Régression non-paramétrique

Philippe Real

16/09/2019

#rmarkdown::render("MNP_DMFinal.Rmd") # html_document: default # pdf_document: default

```
#for manipulate data (transform to dataframe)
install.packages("tidyverse")
install.packages("tibble")
install.packages("sm")
install.packages("KernSmooth")
install.packages("np")
install.packages("stats")
install.packages("ggplot2")
```

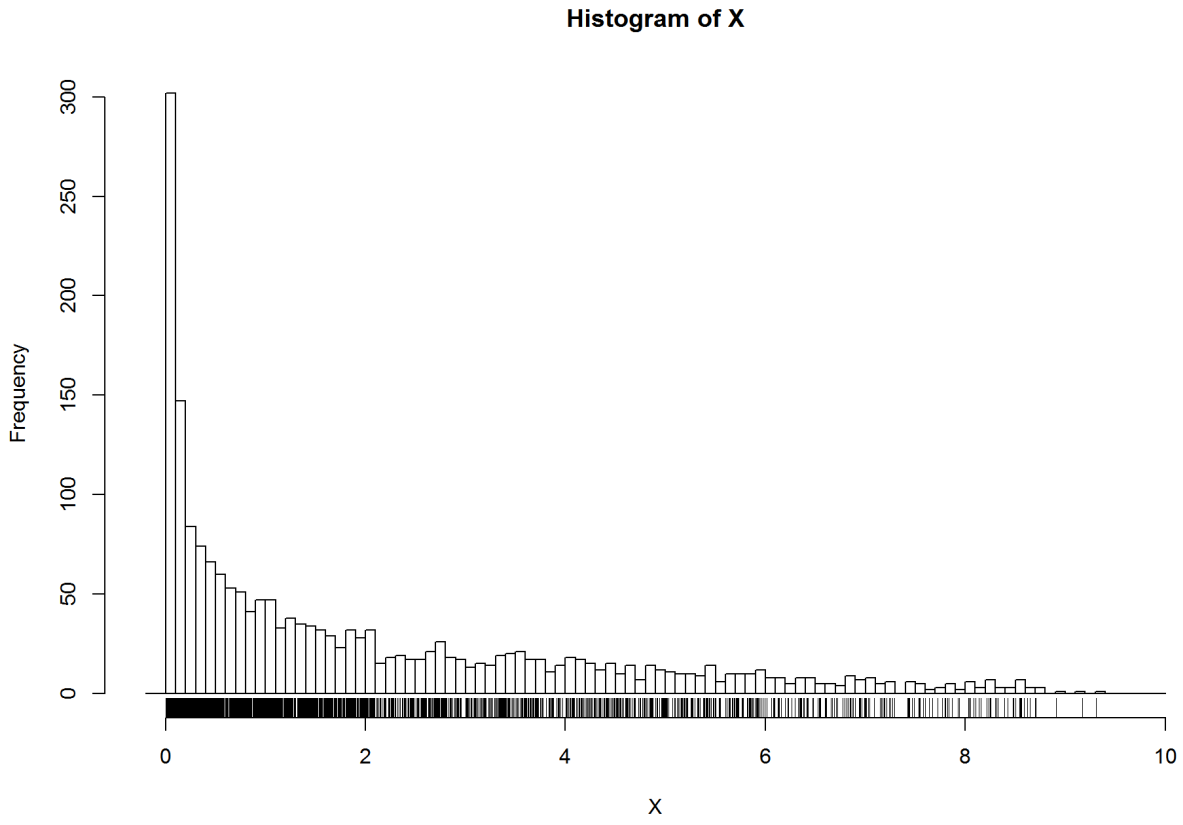
1. Etude de la densitée g des X

1.0 Lecture des données et premières analyses

```
d1 = read.csv("Data1.csv")
summary(d1)
```

```
##       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
```

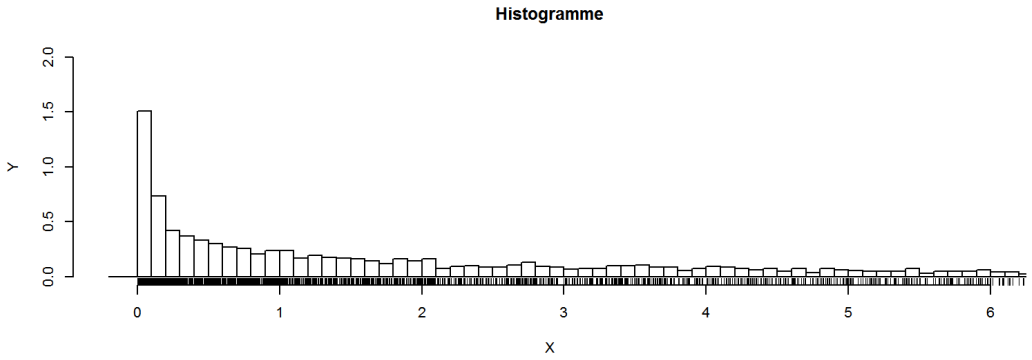
Pour avoir une idée de la densité de X on peut tracer son histogramme.

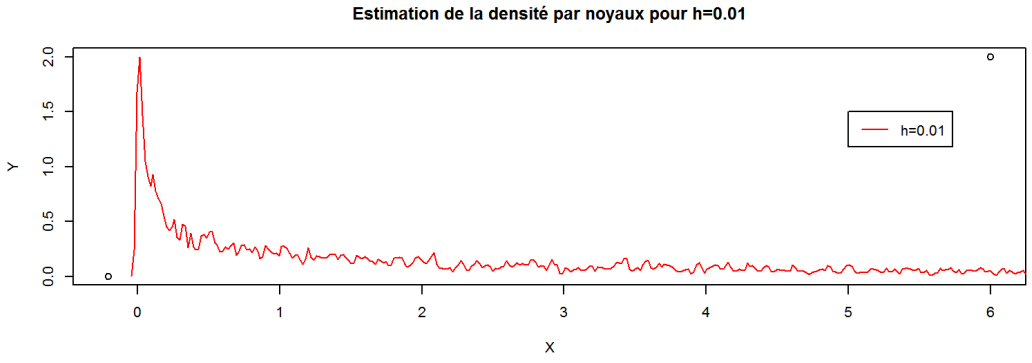


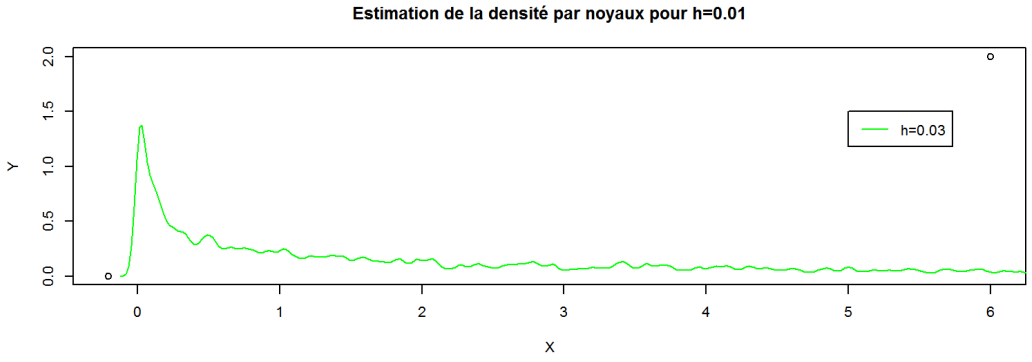
1.1 Estimateur non-paramétrique de g(x)

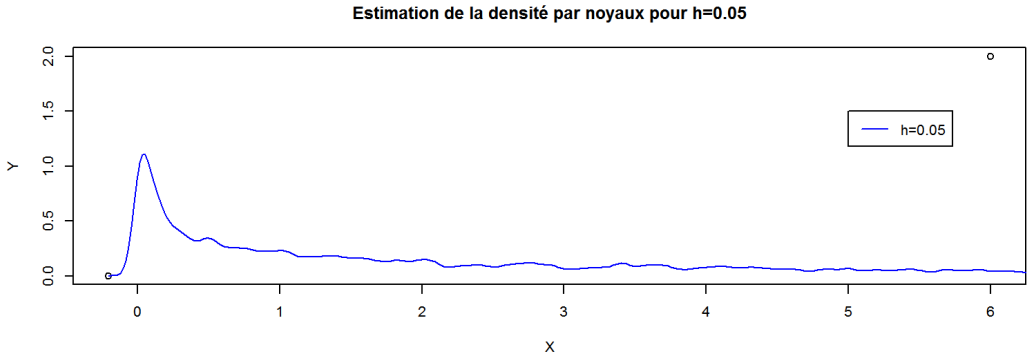
On peut utiliser la fonction bkde du package KernSmooth qui estime la densité par la méthode des noyaux. On prend comme noyau le noyau normal. Ce choix peut sembler arbitraire, mais on a vu que ce n'est pas le choix du noyau qui est le plus important dans l'estimation de la densité. On calcul cette estimateur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

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



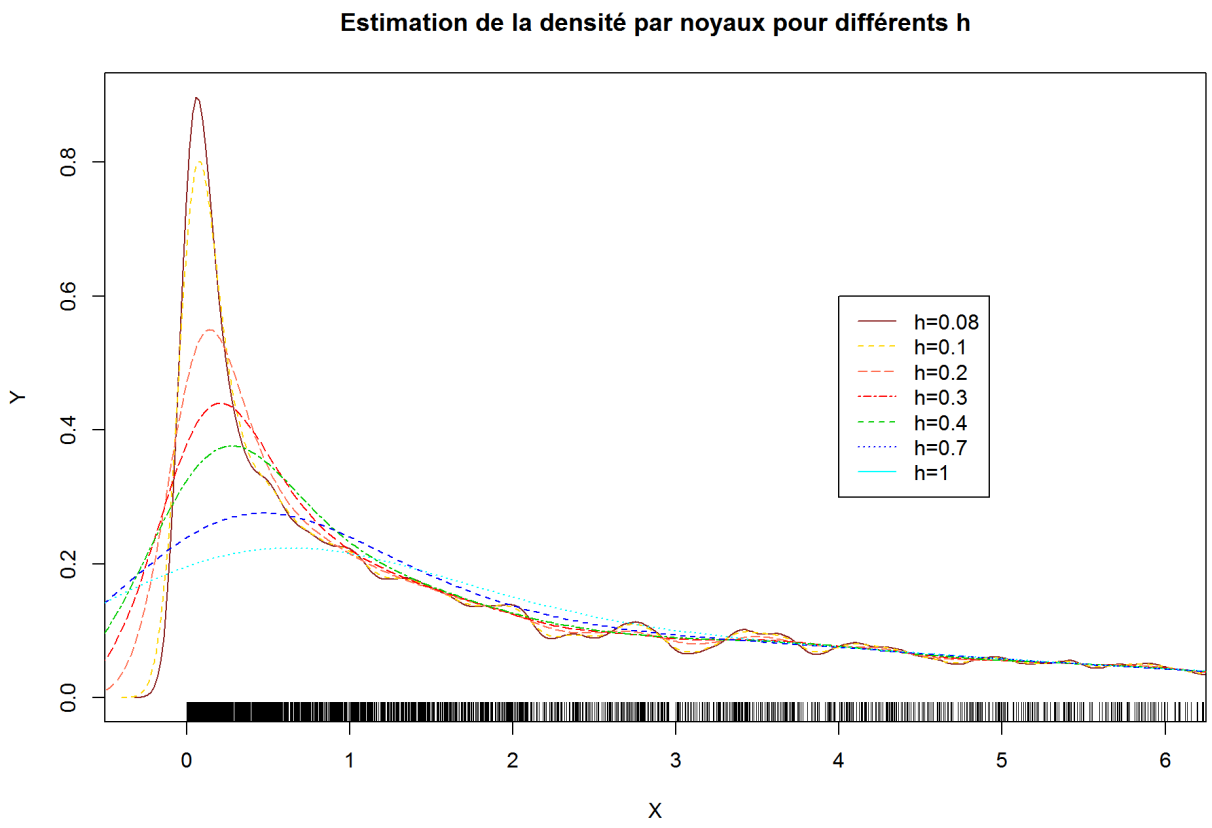






On remarque de fortes oscilations pour des h entre 0.01 et 0.05.

