



Louvain Institute of Data Analysis and
Modeling in economics and statistics

Institute of Statistics, Biostatistics and Actuarial Sciences

LDATS2150: nonparametric statistics-smoothing methods

Project number 1: Density estimation

Author:

WARNAUTS Aymeric ([DATS2M](#)- 87031800)

Professor:

VON SACHS Rainer

1 Task 1

As for this part of the project we will have to compare the performance of a **parametric** and **nonparametric** density estimator we will simulate a sample $\{X_i\}_{i=1}^n$ from a mixture of 2 **Normal distributions** $Z_{1,i} \sim N(2, 1)$ and $Z_{2,i} \sim N(6, 1)$ with $n = 100$ and under the assumption that:

$$P[X = Z_1] = \alpha \quad (1)$$

$$P[X = Z_2] = 1 - \alpha \quad (2)$$

we recall the Normal density as part of a parametric class such that:

$$f_{\mu, \sigma^2}(x) = \frac{1}{\sqrt{2\pi}} e^{\frac{(x-\mu)^2}{2\sigma^2}}$$

and that in our case of mixture we get the following expression for the mixture of two normal random variable:

$$g(x; \mu_1, \sigma_1, \mu_2, \sigma_2, \alpha) = \alpha f(x; \mu_1, \sigma_1) + (1 - \alpha) f(x; \mu_2, \sigma_2)$$

1.1 Histogram Density Estimator

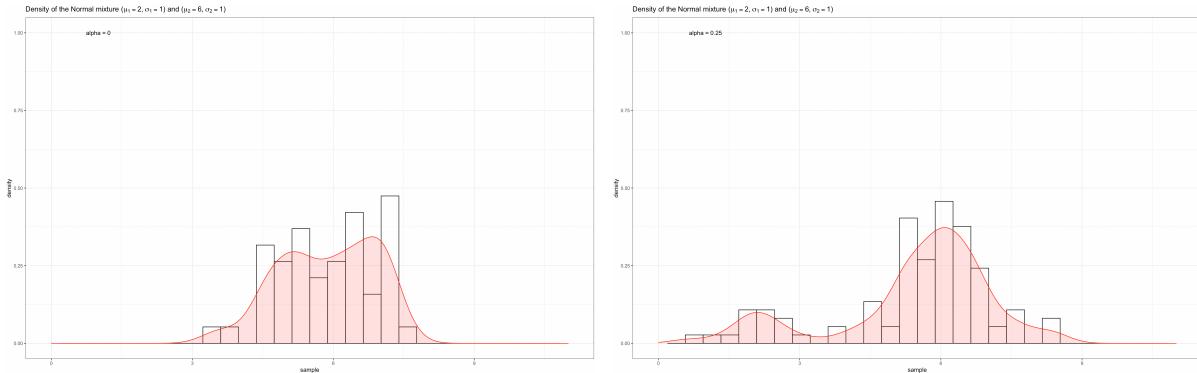
As the **parametric** approach gives accurate estimation of the parameters μ and σ^2 , we will see that it's not robust. This is solved by the a **nonparametric** density function estimation by fitting $f_X(x)$ as a non-negative curve with:

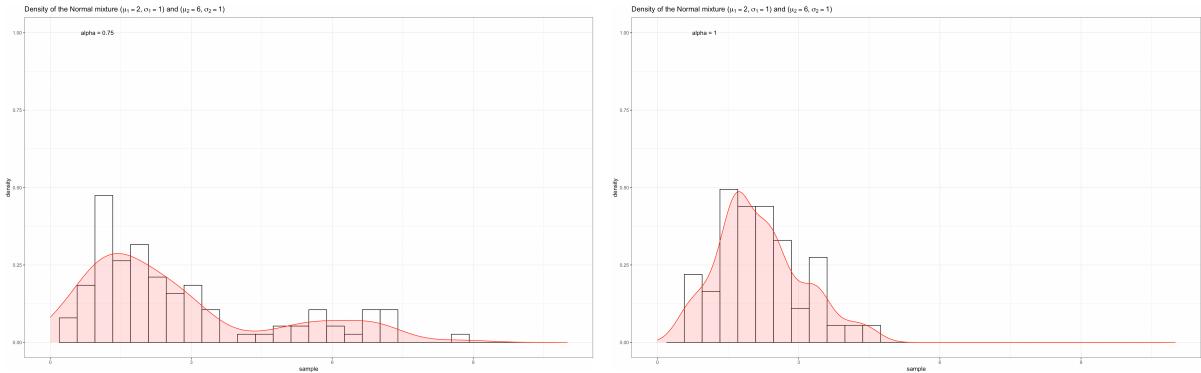
$$\int f_X(x) dx = 1$$

by smoothing the data. Therefore we can build the histogram estimator that is given by:

$$\hat{f}_H(x) = \frac{\nu_k}{nh} \sum_{i=1}^n I_{[t_k; t_k+1]}(X_i)$$

for X_i the class which includes $x \in B_k$, ν_k being the number of observation in class B_k ad h the bin width. We display the histogram density estimator performed by the *R* function *stat bin* from *geom density* in *ggplot* for interesting values of α :





As the *R* function choose by default the number of bins **30** which is not optimal, we can use the *R* function to find the smoothing parameter called the **bandwidth** by some calculation by default respectively ($\alpha : bw \rightarrow (0 : 0.38, 0.05 : 0.52, 0.25 : 0.64, 0.5 : 0.83, 0.75 : 0.79, 0.95 : 0.3, 1 : 0.35)$).

1.2 Kernel Density Estimator

As it motivates the well known **kernel estimator**, that is a method that take a class of length $2h$ around each point such that:

$$\hat{f}_H(x) = \frac{1}{2nh} \sum_{i=1}^n I\{-1 \leq \frac{x - X_i}{h} < 1\}$$

inducing the development of rectangular window under kernel assumptions. This help to prevent the estimator of origin dependence.

Moreover, as we demonstrate under law of large number assumption that:

- $\mathbb{E}[\hat{f}_h(x)] = \frac{F(x+h) - F(x-h)}{2h} \rightarrow f(x)$ if $h = h_n \rightarrow 0$
- $\mathbb{V}[\hat{f}_h(x)] \sim \frac{1}{2nh} f''(x) \rightarrow 0$ if in addition $nh_n \rightarrow \infty$ with $n \rightarrow \infty$

that are the conditions for the mean squared consistency of the estimator $\hat{f}_h(x)$ with simple kernel estimators. The choice of the **bandwidth** h will be crucial as the bias is proportional to it:

