Contents

1 Abstract

This project is focused on Non - Parametric Regression. I began by studying the basics of Simple and Multiple Linear Regression. In Non-Parametric Regression majorly two problems were dealt with. One where the covariates were from the $L^2[0,1]$ space and responses were in $\mathbb{R}$ and other where the covariates were in the $L^2[0,1]$ space and the responses were also in the $L^2[0,1]$ space. In these problems, I established the estimated value and the estimated polynomial respectively and plotted the graphs accordingly.

2 Introduction

2.1 What is Linear Regression?

Linear regression is a statistical method used to model the relationship between a dependent variable (often denoted as Y) and one or more independent variables (often denoted as $X_1, X_2, \ldots, X_n$). It assumes a linear relationship between the independent variables and the dependent variable. The relationship is represented by the equation of a straight line:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_n X_n + \varepsilon$$

Where:

- Y is the dependent variable (the variable you are trying to predict),
- $X_1, X_2, \ldots, X_n$ are the independent variables (also called predictors or features),
- β_0 is the intercept (the value of Y when all X's are zero),
- $\beta_1, \beta_2, \ldots, \beta_n$ are the coefficients (the weights or slopes) of the independent variables,
- ε represents the error term (the difference between the observed value of the dependent variable and the value predicted by the model).

The goal of linear regression is to estimate the values of the coefficients ($\beta_0, \beta_1, \ldots, \beta_n$) that minimize the difference between the observed values of the dependent variable and the values predicted by the model. This is typically done by minimizing the sum of the squared

differences between the observed and predicted values, a method called Ordinary Least Squares (OLS).

Linear regression is widely used in various fields for tasks such as prediction, forecasting, and understanding the relationship between variables.

2.2 Simple Linear Regression

- Involves only one independent variable.
- The relationship between the independent and dependent variables is assumed to be linear.
- The equation is of the form $Y = \beta_0 + \beta_1 X + \varepsilon$, where Y is the dependent variable, X is the independent variable, β_0 is the intercept, β_1 is the coefficient, and ε is the error term.

2.2.1 Estimation of parameters

In simple linear regression, we aim to find the values of β_0 (intercept) and β_1 (slope) that minimize the difference between the observed values of the dependent variable (Y) and the values predicted by the model.

The formula to calculate the coefficients β_0 and β_1 is derived using the method of least squares. The objective is to minimize the sum of the squared differences between the observed values of Y and the values predicted by the regression line.

Here's the formula for calculating β_1, the slope:

$$\beta_1 = \frac{\sum_{i=1}^{n}(X_i - \bar{X})(Y_i - \bar{Y})}{\sum_{i=1}^{n}(X_i - \bar{X})^2}$$

And for β_0, the intercept:

$$\beta_0 = \bar{Y} - \beta_1 \bar{X}$$

Where:

- n is the number of data points.

- X_i and Y_i are the values of the independent and dependent variables respectively for the i^{th} data point.
- $\bar{X}$ and $\bar{Y}$ are the means of the independent and dependent variables respectively.

These formulas essentially calculate the slope and intercept of the line that best fits the data points in a way that minimizes the sum of squared differences between the observed and predicted values. Once β_0 and β_1 are determined, they can be used to predict the value of Y for any given value of X using the equation $Y = \beta_0 + \beta_1 X$.

2.3 Multiple Linear Regression

- Involves two or more independent variables.
- The relationship between the independent and dependent variables is assumed to be linear.
- The equation is of the form $Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_n X_n + \varepsilon$, where Y is the dependent variable, $X_1, X_2, \ldots, X_n$ are the independent variables, β_0 is the intercept, $\beta_1, \beta_2, \ldots, \beta_n$ are the coefficients, and ε is the error term.

2.3.1 Estimator of β

In multiple linear regression, we have multiple independent variables $X_1, X_2, \ldots, X_p$ (where p is the number of predictors), and we aim to estimate the coefficients $\beta_0, \beta_1, \ldots, \beta_p$ of the linear relationship between these variables and the dependent variable Y. These coefficients are typically represented as a vector $\beta = (\beta_0, \beta_1, \ldots, \beta_p)^T$, and the goal is to find the best-fitting line or hyperplane in p-dimensional space.

Now, let's discuss the estimator of β. In multiple linear regression, the least squares estimator of the coefficient vector β is given by:

$$\hat{\beta} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}$$

Where:

- $\mathbf{X}$ is the matrix consisting of $n \times (p+1)$ elements, where n is the number of observations and each row represents a data point with p predictors plus a constant term (intercept).

- $\mathbf{Y}$ is the $n \times 1$ vector of observed values of the dependent variable.
- $(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T$ is the Moore-Penrose pseudoinverse of $\mathbf{X}$.

2.3.2 Unbiasedness of the Estimator:

An estimator is said to be unbiased if, on average, it gives the true parameter value when applied to multiple samples from the population.

For the multiple linear regression least squares estimator $\hat{\beta}$, we can show that it is unbiased by computing its expected value:

$$\mathrm{E}[\hat{\beta}] = \mathrm{E}[(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}]$$

$$= (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathrm{E}[\mathbf{Y}]$$

Since $\mathrm{E}[\mathbf{Y}] = \mathbf{X}\beta$, where β is the true parameter vector, we have:

$$= (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{X}\beta$$

$$= (\mathbf{X}^T\mathbf{X})^{-1}(\mathbf{X}^T\mathbf{X})\beta$$

$$= \beta$$

Therefore, the expected value of the least squares estimator is the true parameter vector β. Hence, the estimator is unbiased.

This property makes the least squares estimator desirable because it means that, on average, it does not overestimate or underestimate the true parameter values when applied to multiple samples from the population.

