Problem Set: Non Parametric Statistics: Group 18

Cesar Conejo Villalobos, Xavier Bryant

3/30/2021

Question 1

Problem Description

Assess S claims of battery failure from temperatures from a sample of previous batteries that experienced failure and from a sample of past battery temperatures in general.

a. Perform a kernel density estimation for temps-7 and temps-other using what you consider is the most adequate bandwidth. Since the temperatures are positive, is it required to perform any transformation?.

b. Is there any important difference on the results from considering the LSCV selector over the DPI selector?

c. It seems that in temps-7 there is a secondary mode. Compute a kernel derivative estimation for temps-7 and temps-other using what you consider are the most adequate bandwidths.

d. Precisely determine the location of the extreme points

e. Check with a kernel second derivative that the extreme points are actually modes.

Preliminary Work

We see the summary of the "Problematic phone" series, phones that have burned due to excessive temperature, and then the "Past phone" series of the working temperature of general phones in the past. We notice that the problematic phones have a much higher maximum value but a similar median. This leads us to suspect that phones that do burn up, do more often, have higher temperatures that lead to their failure.

```
## [1] "Problematic phones"
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    6.85   12.51   15.68   17.34   19.52   62.84
```

```
## [1] "Past phones"
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    5.39   12.69   17.22   17.76   22.01   46.86
```

We can see comparing the histograms in figure 1 that, as we saw in the summaries, the problematic phones have much more disperse temperatures and a longer tail in the direction of higher temperatures. This indeed leads us to suspect that phones that do fail have a section of their series that have higher temperatures. However, phones that fail have temperatures that are normally, for most of the sample, in line with those of the general past phones.

We see in our initial kernel densities with default bandwidths (Figure 2) that there is a tail on the problematic series with a second mode for the high temperatures. There is also some distortion at the mode of the past battery curve, but the tail of high temperatures is not present.

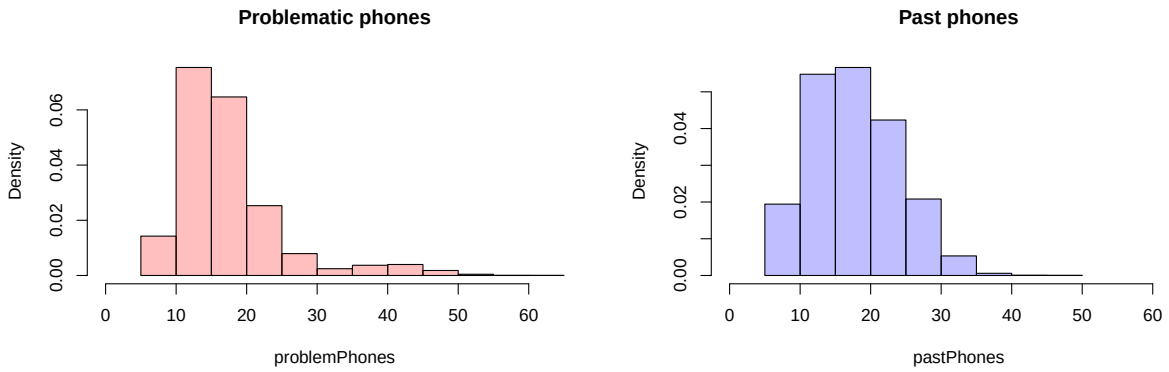

Figure 1: Comparison of histograms: Default bandwidths

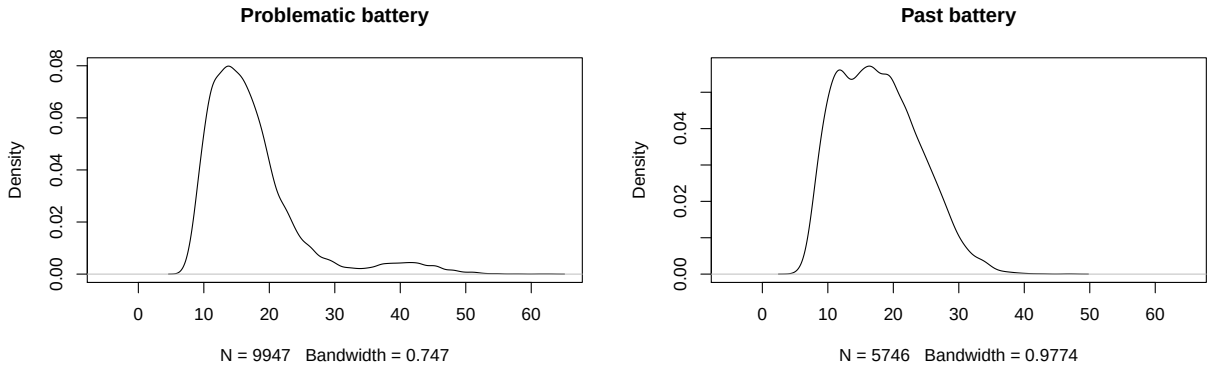

Figure 2: Comparison of densities; Default bandwidths

Part a

Problem Description

Perform a kernel density estimation for temps-7 and temps-other using what you consider is the most adequate bandwidth. Since the temperatures are positive, is it required to perform any transformation?.

Results

We can see these are the default bandwidths that are provided for the density estimates shown above.

```
## [1] "Problematic Temperatures"
## [1] 0.747
## [1] "Past Temperatures"
## [1] 0.977
```

Transformations Transformations must be performed in order to avoid boundary bias, which is to assign probability, with the function mapping, when there is no real density to support it or can be bias towards zero. We are looking at temperatures that can take values close to zero or very high values, therefore this would make us consider a log transformation. Below in Figure 3 and Figure 4, we do not see the shape of the curve change substantially - there are not sections that are mapped by the function but not supported by any density - nor do we see much boundary bias. However, the log transformation does further smooth our data - which for our analysis comparing the overall distribution of batteries - is helpful. This is due to the smaller bandwidth that is enabled due to the transformation, which also advantageous due to asymptotic properties. However, the untransformed series functions as well for our purposes, as a more precise curve is not necessarily important for our general type of analysis, and we don't see significant bias. Therefore, we proceed with the regular series.

