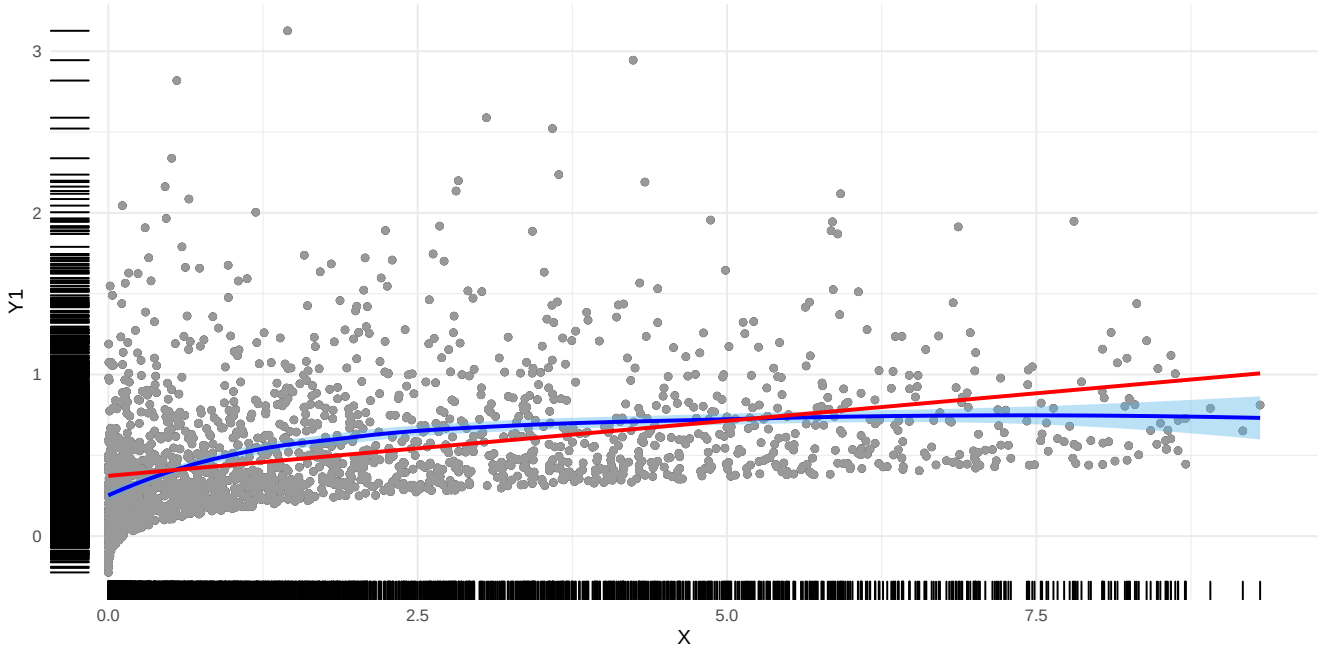
On est dans le cadre de l'estimation non paramétrique, on reprend les hypothèses classiques. On utilise les données de Data1, (X,Y1).

2.1 Linéarité de la fonction r

Pour vérifier la linéarité on va tracer sur un même graphique, le nuage de points ainsi que la droite de régression linéaire Y~X. On va y ajouter la courbe de regression locale de type loess. Avec cette méthode l'ajustement se fait localement, par ajustement d'un polynôme de degès 1 ou 2. Cette dernière courbe indique la tendance globale entre nos 2 variables, que l'on va comparer à la droite de régression linéaire.

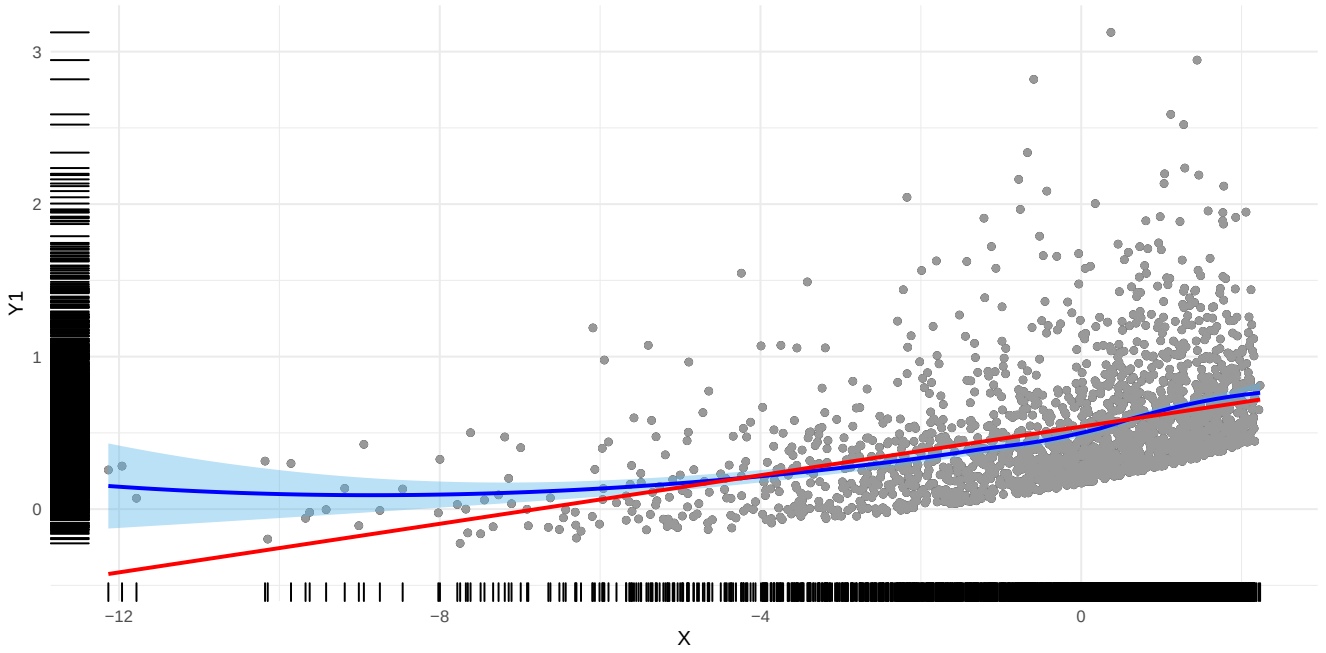
On utilise la fonction de R stat_smooth de la librairie ggplot2 avec l'option method=lm pour construire la régression linéaire simple (en rouge). et l'option method=loess pour construire la régression de type loess (en bleue).

- Tout d'abord on trace Y1 en fonction de X.



Sans transformation, à la vue du graphique, r ne peut être linéaire. La forme de la courbe loess suggère d'utilser la transformation log(X) pour linéariser.

- Maintenant On trace Y1 en fonction de log(X)



Dans la région $X_i \in [-5, 1]$ où l'on retouve la quasi totalité de l'échantillon, on a une bonne adéquation des courbes loess et lm, la transformtion (log(x)) a permis de bien linéariser.

2.2 Construction d'un estimateur non-parametrique de r(x) par noyaux régularisants

Le noyau permet de lisser le bruit ξ_i et localise l'information au niveau d'un point, en y ajoutant un poids. Un peu comme une distribution ou masse de Dirac. Ces méthodes de lissage par noyau consistent à effectuer au voisinage de chaque point une régression locale. Qui dans le cas Nadaraya-Watson est linéaire et est polynomiale dans le cas des méthodes par polynômes locaux. La localisation est assurée par la fonction de poids le noyau K. On le prend ici gaussien $K(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2}$ Plus précisémment en posant $K_h(x_0 - x) = \frac{1}{h} K(\frac{x_0 - x}{h})$ on a en utilisant la représentation $Y_i = r(X_i) + \sigma\xi_i$ que $\frac{1}{n}\sum_{i=1}^n K_h(x_0 - X_i)Y_i = \frac{1}{n}\sum_{i=1}^n K_h(x_0 - X_i)r(X_i) + \frac{1}{n}\sum_{i=1}^n K_h(x_0 - X_i)\sigma\xi_i$ (*) (*) $\approx_{n\to\infty} \mathbb{E}[K_h(x_0 - X_i)r(X_i)] = \int_{\mathbb{R}} K_h(x_0 - x)r(x)g(x)dx) \approx_{h\to 0} r(x_0)g(x_0)$ On a vu au §1 que l'on peut estimer le terme $g(x_0)$ par un estimateur à noyau. Il nous reste donc à estimer $r(x_0)$. L'estimateur de Nadaraya-Watson de $r(x_0)$ est l'estimateur définit par la formule suivante. $\hat{r}_{n,h}(x_0) = \frac{n^{-1}\sum_{i=1}^n K_h(x_0 - X_i)Y_i}{n^{-1}\sum_{i=1}^n K_h(x_0 - X_i)} = \sum_{i=1}^n w_{i,n}(x_0)Y_i$ avec $w_{i,n}(x_0) = \frac{K_h(x_0 - X_i)}{\sum_{i=1}^n K_h(x_0 - X_i)}$ Pour l'estimateur par polynômes locaux, l'approximation localement constante est remplacée par un polynôme.

Pour implémenter ces estimateurs, on commence par déterminer le paramètre de lissage h. Puis à partir des paramètres h obtenus on implémente les méthodes de lissage de Nadaraya-Watson et les méthodes par polynômes locaux.