2.3.3 Convergences

Convergence in Probability:

- Formally, a sequence of random variables $X_1, X_2, X_3, \ldots$ converges in probability to a constant c if, for any small positive number ϵ, the probability that X_n deviates from c by more than ϵ approaches zero as n approaches infinity.

- Mathematically, we say $X_n \xrightarrow{P} c$ if, for any $\epsilon > 0$, $\lim_{n\to\infty} P(|X_n - c| > \epsilon) = 0$
- The least squares estimator $\hat{\boldsymbol{\beta}}$ in multiple linear regression converges in probability to the true parameter vector $\boldsymbol{\beta}$ as the sample size increases indefinitely.

$$\hat{\boldsymbol{\beta}} \xrightarrow{P} \boldsymbol{\beta}$$

Convergence in Distribution:

- Formally, a sequence of random variables $X_1, X_2, X_3, \ldots$ converges in distribution to a random variable X if the CDFs of X_n converge pointwise to the CDF of X as n approaches infinity.
- Mathematically, we say $X_n \xrightarrow{D} X$ if $\lim_{n\to\infty} F_{X_n}(x) = F_X(x)$ for all x where $F_{X_n}(x)$ and $F_X(x)$ are the CDFs of X_n and X respectively.
- The quantity $\sqrt{n}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta})$ converges in distribution to a normal distribution as the sample size n increases indefinitely, which can be expressed as:

$$\sqrt{n}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}) \xrightarrow{d} \mathcal{N}(0, \Sigma)$$

3 Main Body

3.1 Non-parametric Regression

- Non-parametric regression is a type of regression analysis that makes no assumptions about the functional form of the relationship between the independent and dependent variables. Unlike parametric regression, which assumes a specific functional form (such as linear, quadratic, etc.), non-parametric regression allows for more flexibility in modeling complex relationships without specifying an explicit equation. In non-parametric regression, the relationship between the independent and dependent variables is estimated directly from the data.
- Non-parametric regression is often implemented using kernel regression, which is one of the common techniques.

PROBLEM 1 :

The space $L^2[0,1]$ is defined as:

$L^2[0,1] = \left\{ f : [0,1] \to \mathbb{R} \;\middle|\; \left(\int_0^1 (f(t))^2 \, dt \right)^{\frac{1}{2}} < \infty \right\}$

We define m : L^2[0,1]$\to \mathbb{R}$

We generate samples of errors $\{\epsilon_1, \epsilon_2, \epsilon_3, \ldots\}$ from some standard probability distribution.

MODEL : y_j = m (f_j) + ϵ_j

Mathematically, the estimator $\hat{m_n}(f)$ can be expressed as:

$$\hat{m_n}(f) = \frac{\sum_{i=1}^{n} K\left(\frac{m(f-f_i)}{h_n}\right) y_i}{\sum_{i=1}^{n} K\left(\frac{m(f-f_i)}{h_n}\right)}$$

Where:

$\hat{m_n}(f)$ is the estimated value of the dependent variable at point x.

$f \in$L^2[0,1]

$f_j \in L^2[0,1]$

$K(u)$ is the chosen kernel function.

h_n is the bandwidth parameter.

n is the total number of observations.

EXAMPLE :

Here is an example for method 1:

Here, we define m : $L^2[0,1] \to \mathbb{R}$ by

m(f) = ‖ f ‖ = $\left(\int_0^1 (f(t))^2\, dt\right)^{\frac{1}{2}}$, $\forall\ f \in L^2[0,1]$.

We take f_j(t) also belongs to L^2[0,1] , $\forall t \in [0,1] as$

$f_j(t) = t^j, \quad j = \{1, 2, \ldots, 50\}$

We generate samples of errors $\{\epsilon_1, \epsilon_2, \epsilon_3, \ldots\}$ from standard normal distribution. In this case the model will look like: $y_i = m(f_i) + \epsilon_i = \|f_i\| + \epsilon_i, \quad j = \{1, 2, \ldots, 50\}$ We take f(t) = sin(t) $\forall t \in [0, 1]$

Mathematically, the estimator $\hat{m_n}(f)$ can be expressed as:

$$\hat{m_n}(f) = \frac{\sum_{i=1}^{n} K\left(\frac{||f - f_i||}{h_n}\right) y_i}{\sum_{i=1}^{n} K\left(\frac{||f - f_i||}{h_n}\right)}$$

Where:

$\hat{m_n}(f)$ is the estimated value of the dependent variable at point x.

$K(u)$ is the kernel function like Normal and Epanechnikov .

h_n is the bandwidth parameter with values $n^{-\frac{1}{3}}$ and $n^{-\frac{1}{4}}$.

n is the total number of observations.

OBSERVATIONS:

The value of n was taken to be 1e4. There were two kernels used namely Normal and Epanechnikov. The bandwidth was also adjusted to different values such as $n^{-\frac{1}{3}}$ and $n^{-\frac{1}{4}}$.

Initially the bandwidth chosen was $n^{-\frac{1}{3}}$ and the kernel function chosen was Normal. Convergence was visualized using different $f \in$L^2[0,1] such as $sin(t)$ and $tan(t)$.

A **significant bias** was observed in the convergence of $\hat{m_n}(f)$. To avoid the bias (i.e. reduce it) the bandwidth was changed to $n^{-\frac{1}{4}}$.

The **bias decreased** upon doing this but did not vanish.