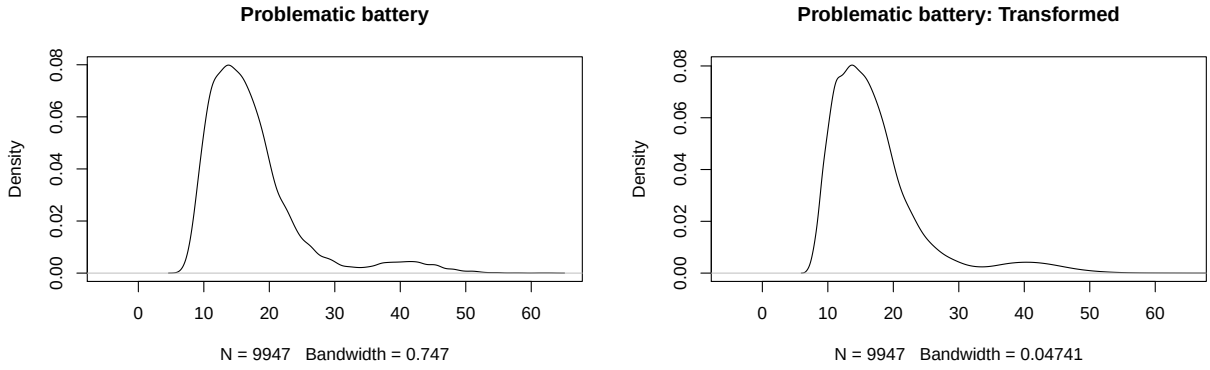

Figure 3: Comparisono default and transformed densities for Problematic series

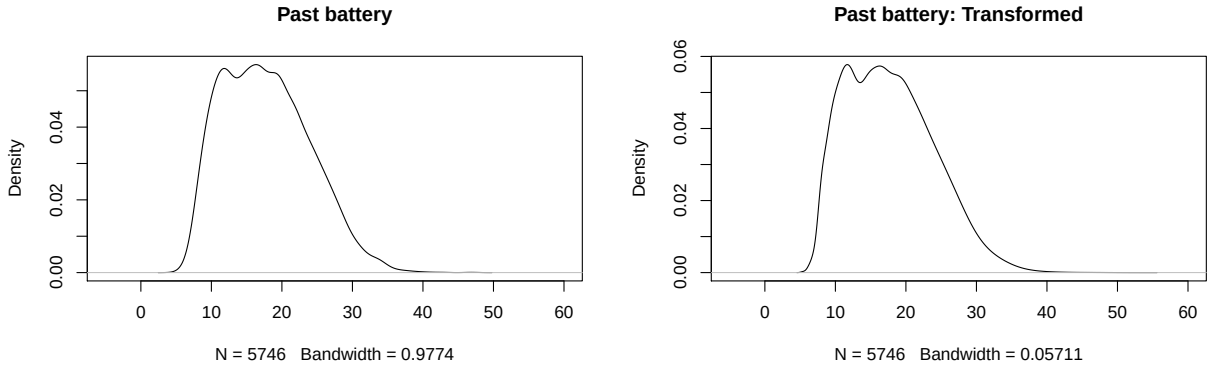

Figure 4: Comparison of default and transformed densities for Past series

Bandwidth Selection We will use several techniques to select the optimal bandwidth. The first technique is "rule of thumb" bandwidth selection derived from minimizing the AMISE with a normal parametric assumption for the unknown $R(f'')$, in zero stage part of the AMISE (asympotic mean integrated squared error) derivation: $h_{AMISE} = [\frac{R(K)}{\mu_2^2(K)R(f'')n}]^{\frac{1}{5}}$. We then look at the Direct Plug In (DPI) selector, which instead of using a normal parameterization in the zero stage, the DPI buries the parametric assumption, usually, with two stages which balances between bias (lower with number of stages) and variance (higher with the number of stages). We then look at two cross-validation techniques, where we attempt to minimize the MISE (mean integrated squared error), through means of cross validation utilizing leave-one-out techniques. The two types of CV analysis are: the Least Squared Cross-Validation (or Unbiased Cross Validation) and the Biased Cross Validation. Numerical optimization is required for the LSCV and therefore we can get trapped in spurious solutions, necessitating the limiting of the bandwidth grid for the $\hat{h}_{LSCV}$. The BCV adapts a hybrid strategy estimating a modification of $R(f'')$ with leave-out-diagonals. It is important that the plot or the grid is reviewed in this selector as well as we are looking for the local minimizer not the global as h, the bandwidth , goes to infinity. We ensure to limit the bandwidth grid for both these functions, when necessary, from the function shown in class.

First we see the bandwidths of the "rule of thumb" selector:

```
## [1] "Past temperatures"
## [1] 1.151
## [1] "Problematic temperatures"
## [1] 0.88
```

Second we see the DPI bandwidths:

```
## [1] "Past temperatures"
## [1] 0.841
## [1] "Problematic temperatures"
## [1] 0.631
```

Third, we have the LSCV bandwidths:

```
## [1] "Past temperatures"
## [1] 0.64052
## [1] "Problem temperatures"
```

```
## [1] 0.6555154
```

Lastly we have the BCV bandwidths:

```
## [1] "Past temperatures"
```

```
## [1] 0.8766243
```

```
## [1] "Problematic temperatures"
```

```
## [1] 0.6149622
```

To summarize we have:

Past temperature bandwidths: - RT: 1.15 - DPI: 0.84 - LSCV: 0.64 - BCV: 0.88

Problematic temperature bandwidths: - RT: 0.87 - DPI: 0.63 - LSCV: 0.66 - BCV: 0.61

Reviewing our results, we see that that the RT gives us bandwidths that are quite large in comparison with the other bandwidth selectors, which is common for non-normal data like our own. From the literature, we know from the DPI selector has a convergence rate that is much faster than the cross-validation technique, and therefore is dominant among academics. We also note that the BCV tends to be more bias, but have substantially less variance than the LSCV, tending to have larger bandwidths. However, we notice that the default values for bandwidth of 0.75 for the problematic series and, for the past series, 0.98 are fairly similar and, considering part b, we decide to use the default bandwidths as, for our purpose of comparing the series at its entirety, the differences are not important.

Part b

Problem Description

Is there any important difference on the results from considering the LSCV selector over the DPI selector?

Results

Bandwidths for LSCV and BCV:

Past temperature bandwidths: - DPI: 0.84 - LSCV: 0.64

Problematic temperature bandwidths: - DPI: 0.63 - LSCV: 0.66

We can see that the bandwidths of LSCV and DPI for the past temperatures differ significantly more than those for the problematic temperatures. The difference between the bandwidths is only 0.03 for the problematic temperatures. It is 0.2 for the past temperatures. In the literature, the DPI has much higher convergence rate than cross-validation methods, like LSCV, however, with very volatile or non-normal data the DPI may over smooth. We may be seeing this in practice as the LSCV does have a smaller bandwidth that is capturing more of disruption around the mode in the past temperatures series. The sample size is also considerably larger in the problematic temperature series so this may allow the two techniques to more closely approach one another. For functional purposes, they are fairly equivalent to us as we are comparing the overall past and problematic series, and both bandwidths more or less find similar curves.

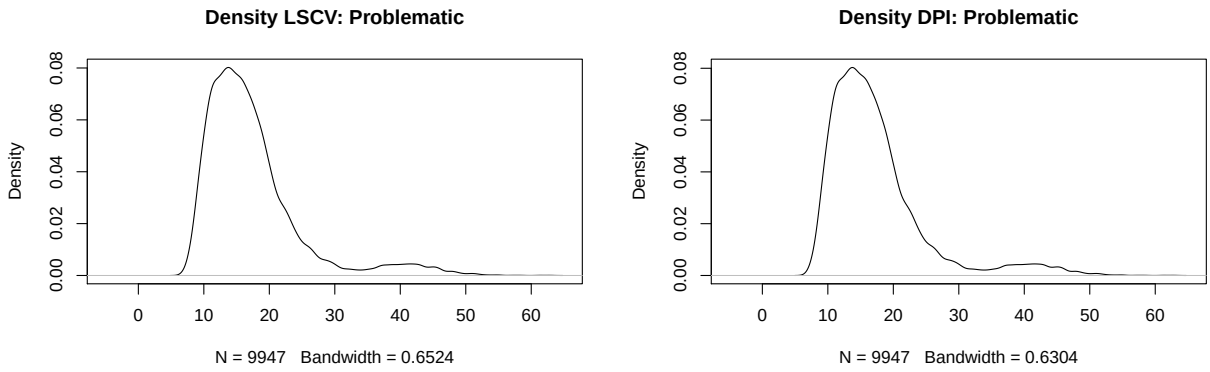

Figure 5: LSCV and DPI densities problematic series

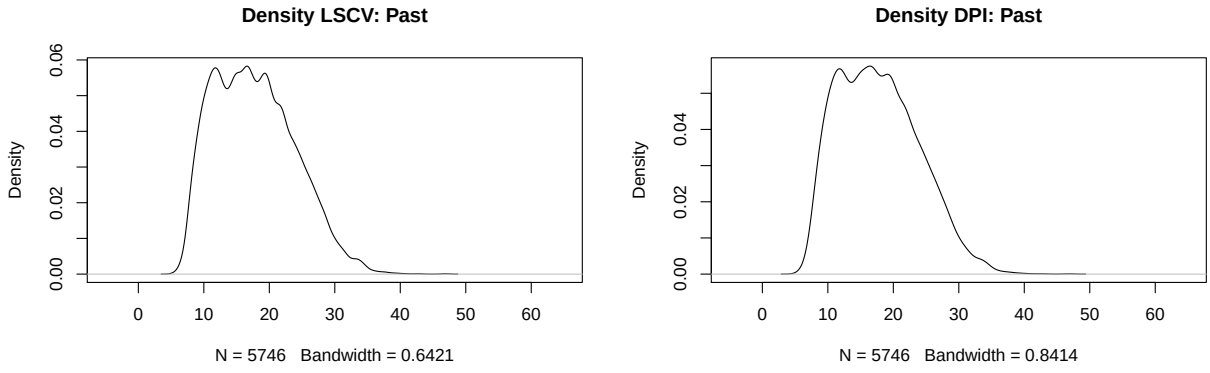

Figure 6: LSCV and DPI densities past series

Part c

Problem Description

It seems that in temps-7 there is a secondary mode. Compute a kernel derivative estimation for temps-7 and temps-other using what you consider are the most adequate bandwidths.

Results

We can see indeed that there is two modes in the problematic series. This second mode is key for our analysis as it likely shows that this mode of high temperatures could be largely responsible for the failure of these cellphones rather than those in general, or the past temperature series. There is further evidence they are modes as the the first derivative crosses the x-axis at these points. We also see a minimum in red.

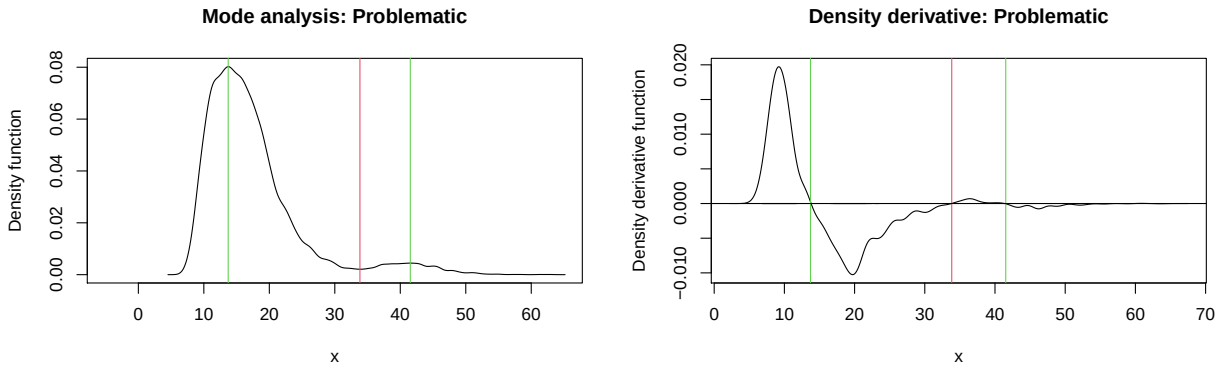

Figure 7: Mode and derivative estimation analysis: Problematic series

We can see the mode identified in the past phone series with the green line. We can see that the first derivative crosses the x-axis several times reflecting the disturbance we have around the mode of the series. However, we can still see that a majority of the data falls in the region around 10 and 20. This general idea is suitable for our analysis.