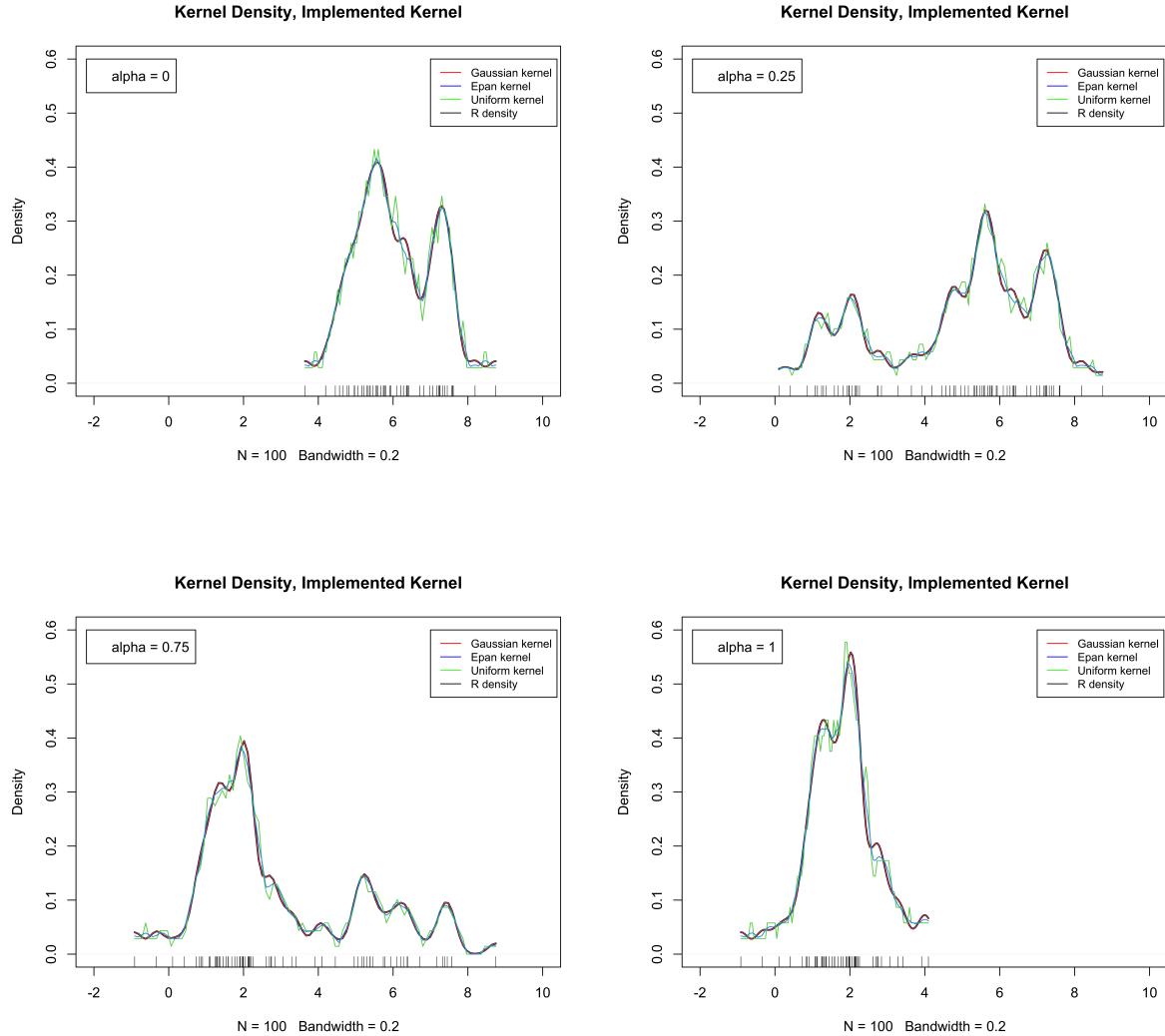
$$\mathbb{E}[\hat{f}(x) - f(x)] = \frac{h^2}{2} f''(x) \mu_2 + o(h^2)$$

and for the variance we get:

$$\mathbb{V}[\hat{f}(x)] = \frac{1}{nh} f(x) \int K^2(u) du + o(\frac{1}{nh})$$

where μ_2 is the second moment of the kernel and f'' can be interpreted as the curvature of the density. And we denote that the bias decreases to zero if h decreases to zero but the variance goes up.

Thus in this section we will compare the performance of different kernels approximation functions, even if comparing the results for the efficiency of such kernel functions, we can assess that we do not lose a lot of efficiency by taking a sub-optimal one. First of all, the built-in kernel density in *R* is scaled such that the bandwidth h is equal to the standard deviation of the smoothing kernel. So, to our estimator be comparable with density, we should also standardize our h . And as explained previously, the variance is equal to the second moment μ_2 .



And the following definitions need to be checked for kernels:

$$\blacktriangleright K_j(k) = \int_{-\infty}^{\infty} u^j k(u) du$$

$$\blacktriangleright R(k) = \int_{-\infty}^{\infty} k(u)^2 du$$

Thus, to build the previous plots we have computed the previous expressions for the **Gaussian**, **Uniform** and **Epanechnikov** kernels in *R*, we display the development for the uniform one below:

$$K_2(k) = \int_{-\infty}^{\infty} u^2 \times 0.5 \times 1_{|u| \leq 1} du \quad (3)$$

$$= 0.5 \int_{-1}^{1} u^2 du \quad (4)$$

$$= \frac{1}{3} \quad (5)$$

and

$$R(k) = \int_{-\infty}^{\infty} (0.5 \times 1_{|u| \leq 1})^2 du \quad (6)$$

$$= \frac{1}{4} \int_{-1}^1 1_{|u| \leq 1} du \quad (7)$$

$$= \frac{1}{2} \quad (8)$$

For the **Gaussian** and **Epanechnikov**, for which the same integrations are to be carried out, you should obtain $K_2(k) = 1$, $R(k) = \frac{1}{2\sqrt{\pi}}$ and $K_2(k) = \frac{1}{5}$, $R(k) = \frac{3}{5}$. This are the values obtained by our *R* function *IntKnorm*, *IntKunif* and *IntKepan*.

1.3 Gaussian Maximum Likelihood Estimator

In this section we will introduce the parametric **Gaussian** estimator because we know that if the true PDF p is close to a Gaussian distribution, then the parametric approach is a good one. But if p is very far away from being a Gaussian, this method is going to give us a huge bias.

When using Maximum Likelihood Estimation to estimate parameters of a Gaussian, we set the mean of the Gaussian to be the mean of the data, and set the standard deviation of the Gaussian to be the standard deviation of the data such that:

$$\hat{\mu}_{MLE} = \frac{1}{n} \sum_{i=1}^n x_i \quad (9)$$

$$\hat{\sigma}_{MLE}^2 = \frac{1}{n} \sum_{i=1}^n (x_i - \hat{\mu})^2 \quad (10)$$

and thus in the case of the **Gaussian mixture** for the likelihood of the observations as a product over the n observations we obtain:

$$P(x|\alpha, \mu_1, \mu_2, \sigma_1^2, \sigma_2^2) = \prod_{i=1}^n (\alpha N(x_i|\mu_1, \sigma_1^2) + (1 - \alpha)N(x_i|\mu_2, \sigma_2^2))$$

and written as a product over the likelihoods for the two part of the mixture:

$$P(x|\alpha, \mu_1, \mu_2, \sigma_1^2, \sigma_2^2) = \prod_{i=1}^n ((\alpha N(x_i|\mu_1, \sigma_1^2))^{1-z_i} + ((1 - \alpha)N(x_i|\mu_2, \sigma_2^2))^{z_i})$$

where z is a binary latent variable, $z = 1$, $z = 2$, respectively for $X = Z_1$ and $X = Z_2$ with $P(X = Z_1) = \alpha$ and $P(X = Z_2) = 1 - \alpha$. And thus, each data point is associated to a subclass z related to Gaussian distribution with different parameters (μ_z, σ_z^2) with probability $P(X = Z_i)$.

And finally the **log-likelihood** to be maximised w.r.t μ_1, σ_1^2 or μ_2, σ_2^2 is:

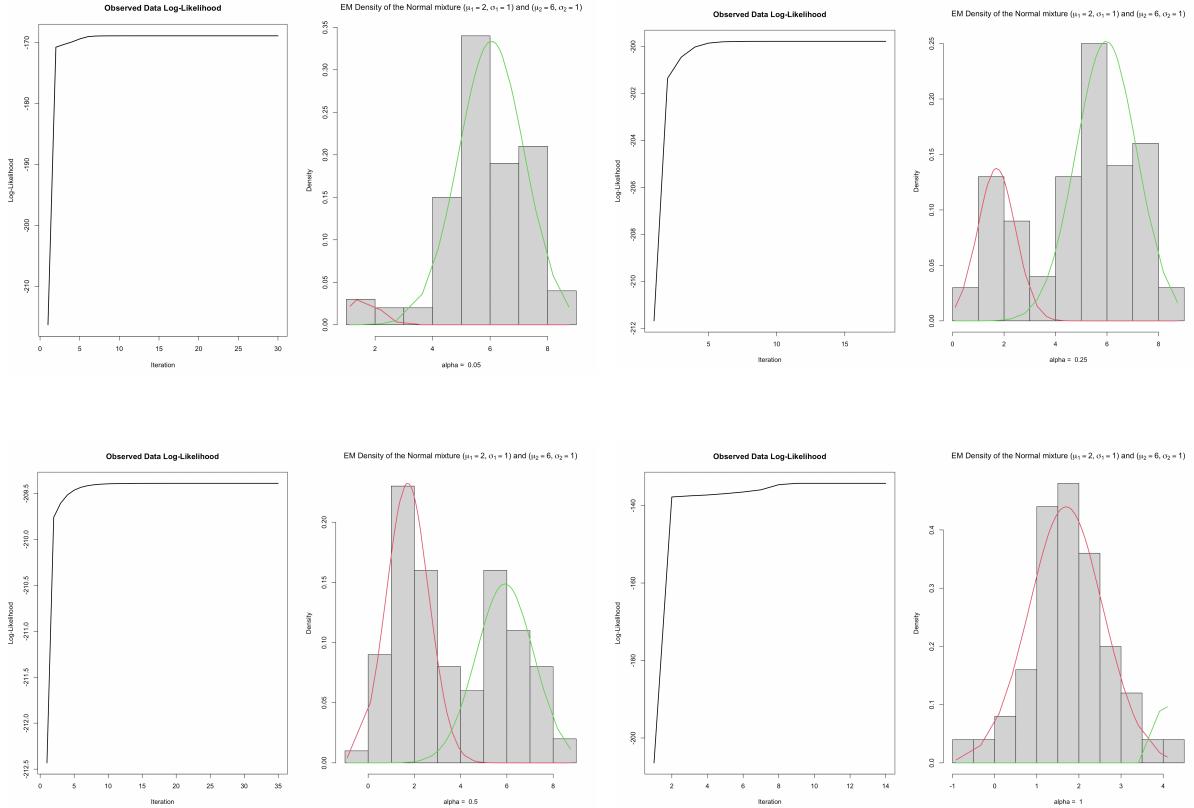
$$\ln(P(x|\alpha, \mu_1, \mu_2, \sigma_1^2, \sigma_2^2)) = \sum_{i=1}^n ((1 - z_i)(\ln \alpha + \ln N(x_i|\mu_1, \sigma_1^2)) + z_i(\ln(1 - \alpha) + \ln N(x_i|\mu_2, \sigma_2^2)))$$