Estimateurs pour h variant de 0.08 à 1



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

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 : trop de biais
- Si h est trop petit (courbes marron du graphique ci-dessus) : trop oscillant, trop de variance.

1.2 Détermination d'un h optimal

Représentation graphique de l'estimation par noyau de la densité ed X: g(x), où h.n est la fenêtre donnee par validation croisee ou par une autre methode que l'on precisera.

1.2.1 Validation croisée pour la densité,

Utilisation de la fonction bw.ucv library stats

```
h_ucv<-bw.ucv(X)
h_ucv
```

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

1.2.2 Règle Silverman

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

```
n<-length(X)
h_sil<-1.06*sqrt(var(X))*n**(-1/5)
h_sil
```

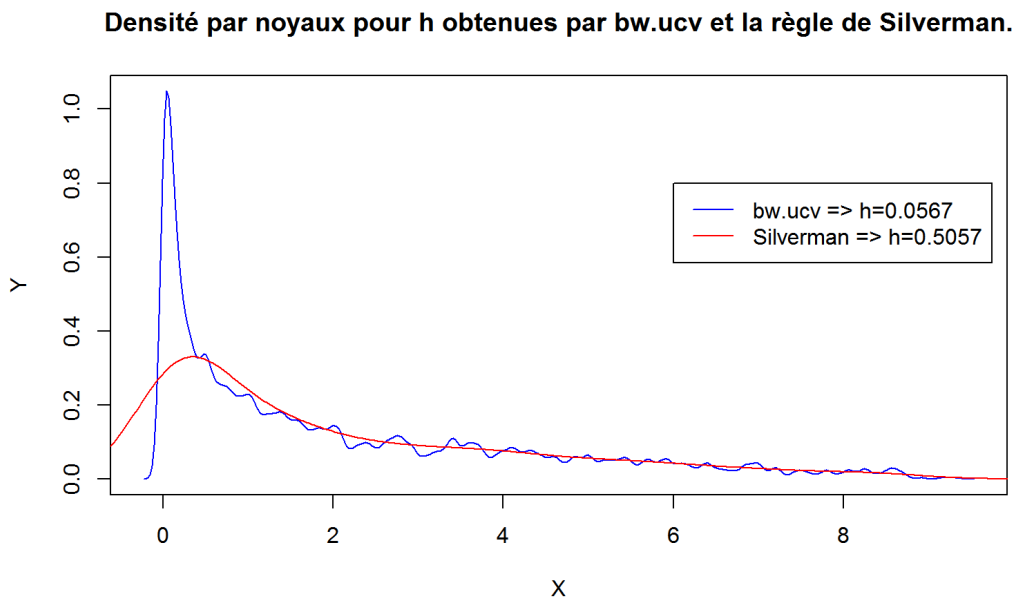
```
## [1] 0.505695
```

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

Fonction bw.nrd

```
h_nrd <- bw.nrd(x = X)
h_nrd
```

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



1.2.3 Méthodes alternatives

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
```

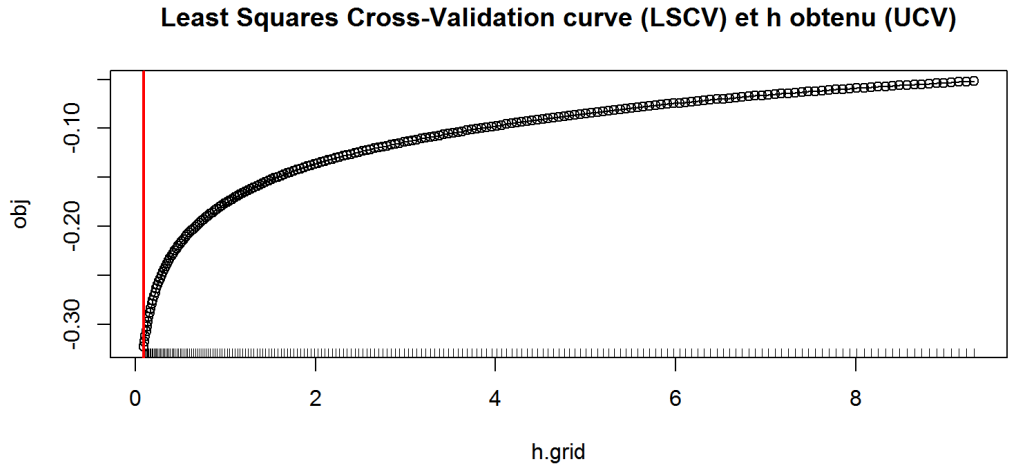
Fonction dpik

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

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

Fonction ucv

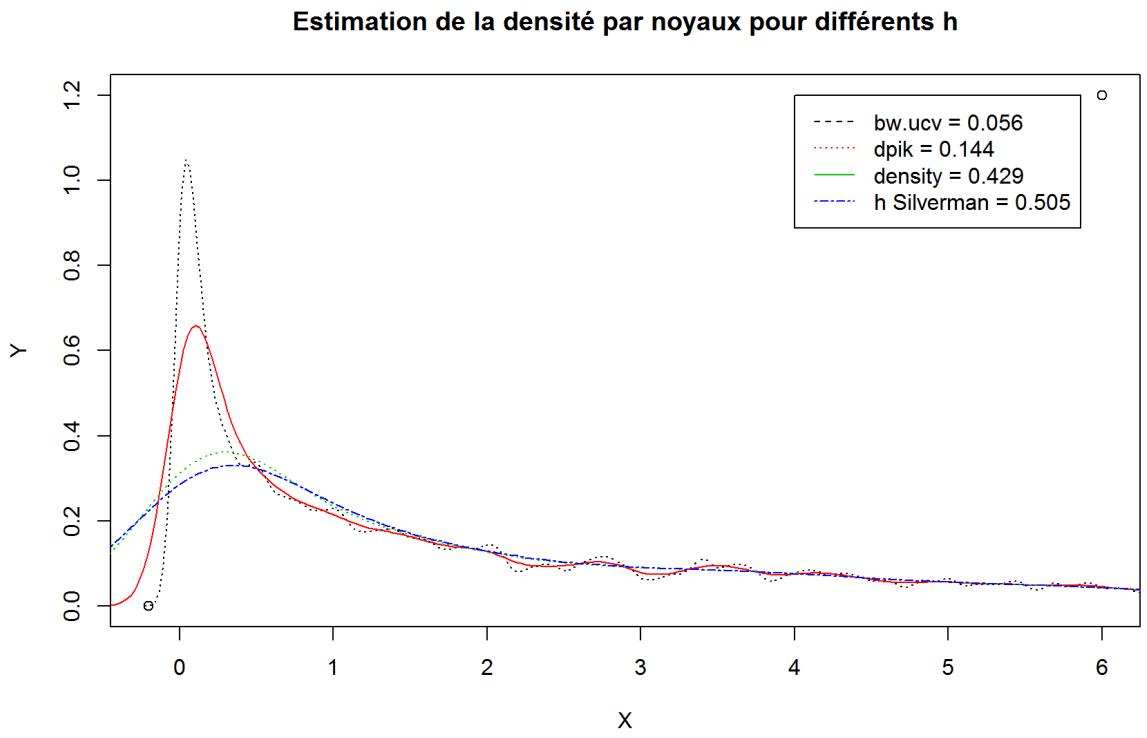
La fenêtre h est obtenu par (UCV) de Least Squares Cross-Validation curve (LSCV) et h obtenu (UCV)




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

Résumé des résultats

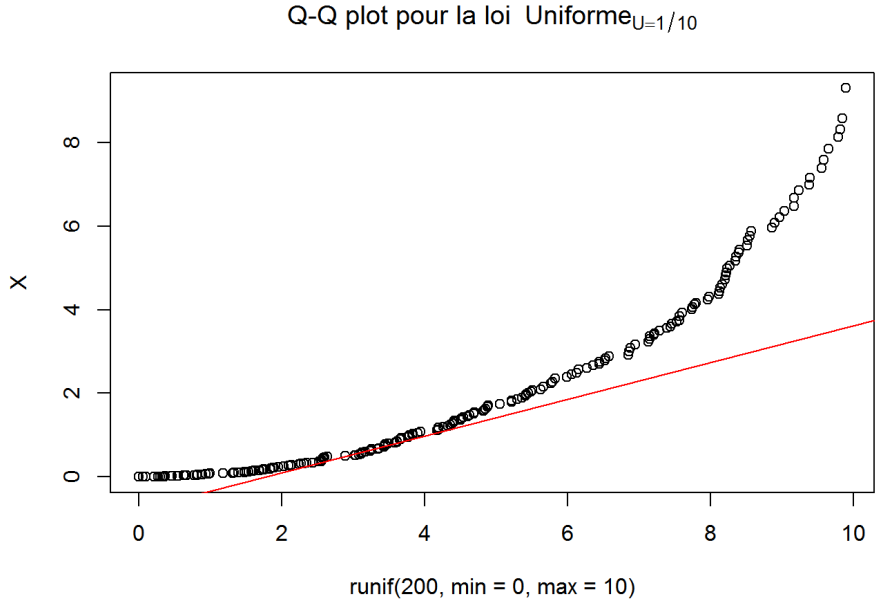
Méthode	valeur de h
bw.ucv	0.0567514
dpik	0.1439489
density	0.4293637
Regle Silverman	0.505695



###

1.3 QQPlot - g(x) densité Uniforme ?

Implementation d'un QQ-plot pour vérifier empiriquement l'hypothese 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 raisonnable. Cependant entre les valeurs en abscisse [2,6] l'hypothèse semble plus crédible. On pourrait diviser l'espace en 3 parties: [0 , 2] [2 , 6] [6 , 10]

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

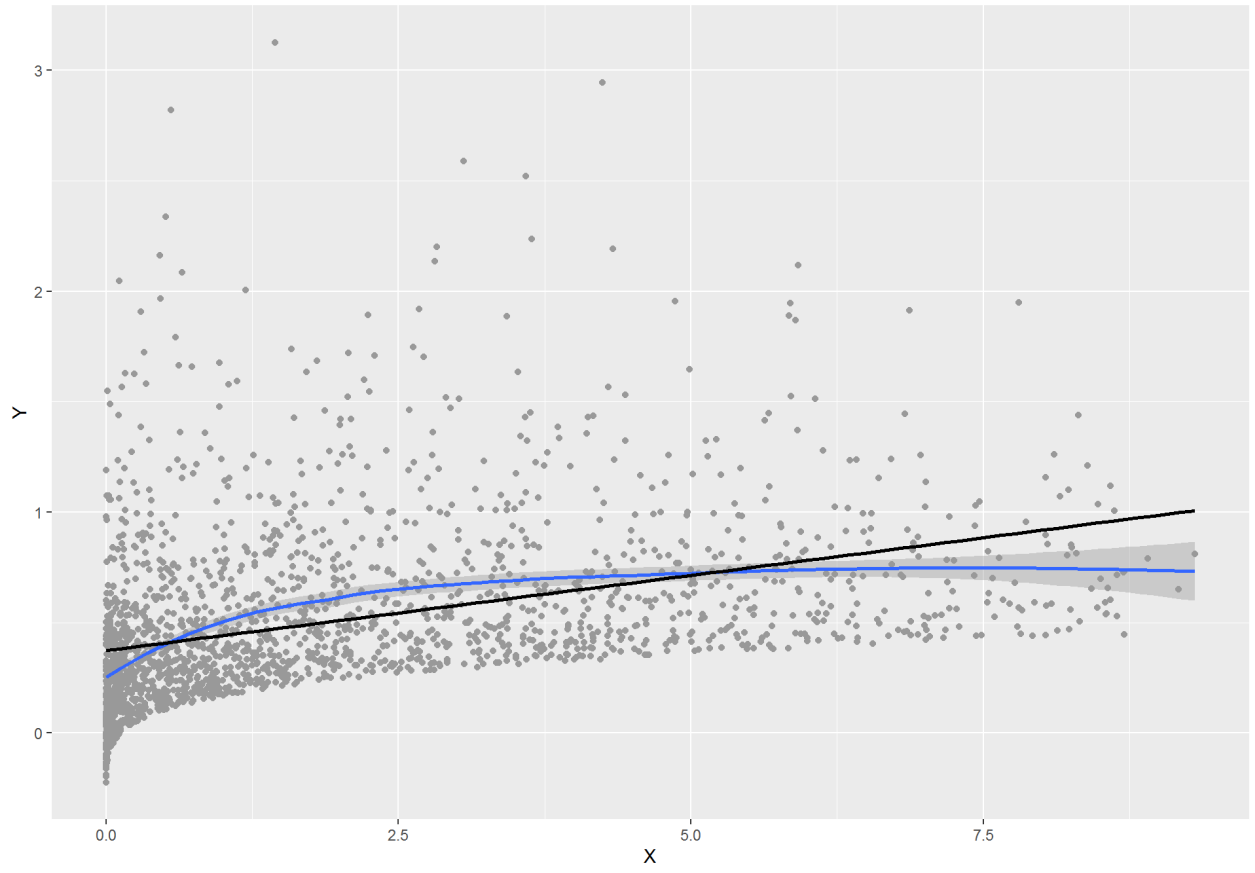
Plus de précision où les données ne sont pas trop dispersées et ou la densité est importante. Donc plus de précision dans l'intervalle [0+dela , 5]
La précision diminue à droite… Dans un voisinage de 0 l'estimation n'est pas précise non plus: difficulté d'estimer où g(x) = 0

2. Reconstruction de r(x)

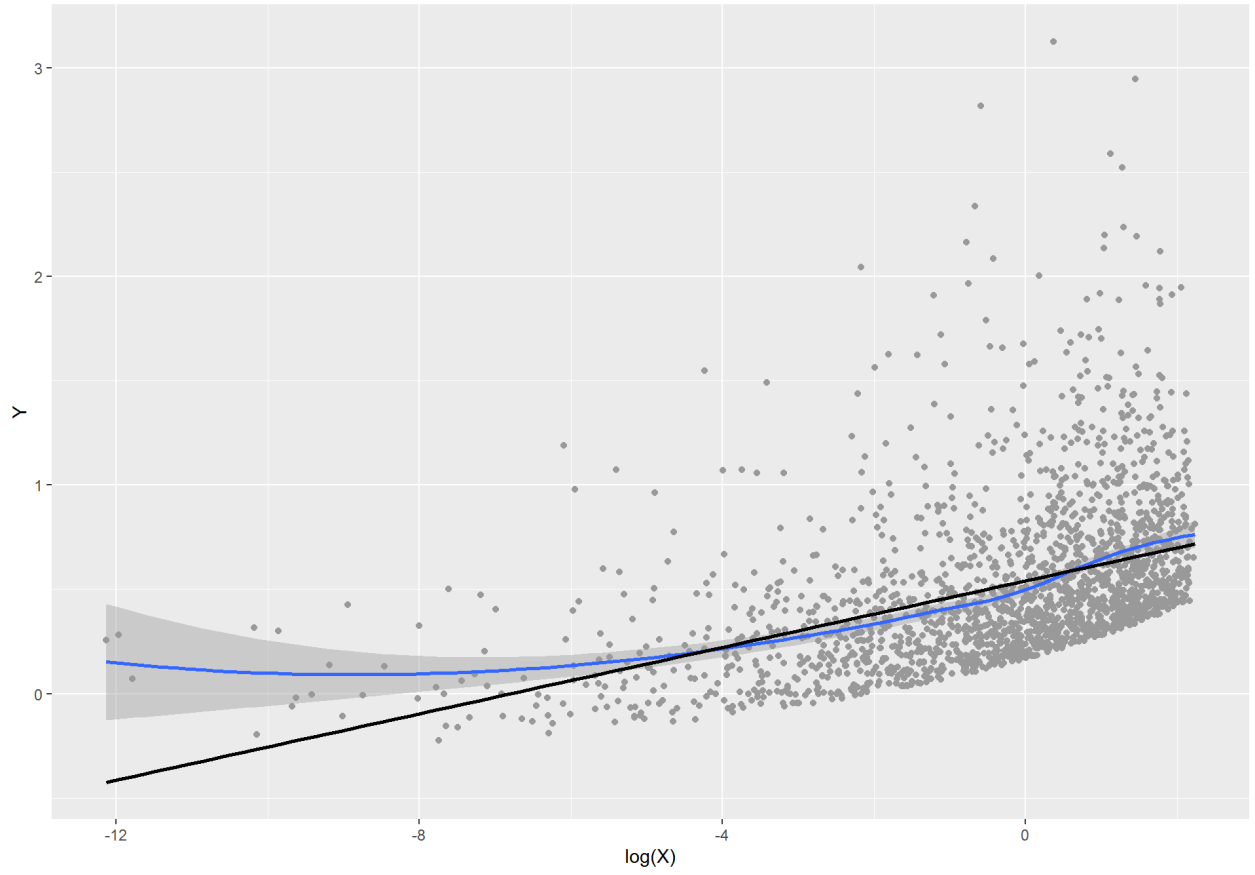
On est dans le cadre de l'estimation non paramétrique, on reprend les hypothèses classiques On utilise les données de Data1, (X,Y)

2.1 La fonction r peut elle être linéaire ?

On trace Y1 en fonction de X Sans transformation, à la vue du graphe, r ne peut être linéaire.



Maintenant On trace Y1 en fonction de log(X)



Dans la région [-5,1] où l'on retouve la quasi totalité de l'échantillon, la transfoarmtion (log(x)) a permis de bien linéariser.

2.2. Construction d'un estimateur non-parametrique de r(x)

Détermination de la fenêtre h

Différentes fenêtres h obtenues à partir de différentes métodes: fonction dpill de R, h de Silverman, validation croisée.