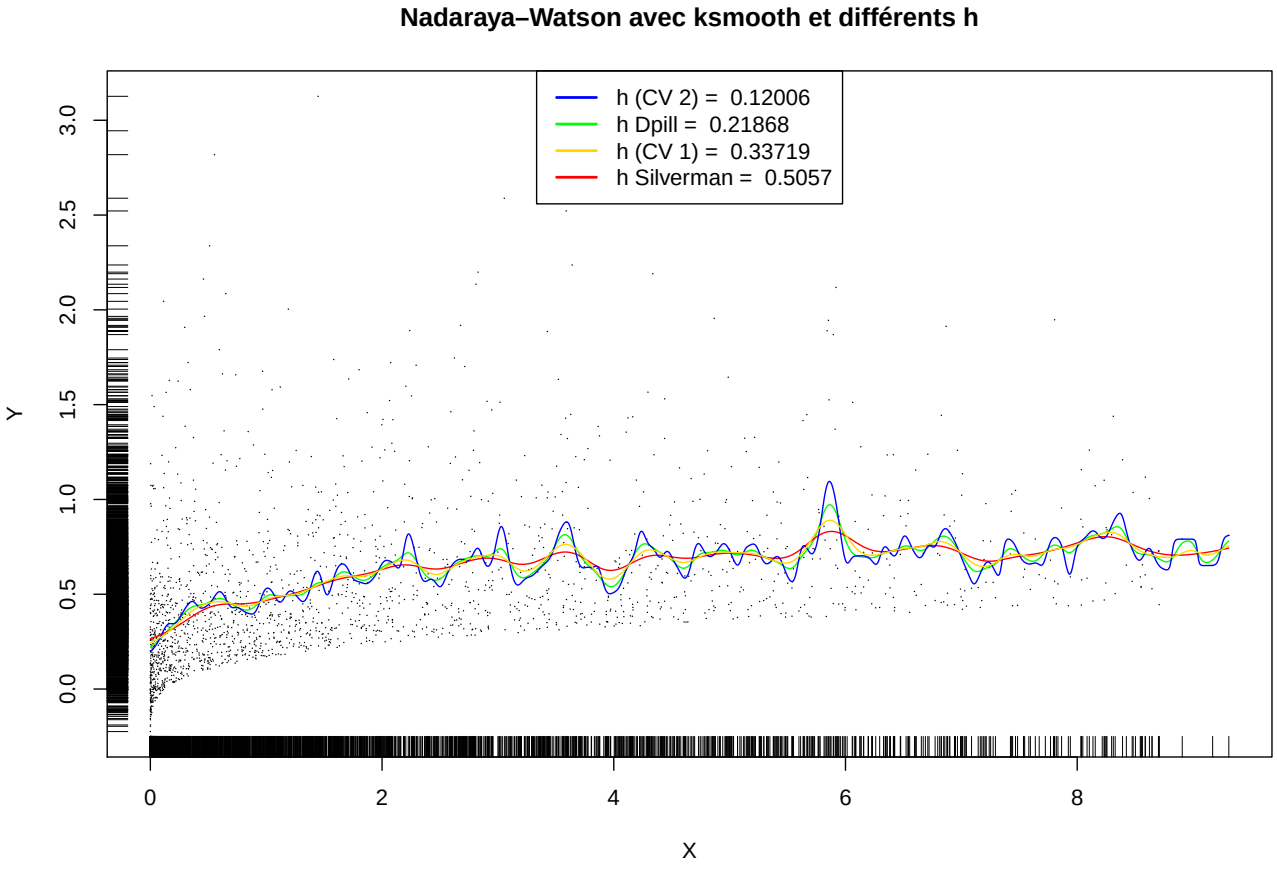
2.2.1 Détermination de la fenêtre h

La validation croisée dans le cas de la régression se décompose comme suit : On définit, pour i=1,2,. . . ,n l'estimateur $\hat{r}_{(-i),n,h}(x_0) = \sum_{j=1}^n Y_j w_{j,n,h,(-i)}$ avec $w_{j,n,h,(-i)}(x_0) = \frac{w_{j,n,h}(x_0)}{\sum_{k\neq i} w_{k,n,h}(x_0)}$ si $j \neq i$ et 0 sinon. On définit le score de validation leave-one-out $\hat{R}(h) = \frac{1}{n}\sum_{i=1}^n (Y_i - \hat{r}_{(-i),n,h}(X_i))^2$ On choisi $\hat{h}$ qui minimise la fonction de score, $\hat{h} = min_h \hat{R}(h)$. Pour la validation croisé, dans le cas de la régression on utilise la fonction codée en R vue en cours CVbwt (h_CV1) et bw.cv.grid (h_CV2). On obtient différentes fenêtres h obtenues à partir des méthodes: fonction dpill de R, h de Silverman, validation croisée.

```
## [1] "h_dpill : 0.21868486029053 h_silver : 0.505695020156574 h_CV1 ; 0.337185929648241 h_CV2 ; 0.120060
```

2.2.2 Estimateurs $\hat{r}$ de r par Nadaraya-Watson avec la librairie stat - fonction : ksmooth

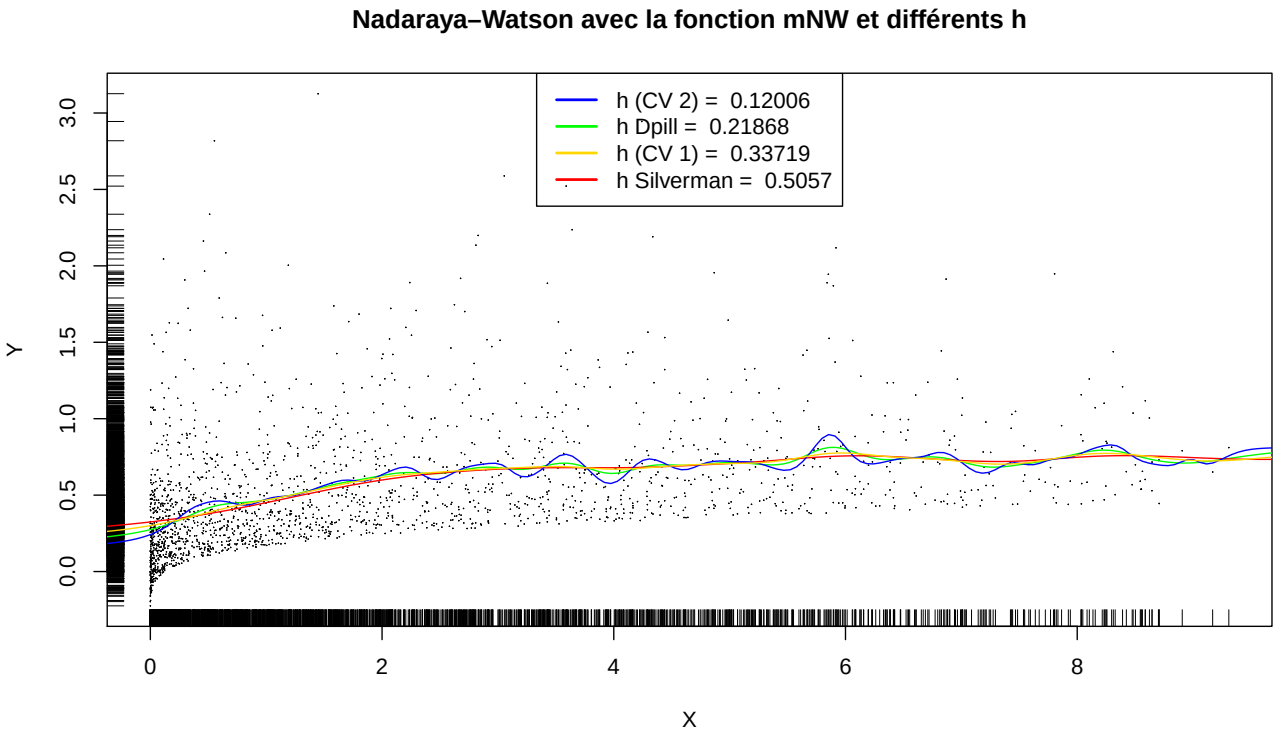
Utilisation de la fonction ksmooth et différentes fenêtre h_dpill=0.2186849, h_silver=0.505695, h_CV1=0.3371859, h_CV2=0.1200601



L'estimateur est sensible au choix de h. L'estimateur de Nadaraya-Watson est très oscillant par construction.