The same steps were repeated for the Epanechnikov kernel and similar results were observed for the bias upon using different bandwidth values as mentioned above. Here are some visualisation :

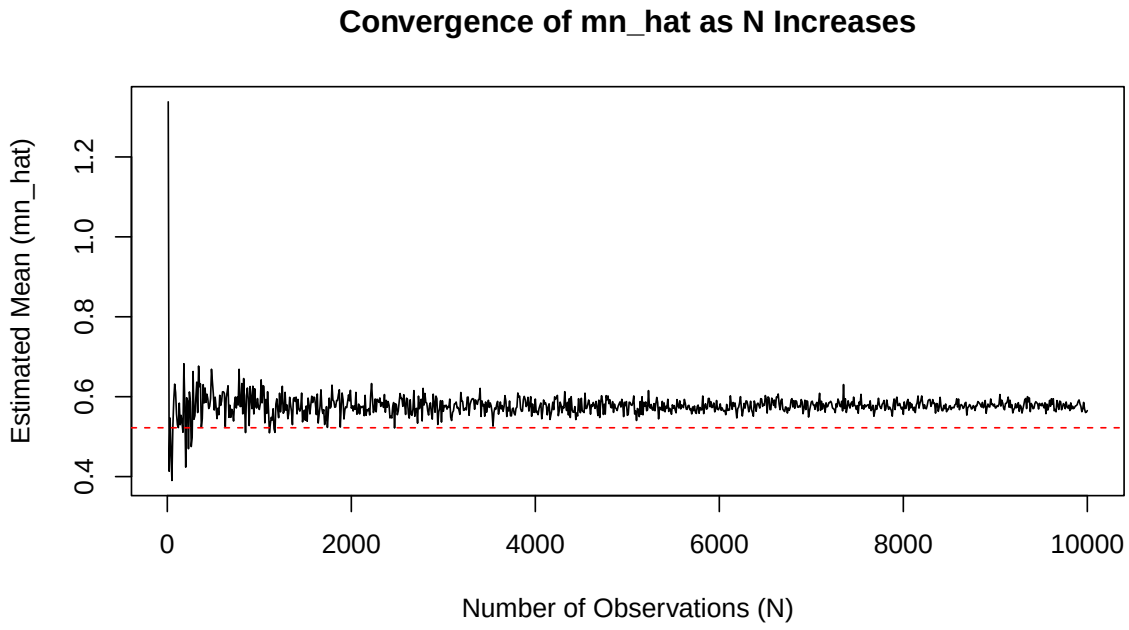

Figure 1. f : sin(t), Normal kernel and bandwidth $n^{-1/3}$

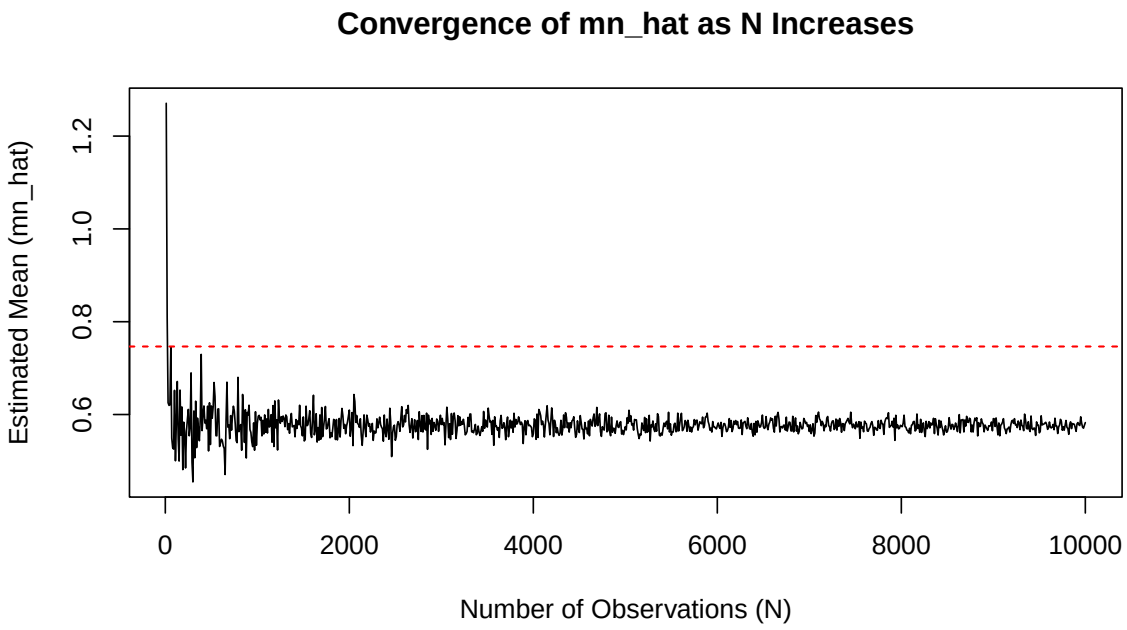

Figure 2. f : tan(t), Normal kernel and bandwidth $n^{-1/3}$

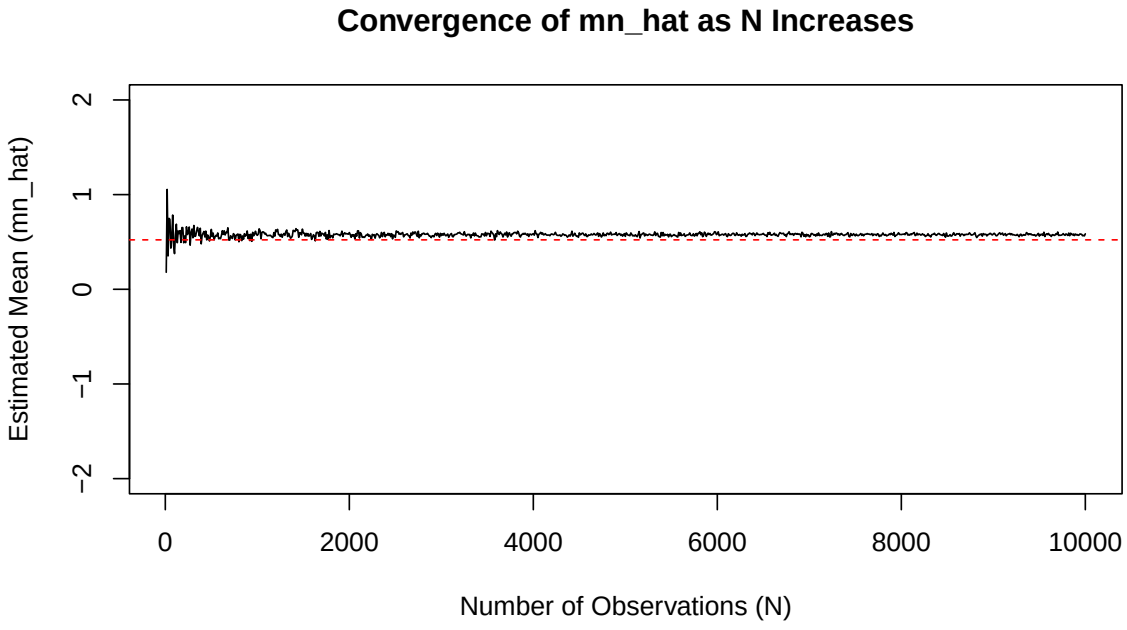

Figure 3. f : sin(t), Normal kernel and bandwidth $n^{-1/4}$

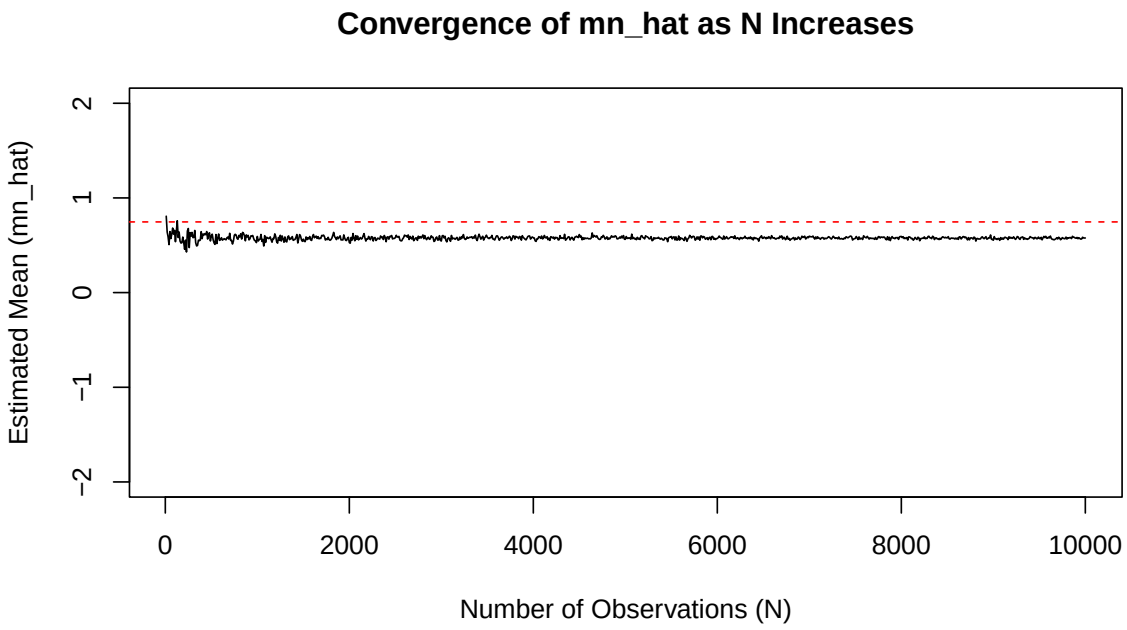

Figure 4. f : tan(t), Normal kernel and bandwidth $n^{-1/4}$

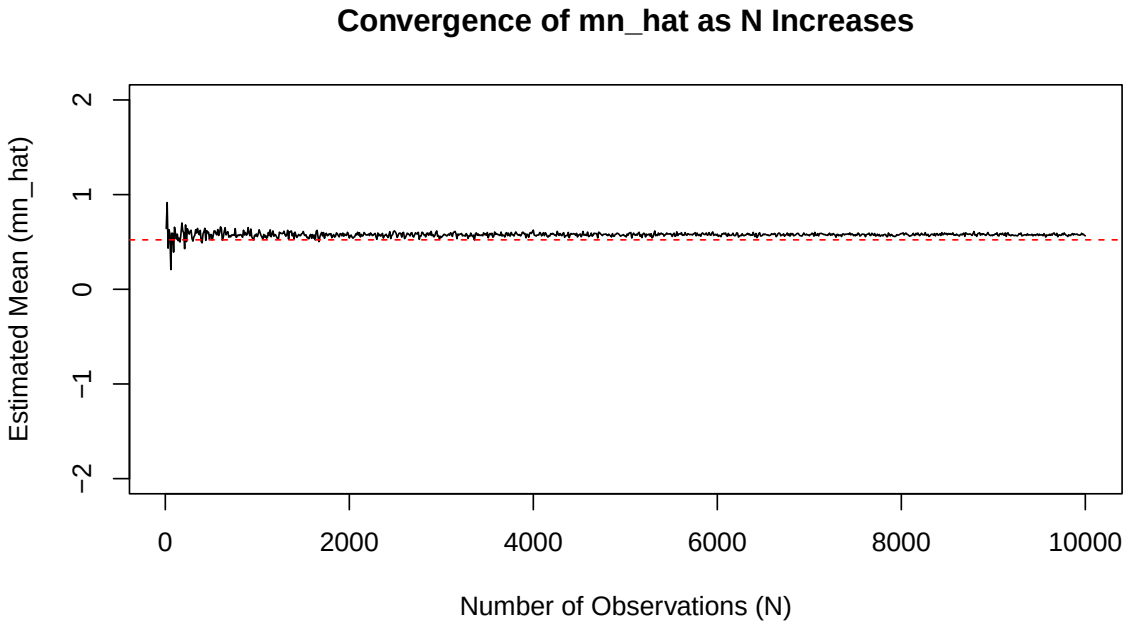

Figure 5. f : sin(t), Epanechnikov kernel and bandwidth $n^{-1/3}$