```
## [1] "h_dpill : 0.218684860290525 h_silver : 0.505695020156574 h_CV1 ; 0.337185929648241 h_CV2 ; 0.120060113212087"
```

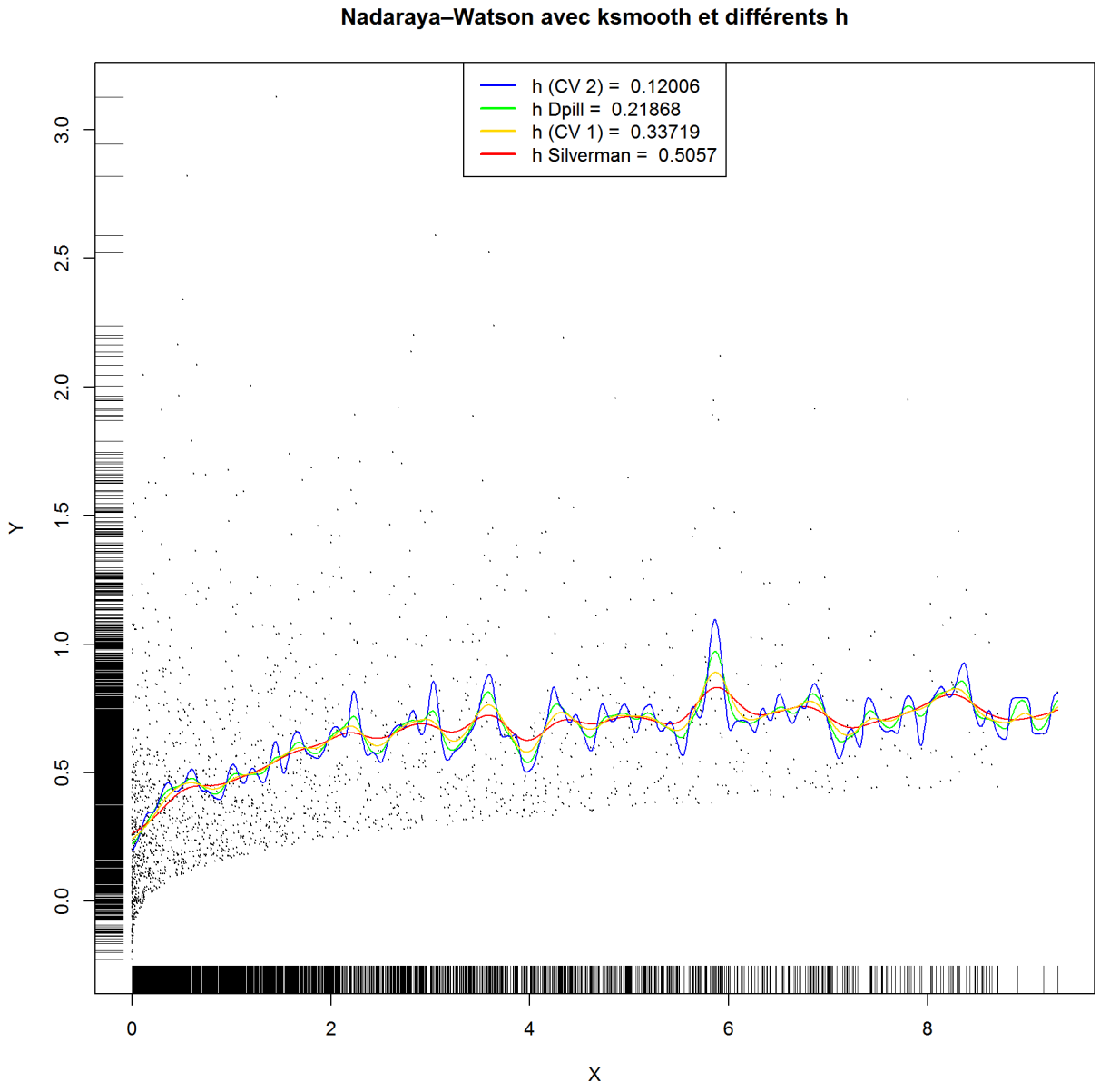
Test: choix de h local

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## 0.005019 0.014504 0.036759 0.059364 0.078701 0.173141
```

On remarque que, la valeur médiane 0.036759 est assez proche du résultat trouvé pour le h global à partir de la validation croisée: h_CV1