Thus the **MLE** for μ_1, σ_1^2 are:

$$\hat{\mu}_1 = \frac{1}{\sum_{i_1}^n (1 - k_i)} \sum_{i=1}^n (1 - k_i) x_i \quad (11)$$

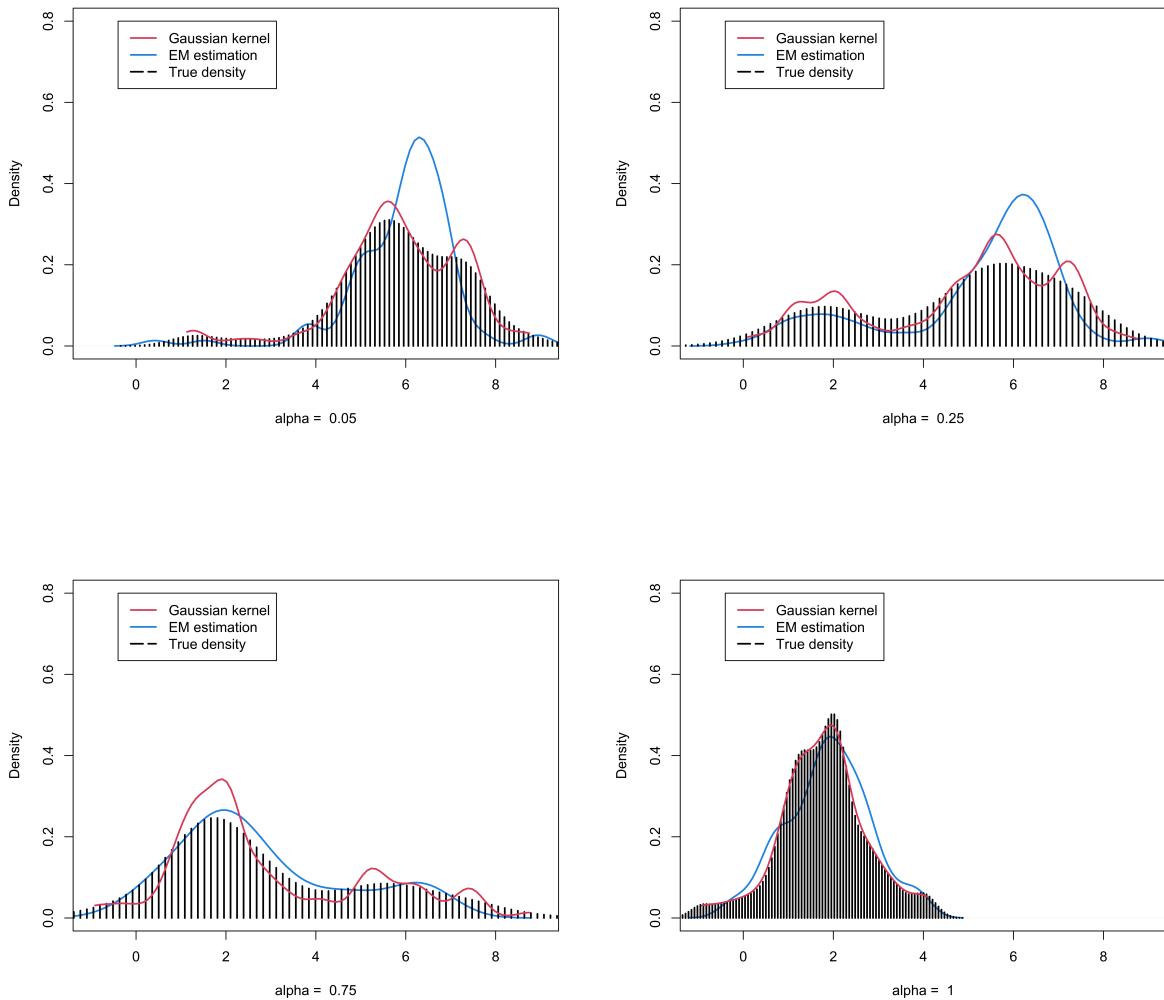
$$\hat{\sigma}_1^2 = \frac{1}{\sum_{i_1}^n (1 - k_i)} \sum_{i_1}^n (1 - k_i) (x_i - \hat{\mu}_1)^2 \quad (12)$$

We will implement the **expectation–maximization (EM) algorithm** that is an iterative method to find (local) maximum likelihood where the model depends on unobserved latent variables. The EM iteration alternates between performing an **expectation** step, which creates a function for the expectation of the log-likelihood evaluated using the current estimate for the parameters, and a **maximization** step, which computes parameters maximizing the expected log-likelihood found on the previous step.



1.4 Comparison of parametric and nonparametric estimators

As it's time to compare the performances of our fitted distribution for the **parametric** Gaussian Maximum Likelihood Estimator and the **nonparametric** Kernel Density Estimator, we will do it graphically. In fact, we simulate the **true density** of our fixture with the true parameters ($\mu_1 = 2, \sigma_1^2 = 1$) and ($\mu_2 = 6, \sigma_2^2 = 1$) and we plot it in **black** on the following plots. And thus, we graphically compare the way our two approaches fit the real density distribution by re-coding the **kernel density estimator** (same implementation as for our kernel density comparison) in *R* and by generating a density with the **EM algorithm** estimated parameters. Moreover, we will display our results to be able to compare the two density estimation methods for different α values. We recall that as we haven't computed our optimal bandwidth and thus even if we can work under an adaptive one, we decided to fix it. At the end of this section, and in order to link the next task to be implemented, we will compare the estimations in terms of the symmetric mean absolute percentage error using the Monte Carlo method. In fact, it has been advanced that the adaptive kernel method outperforms the smoothing with a fixed bandwidth in the majority of models.



Analyzing our results we will point out that the quality of fit of the **Gaussian kernel** density estimator is clearly linked to the choice of the bandwidth as it's clear that it fit well our **Gaussian mixture** for $\alpha = [0.05; 1]$ but the same conclusion can be made for $\alpha = 0.95$ looking at the R estimated bandwidths at the end of the section **Histogram Density Estimator**. A table of the parameter bias will be usefull to understand the behavior of our **EM estimation**:

α	Bias for the parametric distribution parameters			
	parametric EM			
	μ_1	μ_2	σ_1	σ_2
0.05	-0.3933	0.0508	-0.4671	0.1436
0.25	-0.2924	-0.0252	-0.2702	0.1847
0.5	-0.2922	-0.0799	-0.0824	0.2490
0.75	-0.3249	-0.1079	-0.0716	0.2191
0.95	-0.28	-0.9704	-0.0734	0.2989
1	-0.3055	-1.9904	-0.1246	-0.9091

This table helps us to assess how the resulting distribution parameter estimates change depending on the value of the mixture parameter α . In fact, when one of the two mixture parameter α or $1 - \alpha$ is one, the resulting distribution would be just **Gaussian** with respective mean and variance. This is the reason why this algorithm under-performs for $\alpha = [0.05; 1]$. In fact, it's obvious that when $\alpha = 0.05, 0$ and thus that our mixture look like a **Normal distribution** as $N(6(\mu_2), 1(\sigma_2^2))$ such that we get a huge bias for the parameter of the other part of

the mixture. We get the exact opposite behaviour when we perform the algorithm with $\alpha = [0.95; 1]$. To conclude this question we can introduce the mixture model that assume two type of random variable, Z the hidden state variable and X the observed variable. Thus observations are sampled with a two step-procedure. First the class observation is sampled from the distribution of the hidden variable $P_Z(z) \rightarrow z_i$ and then an observation is drawn from the class conditional density for the selected state $P_{X|Z}(X|z_i) = x_i$.