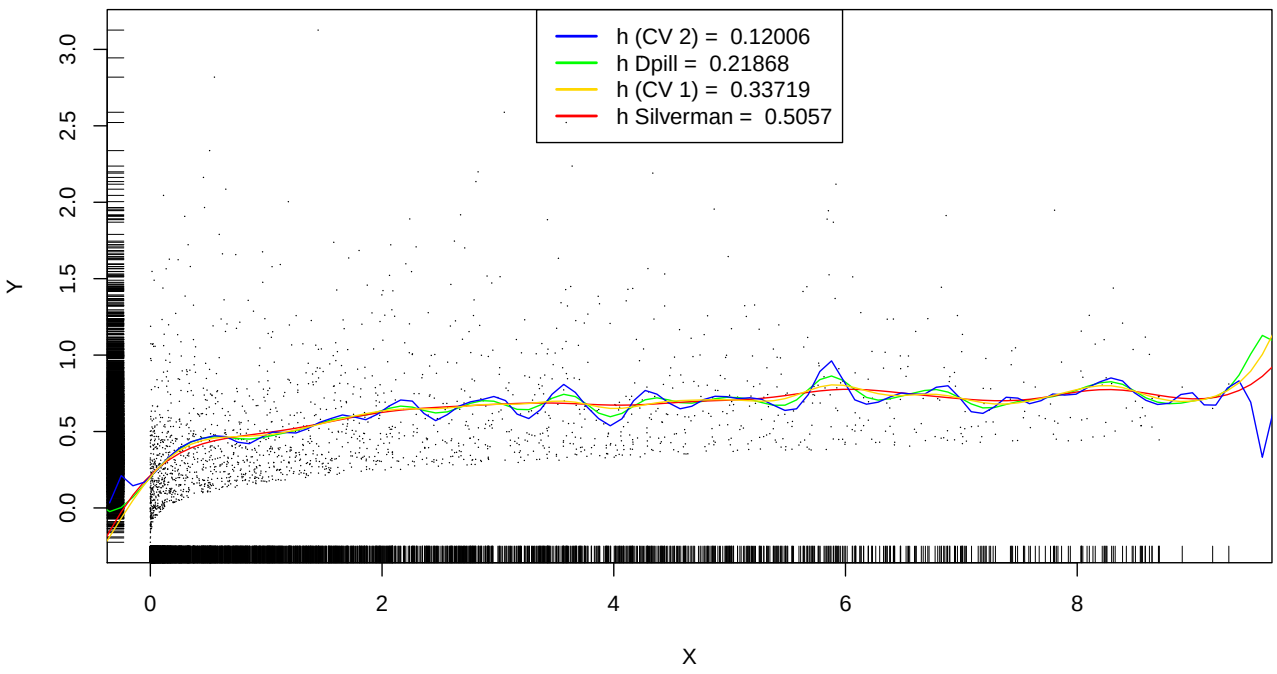
2.2.3 Estimateurs $\hat{r}$ de r par Nadaraya-Watson à partir de la fonction recodée NW

Utilisation de la fonction NW et différentes fenêtre h_dpill=0.2186849, h_silver=0.505695, h_CV1=0.3371859, h_CV2=0.1200601



2.2.4 Estimateurs $\hat{r}$ de r par polynômes locaux

Utilisation de la fonction locpoly du package Kersmooth avec différentes fenêtre h_dpill=0.2186849, h_silver=0.505695, h_CV1=0.3371859, h_CV2=0.1200601 On choisi le degrès 2.



On remarque que cet estimateurest plus régulier que Madaraya-Watson implémenté par ksmooth.

2.3 Estimation de r en regressant Y1 sur log(X)

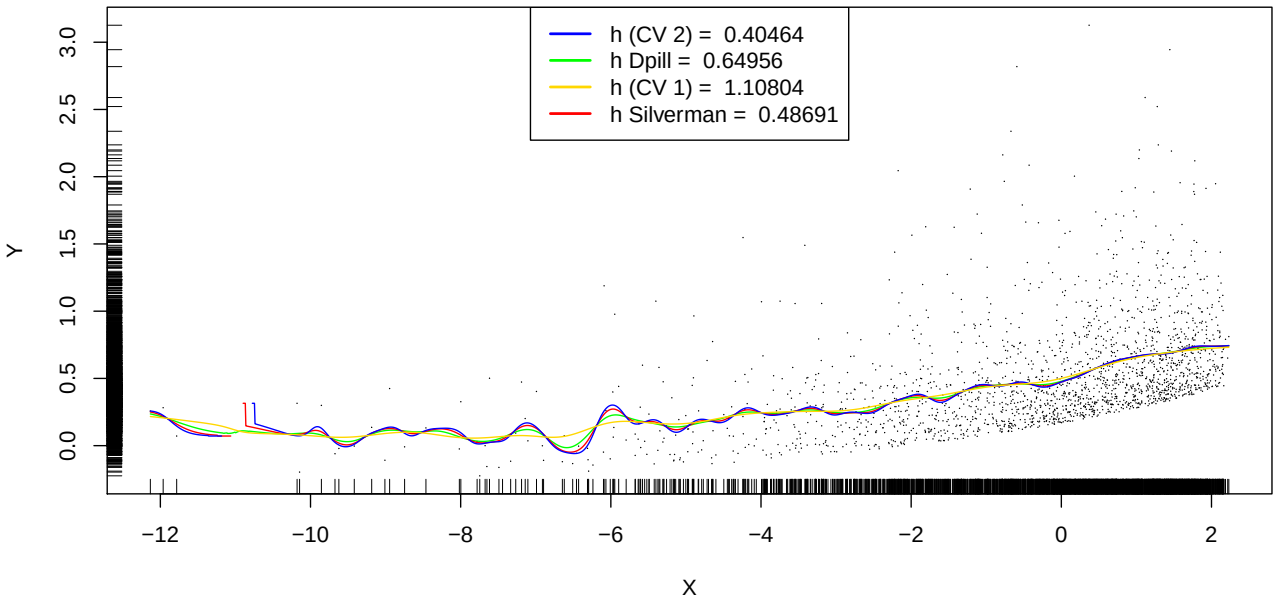
On suit la même démarche qu'au §2.2. On commence par déterminer un h optimal à partir de la fonction dpill, de la règle de Silverman et par validation croisée. Et ensuite on implémente avec le paramètre h calculé l'estimateur de Nataraya-Watson avec la fonction de R ksmooth et l'estimateur par polynômes locaux avec la fonctionde R locpoly.

```
## [1] "h_dpill : 0.649558017143592 h_silver : 0.48691251968111 h_CV1 ; 1.10804020100503 h_CV2 ; 0.4046380
```

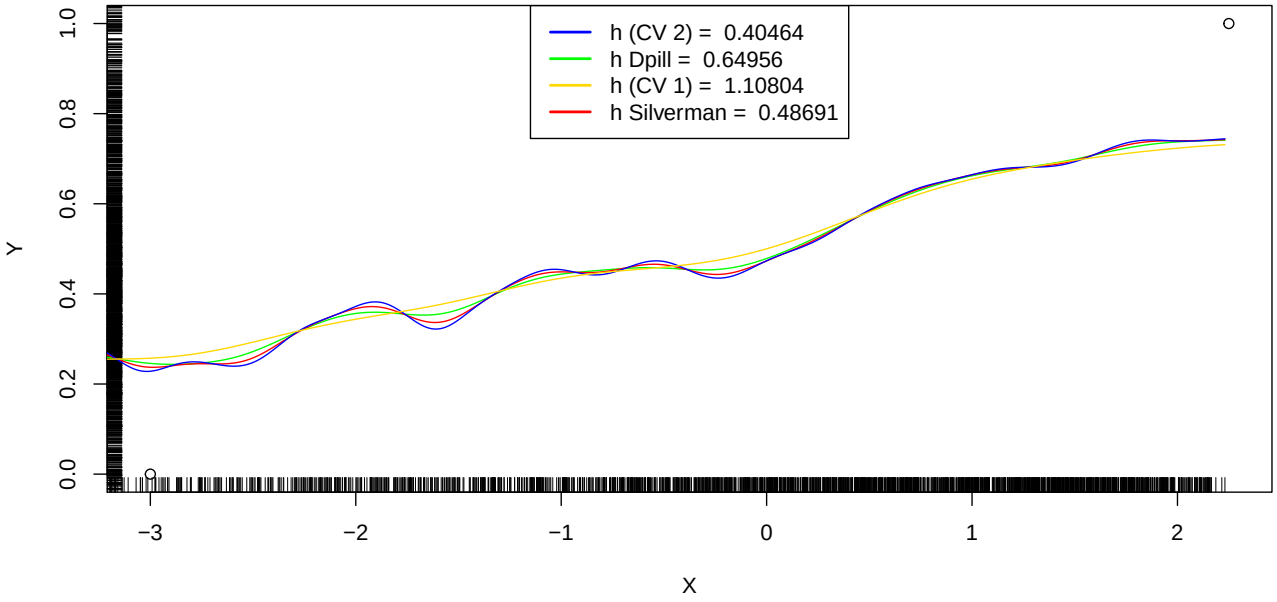
2.3.1 Estimateurs $\tilde{r}$ de r par Nadaraya-Watson avec la librairie stat - fonction : ksmooth

Utilisation de la fonction ksmooth avec différentes fenêtres h_dpill=0.649558, h_silverman=0.4869125, h_CV1=1.1080402, h_CV2=0.4046381