Estimateur de Nadaraya-Watson avec la librairie stat - fonction : ksmooth

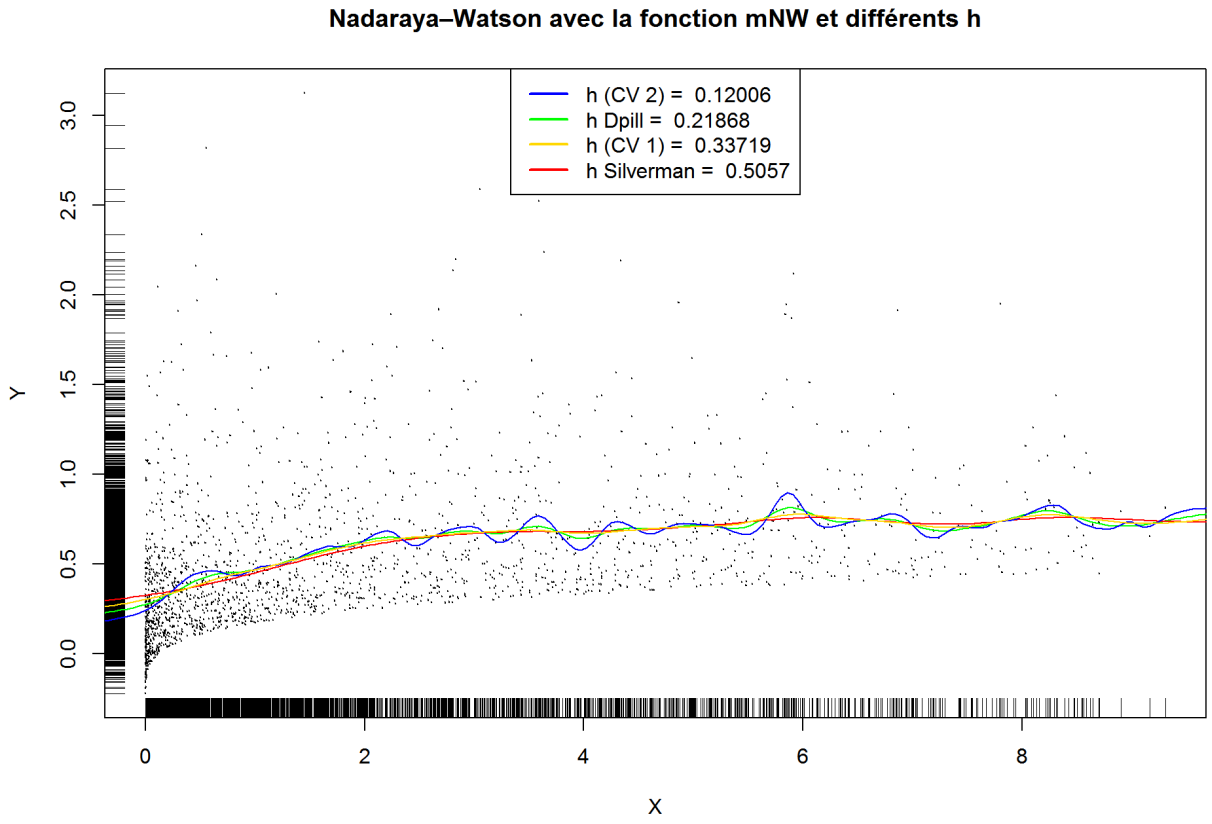
Utilisation de la fonction ksmooth et différentes fenêtre h_dpill=0.2186849, h_silver=0.505695, h_CV1=0.3371859, h_CV2=0.1200601



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

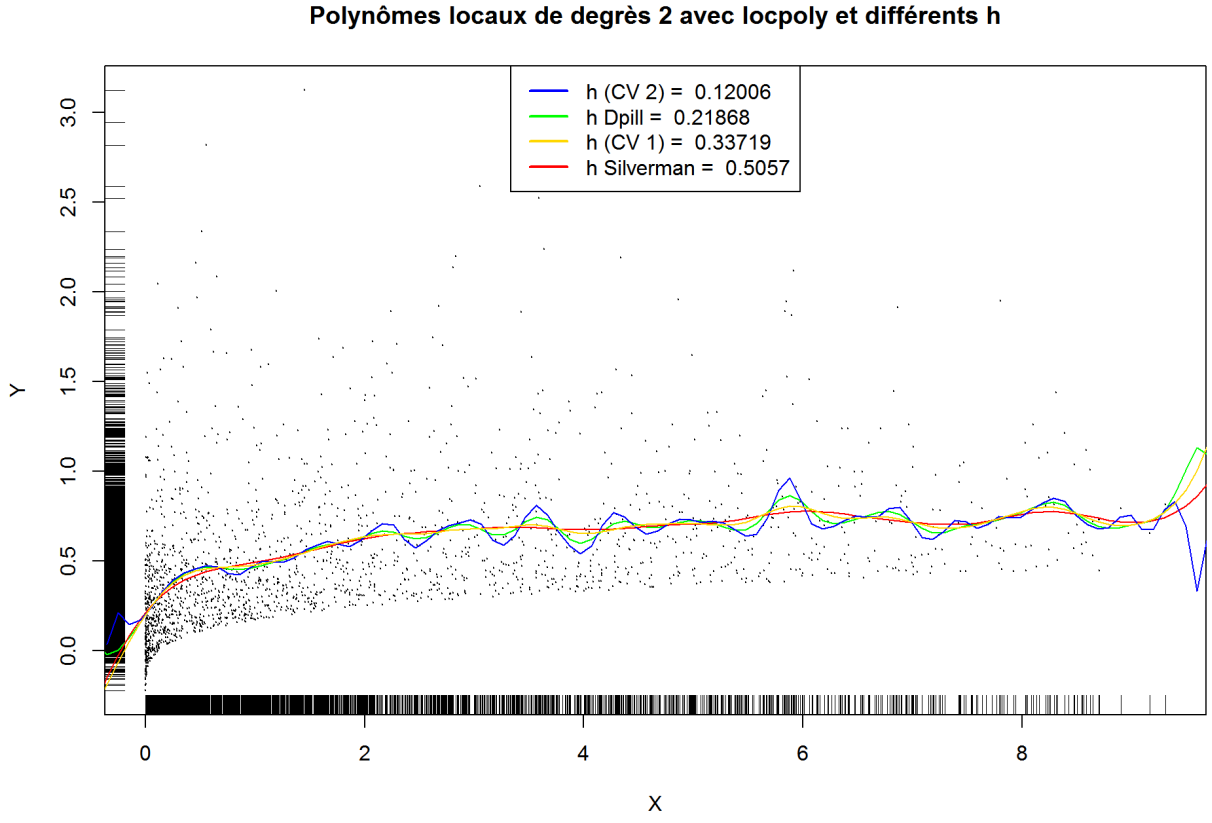
Estimateur de Nadaraya-Watson à partir de la fonction recodé NW

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



Estimateur 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 choisie le degès 2



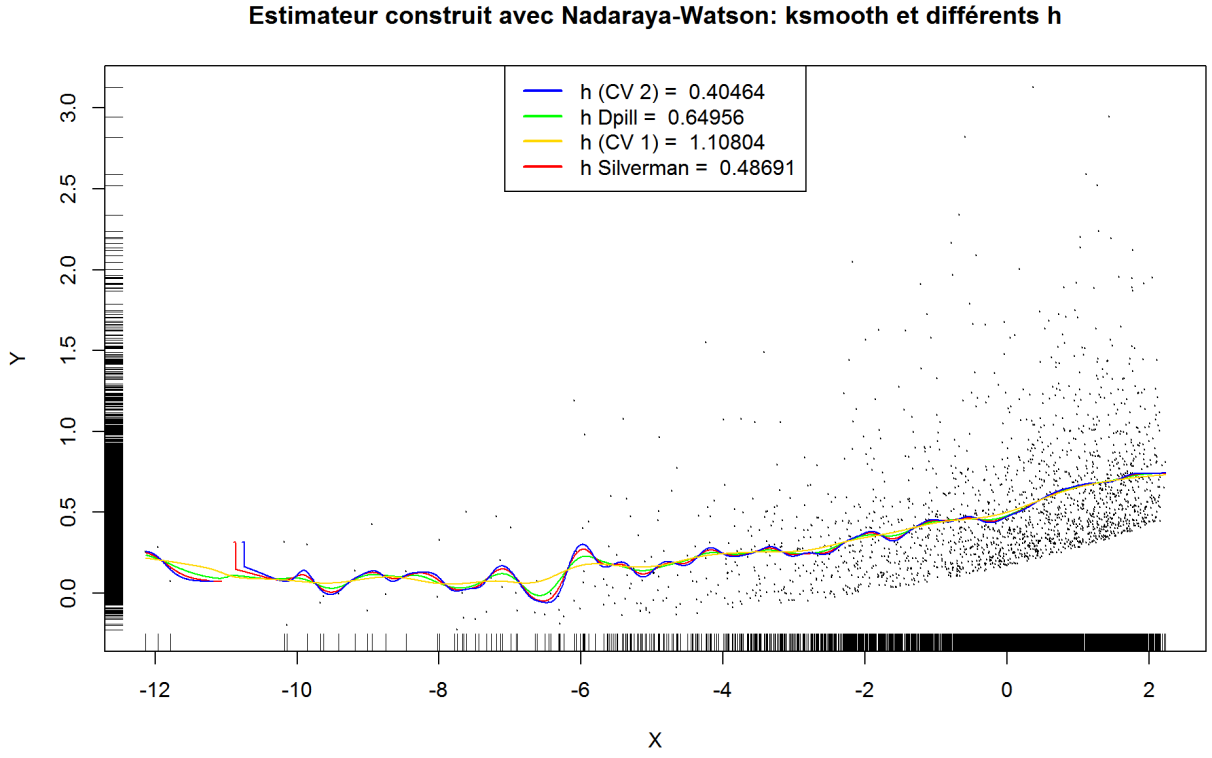
2.3. estimation de r en regressant Y1 sur log(X)

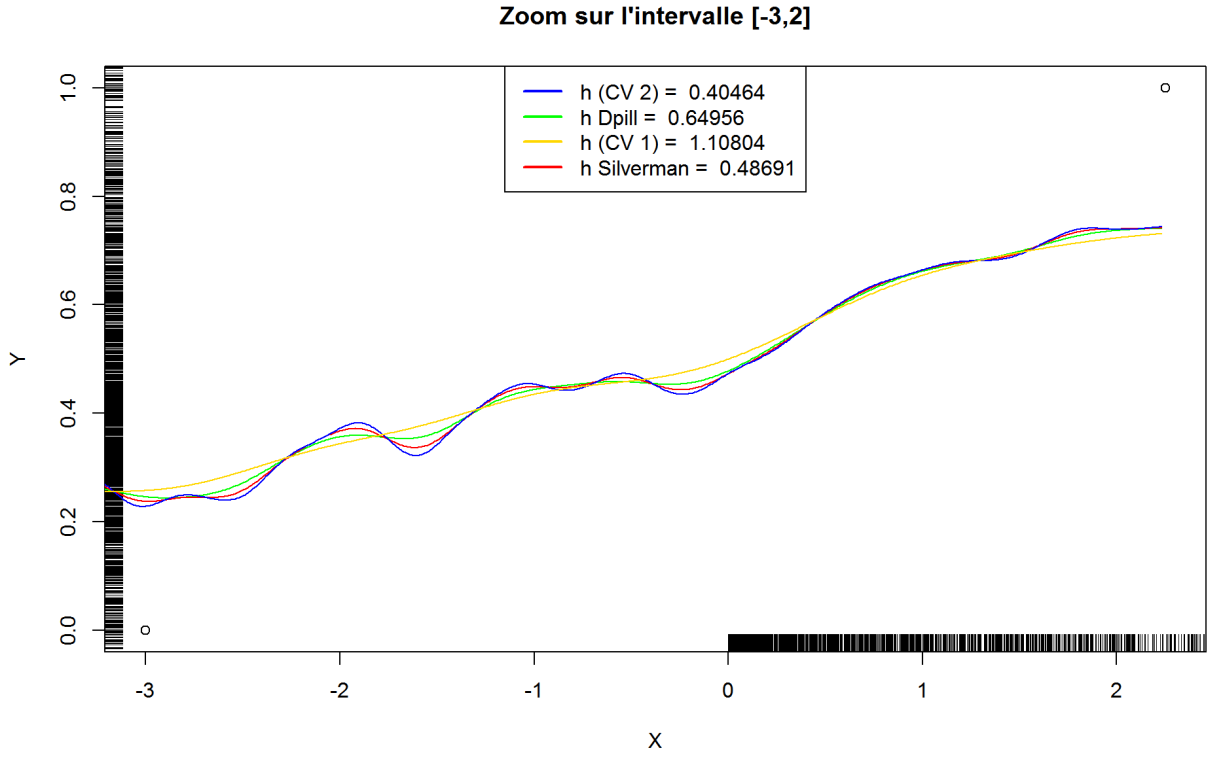
Choix du h optimal à partir de la fonction dpill, de la règle de Silverman et par validation croisée

```
## [1] "h_dpill : 0.649558017143598 h_silver : 0.48691251968111 h_CV1 ; 1.10804020100503 h_CV2 ; 0.404638091616267"
```