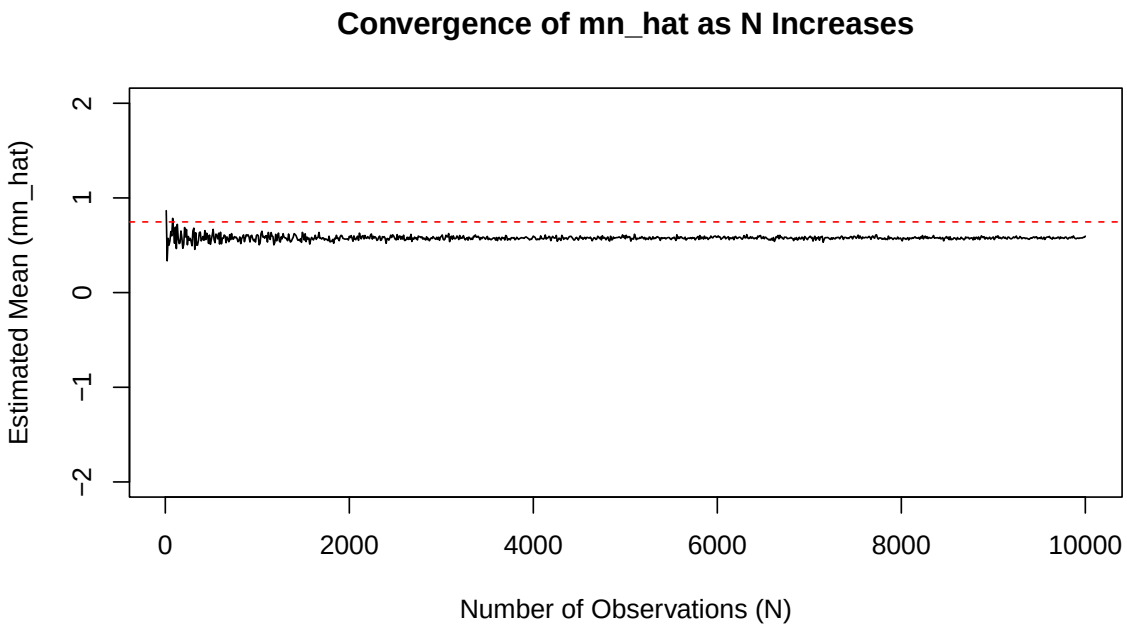

Figure 6. f : tan(t), Epanechnikov kernel and bandwidth $n^{-1/3}$

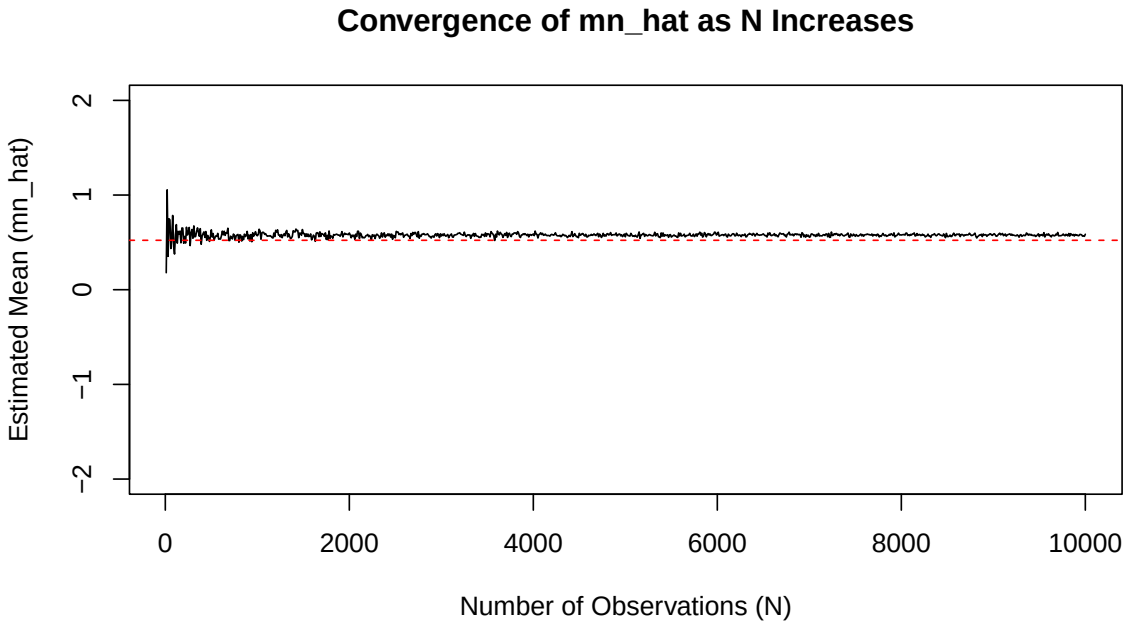

Figure 7. f : sin(t), Epanechnikov kernel and bandwidth $n^{-1/4}$

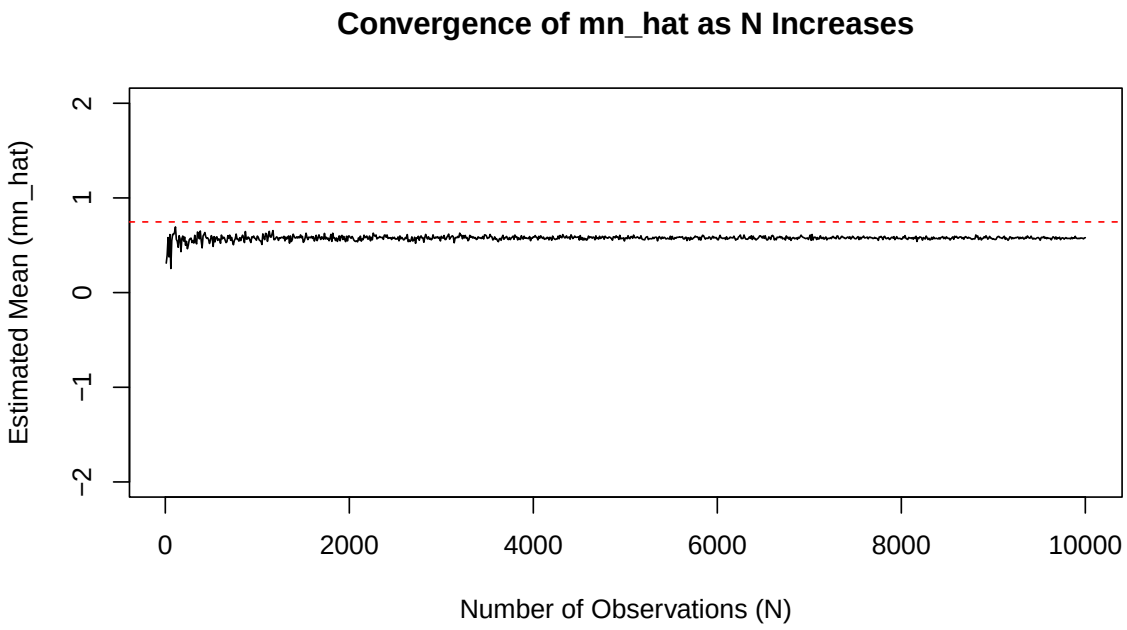

Figure 8. f : tan(t), Epanechnikov kernel and bandwidth $n^{-1/4}$

PROBLEM 2:

We define $m : L^2[0,1] \to L^2[0,1]$

$$L^2[0,1] = \left\{ f : [0,1] \to \mathbb{R} \;\middle|\; \left(\int_0^1 (f(t))^2 \, dt \right)^{\frac{1}{2}} < \infty \right\}$$

{ $e_1, e_2, e_3, \ldots$} is a known orthonormal basis of $L^2[0,1]$.

We fix j such that $1 \leq$ j $\leq$ N, where N is a sufficiently large number.

Define $M_j : L^2[0,1] \to \mathbb{R}$

M_j(f) = <m(f), e_j>