Estimateurs construit avec Nadaraya−Watson: ksmooth et différents h

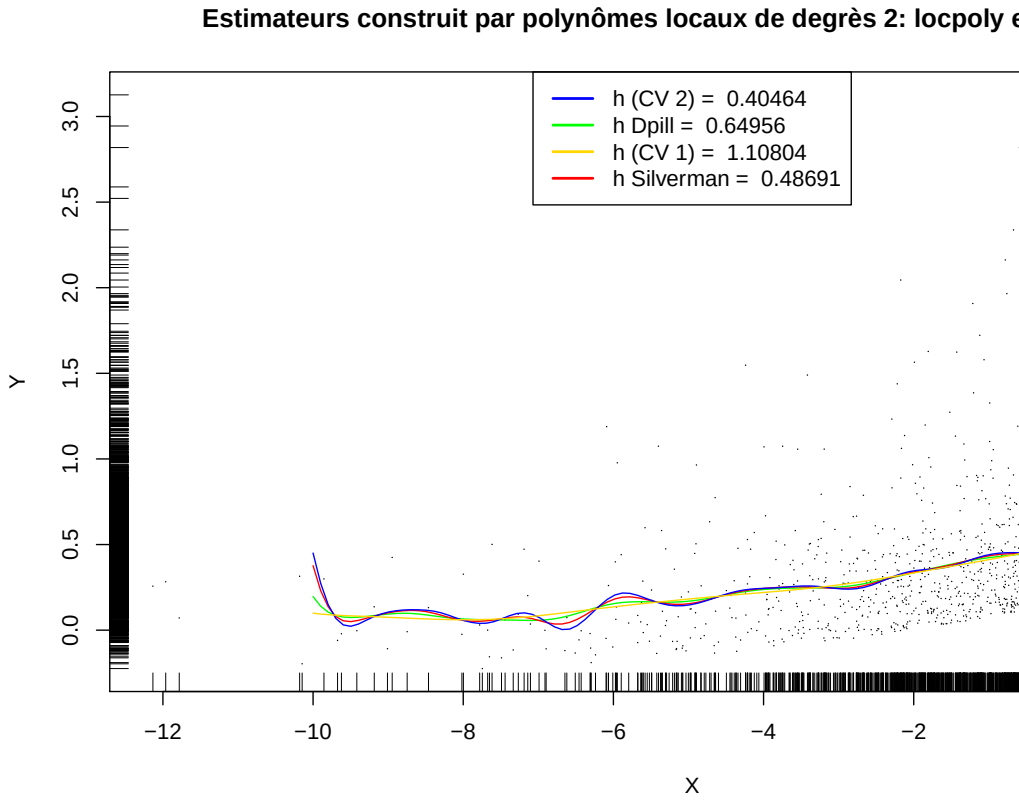


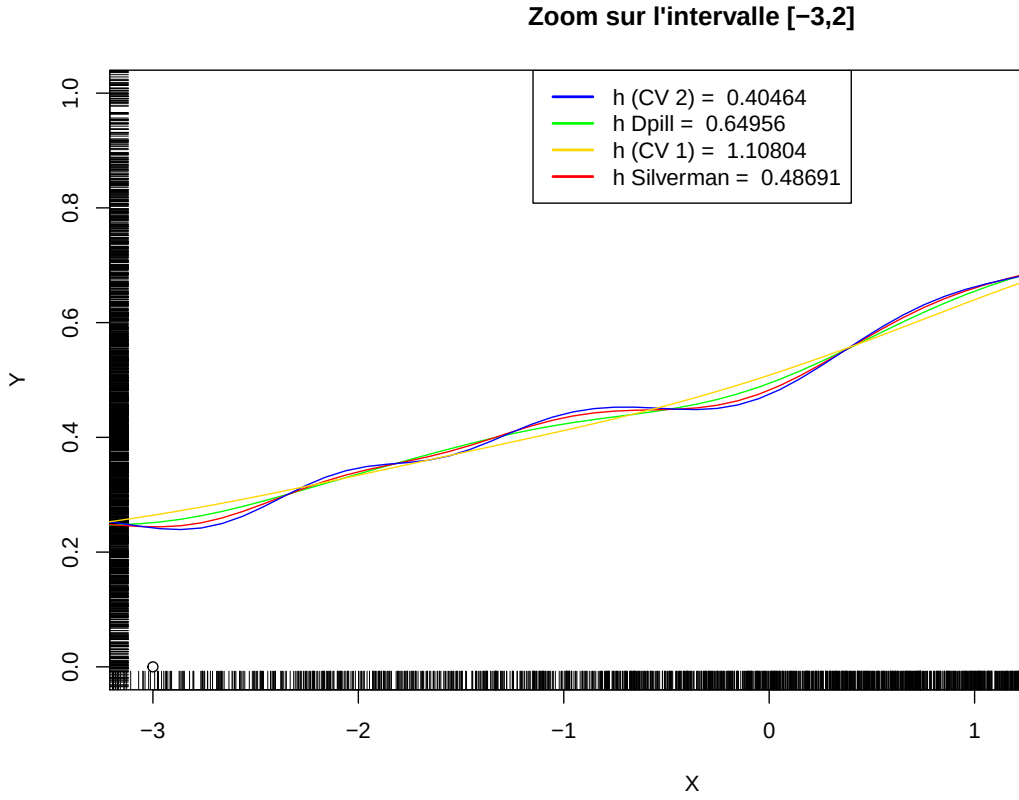
Zoom sur l'intervalle [−3,2]



2.3.2 Estimateurs $\tilde{r}$ de r par polynômes locaux de degrès 2

Utilisation de la fonction locpoly avec différentes fenêtres h_dpill=0.649558, h_silverman=0.4869125,

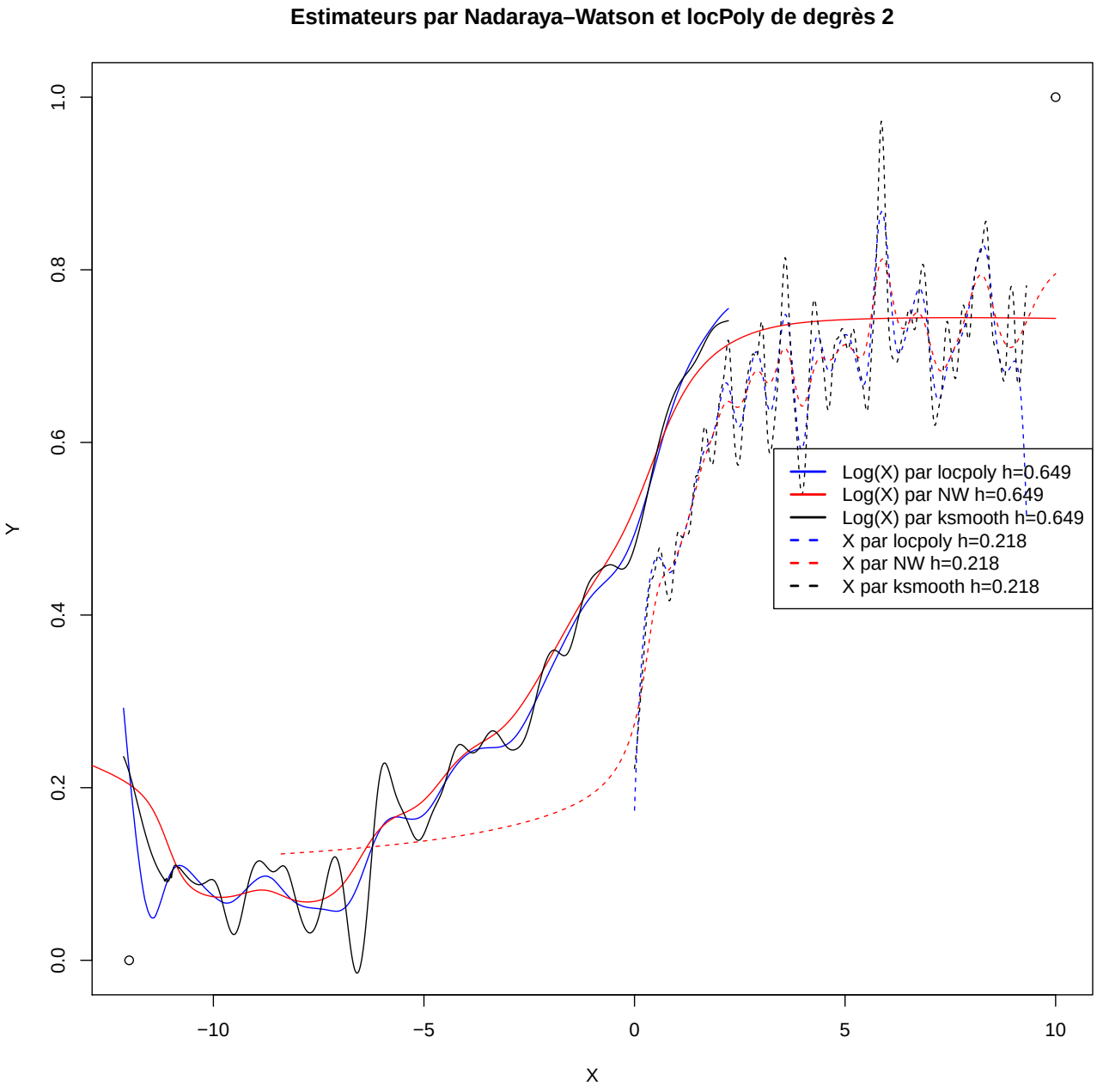




h_CV1=1.1080402, h_CV2=0.4046381

Ces estimateurs par polynômes locaux sont plus régularisants par construction que l'estimateur de Nadaraya-Watson construits avec ksmooth.

2.4 Représentation sur un même graphe de $\hat{r}$ et $\tilde{r}$



2.5 Remarques - Explications

Dans les zones où la densité est élevé ($log(X_i) \in [-4, 2]$ pour log(X) et $X_i \in]0+, 4]$ pour X) On remarque que les estimateurs basés sur la régression de Y_i sur $log(X_i)$ sont bien plus réguliers, quasiment linéaires.

La transformation log(X) a permis de linéariser la zone à forte densité. C'est ce que l'on a remarqué au §2.1.