Estimation de ˜r avec Nadaraya-Watson avec la librairie stat - fonction : ksmooth

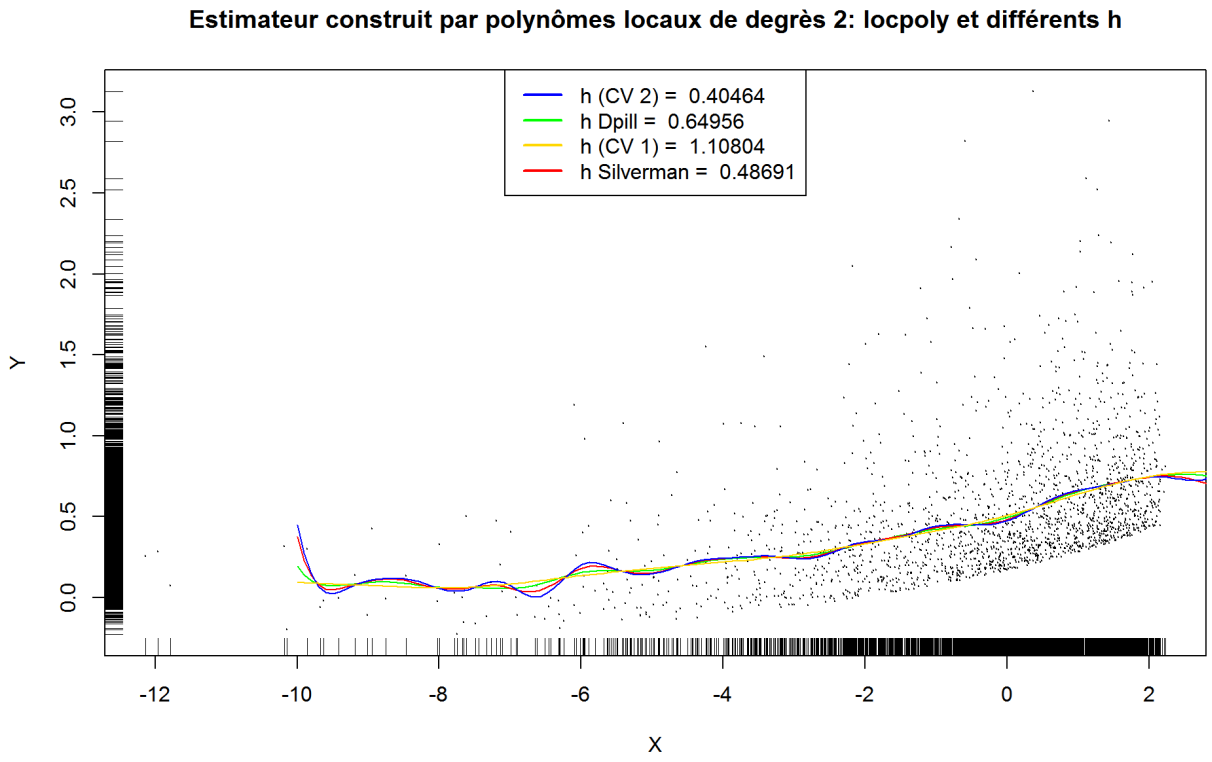
Utilisation de la fonction ksmooth avec différentes fenêtres h_dpill=0.649558, h_silverman=0.4869125, h_CV1=1.1080402, h_CV2=0.4046381

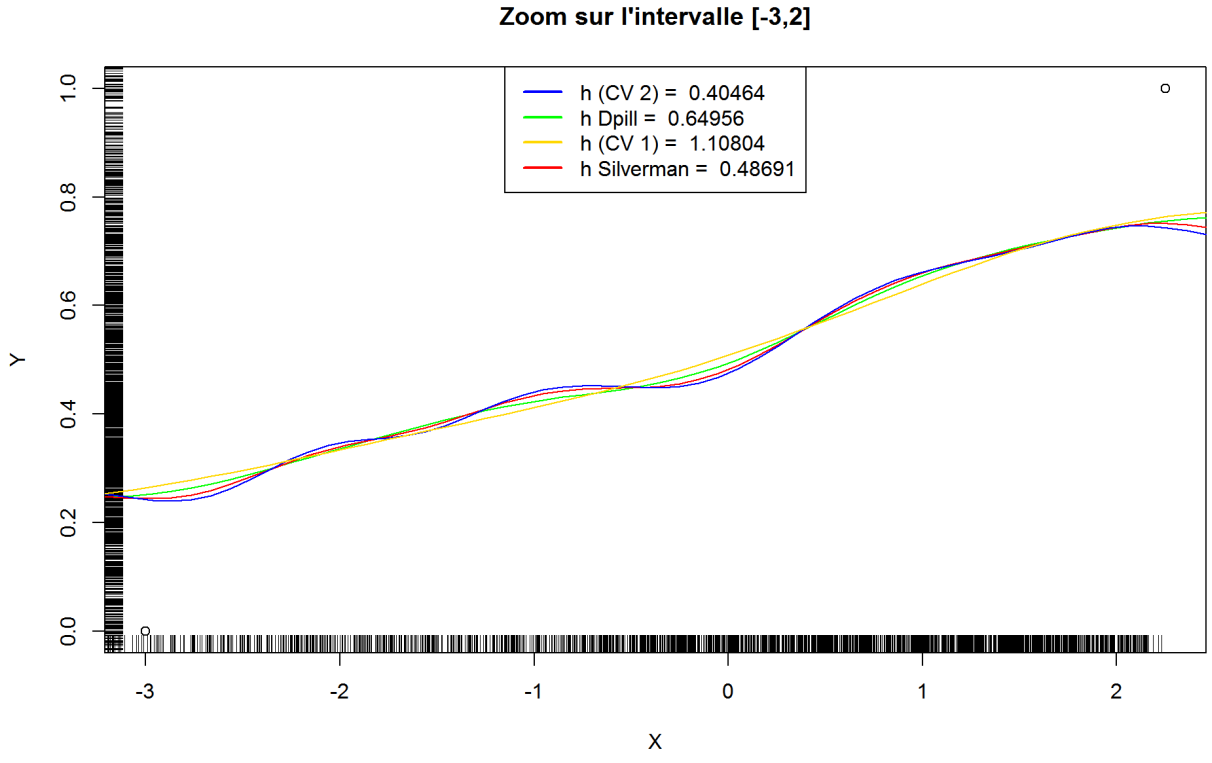




Estimateur de ˜r(x) par polynômes locaux de degrès 2

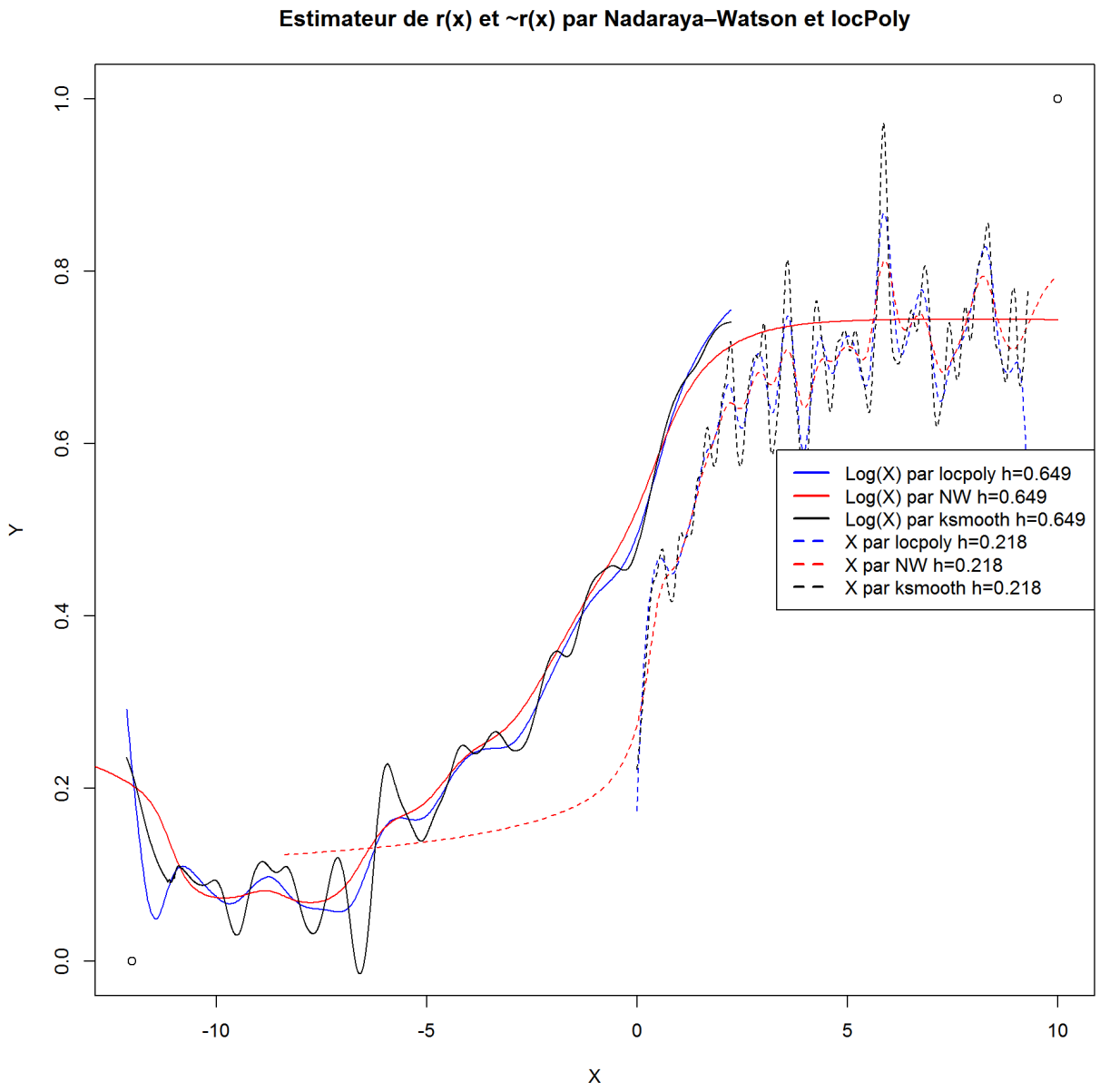
Utilisation de la fonction locpoly avec différentes fenêtres h_dpill=0.649558, h_silverman=0.4869125, h_CV1=1.1080402, h_CV2=0.4046381





Dans la zone à forte densité où est concentrée l'essentiel de l'information [-5,1] les 2 estimateurs sont très proches, quasiment confondus.