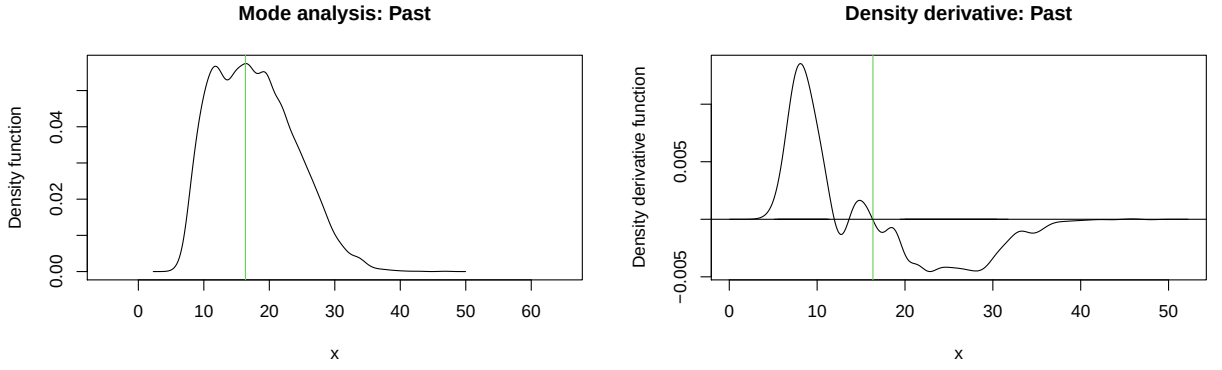

Figure 8: Mode and derivative estimation analysis: Past series

Part d

Problem Description

Precisely determine the location of the extreme points.

Results

We can see the extreme points of the problematic series for the two modes (13.72, 0.08) and (41.53, 0.01) as well as the minimum point (33.83, 0.002). For the past series, we can see the mode at (31.36, 0.007).

```
## [1] "Problematic temperatures"
## [1] "Mode 1: x-value"
## [1] 13.72123
## [1] "Mode 2: x-value"
## [1] 41.53123
## [1] "Minimum: x-value"
## [1] 33.83911
## [1] "Mode 1: y-value"
## [1] 0.0798685
## [1] "Mode 2: y-value"
## [1] 0.004438786
## [1] "Minimum: y-value"
## [1] 0.002133435
## [1] "Past temperatures"
## [1] "Mode 1: x-value"
## [1] 31.35863
## [1] "Mode 1: y-value"
## [1] 0.006814199
```

Part e

Problem Description

Check with a kernel second derivative that the extreme points are actually modes.

Results

We can confirm that the points of the modes for the problematic series are indeed the modes as the they remain on the opposite side of the second derivative estimator. A mode should be on the negative side of the second derivative and a minimum should be on the positive side. We see this is true for both the modes and minimum of the problematic series.

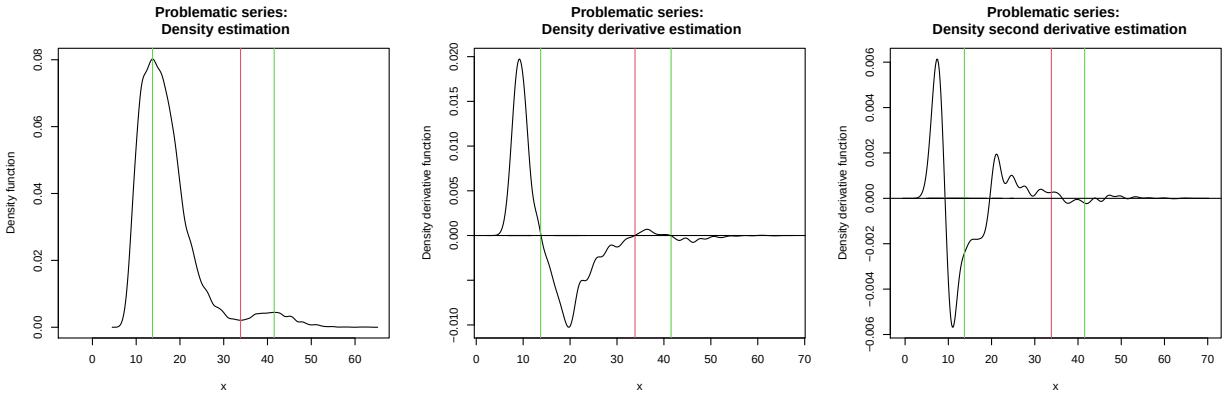

Figure 9: Second derivative analysis: Problematic series

We see this is also true for the past series, where the mode is on the negative side of the second derivative estimator.

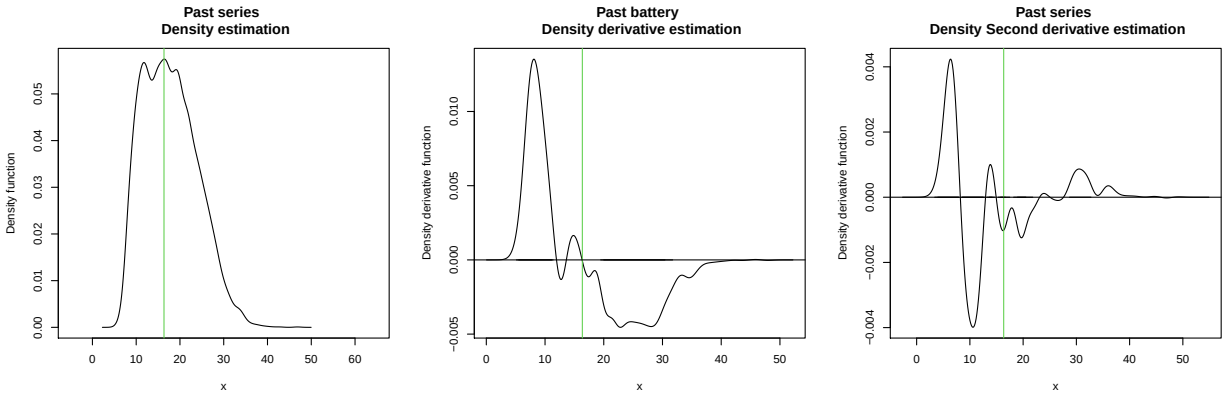

Figure 10: Second derivative analysis: Past series

Question 2

Problem Description

Compare the MISE and AMISE criteria in three densities in a `nor1mix` of your choice.

1. Code (2.33) and the AMISE expression for the normal kernel, and compare the two error curves.
2. Compare them for $n = 100, 200, 500$, adding a vertical line to represent the h_{MISE} and h_{AMISE} bandwidths. Describe in detail the results and the major takeaways.

Response

For this exercise, we require some mathematical ideas that we will develop briefly.

We start with the KDE estimator $\hat{f}(x;h) = \sum_{i=1}^{n} K_h(x - X_i)$. The expectation and variance for this estimator are given by the following expressions:

(1) $E[\hat{f}(x;h)] = (K_h * f)(x)$.

(2) $\text{Var}[\hat{f}(x;h)] = \frac{1}{n}((K_h^2 * f)(x) - (K_h * f)^2(x))$

Then, we develop some asymptotic expressions for (1) and (2):

(3) $E[\hat{f}(x;h)] - f(x) = \text{Bias}[\hat{f}(x;h)] = \frac{1}{2}\mu_2(K)f''(x)h^2 + o(h^2)$

(4) $\text{Var}[\hat{f}(x;h)] = \frac{R(K)}{nh}f(x) + o((nh)^{-1})$

Then, from equations (3) and (4) we obtain the following expression for the MSE:

(5) $\text{MSE}[\hat{f}(x;h)] = \frac{\mu_2^2(K)}{4}(f''(x))^2 h^4 + \frac{R(K)}{nh}f(x) + o(h^4 + (nh)^{-1})$

It is important to note that in (3), (4) and (5) we define K and $R(K)$ as:

(6) Second order moment of K: $\mu_2(K) := \int z^2 K(z)dz$

(7) Squared integral of kernel: $R(K) := \int (K(x))^2 dx$

We are able to define $\text{MISE}[\hat{f}(\cdot;h)]$ as global error criteria for measuring the performance of $\hat{f}$ in relation to the target density f by taking the the intergral of the of MSE.