3 Etude de la densité μ des ξ_i

3.1 A partir du jeu de donnees Data1

3.1.1 Distribution approximative de $\tilde{\xi}_i$

```
##       X.1               X                  Y1          
##  Min.   :   1.0   Min.   :0.000005   Min.   :-0.2244  
##  1st Qu.: 250.8   1st Qu.:0.234186   1st Qu.: 0.2545  
##  Median : 500.5   Median :1.035451   Median : 0.4186  
##  Mean   : 500.5   Mean   :1.997154   Mean   : 0.5045  
##  3rd Qu.: 750.2   3rd Qu.:3.332095   3rd Qu.: 0.6604  
##  Max.   :1000.0   Max.   :8.707688   Max.   : 2.9446
```

On a la représentation suivante $Y_i - r(X_i) = \sigma\xi_i$ (cas homoscédastique σ est constant)

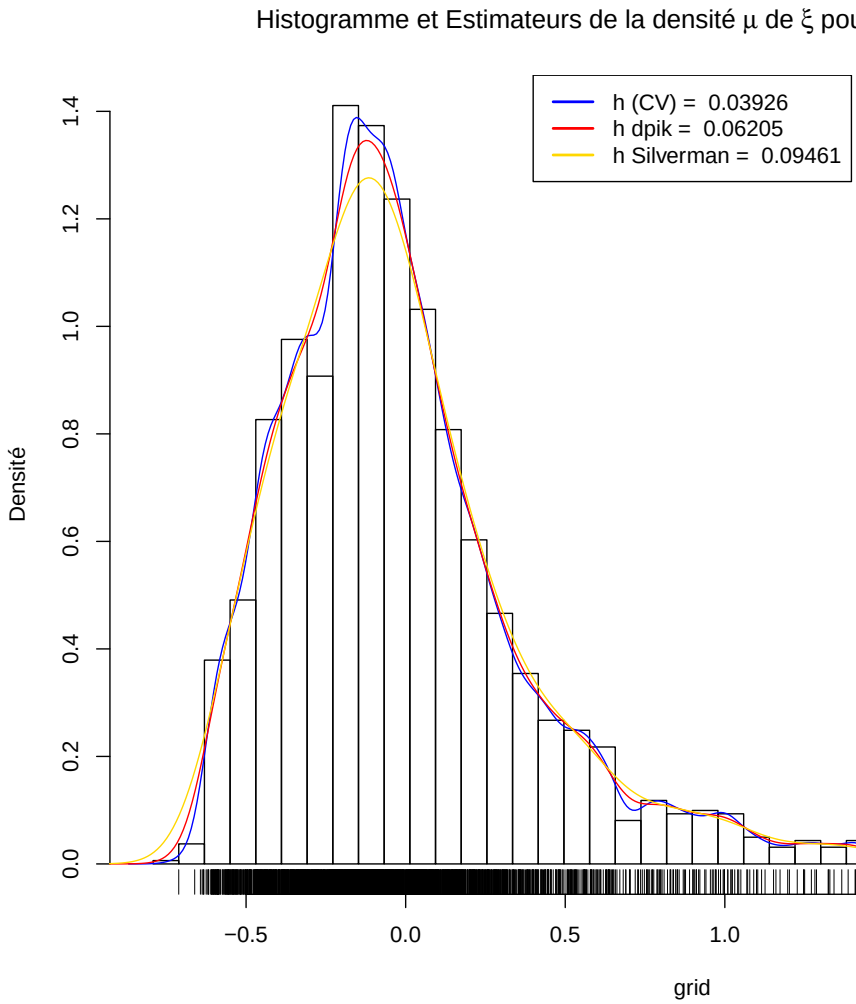
Par définition $\tilde{\xi}_i = Y_i - \hat{r}_{n,h}^{(-)}(X_i)$ où $\hat{r}_{n,h}^{(-)}(x)$ est un estimateur de r(x).

La distribution approximative de $\tilde{\xi}_i$ est celle de ξ_i à la constante multiplicative près σ.

3.1.2 Représentation de la densité $\mu(x)$ ξ_i

A partir du h établi a la question 2.2: h_dpill=0.2186849 et de l'estimateur $\hat{r_h}$ on calcul un h optimal pour $Y_i - \hat{r_h}(X_i)$ On obtient:

```
## [1] "h_dpik : 0.0620479371982669 h_silver : 0.0946066036212771 h_density ; 0.0638864271238886 h_ucv ; 0
```



On estime alors la densité μ avec la fonction bkde de R.

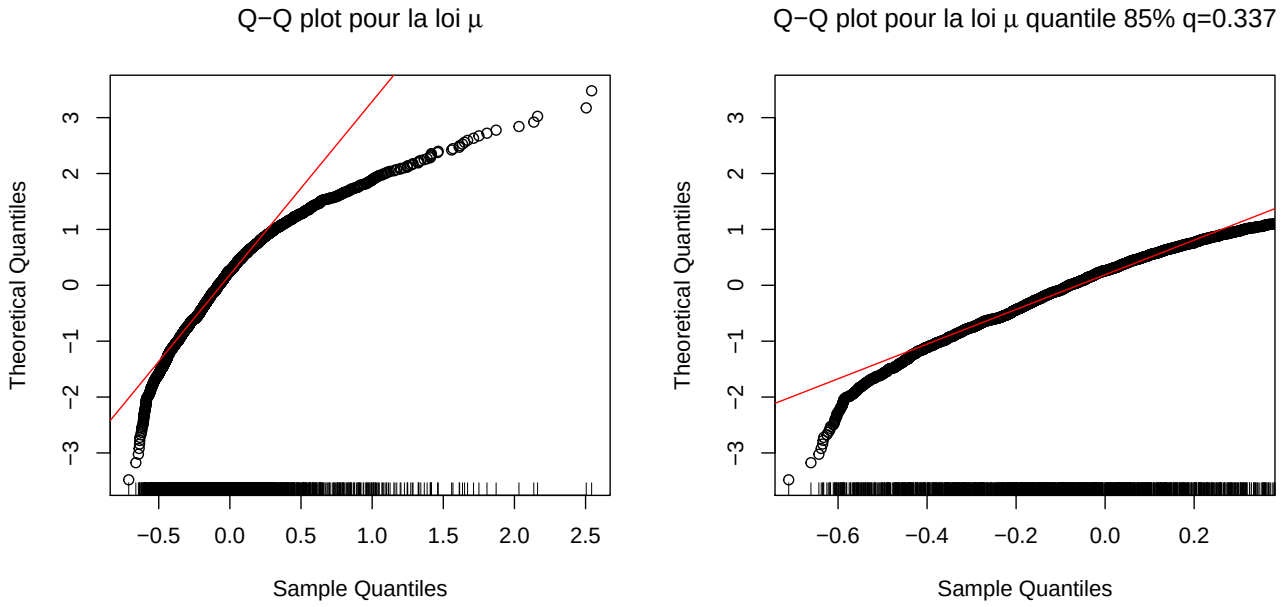
3.1.3 Interêt d'avoir decoupé le jeu de donnees selon *J+* et *J-*

On a ainsi un jeux de données d'apprentissage et de test. On peut utiliser le jeux de données d'apprentissage pour estimer et construire nos estimateurs, le jeux de test pour calculer une erreur de prédiction. A partir de cette erreur de prédiction on a un critère pour choisir le meilleur estimateurs

3.1.4 La densité μ peut-elle être gaussienne

On peut déjà remarquer qu'à la vue des graphiques des estimateurs de la densité $\tilde{\xi}_i$, on a trop de dissymétrie pour avoir une gaussienne. On va le préciser avec le protocole empirique de verification ci-après que l'on va appliquer aux données $\hat{r}_{n,h}^{(-)}(X_i) - Y_i = \tilde{\xi}_i$

- test du QQPlot



Le QQPlot, graphique d'adéquation des quantiles rejette l'hypothèse de normalité si l'on regarde la globalité de l'échantillon. C'est moins sûre si l'on regarde le sous ensemble qui correspond au quantile 85% qui vaut 0.337 (2ème graphique).

- test de shapiro

```
##
##  Shapiro-Wilk normality test
##
## data:  estimMu
## W = 0.89345, p-value < 2.2e-16
```

Le test de Shapiro-Wilk donne une p-value $<$ 2.2e-16. L'hypothèse de normalité est rejetée.