And so we can denote that as the pdf of a mixture model can be written as:

$$P_X(x) = \sum_{z=1}^Z P_X|Z(x|z)\pi_z$$

then this expression look like the kernel density estimate that is a mixture estimate of n components where mixture components are $\frac{1}{nh^d}\phi(\frac{x-x_i}{h})$ and the mixture weights are uniform such that $\pi_z = 1/n$. Thus in our case, the mixture density estimate may be performed by just picking the right number of Gaussians. For the parametric model, we can interpret this as a mixture of 1 component with the weight = 1 and the mixture component being the parametric density itself.

Thus the mixture model provide a connection between the **nonparametric** and **parametric** estimations. This estimation method for mixture density estimations is appropriated since it provide more degrees of freedom and then **less bias** than for the **parametric** one and gets much smaller number of components than the **non parametric kernel density estimation** and thus **less variance**. This usually leads to a better fit.

2 Task 2

For this task we have to choose two interesting values for the attribution parameter α and as we want to assess for the performances of such estimators in case of bimodality we will choose $\alpha = (0.25, 0.75)$ using these as mixture parameter. As these can be considered as symmetric with respect to the balanced mixture attribution parameter $\alpha = 0.5$, we will compare how the parameters that define the shape of the density have an impact on our results. Moreover, we will go further than the previous obtained results with the “rule-of-thumb” (i.e. normal reference) bandwidth selection procedure. Finally, we will discuss how well it works in these two cases by performing a **Monte Carlo** simulation study on the bias, variance and MSE behaviour of the estimators at selected interesting points x .

2.1 Simple rule of reference to a normal distribution

First of all let's recall how this "plug-in" bandwidth selection works as by **Taylor expansion** of the density f we obtain the following useful conditions on the kernel:

- $\int K(u)du = 1$
- $\int uK(u)du = 0$ (if K is symmetric)
- $0 < \mu_2 := \int u^2K(u)du < \infty$

and putting the respective leading terms h^2 and $\frac{1}{nh}$ together we get:

- $MSE[\hat{f}(x)] = \frac{1}{nh}f(x)R(K) + \frac{h^4}{4}(f''(x))^2\mu_2^2 + o(h^4 + \frac{1}{nh})$
- $MISE[\hat{f}_h(x)] = \frac{1}{nh}R(K) + \frac{h^4}{4} \int (f''(x))^2dx\mu_2^2 + o(h^4 + \frac{1}{nh})$

and thus we can choose the smoothing parameter h by minimisation of the AMSE/AMISE (=leading terms of the MSE/MISE), therefore we obtain respectively:

$$\blacktriangleright h_{AMSE}(x) = \left(\frac{f(x)R(K)}{(f''(x))^2\mu_2^2}\right)^{1/5}n^{-1/5}$$

$$\blacktriangleright h_{AMISE} = \left(\frac{R(K)}{R(f'')\mu_2^2} \right)^{1/5} n^{-1/5}$$

and if h_{MISE} is the bandwidth which would minimize the exact **MISE**, then:

$$\lim_{n \rightarrow \infty} \frac{h_{MISE}}{h_{AMISE}} = 1$$

As we get the previous expression of h_{AMISE} that is linked to the unknown function f'' and thus on f , we calculate $R(f'')$ for $f =$ density of $N(\mu, \sigma^2)$:

$$R(f'') = \frac{3}{8\sqrt{\pi}\sigma^5}$$

and then we get the bandwidth of type "normal reference":

$$\hat{h}_{NR} = \left(\frac{8\sqrt{\pi}R(K)}{3\mu_2^2} \right)^{1/5} \hat{\sigma} n^{-1/5}$$

where $\hat{\sigma}$ is given as $\min(S, \frac{R}{1.349})$, S being the Standard deviation and R is the empirical interquartile range. We have already developed the $R(k)$ expression in the previous section and the second moment $\mu_2 = \int_{-\infty}^{\infty} u^2 K(u) du$.

As we will try to find this value for the generation of $\{X_i\}_{i=1}^n$ with our two selected mixture parameters, we could have recoded the expression of \hat{h}_{NR} but there is only a little difference between this result and the result of `bw.nrd()`, the built-in function in *R*. We will display the results in the following table:

Bandwidth selection with Simple rule of reference		
	$\alpha = 0.25$	$\alpha = 0.75$
\hat{h}_{NR}	0.911	0.927

2.2 Monte Carlo simulation study

In this section we will perform the development of an MCS in 3 steps for our analysis of the **bias**, **variance** and **MSE** behaviour of the estimators at selected interesting points x . The MC method is often criticized for its slow convergence. In particular, given a fixed number of points n , random samples from a Normal mixture are usually not inclusive enough and often do not accurately represent the full mixture population.

We will consider a sequence of different sample sizes n and compute the 3 metrics only for particular points of x . Our analysis will compare the results obtained for boundary points and an interior points of our density estimators. In fact, by setting a very high number of repetition ($K = 10000$) for our MCS algorithm, we will try to estimate these metrics by averaging our results. Once again, we will use the previous re-coded mixture distribution to generate $\{X_i\}_{i=1}^n$ and we will use the two selected mixture parameters $\alpha = (0.25, 0.75)$. Thus by generating 10000 mixture distribution for each α , we will compare the bandwidth selection procedure and discuss how well it works.

Let's point out that we select the **Gaussian kernel density estimator** for our implementation as we know that for the Rule of Thumb constants stay around 1 for higher order kernels and thus can be simplified such that $h = \hat{\sigma}C_v(k)n^{-1/2v+1} = \hat{\sigma}n^{-1/2v+1}$. In addition, we do not lose a lot by taking a sub-optimal kernel because the choice of the optimal bandwidth is much more important.

In fact, at each iteration we recompute the \hat{h}_{NR} of our generated mixture and stack it to compute an average for different sample sizes $n = (25, 50, 75, 100, 200, 500, 1000)$. Moreover, as we will try to compare our results for the **parametric** and **nonparametric** density estimations, we will compare the results of the **EM density estimator** by plugging the estimated parameters in our mixture generator. We will stack these estimated parameters through our iterative Monte Carlo simulation and also take an average of the results for each sample sizes. For our **Kernel density estimation** we will use the **Gaussian kernel** by plugging the computed "rule-of-thumb" (i.e. normal reference) bandwidth.

We recall the criteria to compare estimators following a point-wise criteria as we will implement these metrics:

- The **bias** of an estimator: $Bias[\hat{f}(x)] := \mathbb{E}[\hat{f}(x) - f(x)]$ and its asymptotic behaviour.
- The **MSE** at given point x : $MSE[\hat{f}(x)] = \mathbb{E}[(\hat{f}(x) - f(x))^2] = \mathbb{V}[\hat{f}(x)] + (Bias[\hat{f}(x)])^2$. And the estimator is called consistent if its MSE ends to zero fo $n \rightarrow \infty$

When we develop the notions of bias and variance of the kernel estimator then we mean the random quantity, $\hat{f}_n(x)$ at a fixed value of x with a gaussian mixture distribution for this project. Moreover, it has been advanced that the the kernel density estimator is biased such that:

$$\mathbb{E}[\hat{f}(x) - f(x)] = (K_h * f)(x) - f(x) \approx \frac{h^2}{2} f''(x) \mu_2 \neq 0$$

with a bias that tends to 0 when h is small. But it's important to observe that in practice for a given sample size the bandwidth h cannot be chosen too small, because otherwise none of the observations will fall into the bin.

2.2.1 MSE behaviour

As we will analyze the behaviour of our MSE through the sample sizes let's point out that the points at which the MSE has been computed have not been chosen randomly. In fact, as we want to compare boundary and interior points we choose the two extreme points ($1, n$) and the quartiles as we want to understand how our mixture unbalanced distribution has an impact on our results.