(8) $\text{MISE}[\hat{f}(\cdot;h)] = \int \text{MSE}[\hat{f}(x;h)]$

Therefore, we obtain the following asymptotic expansion for the MISE:

(9) $\text{MISE}[\hat{f}(\cdot;h)] = \frac{1}{4}\mu_2^2(K)R(f'')h^4 + \frac{R(K)}{nh} + o(h^4 + (nh)^{-1})$

We define the dominant part of equation (9) (the little o will asymptotically approach zero more quickly) as $\text{AMISE}[\hat{f}(\cdot;h)]$. In particular:

(10) $\text{AMISE}[\hat{f}(\cdot;h)] = \frac{1}{4}\mu_2^2(K)R(f'')h^4 + \frac{R(K)}{nh}$

with the expression $R(f'')$ given by:

(11) $R(f'') = \int (f''(x))^2 dx$

We can see, finally, the bandwidth that minimizes the AMISE is:

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

Now, we consider our particular case of study. In this case, we use the following assumptions to *reduce* our analysis and proceed in our analysis:

a) A normal kernel $K_h(\cdot)$ with distribution $\mathcal{N}(0,1)$

b) The density function f is based on the family of normal r-mixtures, therefore we obtain:

(13) $f(x;\mu,\sigma,\boldsymbol{w}) = \sum_{j=1}^{r} w_j \phi_{\sigma_j}(x-\mu_j)$

where $w_j \geq 0$, $j = 1, ..., r$ and $\sum_{j=1}^{r} w_j = 1$.

With these two expressions, we can obtain a specific value for the AMISE in equation (10). Utilizing assumption a), we obtain the following expressions for equations (6) and (7):

(6.1) Second order moment of K: $\mu_2(K) = 1$

(7.1) Squared integral of kernel: $R(K) = \frac{1}{2\sqrt{\pi}}$

Expression for equation (11) can also be derived following *Theorem 4.1* from *Marron and Wand (1992)*

(11.1) $R(f^{''}) = \int \left(f^{''}(x)\right)^2 dx = \sum_{j=1}^{k} \sum_{j'=1}^{k} w_j w_{j'} \phi^4_{\sigma_{jj'}}(\mu_j - \mu_{j'})$

Where $\sigma_{jj'} = (\sigma_j^2 + \sigma_{j'}^2)^{1/2}$. Additionally, we use the *Probabilist Hermite Polynomial* of order 4: $\phi^4(x) = \phi(x)H_4(x) = \phi(x)(x^4 - 6x^2 + 3)$. With this expression, we obtain the reduced form of the AMISE:

(10.1) $\text{AMISE}[\hat{f}(\cdot;h)] = \frac{1}{4}R(f^{''})h^4 + \frac{1}{2nh\sqrt{\pi}}$

With optimal bandwidth h_{AMISE} given by:

(12.1) $h_{AMISE} = \left[\frac{(2\sqrt{\pi})^{-1}}{R(f^{''})n}\right]^{1/5}$

Finally, under this assumptions, we obtain a explicit and exact MISE expression of equation (8):

(14) $\text{MISE}_r[\hat{f}(\cdot;h)] = (2\sqrt{\pi}nh)^{-1} + \boldsymbol{w}^{'}\{(1-n^{-1})\Omega_2 - \Omega_1 + \Omega_0\}\boldsymbol{w}$

Where:

(15) $(\Omega_a)_{i,j} = \phi_{(ah^2+\sigma_i^2+\sigma_j^2)^{1/2}}(\mu_i - \mu_j)$ for $i, j = 1, ..., r$

Finally, we can proceed evaluating numerically with equation (14) and obtain:

(16) $\arg\min_{h>0} \text{MISE}[\hat{f}(\cdot;h)]$

With this mathematical review, we are able to compare the MISE and AMISE criteria. We code the following functions in `R`:

- `omega_a()`: Computes the matrix $(\Omega_a)_{i,j}$ defined in equation (15).
- `omega()`: Computes the `scalar` given by the following vector and matrix operations of equation (14): $\Omega = \boldsymbol{w}^{'}\{(1-n^{-1})\Omega_2 - \Omega_1 + \Omega_0\}\boldsymbol{w}$
- `MISE()`: Useful for computing the expression of equation (14).
- `AMISE()`: Useful for computing the expression of equation (10.1). We also code the auxiliary functions `Hermite_4()` and `R_f2()` related to equation (11.1).
- Value of h_{AMISE} is obtained from the code function `h_AMISE_n()`. Value of h_{MISE} is obtained numerically from equations (15) and (15) using `optimize()`.

We consider in particular the following three densities of the `nor1mix` package whose parameter are available in R Help

- `MW.nm1`: Gaussian $\mathcal{N}(0,1)$
- `MW.nm2`: Skewed $.2\mathcal{N}(-.3, 1.44) + .2\mathcal{N}(.3, .64) + .6\mathcal{N}(1, 4/9)$
- `MW.nm6`: Bimodal $.5\mathcal{N}(-1, 4/9) + .5\mathcal{N}(1, 4/9)$

In the case using the data set `nor1mix::MW.nm1`, figure 11 shows the histograms for the sample distribution for $n = 100, 200, 500$ and shows the population value in the red density curve. As expected, when we increase the number of observations the sample histogram approaches the distribution of the population density curve.

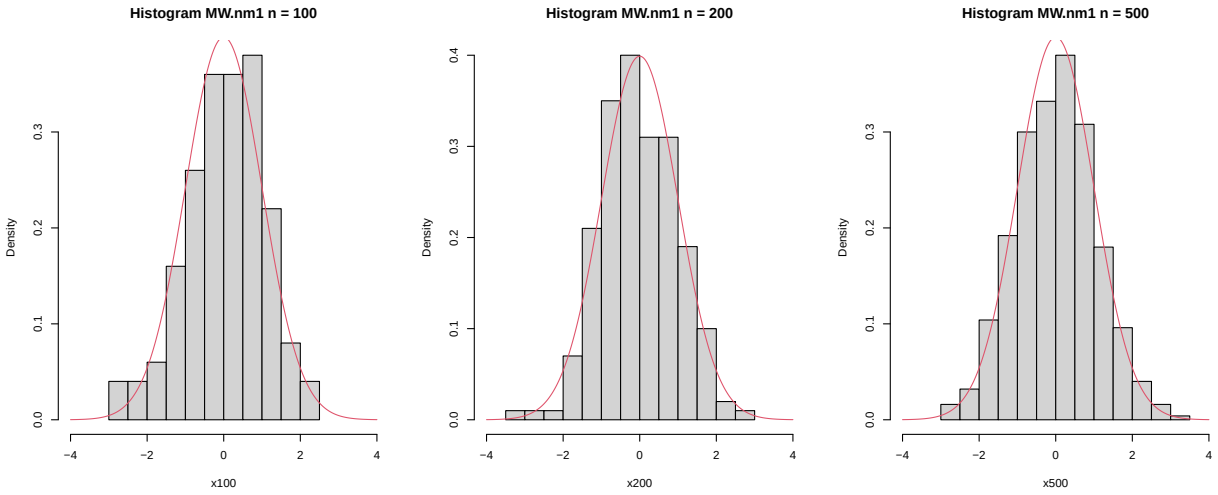

Figure 11: Histograms from sample obtained from n = 100, 200 and 500 for object MW.nm1

Figure 12 below shows a graphical representation of the MISE and AMISE related to `nor1mix::MW.nm1` for bandwidth ranges between 0.04 and 0.85. The optimal values of h_{AMISE} are 0.422, 0.367 and 0.306. On the other hand,optimal values of h_{MISE} are 0.445, 0.383 and 0.315 for $n = 100, 200, 500$ respectively. In the three curves, we can see that for each sample size, we have that $h_{\text{AMISE}} < h_{\text{MISE}}$. So, the asymptotic approximation underestimate the value of h_{MISE}.

Additionally, when we reduce the bandwidth size, the MISE and AMISE values are quite similar. Additionally, we can see the effect of increasing the number of samples n in the y axis of the plot. (The scale is reduced). So, the difference between both gets closer as the sample size get larger.

In other words, when we increase the number of samples n and reduce the value of the bandwidth, the curves for MISE and AMISE is quite similar.

Figure 13 shows the histograms for the skewed object `MW.nm2`. In figure 14 we observe the comparisons of MISE(h) and AMISE(h). Similar to the previous graphics, the approximation of these curves is suitable for small h but lower for large h. However, in this case, we notice how the asymptotic approximation slightly overestimate the value of h_{MISE}, but the difference between h_{MISE} and h_{AMISE} are closer than scenario given by `MW.nm1`.

Specifically, values of h_{AMISE} are given by 0.375, 0.327 and 0.272. On the other hand,optimal values of h_{MISE} are 0.366, 0.313 and 0.256 for $n = 100, 200, 500$ respectively.

For the object `MW.nm6` Figure 15 shows the histograms for the Bimodal object. Like figure 11, increasing the number of observations conducts to a better fitting to the real population.

In figure 16 we see a similar phenomena than figure 12. In this case, the asymptotic MISE underestimate the value for MISE. The minimizers of AMISE(h) are given by h_{AMISE} are 0.352, 0.306 and 0.255. On the other hand, minimizers of MISE(h) corresponds to 0.385, 0.322 and 0.258 for $n = 100, 200, 500$ respectively.