Représentation sur un même graphe de ˆr(x) et ˜r(x)



2.4. Remarques? Explications?

Régularité proportionalité?

3. Etude de la densitée μ(x) des ξi

3.1 A partir du jeu de donnees Data1

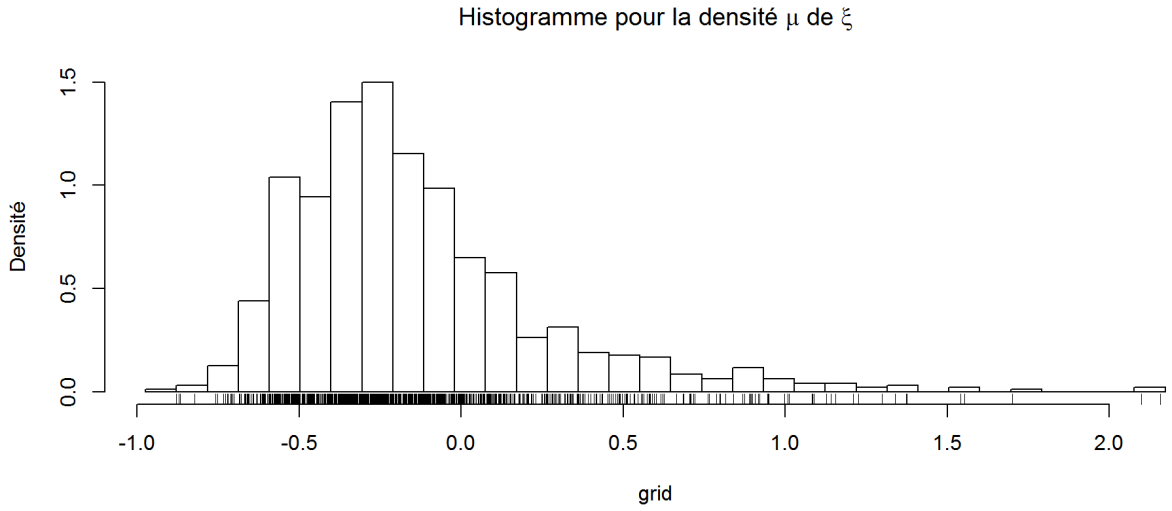
3.1.1 Distribution approximative de $\tilde{\xi_i}$

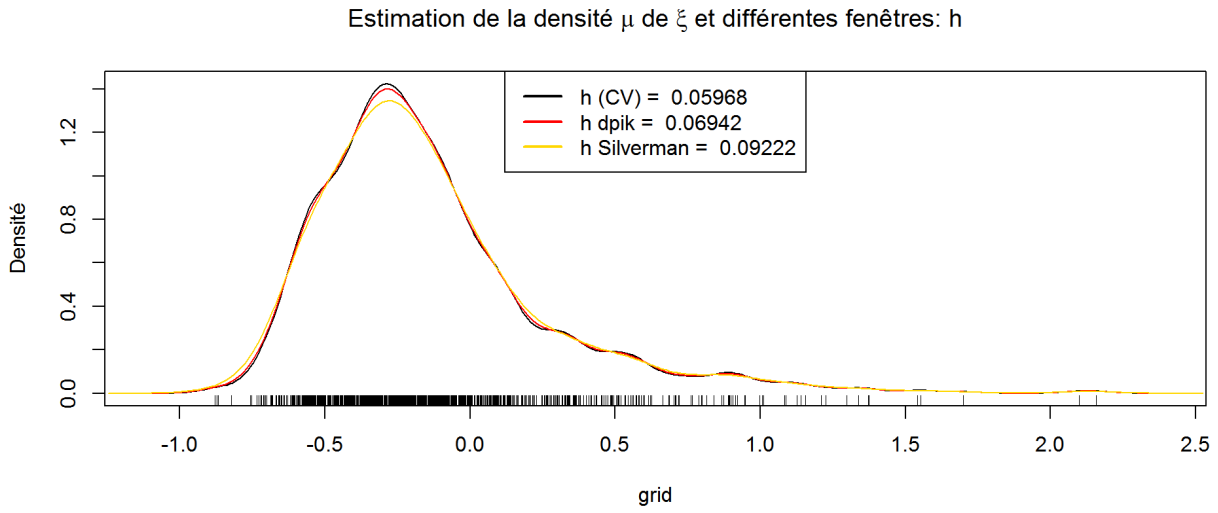
On a la représentation suivante : $Y_i - r(X_i) = \sigma * \xi_i$ (cas homoscédastique σ est constant)
Par définition $\tilde{\xi_i} = Y_i - \hat{r}(X_i)$ où $\hat{r}(x)$ est un estimateur de r(x).
La distribution approximative de $\tilde{\xi_i}$ est celle de ξ_i à la constante multiplicative près: σ

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

choix de h établi a la question 2.2: h_dpill=0.2186849

```
## [1] "h_dpik : 0.0694249883767452 h_silver : 0.092216198597302 h_density ; 0.0681078157621856 h_ucv ; 0.0596771927200419"
```





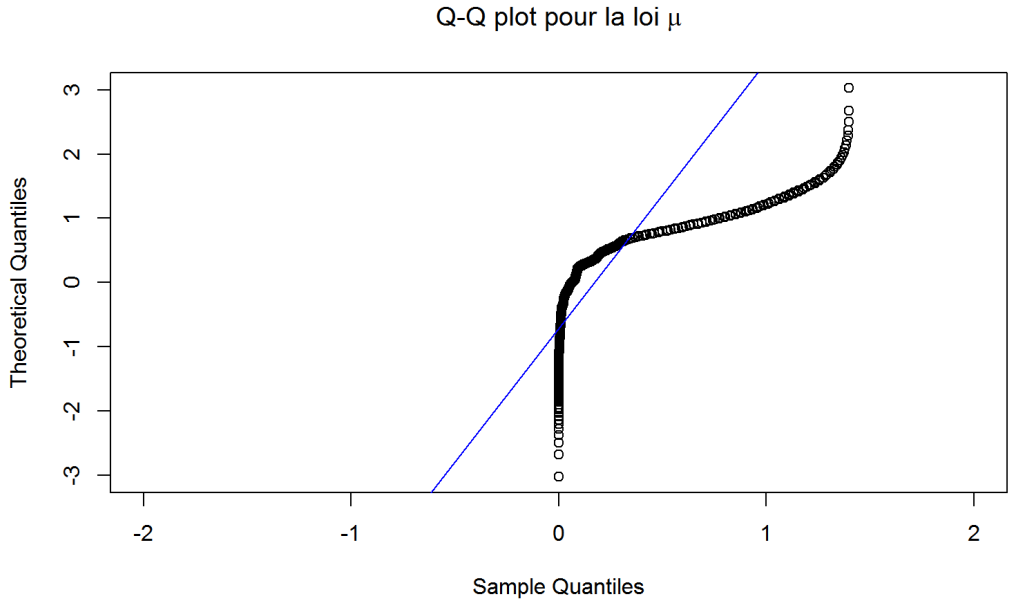
3.1.3. (Facultatif.) Quel est l'interêt d'avoir decoupé le jeu de donnees selon J+ et J- ?

On a aisni un jeux de données d'apprentissage et de test. On peut utiliser le jeux de données d'apprentissage pour estimer et construire nos modèles, 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 modèle.

3.1.4. La densitée μ peut-elle être gaussienne ?

Protocole empirique de verification http://www.biostat.ulg.ac.be/pages/Site_r/Normalite.html (http://www.biostat.ulg.ac.be/pages/Site_r/Normalite.html)

- 1. test du QQPlot



Le QQPlot graphique d'adéquation des quantiles rejette l'hypothèse de normalité.

- 2. test de shapiro

```
##
##  Shapiro-Wilk normality test
##
## data:  nu_hdpik$y
## W = 0.69508, p-value < 2.2e-16
```

Le test de Shapiro-Wilk donne une probabilité de dépassement de p-value < 2.2e-16, nettement < à 0.05. L'hypothèse de normalité est rejettéee.

- 3. test de Kolmogorov-Smirnoff Dans ce cas-ci également, il existe dans R une commande pour tester l'ajustement de données à une loi normale via le test de Kolmogorov-Smirnov:

```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  nu_hdpik$y
## D = 0.26571, p-value < 2.2e-16
## alternative hypothesis: two-sided
```

p-value faible rejetté.

3.1.5. homoscedasticité du modèle

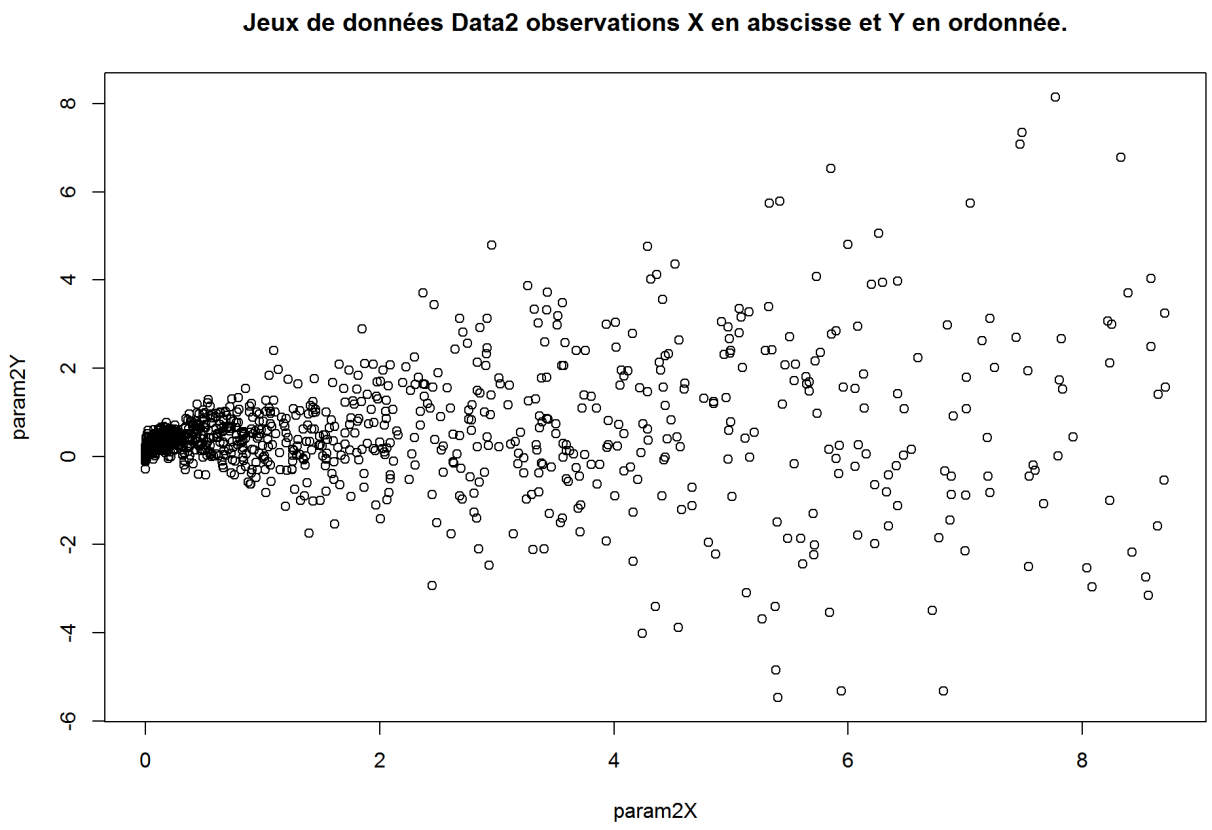
(Facultatif.) Comment peut-on tester si le modele est bien homoscedastique ? On peut tracer un graphe des résidus.