We estimate $\hat{M}_{j,n}$ from data (Y_i, f_i), $i = \{1, 2, 3, \ldots\}$

MODEL : <Y, e_j> = M_j(f) + <ϵ, e_j>

We generate samples of errors $\{\epsilon_1, \epsilon_2, \epsilon_3, \ldots\}$ from some standard probability distribution.

Mathematically, the estimator m(.) can be expressed as:

m(f) = $\sum_{i=1}^{n} \hat{M}_{j,n} e_j$

EXAMPLE:
The function m has the domain as well as range as $L^2[0,1]$.

The orthonormal basis of $L^2[0,1] : \{e_1, e_2, e_3, \ldots\}$, is taken to be the Legendre Polynomials of degree 0 to $N-1$.

We take (m(f))(x) := f(x) sin(x) for all x in [0,1].

We take various choices of f such as polynomials, standard trigonometric functions (sine, cosine etc.)

M is defined by the choice of m.

We fix j such that $1 \leq$ j $\leq$ N, where N is a sufficiently large number.
Define $M_j : L^2[0,1] \to \mathbb{R}$
M_j(f) = <m(f), e_j>
We estimate $\hat{M}_{j,n}$ from data (Y_i, f_i), $i = \{1, 2, 3, \ldots\}$

MODEL : <Y, e_j> = M_j(f) + <ϵ, e_j>
We will use the Nadaraya Watson estimator from Method 1.

$$\hat{M}_{j,n}(f) = \frac{\sum_{i=1}^{n} K\left(\frac{|| < Y_i, e_j > - < f, e_j > ||}{h_n}\right) < Y_i, e_j >}{\sum_{i=1}^{n} K\left(\frac{|| < Y_i, e_j > - < f, e_j > ||}{h_n}\right)}$$

$K(u)$ is the Normal kernel function .

h_n is the bandwidth parameter with value $n^{-\frac{1}{5}}$ Proposed estimator of m(.) is,

m(f) = $\sum_{i=1}^{n} \hat{M}_{j,n} e_j$

OBSERVATIONS:

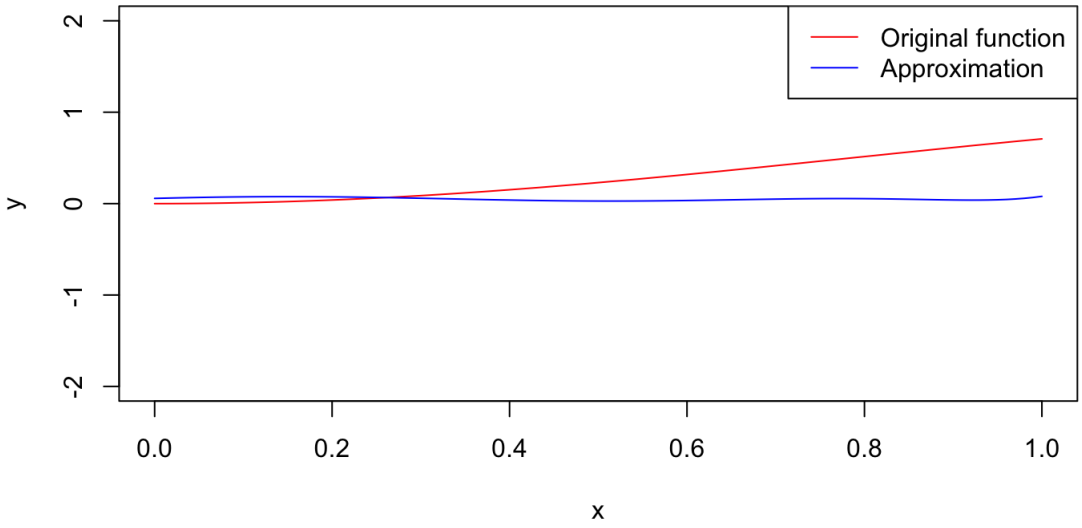

Figure 1. Taking m(f)= f

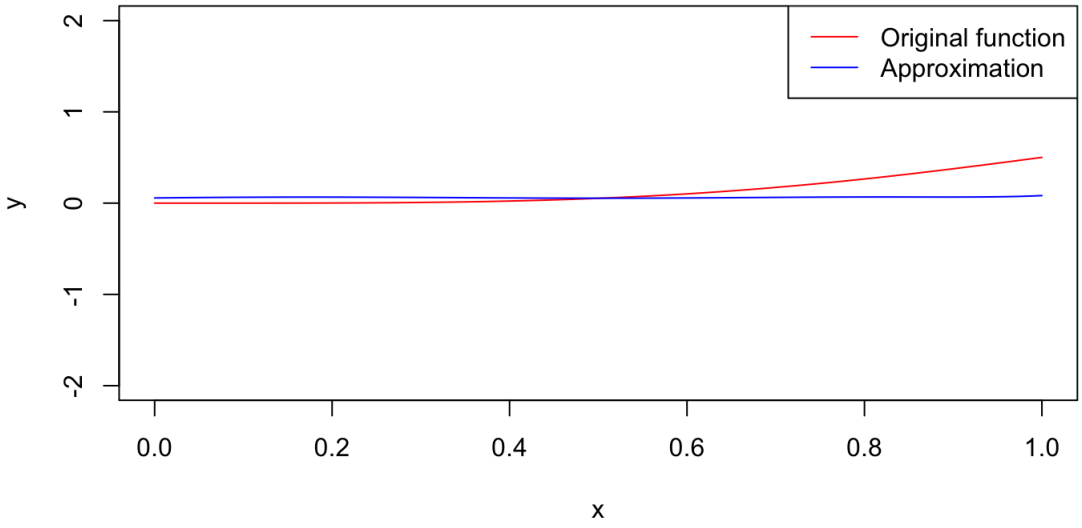

Figure 2. Taking m(f) = f^2

4 Codes

PROBLEM 1:

The code for convergence in non-parametric regression with normal kernel and bandwidth $N^{-1/4}$:

```
f <- function(x)
{
  return((exp(x))^2)
}
convergence <- function(N)
{
  errors <- rnorm(N)
  hn <- N^(-1/4)
  ctr <- 0
  ctr1 <- 0
  fun <- function(x)
  {
    ctr <- ctr+1
    return((x^ctr)^2)
  }
  y_i <- numeric(N)
  for(i in 1:N)
  {
    r <- integrate(fun,0,1)
    y_i[i] <- sqrt(r$value) + errors[i]
  }

  func <- function(x)
  {
    ctr1 <- ctr1 + 1
    return(((exp(x))-(x^ctr1))^2)
  }
  num <- numeric(N)
  denom <- numeric(N)
  for(i in 1:N)
  {
```

```
   result <- integrate(func,0,1)
   num[i] <- dnorm(sqrt(result$value)/hn)*y_i[i]
   denom[i] <- dnorm(sqrt(result$value)/hn)
  }
  mn_hat <- sum(num)/sum(denom)
  return(mn_hat)
}
real <- integrate(f,0,1)
real <- sqrt(real$value)
############################################################

N_values <- seq(10, 10000, by = 10)
# Calculate mn_hat for each N value
mn_hat_values <- sapply(N_values, convergence)

# Plot the convergence behavior
plot(N_values, mn_hat_values, type = "l",
     xlab = "Number of Observations (N)", ylab = "Estimated Mean (mn_hat)",
     main = "Convergence of mn_hat as N Increases", ylim = c(-2,2))
abline(h = real, col = "red", lty = 2)  # True value line
```