In this case, we observe how for $n = 500$ the approximation is the same for the first decimal point.

In general terms, we have the following conclusions:

- From equation (10.1), AMISE(h) $\to \infty$ as $h \to \infty$. However, in the case of MISE(h), we see how MISE(h) increase more slowly than AMISE(h).

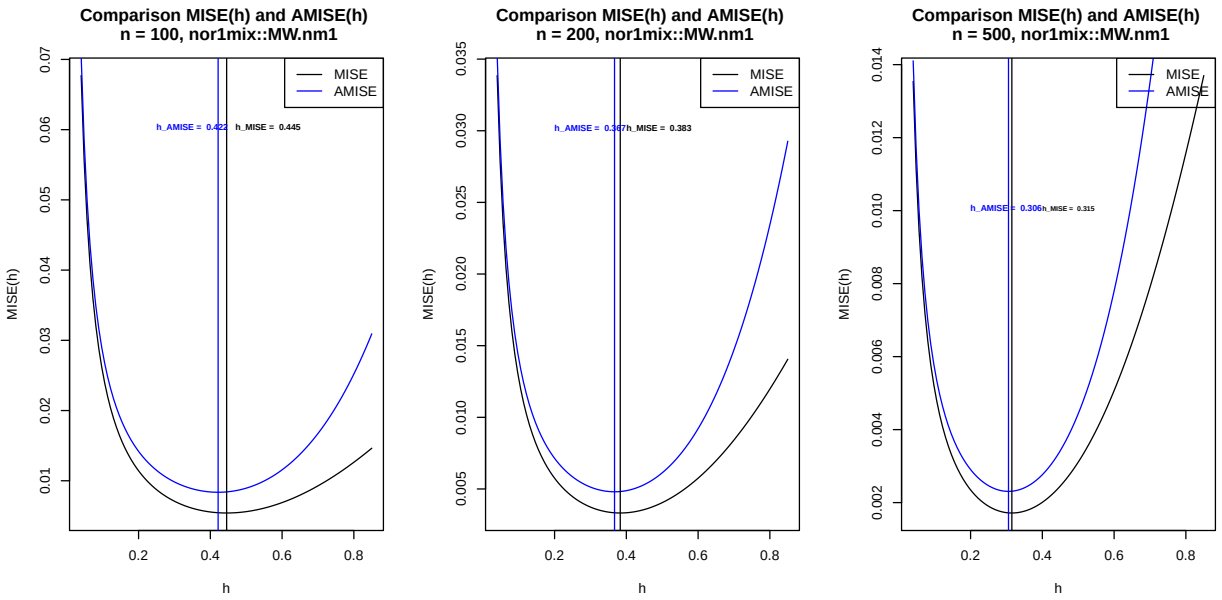

Figure 12: MISE and AMISE for range bandwith between 0.04 and 0.85

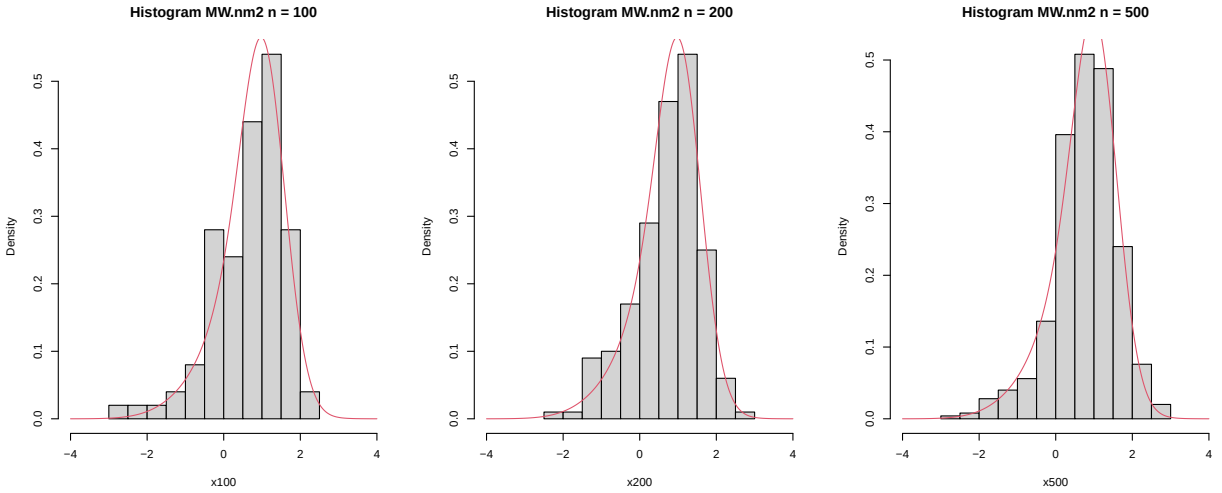

Figure 13: Histograms from sample obtained from n = 100, 200 and 500 for object MW.nm2 (Skewed)

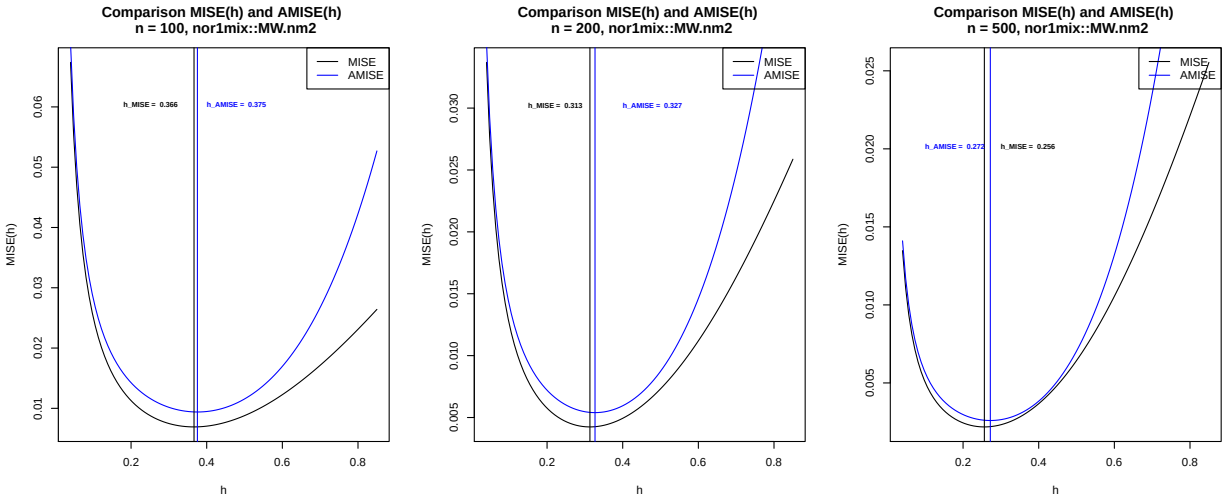

Figure 14: MISE and AMISE for range bandwith between 0.04 and 0.85

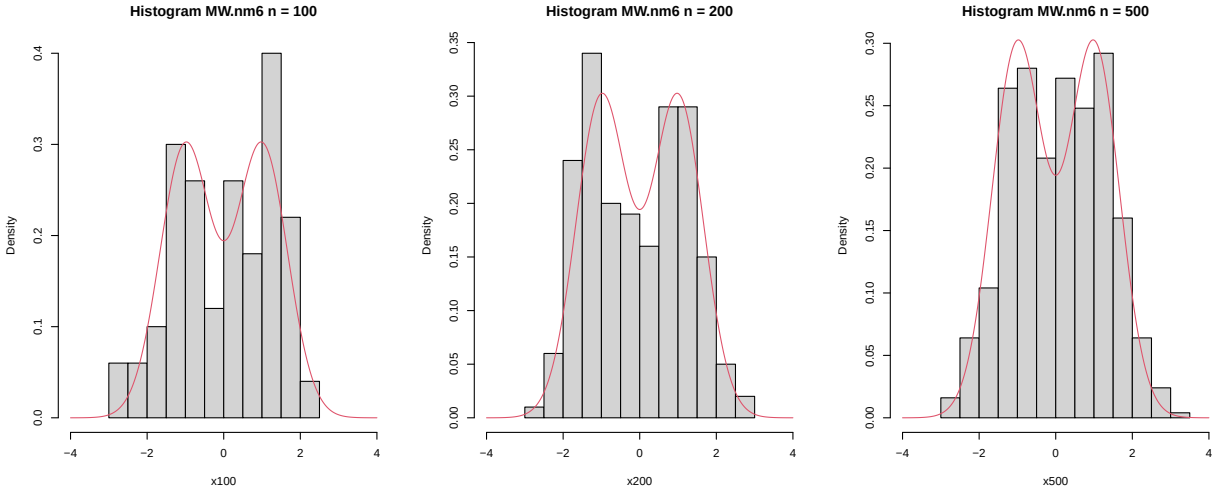

Figure 15: Histograms from sample obtained from n = 100, 200 and 500 for object MW.nm6 (Bimodal)