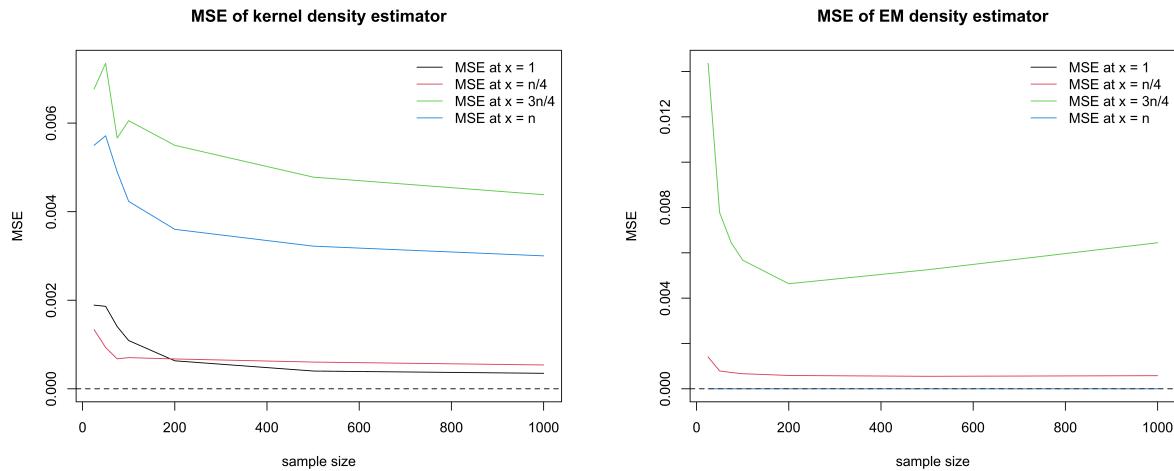
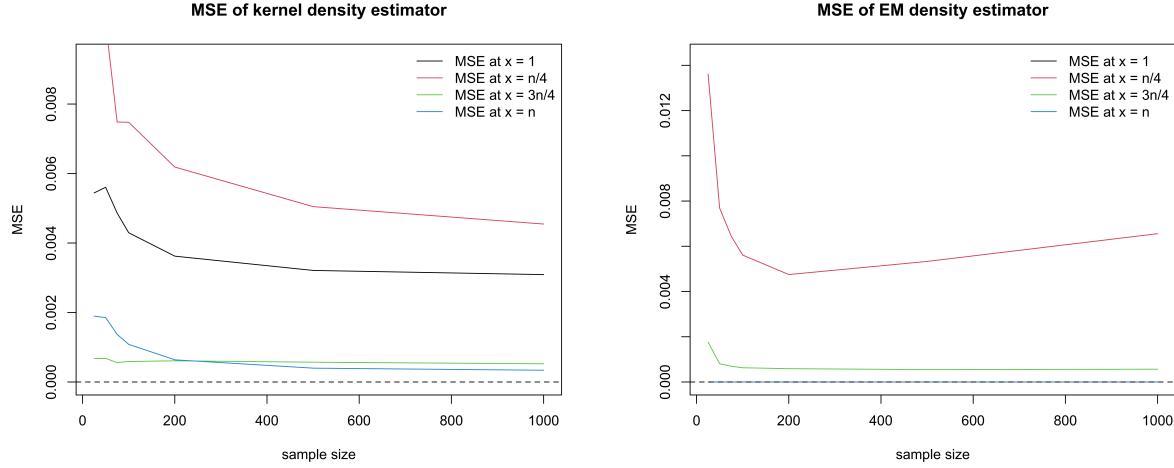


Figure 9: $\alpha = 0.25$

The first conclusion is that we denote that as n increases, MSE of the Kernel density estimator decreases for both boundary and interior points. This is due to the convergence behaviour of the estimated parameters as for $MSE(\hat{f}(x))$ to decline as $n \rightarrow \infty$ we must have $h \rightarrow 0$ and $nh \rightarrow \infty$: That is, the bandwidth must decrease, but not at a rate faster than sample size. We can check this behaviour in our table summary in appendices. And thus we denote that the decreasing behaviour of the **MSE** is due to the decreasing bandwidth when the sample size increase. In fact we know that the the means squared error depends on the density estimator variance $\mathbb{V}[\hat{f}(x)]$ that depends itself to the chosen bandwidth.

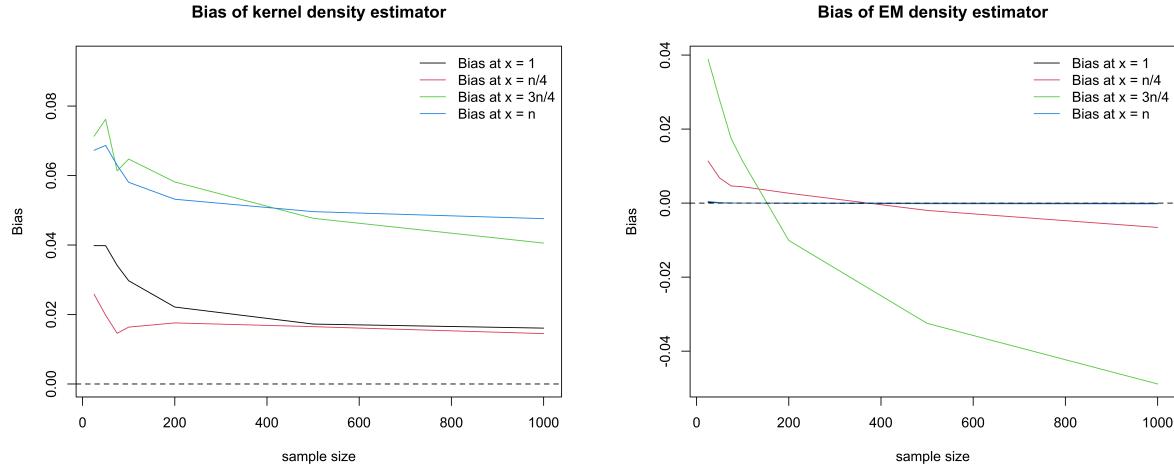
Moreover, the MSE of our the Kernel density estimator at boundary points are lower comparing to the interior point at $x = \frac{3n}{4}$ but it's not true for the point at $\frac{n}{4}$. This is due to the implementation of the bandwidth as it computes the **standard deviation** of the mixture and the **Interquartile Range**. In fact, as we know that the Gaussian reference bandwidth can be a good choice when the density is weakly Gaussian with $\alpha = (0, 0.5, 1)$, in other cases, it can lead to significant **oversmoothing** of the density estimates.

Figure 10: $\alpha = 0.75$

As we launch the same Monte Carlo procedure for our mixture with $\alpha = 0.75$ we obtain the exact opposite behaviour for our **MSE** analysis, what's expected. In fact, in this case, when our distribution points have a higher probability to follow $N(2, 1)$ we get a higher mean squared error for the $\frac{n}{4}$ interior point. Moreover we see the opposite behaviour for the boundary points located at $x = 1$ and $x = n$. In this case we get a higher mean squared error for the first distribution point.

2.2.2 Bias behaviour

Looking at the **Bias** at interior and boundary points, as we have previously defined the expectation of the density estimator that also depends on the choice of the bandwidth, seen as an average of $f(z)$ locally about x , we denote that smaller bandwidths imply reduced bias.

Figure 11: $\alpha = 0.25$

Looking at our results performed with the "simple rule of thumb" bandwidth selector, we conclude that this estimated bandwidth decrease with the sample size. Moreover, as the density estimator $\hat{f}(x)$ that smooth the data local to $X_i = x$. And thus, the bias results from this smoothing, and is larger the greater the curvature in

$f(x)$. That's the behaviour we denote on the following plot respectively with the bias located at $\frac{3n}{4}$ and $\frac{n}{4}$ when $\alpha = 0.25$ and $\alpha = 0.75$. In fact this skewed densities imply local pics.