3.2 A partir du jeu de donnees Data2

On cherche a estimer μ(x) et $\sigma^2(x)$. Pour cela, on coupe a nouveau l'echantillon en deux et on considere a nouveau ξi:

```
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
```



3.2.1. Justifier qu'en regressant ξi sur X_i on obtient un estimateur de $\sigma^2(x)$

Par définition $\tilde{\xi}_i = Y_i - \hat{r}(X_i)$ où $\hat{r}(x)$ est un estimateur de r(x). $\tilde{\xi_i^2} = (Y_i - \hat{r}(X_i))^2$

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

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 le résultat.

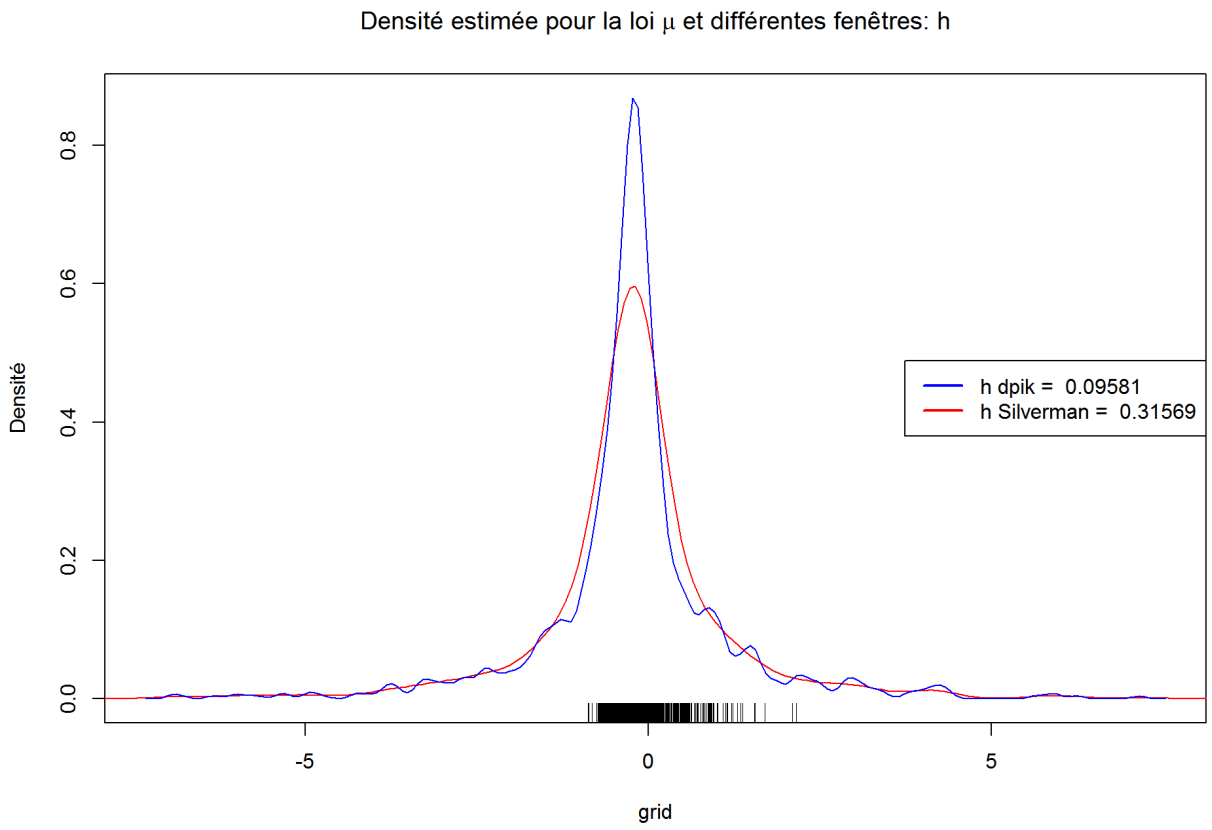
Implémentation et visualisation

A partir de la fenêtre h etablie à la question 2.2: h=0.2186849 on calcul un h optimal pour estimer la densité de: $\tilde{\xi}_i = Y_i - \hat{r}(X_i)$

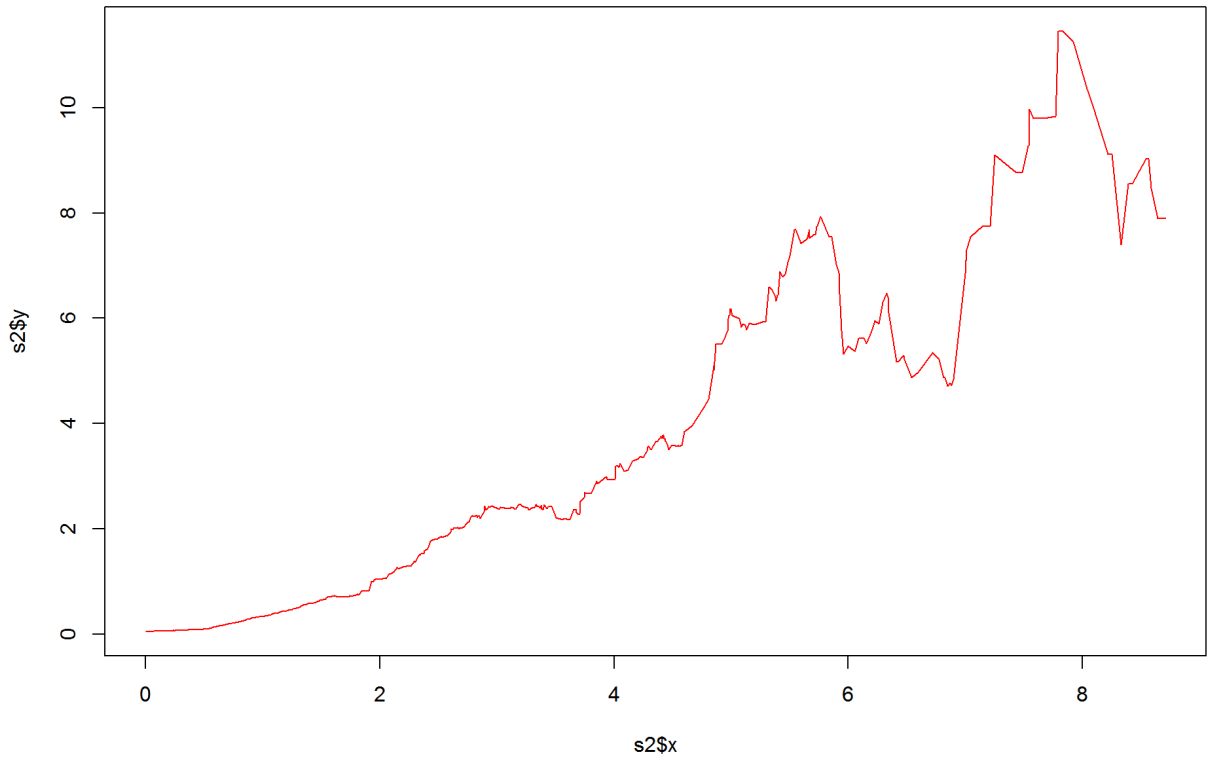
On ne conserve que le h obtenu par la fonction dpik (quasimment identique à celui obtenu par CV) et la règle de Silverman

```
## [1] "h_dpik : 0.095813007683831 h_silver : 0.315688530463266 h_ucv : 0.0958357155156441"
```

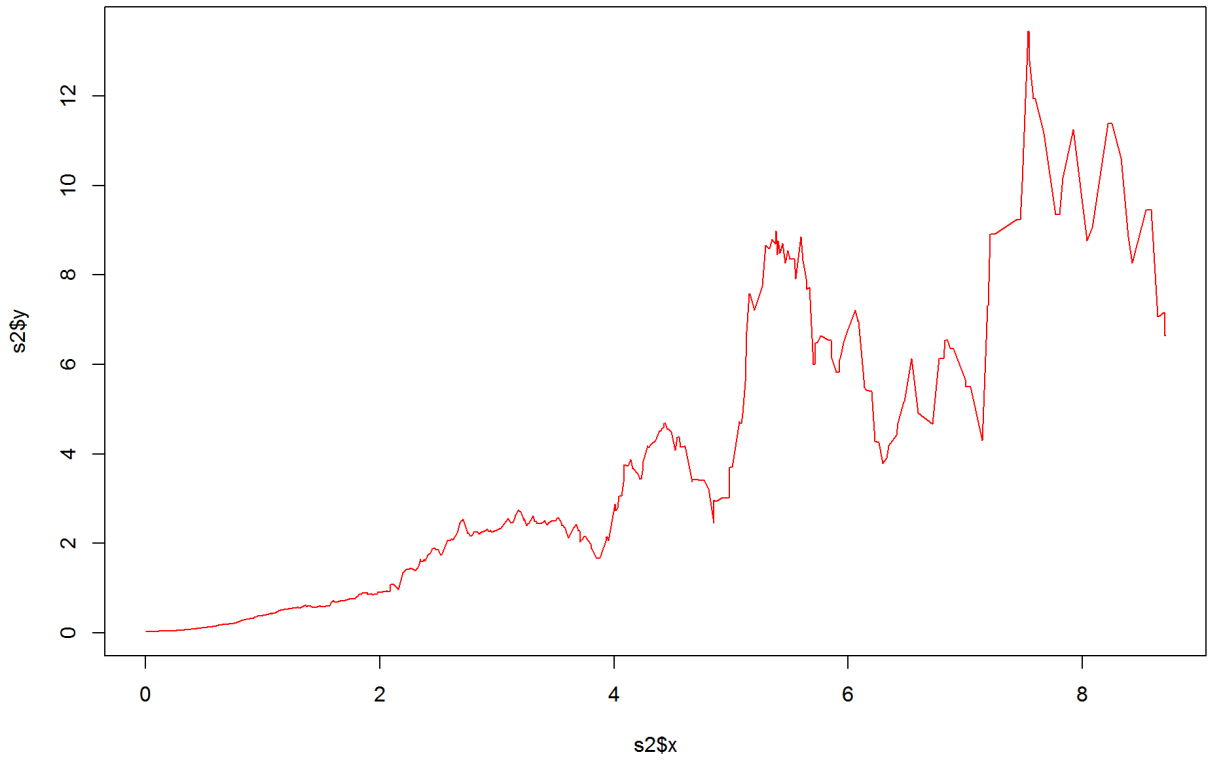
Puis à partir de la fonction bkde une estimation de la densité de : $\tilde{\xi}_i = Y_i - \hat{r}(X_i)$ Que l'on trace en fonction des différentes fenêtres obtenues précédemment.