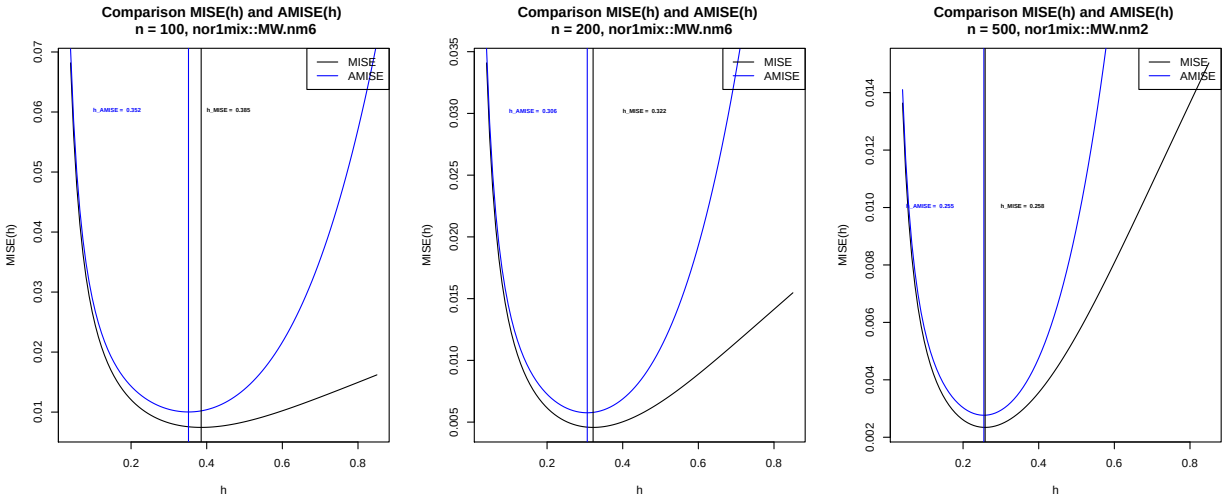

Figure 16: MISE and AMISE for range bandwith between 0.04 and 0.85

Question 3

Problem Description

Adapt the `np_pred_CI` function to include the argument `type_boot`, which can take either the value `"naive"` or `"wild"`.

1) If `type_boot = "wild"`, then the function must perform the wild bootstrap algorithm, implemented from scratch following substeps i–iv.

2) Compare and validate the correct behavior of the confidence intervals, for the two specifications of type_boot, in the model considered in Exercise 5.8 (without performing the full simulation study).

Response

For this question we will describe briefly the necessary steps related to the implementation of the wild-bootstrap:

First of all, for a given sample with n observations, the predicted (fitted) values are given by the expression: $\hat{Y}_i := \hat{m}(X_i; q, h)$ with $i = 1, ..., n$.

Then, the uncertainty of $\hat{m}(x, q, h)$ (this includes the consideration of new set of samples) can be measure based on two approaches:

1. Asymptotic approach: $\hat{m}(x; q, h) \pm \hat{se}(\hat{m}(x; q, h))$ where $\hat{se}(\hat{m}(x; q, h))$ is the asymptotic estimation of the standard deviation of $\hat{m}(x; q, h)$

2. Bootstrap: In this case, we consider two types of bootstrap:

   - Naive Bootstrap: From the original sample, we create a new set of IID observations called the *Bootstrap sample*

   - Wild Bootstrap: This technique is focused on resampling the residuals. It is similar to the *residual bootstrap* in which we fixed the covariate X_i and generate a new value of Y_i using the fitted model and the *noise* from sampling the residuals. In addition, we generate IID random variables V_i (random variable with mean zero and variance 1) and then we can find the perturbed residuals

by multiplying this variable against the residults. This modification improves the stability of the computations, especially in the presence of heteroskedasticity.

The algorithm of the computation of the wild bootstrap is given in the following steps:

1. Compute $\hat{m}(x;q,h) = \sum_{i=1}^{n} W_i^q(x) Y_i$ from the original sample $(X_1, Y_1), ..., (X_n, Y_n)$
2. Enter the wild bootstrap. For $b = 1, ..., B$.
   i. Simulate $V_1^{*b}, ..., V_n^{*b}$ to be iid copies of V such that $E[V] = 0$ and $Var[V] = 1$,
   ii. Compute the *pertubed residuals* $e_i^{*b} = \hat{e}_i V_i^{*b}$ where $Y_i^{*b} := \hat{m}(X_i;q,h) + e_i^{*b}$ for $i = 1, ..., n$.
   iii. Obtain the bootstrap sample: $(X_1, Y_1^{*b}), ..., (X_n, Y_n^{*b})$ where $Y_i^{*b} := \hat{m}(X_i;q,h) + e_i^{*b}$ with $i = 1, ..., n$.
   iv. Compute $\hat{m}^{*b}(x;q,h) = \sum_{i=1}^{n} W_i^q(x) Y_i^{*b}$ from $(X_1^*, Y_1^{*b}), ..., (X_n^8, Y_n^{*b})$

As a result, we modified the function `np_pred_CI`. The inputs of the original function are the following:

1. `npfit`: A `np::npreg` object (npfit)
2. `exdat`: Values of the predictors were to carry out prediction (exdat)
3. `B`: Number of bootstrap iterations.
4. `conf`: Range of confidence interval
5. `type_CI`: Type of confidence interval. (Normal standard or quantiles)

Additionally, we add three new inputs:

6. `type_boot`: Type of bootstrap procedure. Options: `Naive` and `Wild` bootstrap
7. `perturbed_res`: For the case of `Wild` bootstrap, we can choose the type of perturbation. As explained before, any random variable with mean 0 and variance 1 can be used. As a result, we use two possible perturbation:

   7.1. `normal`: Based on the normal distribution $V_i \sim \mathcal{N}(0,1)$

   7.2 `golden`: Based on the *golden section binary variable* $P[V = 1 - \phi] = p$, $P[V = \phi] = 1 - p$ and $p = \frac{\phi+2}{5}$
8. `seed`: Used for reproducibility. Default option is `seed = 42`.

The code of the new version of `np_pred_CI` is given below. We create a new `if` condition, in order to compute the type of bootstrap resampling method chosen by the user. In the case of the `wild` bootstrap, we focus on resampling the residuals obtained from the difference of the fitted value $\hat{Y}_i$ and the *real* values Y_i. Moreover, inside the option `type_boot == wild`, there is an `if` condition with the two possible alternatives for computing the perturbed residuals.

```
# Function to predict and compute confidence intervals for m(x).

# 1) Inputs

## 1.1) npfit: A np::npreg object (npfit)
## 1.2) exdat: Values of the predictors where to carry out prediction (exdat)
## 1.3) B:     Number of bootstrap iterations.
## 1.4) conf:  Range of confidence interval
## 1.5) type_CI: Type of confidence interval. (Based on normal standard or quantiles)
## 1.6) type_boot: Type of bootstrap procedure. Options Naive and Wild bootstrap
## 1.7) perturbed_res: Valid only for Wild Bootstrap. Type of perturbation on the residuals.
## Options are "normal"or "golden"
## 1.8) seed: Used for reproducibility. Default option is seed = 42.
```

```r
# 2) Outputs

## 2.1) exdat: Values of the predictors where to carry out prediction
## 2.2) m_hat: Predicted regression
## 2.3) lwr:   Lower confidence interval
## 2.4) upr:   Upper confidence interval

np_pred_CI <- function(npfit,
                       exdat,
                       B = 200,
                       conf = 0.95,
                       type_CI = c("standard", "quantiles")[1],
                       type_boot = c("naive", "wild")[1],
                       perturbed_res = c("normal", "golden")[1],
                       seed = 42) {

  # Fix seed
  set.seed(seed)

  # Extract predictors
  xdat <- npfit$eval

  # Extract response, using a trick from np::npplot.rbandwidth
  tt <- terms(npfit$bws)
  tmf <- npfit$bws$call[c(1, match(c("formula", "data"),
                                   names(npfit$bws$call)))]
  tmf[[1]] <- as.name("model.frame")
  tmf[["formula"]] <- tt
  tmf <- eval(tmf, envir = environment(tt))
  ydat <- model.response(tmf)

  # Predictions m_hat from the original sample
  m_hat <- np::npreg(txdat = xdat,
                     tydat = ydat,
                     exdat = exdat,
                     bws   = npfit$bws)$mean

  if (type_boot == "naive") {

    # Function for performing naive bootstrap
    boot_function_naive <- function(data, indices) {
      np::npreg(txdat = xdat[indices,],
                tydat = ydat[indices],
                exdat = exdat,
                bws   = npfit$bws)$mean
    }

    # Carry out the bootstrap estimator
    m_hat_star <- boot::boot(data = data.frame(xdat),
                             statistic = boot_function_naive,
                             R = B)$t

  } else if (type_boot == "wild") {
```

```r
# Sample size of the predictors
n <- length(xdat)

# Y fitted
Y_hat <- npfit$mean

# Ordinary residuals
residuals_O <- Y_hat - ydat

# Type of perturbation
if(perturbed_res == "normal"){

  # Function for performing wild bootstrap
  boot_function_wild <- function(data, indices) {

    # Step i: Simulate V_{i} copies of V (Mean 0 and variance 1)
    V_n <- rnorm(n)

    # Step iii. Obtain the bootstrap sample
    ydat_bt <- Y_hat + data[indices]*V_n

    np::npreg(txdat = xdat,
              tydat = ydat_bt,
              exdat = exdat,
              bws = npfit$bws)$mean
  }

  # Step iv. Carry out the wild bootstrap estimator
  m_hat_star <- boot::boot(data = residuals_O,
                           statistic = boot_function_wild,
                           R = B)$t

} else if(perturbed_res == "golden"){

  # Function for performing wild bootstrap
  boot_function_wild <- function(data, indices) {

    # Step i: Simulate V_{i} copies of V (Mean 0 and variance 1)
    phi <- (1 + sqrt(5))/2
    prob <- (phi + 2)/5

    golden <- sample(x = c(1-phi,phi), size = n, prob = c(prob, 1 - prob), replace=T)

    # Step iii. Obtain the bootstrap sample
    ydat_bt <- Y_hat + data[indices]*golden

    np::npreg(txdat = xdat,
              tydat = ydat_bt,
              exdat = exdat,
              bws = npfit$bws)$mean
  }

  # Step iv. Carry out the wild bootstrap estimator
```

```
      m_hat_star <- boot::boot(data = residuals_0,
                               statistic = boot_function_wild,
                               R = B)$t

    }

    else{stop("Incorrect type of peturbation")}

  }else{stop("Incorrect type_boot")}

  # Confidence intervals
  alpha <- 1 - conf

  if (type_CI == "standard") {

    z <- qnorm(p = 1 - alpha / 2)
    se <- apply(m_hat_star, 2, sd)
    lwr <- m_hat - z * se
    upr <- m_hat + z * se

  } else if (type_CI == "quantiles") {

    q <- apply(m_hat_star, 2, quantile, probs = c(alpha / 2, 1 - alpha / 2))
    lwr <- q[1, ]
    upr <- q[2, ]

  } else {
    stop("Incorrect type_CI")
  }
  # Return evaluation points, estimates, and confidence intervals
  return(data.frame("exdat" = exdat, "m_hat" = m_hat, "lwr" = lwr, "upr" = upr))
}
```