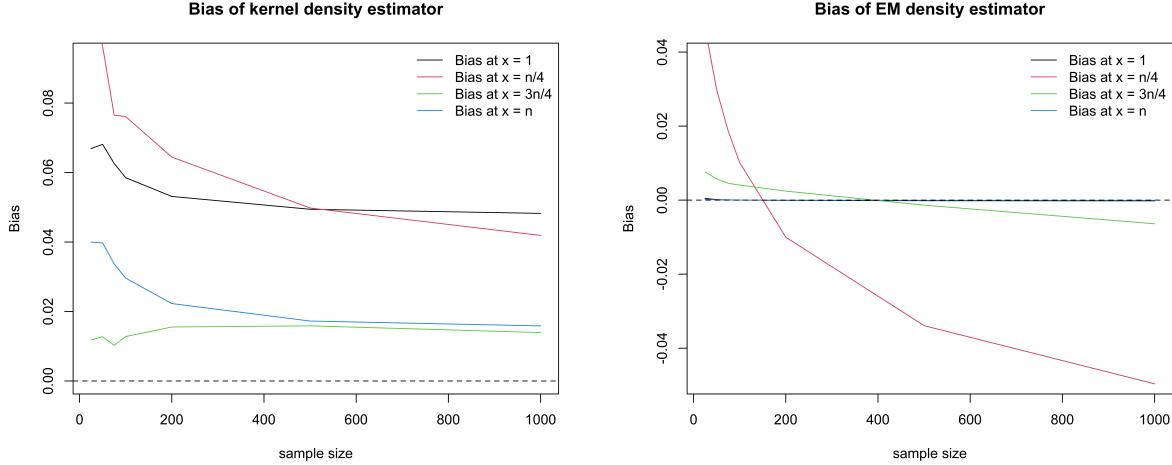


Figure 12: $\alpha = 0.75$

2.2.3 Variance behaviour

Now we will tackle the variance analysis and as we display its analytical derivation:

$$\int_{-\infty}^{\infty} x^2 \hat{f}(x) dx - \left(\int_{-\infty}^{\infty} x \hat{f}(x) dx \right)^2 = \sigma^2 + h^2 k_2(k)$$

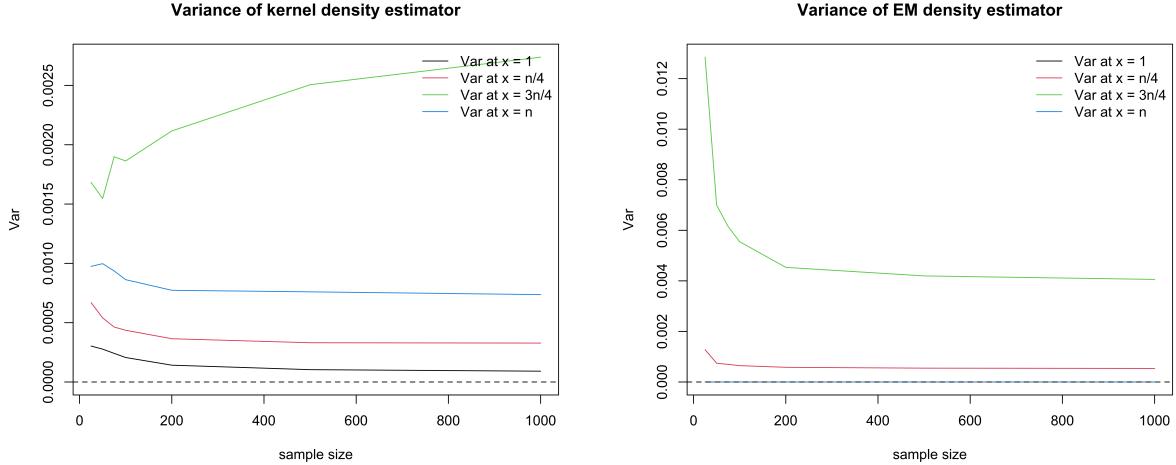
Thus the density estimate inflates the sample variance by the factor $h^2 k_2(k)$ and moreover the variance is asymptotically approximated by:

$$\mathbb{V}[\hat{f}(x)] \approx \frac{1}{nh} f(x) \int K^2(u) du$$

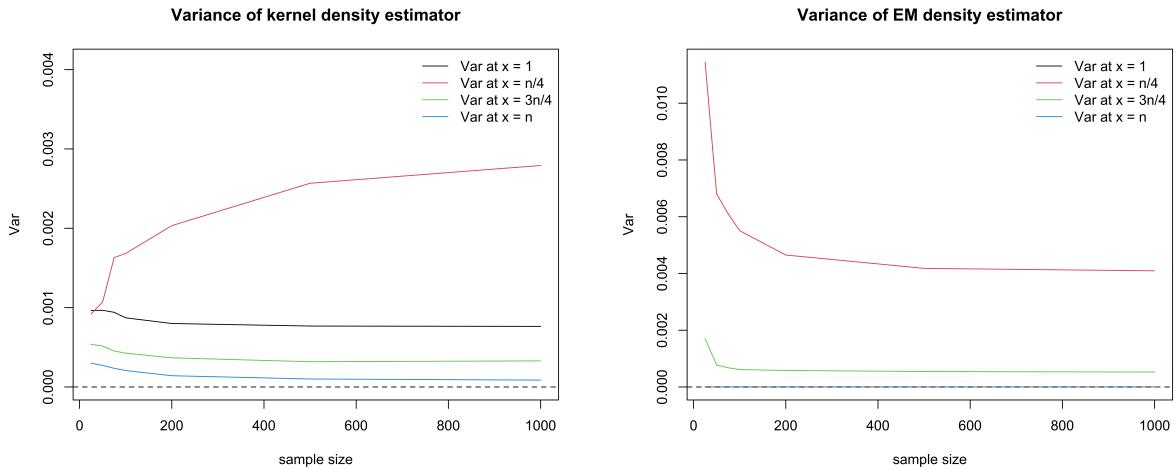
and thus the variance is proportional to the reciprocal of the "effective" sample size nh . And thus once again as the leading term here $\frac{1}{nh}$ is going to 0 because of the different rate of convergence of n and h we expect a decreasing variance when $n \rightarrow \infty$. Practically, as we have seen before, the density estimator variance expression can be found analytically with the previously computed **MSE** and **Bias** such that:

$$\mathbb{V}[\hat{f}(x)] = MSE[\hat{f}(x)] - (Bias[\hat{f}(x)])^2$$

at a given point x .

Figure 13: $\alpha = 0.25$

The obtained results are the keys for the understanding of our **MSE** and **Bias** interpretations. In fact as we look at the variance expression we see that the choice of the non-parametric kernel is what induces this!

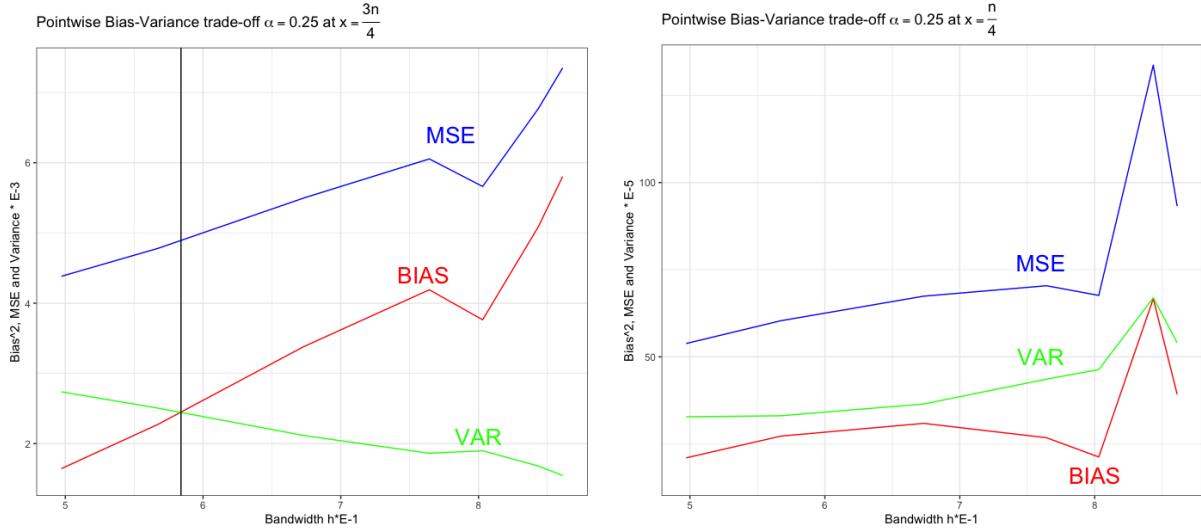
Figure 14: $\alpha = 0.75$

And thus, as it time to have some conclusion, we clearly see with our previous analysis that nonparametric density estimators cannot achieve the parametric rate, but get the closer to it, the "smoother" their construction is.