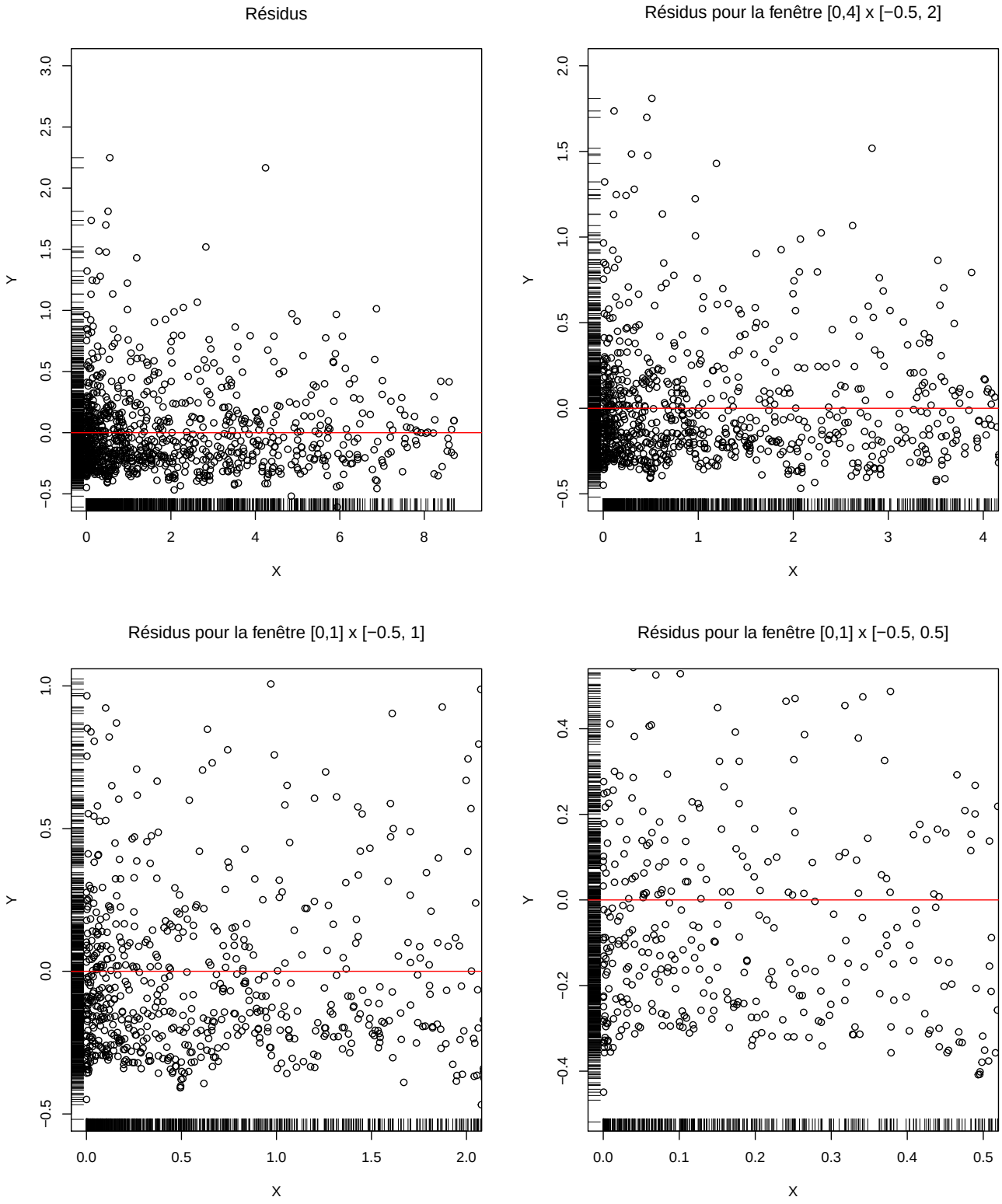
- test de Kolmogorov-Smirnoff

```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  estimMu
## D = 0.10503, p-value < 2.2e-16
## alternative hypothesis: two-sided
```

la p-value n'est pas significative l'hypothèse de normalité est rejetée. La densité μ n'est donc pas gaussienne.

3.1.5 homoscédasticité du modèle

Pour tester que le modele est bien homoscédastique, on peut tracer un graphe des résidus.



On ne remarque pas de structures particulières ou de tendances. La répartition est assez uniforme, en particuliers dans la zone de forte densité (proche de x=0). Ce qui nous ammène à penser que l'hypothèse d'homoscédasticité est bien vérifiée.

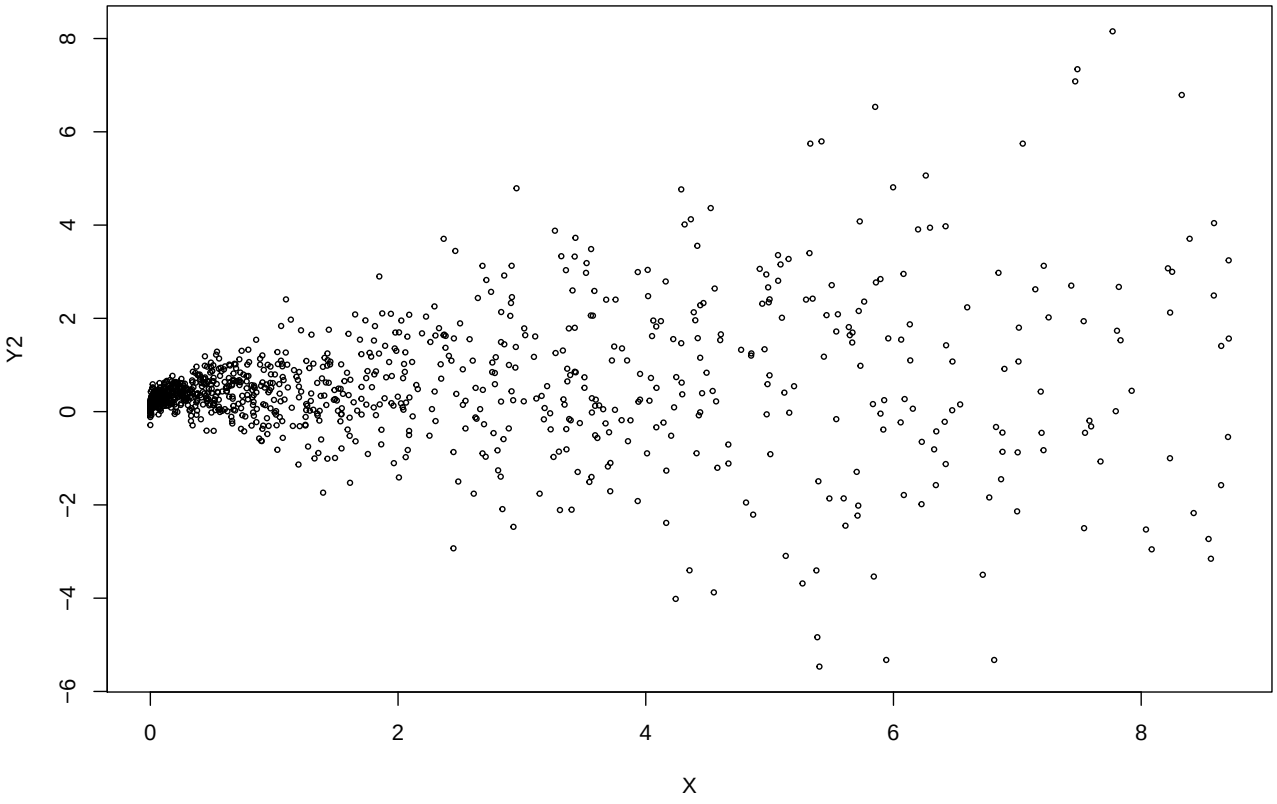
3.2 A partir du jeu de donnees Data2

On cherche à estimer μ et σ^2. Pour cela, on coupe a nouveau l'echantillon en deux et on considère à nouveau $\tilde{\xi}_i$

```
summary(d2)
```

```
##       X.1               X                  Y2          
##  Min.   :   1.0   Min.   :0.000005   Min.   :-5.46738  
##  1st Qu.: 500.8   1st Qu.:0.258074   1st Qu.: 0.08333  
##  Median :1000.5   Median :1.192414   Median : 0.35947  
##  Mean   :1000.5   Mean   :2.029447   Mean   : 0.53951  
##  3rd Qu.:1500.2   3rd Qu.:3.318174   3rd Qu.: 0.87849  
##  Max.   :2000.0   Max.   :9.308684   Max.   :10.15297
```

Jeux de données Data2 observations X en abscisse et Y2 en ordonnée.



3.2.1 Justifier qu'en regressant ξ_i sur X_i on obtient un estimateur de σ^2

Par définition $\tilde{\xi}_i = Y_i - \hat{r}(X_i)$ où $\hat{r}(x)$ est un estimateur de r(x).

Ainsi $\tilde{\xi}_i^2 = (Y_i - \hat{r}(X_i))^2$

En remplaçant par l'expression de $Y_i = r(X_i) + \sigma(X_i)\xi_i$ on obtient

$$\tilde{\xi}_i^2 = (r(X_i) + \sigma(X_i)\xi_i - \hat{r}(X_i))^2$$

Puis en développant

$$\tilde{\xi_i^2} = (r(X_i) - \hat{r}(X_i))^2 + \sigma(X_i)^2\xi_i^2 + 2(r(X_i) - \hat{r}(X_i))\sigma(X_i)\xi_i$$

On conditionne par rapport à X_i et on utilise l'hypothèse d'indépendance de ξ_i

$$E(\tilde{\xi_i^2} \mid X_i) = E((r(X_i) - \hat{r}(X_i))^2 \mid X_i) + E(\sigma(X_i)^2 \mid X_i)E(\xi_i^2) + 2E((r(X_i) - \hat{r}(X_i))\sigma(X_i)) \mid X_i)E(\xi_i)$$

Par hypothèse $E(\xi_i) = 0$ et $E(\xi_i^2) = 1$ on a donc finalement:

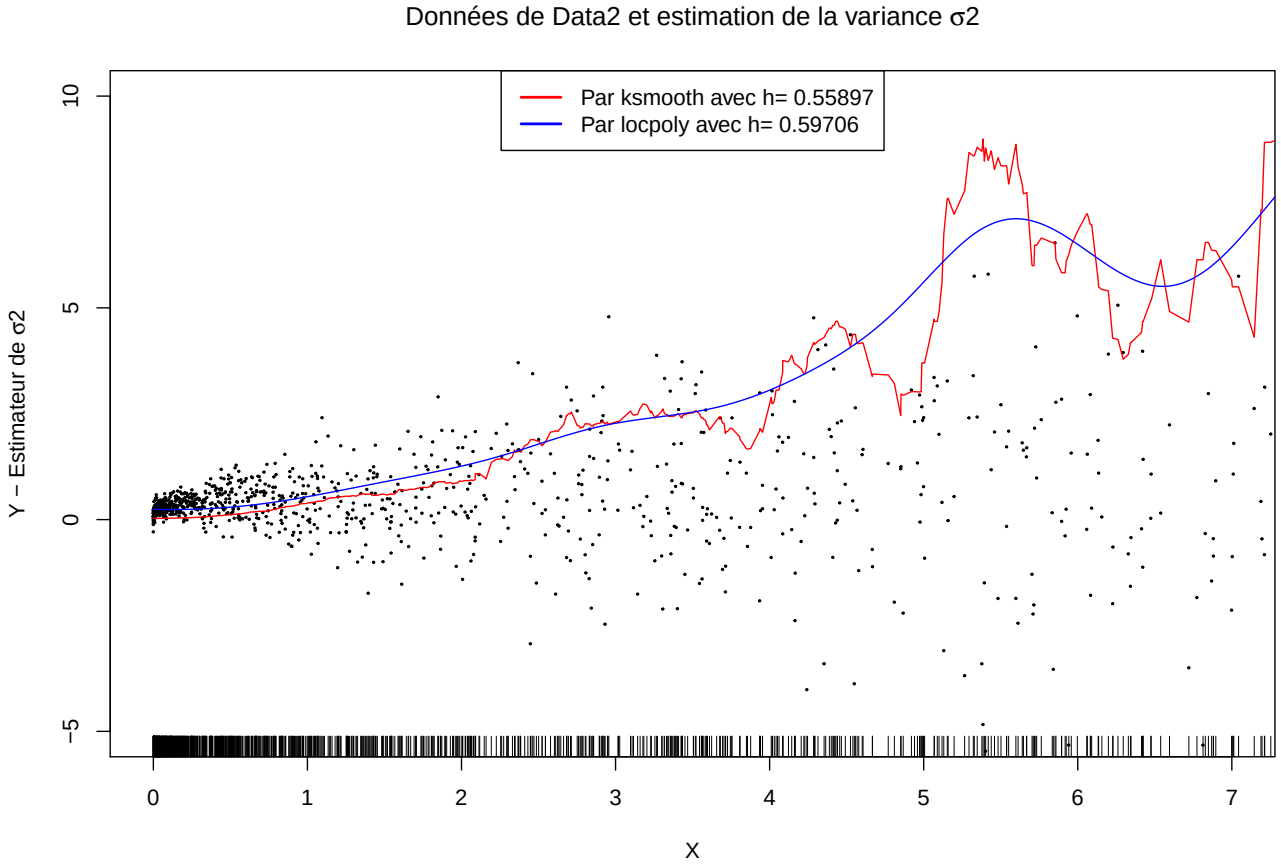
$$E(\tilde{\xi_i^2} \mid X_i) = E((r(X_i) - \hat{r}(X_i))^2 \mid X_i) + E(\sigma(X_i)^2 \mid X_i)$$

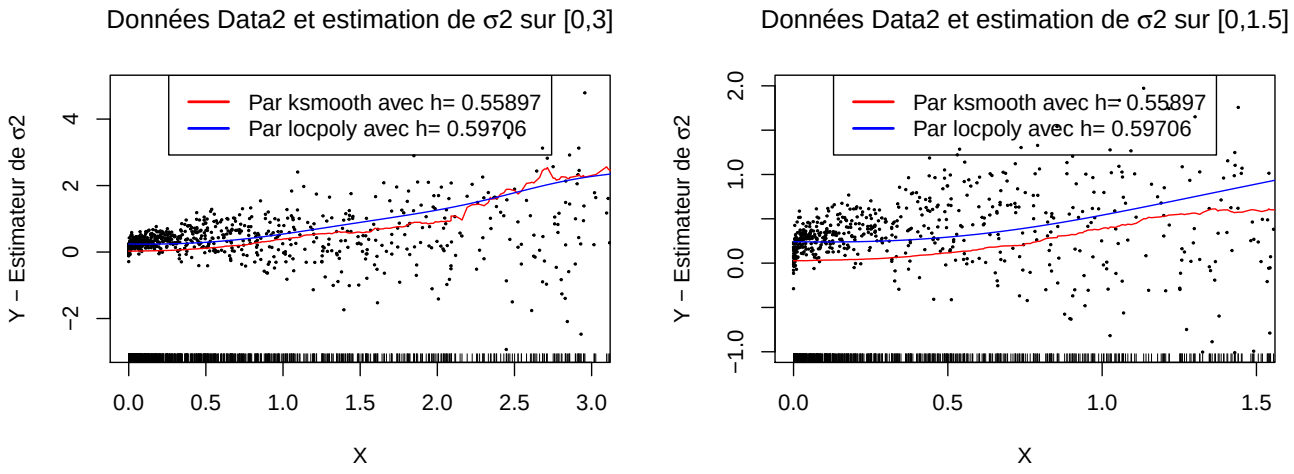
Ce qui donne $E(\tilde{\xi_i^2} \mid X_i) = (r(X_i) - \hat{r}(X_i))^2 + \sigma(X_i)^2$

Comme $\hat{r}(X_i)$ est un estimateur de $r(X_i)$ on a bien le résultat.

3.2.2 Implémentation et visualisation

A partir de l'estimateur $\hat{r}(X_i)$ on va construire un estimateur de $\sigma(X_i)^2$ Pour celà on construit un estimateur en regressant sur X_i le carré des résisus: $(Y_i - \hat{r}(X_i))^2$ On utilise la fonction ksmooth et locpoly (degrès 2).





En comparant au jeux de données (nuage de points Data2) on retrouve bien le résultat attendu. Peu de variance où la densité est élevé dans l'intervalle [0+,1]. Les points sont proches les uns des autres. Ensuite la variance augmente en même temps que la densité des points diminue. Avec un premier saut à partir de x=2 puis x=4. Ensuite la densité des observations est faible.

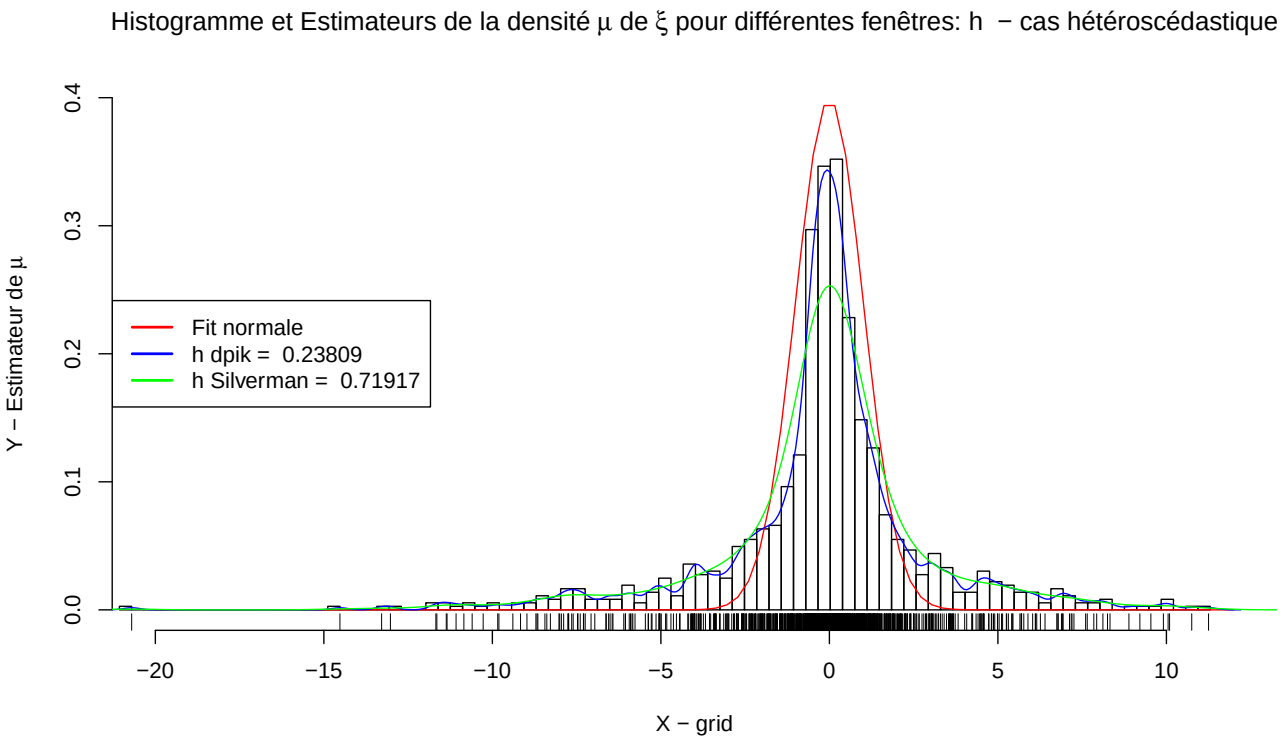
3.2.3 La densité μ peut-elle être gaussienne