Finally, we compare and validate the correct behavior of the confidence intervals, for the two specifications of `type_boot` and the two types of perturbations. For this purpose, we simulate the following sample of size $n = 100$ from the regression model $Y = m(x) + \epsilon$ where $m(x) = 0.25x^2 - 0.75x + 3$, with $X \sim \mathcal{N}(0, 1.5^2)$ and $\epsilon \sim (0, 0.75^2)$. Figure 17 shows the simulated sample. In this case the simulated observations are concentrated in the left side of the plot.

Then, we fit a model using the function `npregbw()` with `regtype = "lc"` and the final model is created with the function `npreg()`. For reference, figures 18 and 19 shows the confidence interval under normal approximation and quantile compute by `np::npplot`. In particular, we focus on the right side of the confidence intervals. In the case of quantile confidence interval (figure 19) the upper confidence bound is relatively close to the fitted regression $\hat{m}(X_i; q, h)$

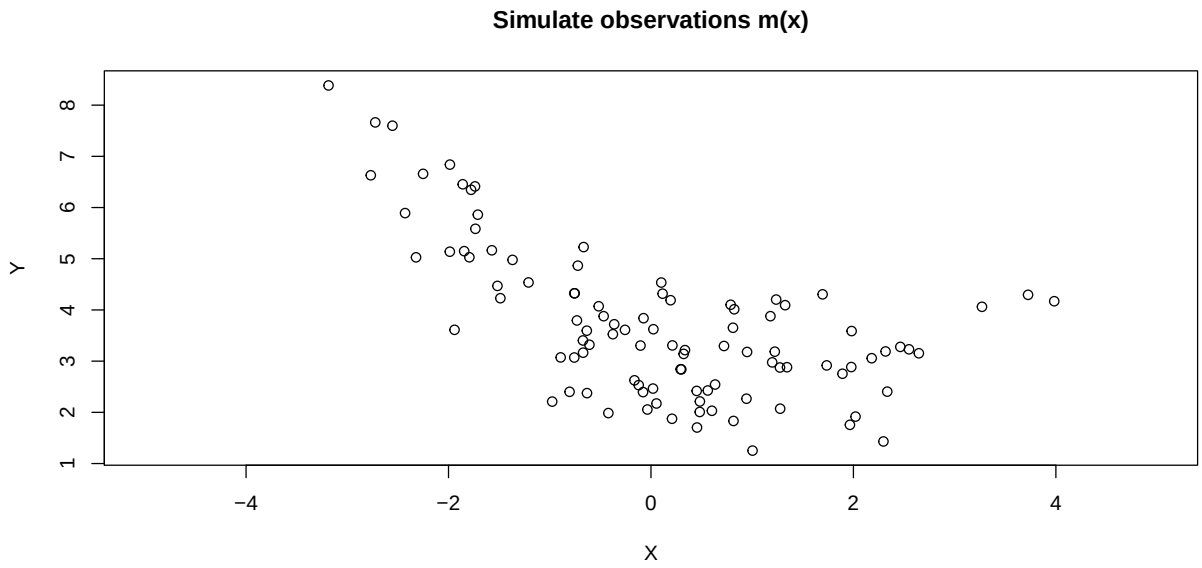

Figure 17: Simulation of 100 observations for m(X). Seed = 12345

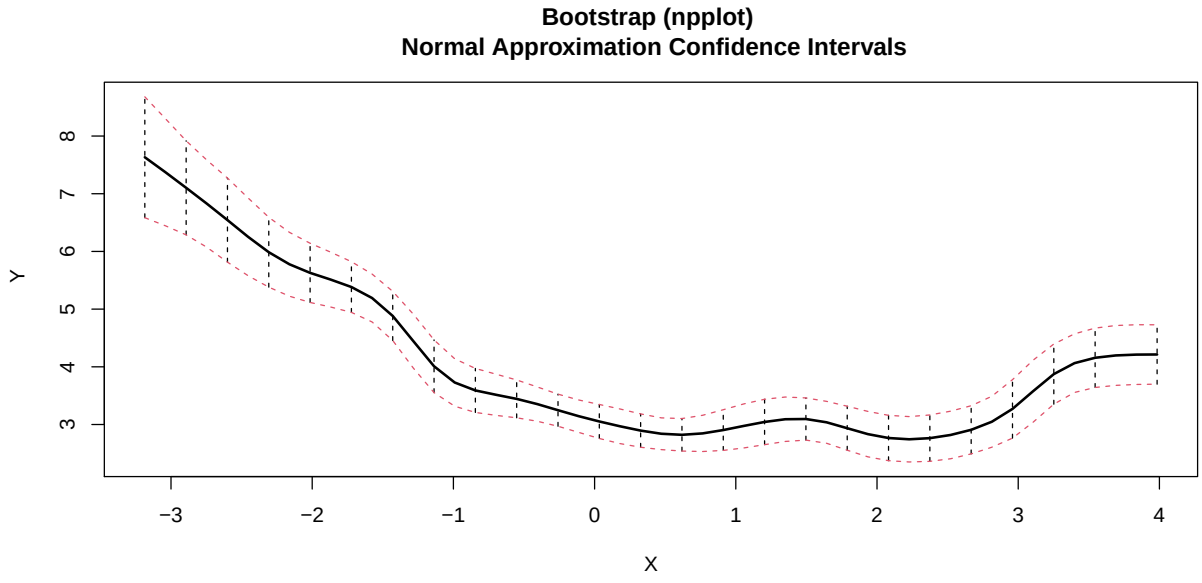

Figure 18: Bootstrap intervals: Default option npplot

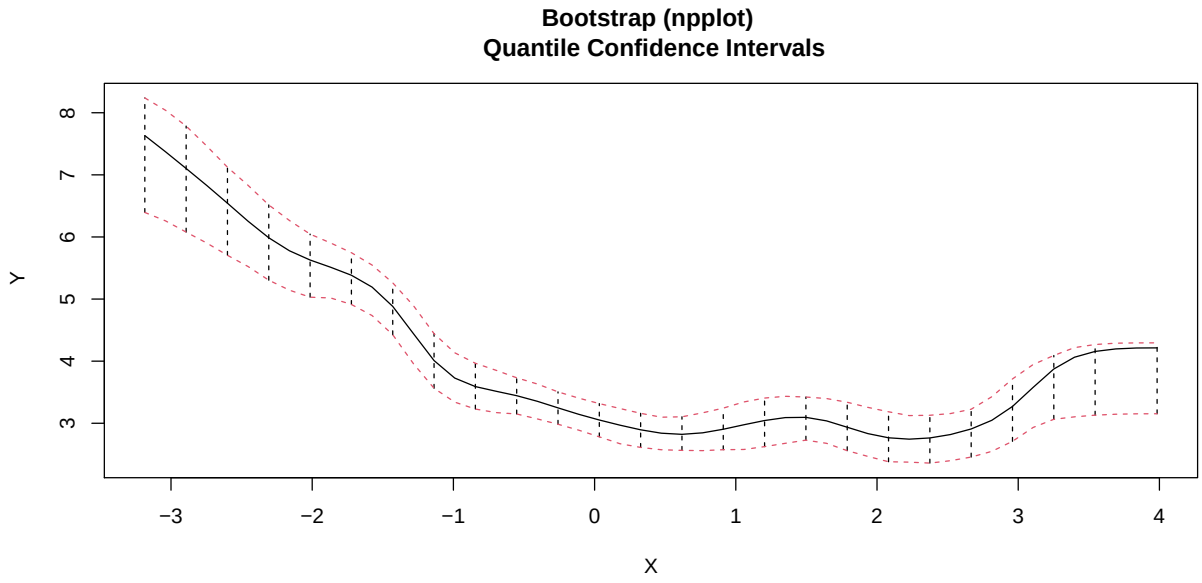

Figure 19: Bootstrap intervals: Default option npplot