2.2.4 Impact of the choice of "normal rule" bandwidth

We have demonstrated previously that if the true density is normal, then the computed normal reference bandwidth will be optimal and if the true density is reasonably close to the normal, then the bandwidth will be close to optimal and that's interesting when we are dealing with a mixture of normal distributions because we can choose $\alpha = (0, 1)$ to compare our results. In fact in these cases our mixture is close to respectively a $\mathbb{N}(6, 1)$ and $\mathbb{N}(2, 1)$. But this has not been implemented as the results are already well known for this distributions.

We will display the **pointwise bias-variance trade off** to conclude our bandwidth selection procedure analysis:



We will not display the behaviour of our bias, variance and mean squared error for $\alpha = 0.75$ since as previously concluded we expect the exact opposite behaviour for our bandwidth at $x = \frac{3n}{4}$ and $x = \frac{n}{4}$.

We clearly denote that the shape of our metrics seems convenient for the pointwise plot at $x = \frac{3n}{4}$ but not for the plot at $x = \frac{n}{4}$ since the metrics doesn't cross at optimum. this behaviour is due to the choice of the global "Rule-of-thumb" bandwidth selector. In fact as for the AMSE/AMISE-optimal bandwidth we should work with a $h = h(x)$, **variable local smoothing parameter**.

2.3 Improvements

In comparison with the Monte Carlo method, the Quasi Monte Carlo method significantly improves the estimation results. The MSE method offers better performance in density fitting than the QMC method. In fact the MSE method is capable of re-constituting the four different MixN densities with low L2-distances. In addition, MSE-RPs from MixN provide outstanding performance for kernel density estimation. MSE-RPs are the theoretical counterparts of cluster means obtained by a k-means algorithm.

For the **Bias** analysis, when higher-order kernels are used and thus that the density has enough derivatives, the bias is proportional to h^r which is of lower order than h^2 . Thus the bias of estimates using higher-order kernels is of lower order than estimates from second-order kernels, and this is why they are called bias-reducing kernels. In fact, as we want to reduce the bias under the assumption that the derivative of order r is continuous in x , then a kernel of order r is such that:

$$\int u^j K(u) du = \begin{cases} 1 & j = 0 \\ 0 & ij = 1, \dots, r-1 \\ \mu_r < \infty & j = r \end{cases}$$

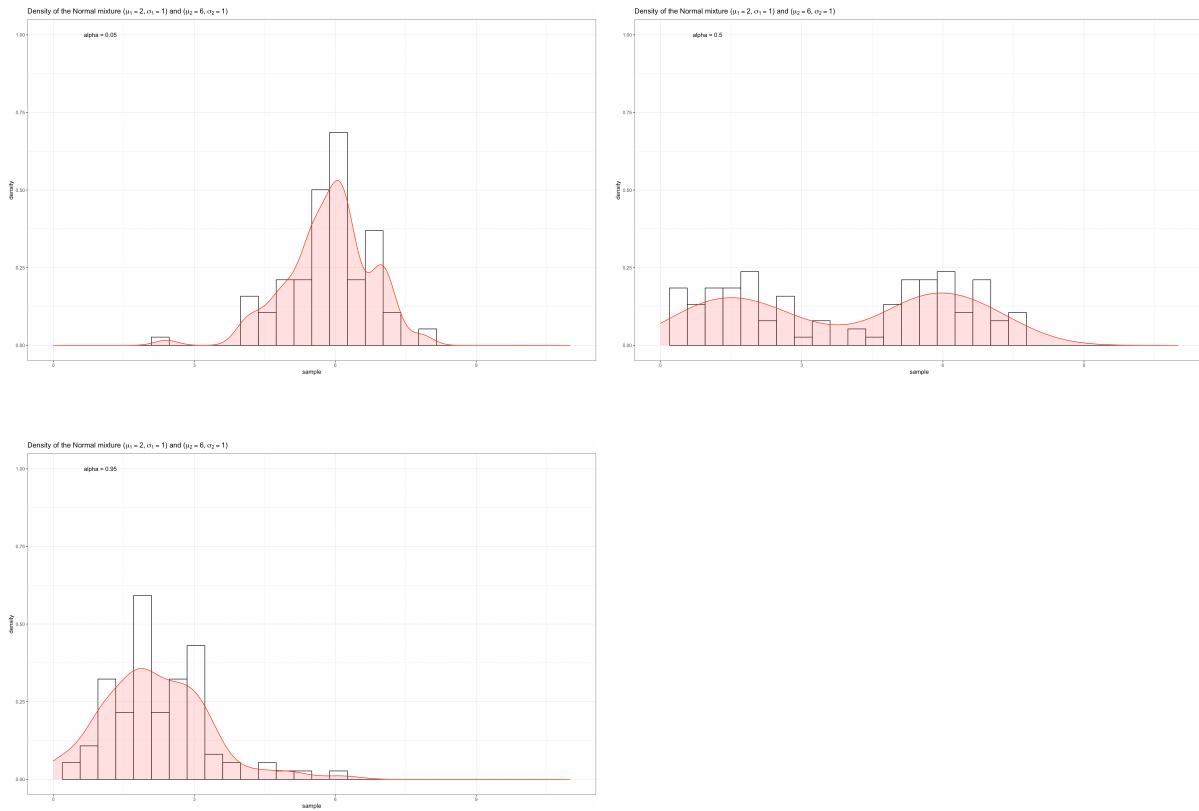
and thus,

$$Bias(\hat{f}(x)) = \frac{h^r}{r!} \mu_r f^{(r)}(x) + O(h^{r+1})$$

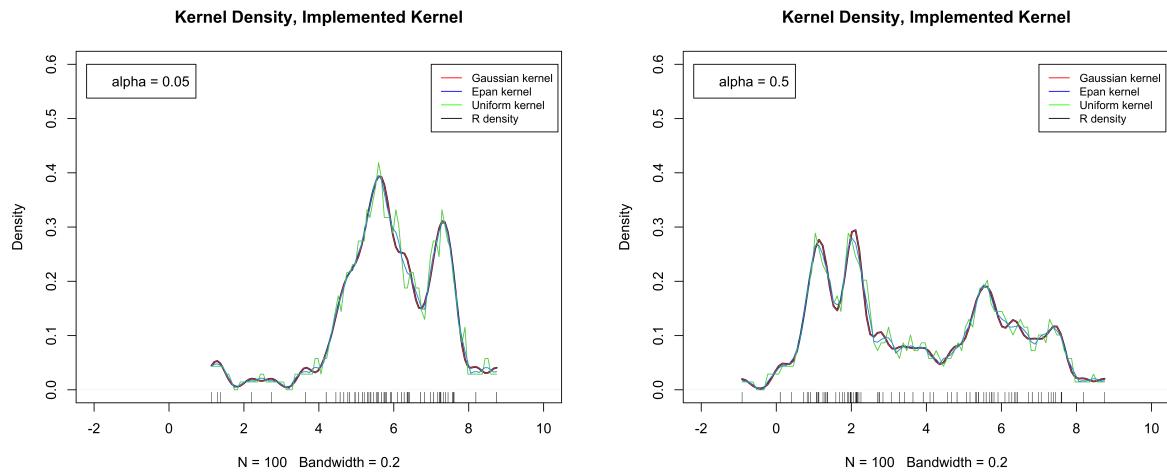
where the bias is reduced but the variance does not change. In fact the larger r , the closer one gets to the parametric rate n^{-1} . But $f(x)$ has to be very regular, and the estimator \hat{f} can become negative. As for the density estimation, the *AMSE* and *AMISE* behave as $\sim n^{-4/5}$ if $n \rightarrow \infty$, for the **nonparametric** regression, if m has r continuous derivatives, and if the kernel is at least of order r , the rate of convergence is of order $n^{\frac{-2r}{2r+1}}$ and the optimal bandwidth is of order $n^{\frac{-1}{2r+1}}$.