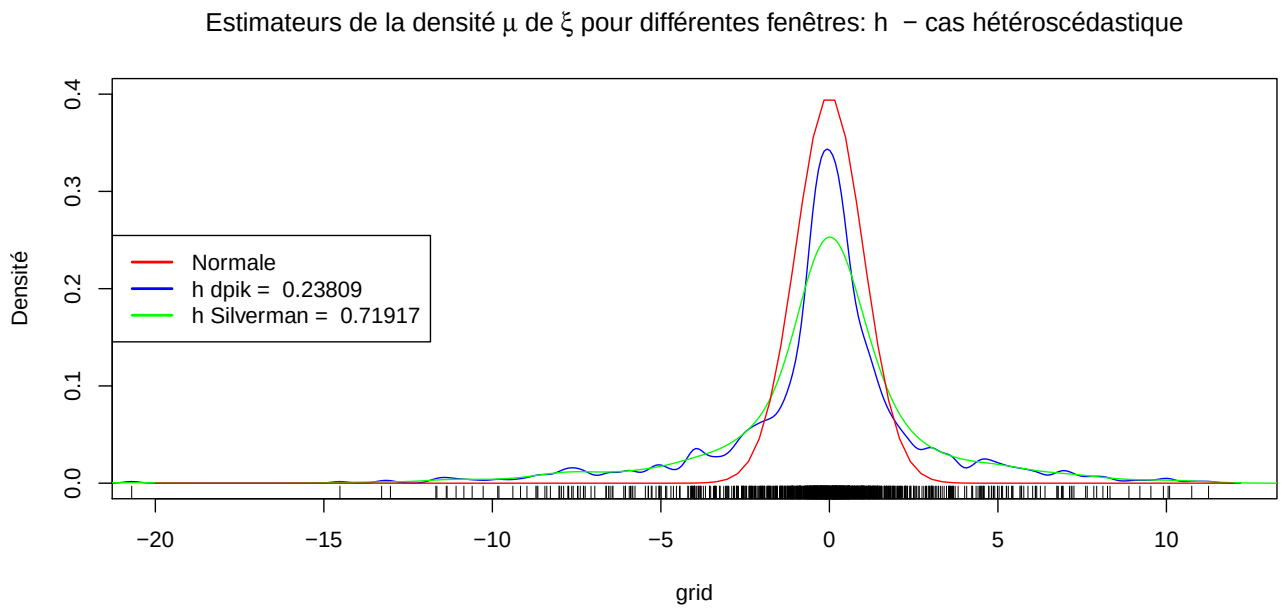
Avant d'appliquer les tests classiques (QQPlot, Test de Shapiro et KS) on va regarder comment se comportent les estimateurs à noyaux gaussiens. On estime la densité μ des ξ_i à partir de l'estimation $(Y_i - \hat{r}(X_i))/\hat{\sigma}^2(X_i)$ où $\hat{\sigma}^2(X_i)$ est l'estimateur de $\sigma^2(X_i)$ obtenu précédement.

- On recherche tout d'abord le h optimal avec les méthodes habituelles: fonction dpik, règle de Silverman, fonction density, validation croisée.

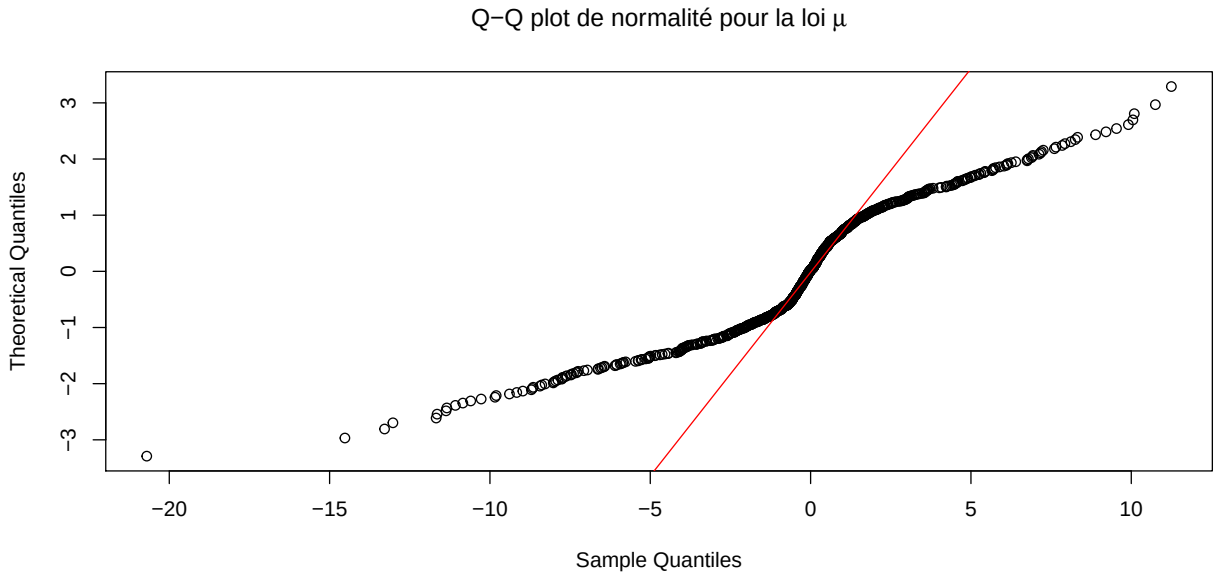
```
## [1] "h_dpik : 0.238094752525185 h_silver : 0.719173399214354 h_density ; 0.313922107129252 h_ucv ; 0.21
```

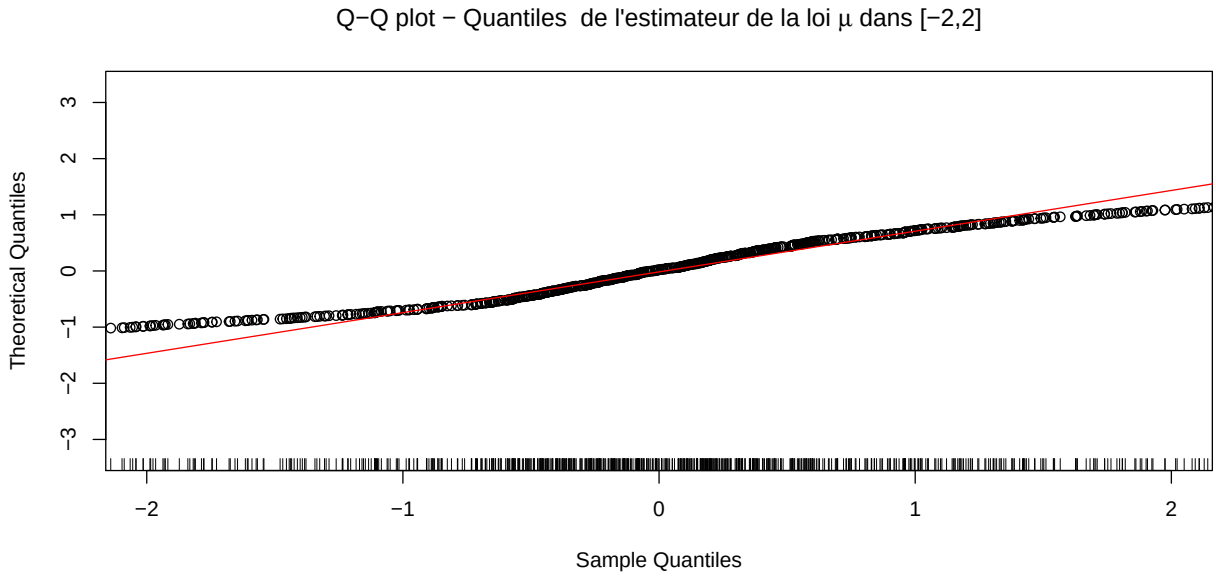
- On estime la densité à partir de la fonction bkde que l'on représente graphiquement:





A la vue des graphiques les estimateurs de la densité μ semblent se rapprocher d'une gaussienne. On a gagné en symétrie et sur la forme générale par rapport au cas précédent étudié au §3.1.4. Maintenant on va vérifier l'hypothèse de normalité des données $(Y_i - \hat{r}(X_i))/\hat{\sigma}^2(X_i)$ avec un QQPlot et des tests.





- test du QQPlot

L'hypothèse de normalité semble rejetée sur la globalité. Mais si on regarde le sous intervalle [-2,2] des quantiles de l'estimation de la loi μ on a une bonne adéquation à la loi normale. De plus sur cette intervalle on a environ 75% des individus (lois quasi symétrique et 2 quantile à 86.5%). Par contre sorti de cette intervalle on diverge rapidement.

- test de Shapiro

```
##
##  Shapiro-Wilk normality test
##
## data:  estMu_ks
## W = 0.88679, p-value < 2.2e-16
```

Le test de Shapiro-Wilk donne une de p-value significative $<$ 2.2e-16. L'hypothèse de normalité est rejetée.

- test de Kolmogorov-Smirnoff

```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  estMu_ks
## D = 0.15787, p-value < 2.2e-16
## alternative hypothesis: two-sided
```

La p-value est significative l'hypothèse de normalité est rejetée. D'après ces derniers tests la densité μ ne peut être gaussienne.