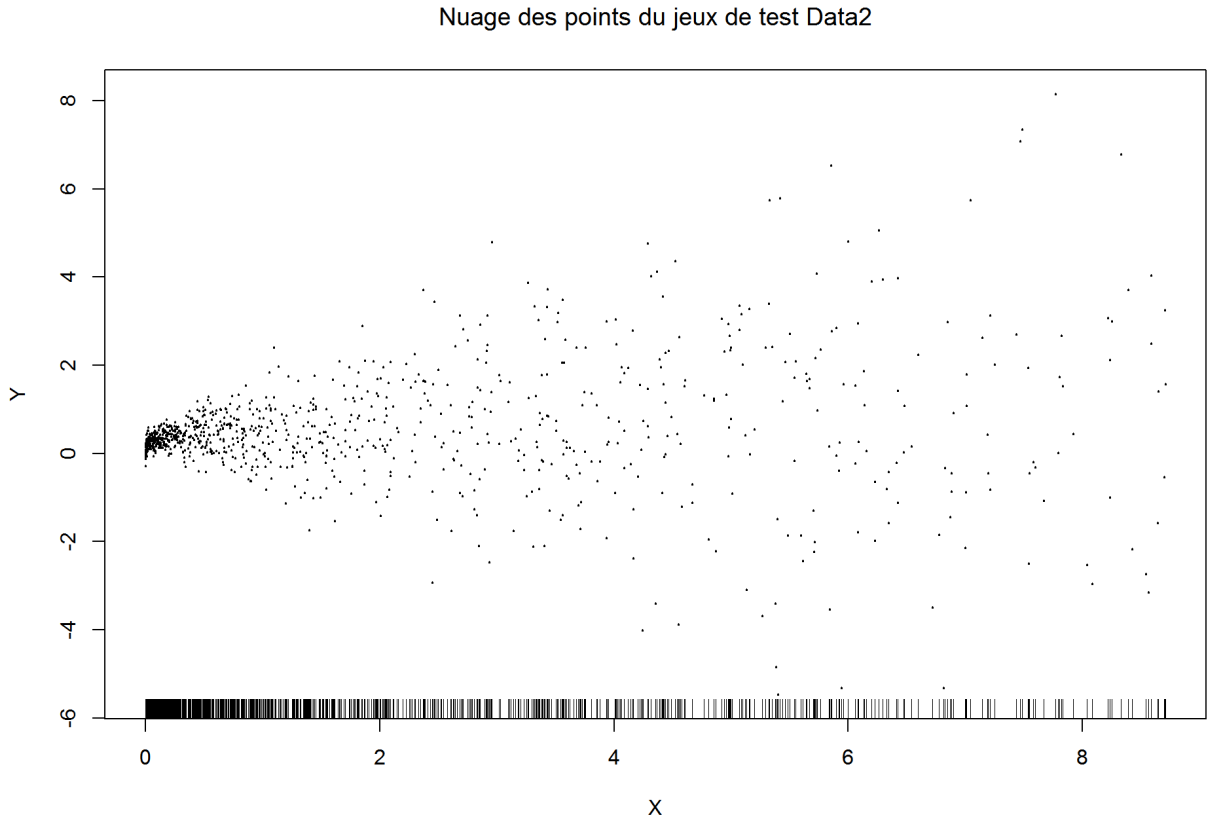
A partir de la fonction de densité estimée on va construire un estimateur de $sigma(X_i)^2$ On calcul par polynômes locaux de degès 2: locpoly la régression de de ξ^2 sur X_i

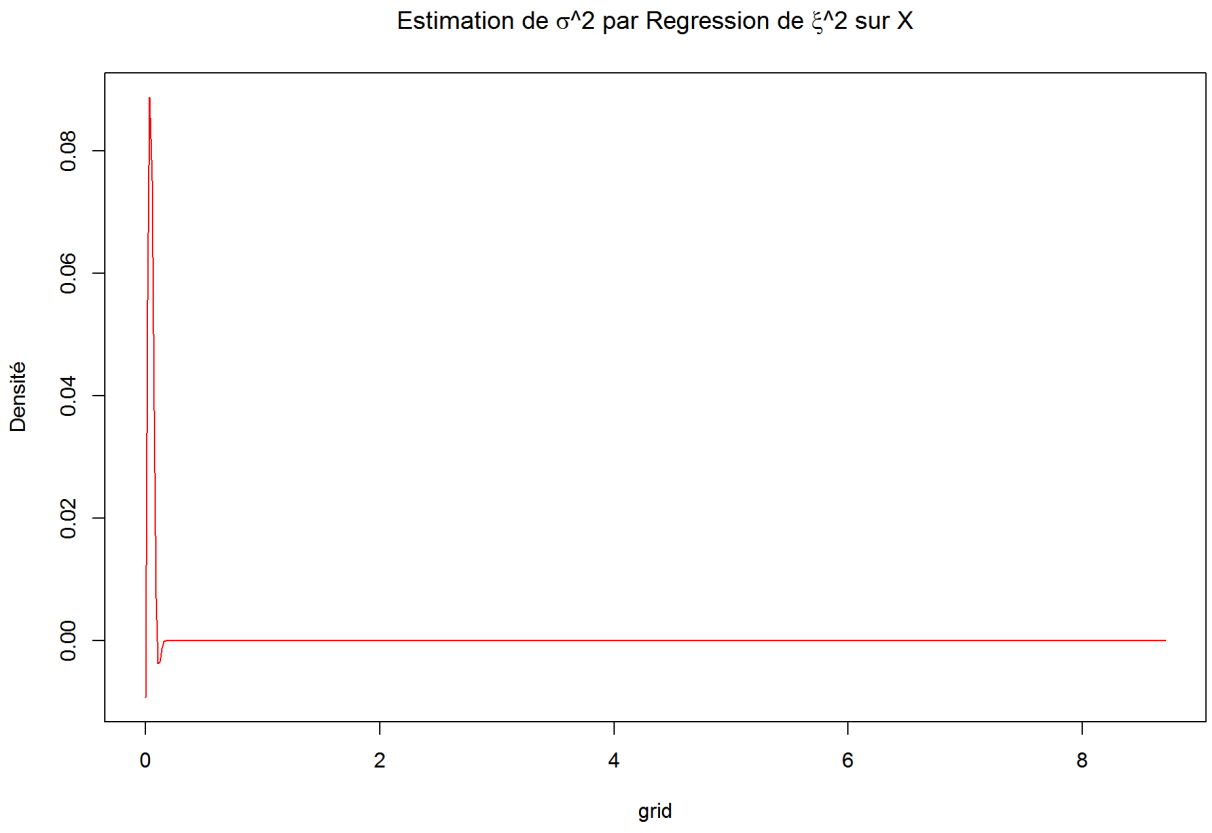



```
## [1] "Err prév initiale : 21.1249870291304 Err prév cas Het : 15.3557484542386"
```




```
## [1] "Err prév initiale : 21.1249870291304 Err prév cas Het : 14.8625877534132"
```






```
## [1] "Err prév initiale : 28.6381132277841 Err prév cas Het : 21.6330356697481"
```

`

En comparant avec le jeu de donnees (Figure 1 a droite), retrouve-t-on un resultat attendu:

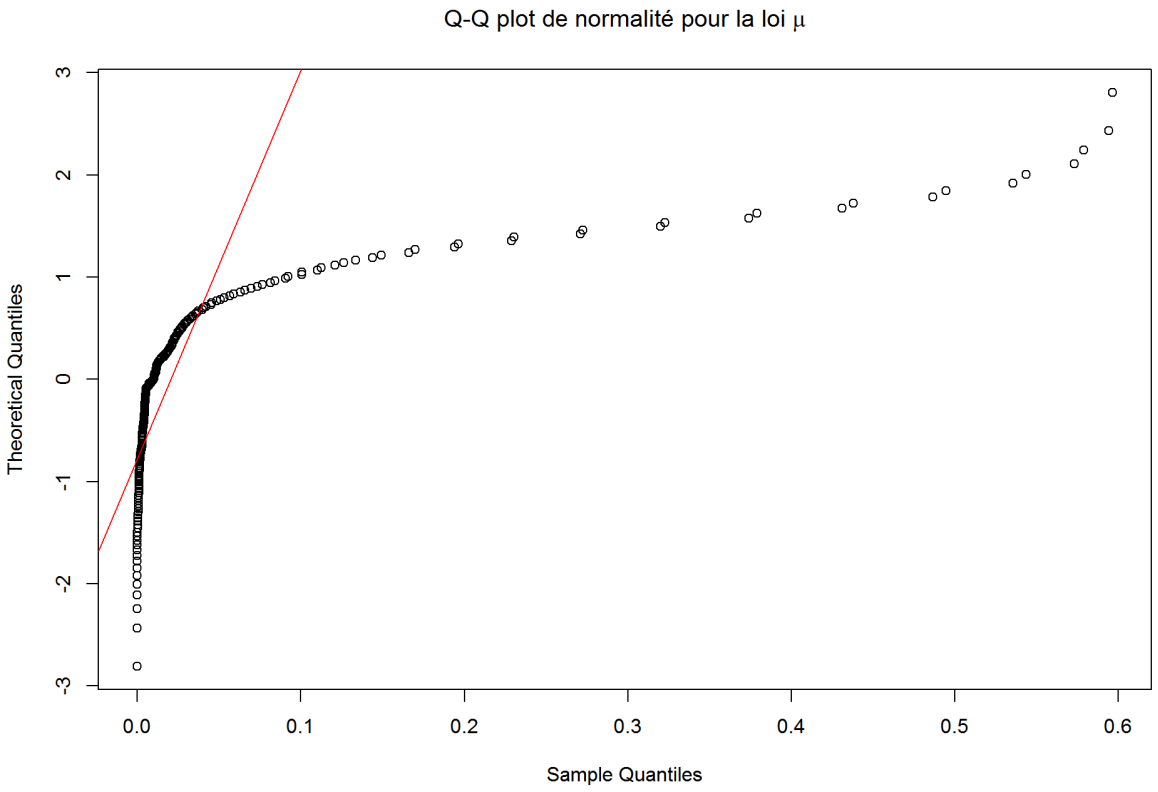
=> symétrie, centrée en 0.
Variance importnate au voisinage de 0 et faible après.

3.2.2. La densitée x->μ(x) peut-elle etre gaussienne ?

Proposer un protocole pour le verifier empiriquement et l'implementer. On pourra penser a renormaliser ksi par la fonction estimee a la question precedente et s'aider des questions de la Section 3.1.

http://www.biostat.ulg.ac.be/pages/Site_r/Normalite.html (http://www.biostat.ulg.ac.be/pages/Site_r/Normalite.html)

- 1- test du QQPlot



=> non ou bien gaussien par morceau

- 2- test de shapiro

```
shapiro.test(mu_sil2$y)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  mu_sil2$y
## W = 0.51229, p-value < 2.2e-16
```

Le test de Shapiro-Wilk donne une probabilité de dépassement de p-value < 2.2e-16, nettement < à 0.05. L'hypothèse de normalité est rejettéee.

- 3- test de Kolmogorov-Smirnoff Dans ce cas-ci également, il existe dans R une commande pour tester l'ajustement de données à une loi normale via le test de Kolmogorov-Smirnov:

```
ks.test(mu_sil2$y,"pnorm",mean(mu_sil2$y),sd(mu_sil2$y))
```

```
##
##  One-sample Kolmogorov-Smirnov test
##
## data:  mu_sil2$y
## D = 0.32093, p-value < 2.2e-16
## alternative hypothesis: two-sided
```

p-value faible rejetté