Then, we compute 95% confidence intervals for $m(x)$ along `x <- seq(-5, 5, by = 0.1)`. Figure 20 shows the case of the naive bootstrap. In this case, we see how the confidence intervals of the default quantile option of `np::npplot` coincide perfectly with the confidence interval given by `np_pred_CI` with the option `type_bot = naive`. Additionally, the light blue and green lines represents the extension of the predicted values for $m(x)$ in the grid between -5 and 5.

Finally, figures 21 and 22 shows the confidence intervals in case of wild bootstrap with the normal and golden perturbation. In this case, we notice how the confidence interval is in the function of the number of simulated observations. In the case of the tails, especially on the right side, the confidence intervals are wider in relation to the confidence intervals given by the `naive` bootstrap. On the other hand, in the sections of the graph where there is a concentration of simulated observations, the fitted lower and upper confidence intervals are similar for the two bootstrap methodologies.

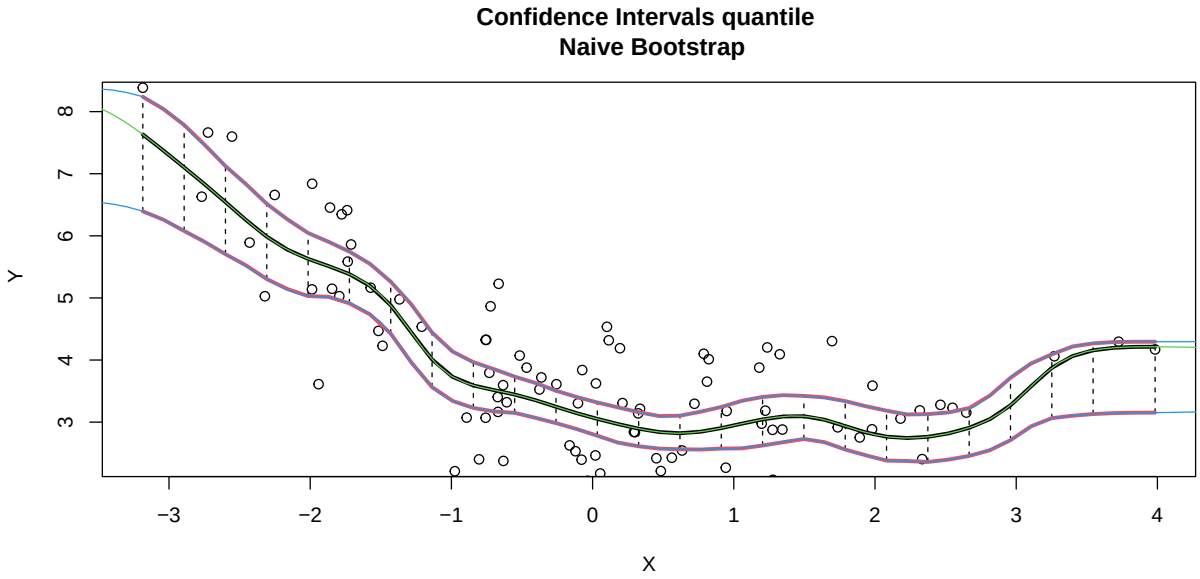

Figure 20: Function np_pred_CI: Naive Bootstrap

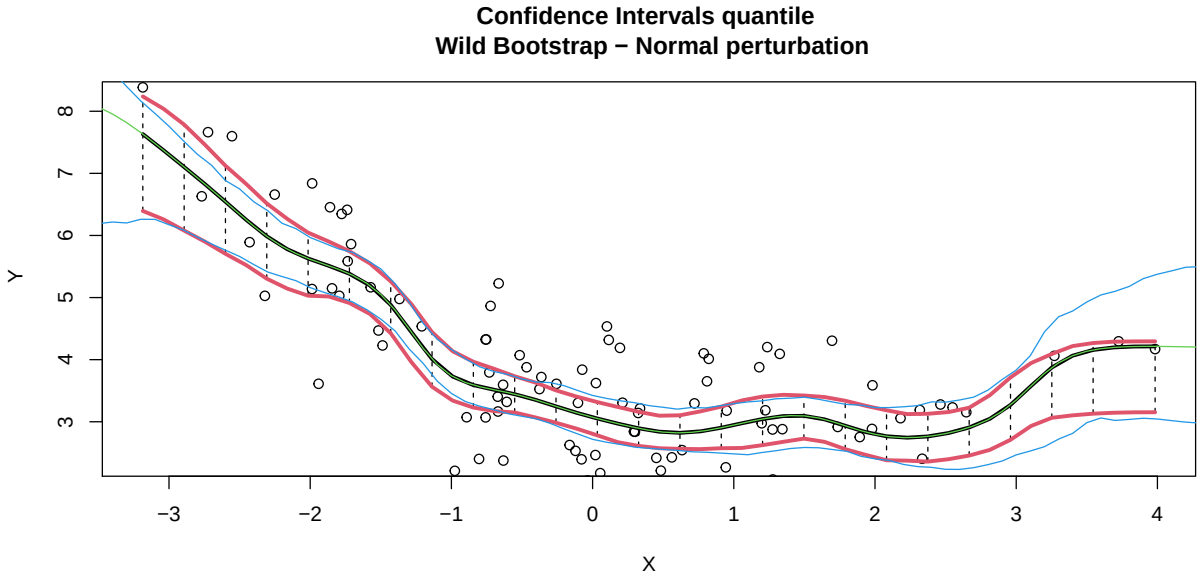

Figure 21: Function np_pred_CI: Wild Bootstrap and normal perturbation

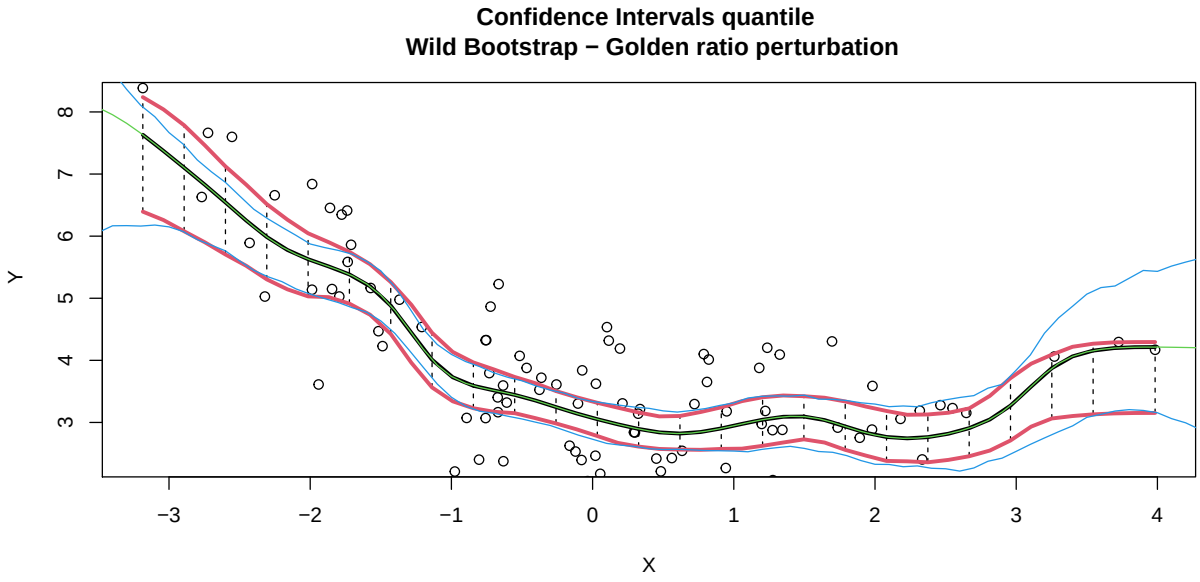

Figure 22: Function np_pred_CI: Wild Bootstrap and golden error perturbation