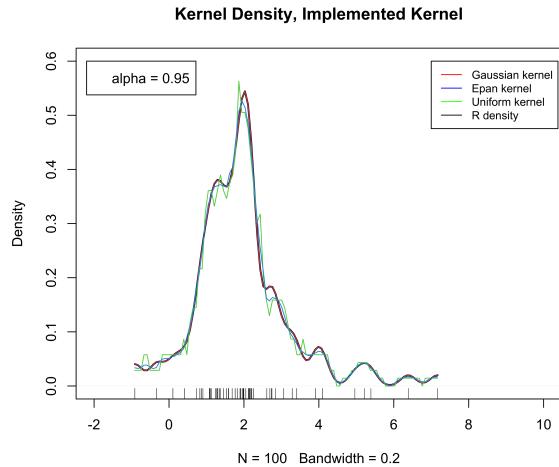
3 Appendices

3.1 Histogram density estimator



3.2 Kernel Density Estimator





3.3 Summary Monte Carlo

Monte Carlo simulation for 0.25

	n = 25	n = 50	n = 75	n = 100	n = 200	n = 500	n = 1000
K MSE at x = 1	0.0018893	0.0018621	0.0014066	0.0010898	0.0006324	0.0004018	0.0003497
K MSE at x = n/4	0.0013367	0.0009325	0.0006762	0.0007041	0.0006740	0.0006036	0.0005383
K MSE at x = 3n/4	0.0067695	0.0073503	0.0056625	0.0060547	0.0054985	0.0047809	0.0043834
K MSE at x = n	0.0054993	0.0057135	0.0049003	0.0042326	0.0036015	0.0032216	0.0030020
EM MSE at x = 1	0.0000001	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000
EM MSE at x = n/4	0.0014047	0.0007880	0.0007187	0.0006659	0.0005879	0.0005512	0.0005793
EM MSE at x = 3n/4	0.0143624	0.0077738	0.0064512	0.0056767	0.0046363	0.0052508	0.0064453
EM MSE at x = n	0.0000004	0.0000001	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000
Estimated SR bw	0.8434201	0.8611835	0.8030849	0.7643722	0.6732185	0.5673303	0.4972401
mu1	2.0594037	2.0384228	2.0464391	2.0391510	2.0399718	2.0472018	2.0432241
mu2	6.0108581	6.0084773	6.0115278	6.0109562	6.0109808	6.0144289	6.0146720
sigma1	0.9300073	0.9377319	0.9578176	0.9711744	0.9874157	1.0015494	1.0004253
sigma2	0.9691841	0.9718056	0.9696602	0.9720924	0.9742563	0.9714268	0.9747872
kern	0.0711115	0.0585982	0.0481271	0.0403711	0.0271217	0.0191003	0.0170297

Monte Carlo simulation for 0.25

	n = 25	n = 50	n = 75	n = 100	n = 200	n = 500	n = 1000
K Bias at x = 1	0.0398292	0.0398120	0.0341389	0.0297234	0.0221490	0.0172563	0.0160797
K Bias at x = n/4	0.0258260	0.0198045	0.0145882	0.0163696	0.0175847	0.0165096	0.0145010
K Bias at x = 3n/4	0.0713218	0.0761825	0.0613471	0.0647414	0.0581557	0.0476967	0.0405575
K Bias at x = n	0.0672648	0.0686736	0.0629616	0.0580517	0.0531821	0.0496158	0.0475962
EM Bias at x = 1	0.0002410	0.0000487	0.0000207	0.0000091	-0.0000134	-0.0000378	-0.0000548
EM Bias at x = n/4	0.0113634	0.0067849	0.0046384	0.0044251	0.0026734	-0.0019716	-0.0065793
EM Bias at x = 3n/4	0.0388959	0.0278604	0.0174308	0.0112973	-0.0100899	-0.0324780	-0.0488680
EM Bias at x = n	0.0004476	0.0001340	0.0000514	0.0000106	-0.0000781	-0.0001537	-0.0001921

Monte Carlo simulation for 0.25

	n = 25	n = 50	n = 75	n = 100	n = 200	n = 500	n = 1000
K Var at x = 1	0.0003030	0.0002771	0.0002411	0.0002063	0.0001418	0.0001040	0.0000911
K Var at x = n/4	0.0006697	0.0005403	0.0004634	0.0004361	0.0003648	0.0003310	0.0003280
K Var at x = 3n/4	0.0016827	0.0015466	0.0018990	0.0018632	0.0021164	0.0025060	0.0027385
K Var at x = n	0.0009747	0.0009975	0.0009361	0.0008626	0.0007732	0.0007599	0.0007366
EM Var at x = 1	0.0000001	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000
EM Var at x = n/4	0.0012756	0.0007420	0.0006972	0.0006464	0.0005808	0.0005473	0.0005360
EM Var at x = 3n/4	0.0128495	0.0069976	0.0061474	0.0055491	0.0045345	0.0041959	0.0040572
EM Var at x = n	0.0000002	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000

Monte Carlo simulation for 0.75

	n = 25	n = 50	n = 75	n = 100	n = 200	n = 500	n = 1000
K MSE at x = 1	0.0054416	0.0056046	0.0048626	0.0042958	0.0036225	0.0032114	0.0030909
K MSE at x = n/4	0.0125095	0.0103540	0.0074822	0.0074763	0.0061871	0.0050482	0.0045469
K MSE at x = 3n/4	0.0006765	0.0006789	0.0005578	0.0005901	0.0006093	0.0005703	0.0005235
K MSE at x = n	0.0018977	0.0018514	0.0013714	0.0010839	0.0006387	0.0003988	0.0003381
EM MSE at x = 1	0.0000004	0.0000001	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000
EM MSE at x = n/4	0.0136158	0.0076928	0.0064534	0.0056110	0.0047503	0.0053307	0.0065575
EM MSE at x = 3n/4	0.0017634	0.0008080	0.0006973	0.0006294	0.0005883	0.0005474	0.0005669
EM MSE at x = n	0.0000001	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000
Estimated SR bw	0.8461822	0.8594501	0.8025137	0.7645264	0.6723683	0.5666276	0.4963150
mu1	1.9897583	1.9891583	1.9902142	1.9844033	1.9892586	1.9879942	1.9875304
mu2	5.9559744	5.9579423	5.9587145	5.9591557	5.9623503	5.9588146	5.9615105
sigma1	0.9726074	0.9702086	0.9712566	0.9733901	0.9732875	0.9733008	0.9728014
sigma2	0.9287572	0.9340402	0.9580003	0.9715228	0.9871238	0.9968913	1.0019652
kern	0.1803585	0.1216482	0.0979978	0.0848160	0.0661887	0.0547938	0.0510947

Monte Carlo simulation for 0.75

	n = 25	n = 50	n = 75	n = 100	n = 200	n = 500	n = 1000
K Bias at x = 1	0.0669137	0.0681039	0.0626207	0.0585169	0.0531251	0.0494305	0.0482427
K Bias at x = n/4	0.1076498	0.0963495	0.0765044	0.0761112	0.0644516	0.0498054	0.0418991
K Bias at x = 3n/4	0.0118165	0.0127412	0.0102495	0.0127907	0.0155430	0.0158724	0.0139398
K Bias at x = n	0.0399694	0.0397475	0.0337030	0.0296074	0.0222903	0.0172585	0.0158570
EM Bias at x = 1	0.0004601	0.0001375	0.0000525	0.0000063	-0.0000775	-0.0001542	-0.0001945
EM Bias at x = n/4	0.0465713	0.0298563	0.0187091	0.0101376	-0.0100220	-0.0339127	-0.0496365
EM Bias at x = 3n/4	0.0076079	0.0057518	0.0045441	0.0040610	0.0024424	-0.0013626	-0.0063874
EM Bias at x = n	0.0002440	0.0000482	0.0000231	0.0000091	-0.0000141	-0.0000378	-0.0000547

Monte Carlo simulation for 0.75

	n = 25	n = 50	n = 75	n = 100	n = 200	n = 500	n = 1000
K Var at x = 1	0.0009642	0.0009664	0.0009413	0.0008716	0.0008002	0.0007680	0.0007636
K Var at x = n/4	0.0009210	0.0010707	0.0016293	0.0016834	0.0020331	0.0025676	0.0027914
K Var at x = 3n/4	0.0005368	0.0005165	0.0004528	0.0004265	0.0003677	0.0003184	0.0003292
K Var at x = n	0.0003002	0.0002715	0.0002355	0.0002073	0.0001419	0.0001010	0.0000866
EM Var at x = 1	0.0000002	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000
EM Var at x = n/4	0.0114469	0.0068014	0.0061034	0.0055082	0.0046498	0.0041807	0.0040937
EM Var at x = 3n/4	0.0017055	0.0007749	0.0006766	0.0006129	0.0005823	0.0005455	0.0005261
EM Var at x = n	0.0000001	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000