The code for convergence in non-parametric regression with Epanechnikov kernel and bandwidth $N^{-1/4}$:

```
f <- function(x)
{
  return((sin(x))^2)
}
convergence <- function(N)
{
  errors <- rnorm(N)
  hn <- N^(-1/4)
  ctr <- 0
  ctr1 <- 0
  fun <- function(x)
  {
    ctr <- ctr+1
    return((x^ctr)^2)
  }
  y_i <- numeric(N)
  for(i in 1:N)
  {
    r <- integrate(fun,0,1)
    y_i[i] <- sqrt(r$value) + errors[i]
  }

  func <- function(x)
  {
    ctr1 <- ctr1 + 1
    return(((sin(x))-(x^ctr1))^2)
  }
  num <- numeric(N)
  denom <- numeric(N)
  for(i in 1:N)
  {
    result <- integrate(func,0,1)
    u <- sqrt(result$value)/hn
    num[i] <- 0.75*(1-u^2)*y_i[i]
```

```
    denom[i] <- 0.75*(1-u^2)
  }
  mn_hat <- sum(num)/sum(denom)
  return(mn_hat)
}
real <- integrate(f,0,1)
real <- sqrt(real$value)
############################################################

N_values <- seq(10, 10000, by = 10)
# Calculate mn_hat for each N value
mn_hat_values <- sapply(N_values, convergence)

# Plot the convergence behavior
plot(N_values, mn_hat_values, type = "l",
     xlab = "Number of Observations (N)", ylab = "Estimated Mean (mn_hat)",
     main = "Convergence of mn_hat as N Increases", ylim = c(-2,2))
abline(h = real, col = "red", lty = 2)  # True value line
```

PROBLEM 2:

The code for convergence in non-parametric regression with m(f) = f

```
library(polynom)
library(orthopolynom)
m_f <- function(x) {
  return(sin(x) * sin(x))
}
convergence <- function(N, x)
{

  onb <- legendre.polynomials(N, normalized = TRUE)
  Mj_f <- numeric(N)
  errors <- rnorm(N)
  e <- numeric(N)
  y <- numeric(N)
  hn <- N^(-1/5)

  for (j in 1:N)
  {
    integrand <- function(x)
    {
      m_f(x) * predict(onb[[j]], x)
    }
    error <- function(x) {
      errors[j] * predict(onb[[j]], x)
    }
    Mj_f[j] <- integrate(integrand, lower = 0, upper = 1)$value
    e[j] <- integrate(error, lower = 0, upper = 1)$value
    y[j] <- Mj_f[j] + e[j]
  }

  Mhat_j <- numeric(N)

  for (j in 1:N) {
    num <- numeric(N)
    denom <- numeric(N)
```

```
    for (i in 1:N) {
      t <- function(x) {
        y[i] * predict(onb[[j]], x^2)
      }
      foo <- integrate(t, lower = 0, upper = 1)$value
      num[i] <- dnorm((foo - Mj_f[j]) / hn) * foo
      denom[i] <- dnorm((foo - Mj_f[j]) / hn)
    }
    Mhat_j[j] <- sum(num) / sum(denom)
  }

  sum_result <- 0

  for (j in 1:N)
  {
    onb[[j]] <- Mhat_j[j] * onb[[j]]
    sum_result <- sum_result + onb[[j]]
  }
  return(as.function(sum_result)(x))
}
#########################################################
x <- seq(0, 1, by = 0.01)
y <- convergence(4, x)
plot(x, m_f(x), type = "l", col = "red", ylim = c(-1, 1), ylab = "y")
lines(x, y, col = "blue")
legend("topright", legend = c("Original function", "Approximation"),
        col = c("red", "blue"), lty = 1)
```

The code for convergence in non-parametric regression with m(f) = f^2

```
library(polynom)
library(orthopolynom)
m_f <- function(x)
{
  return(sin(x)^2*sin(x)^2)
}
convergence <- function(N, x)
{

  onb <- legendre.polynomials(N, normalized = TRUE)
  Mj_f <- numeric(N)
  errors <- rnorm(N)
  e <- numeric(N)
  y <- numeric(N)
  hn <- N^(-1/5)

  for (j in 1:N)
  {
    integrand <- function(x)
    {
      m_f(x) * predict(onb[[j]], x)
    }
    error <- function(x) {
      errors[j] * predict(onb[[j]], x)
    }
    Mj_f[j] <- integrate(integrand, lower = 0, upper = 1)$value
    e[j] <- integrate(error, lower = 0, upper = 1)$value
    y[j] <- Mj_f[j] + e[j]
  }

  Mhat_j <- numeric(N)

  for (j in 1:N) {
    num <- numeric(N)
    denom <- numeric(N)
```

```
    for (i in 1:N) {
      t <- function(x) {
        y[i] * predict(onb[[j]], x^2)
      }
      foo <- integrate(t, lower = 0, upper = 1)$value
      num[i] <- dnorm((foo - Mj_f[j]) / hn) * foo
      denom[i] <- dnorm((foo - Mj_f[j]) / hn)
    }
    Mhat_j[j] <- sum(num) / sum(denom)
  }

  sum_result <- 0

  for (j in 1:N)
  {
    onb[[j]] <- Mhat_j[j] * onb[[j]]
    sum_result <- sum_result + onb[[j]]
  }
  return(as.function(sum_result)(x))
}
############################################################
x <- seq(0, 1, by = 0.01)
y <- convergence(10, x)
plot(x, m_f(x), type = "l", col = "red", ylim = c(-2, 2), ylab = "y")
lines(x, y, col = "blue")
legend("topright", legend = c("Original function", "Approximation"),
        col = c("red", "blue"), lty = 1)
```

References

[1] Linear Regression Analysis" by George Seber and Alan Lee (2012)
https://www.academia.edu/32085934/Linear_Regression_Analyisis_2nd_edition_George_A_F_Seber_Alan_J_Lee_pdf

[2] Hogg, R. V., McKean, J., Craig, A. T. (2005). Introduction to Mathematical Statistics. Pearson Education
https://minerva.it.manchester.ac.uk/~saralees/statbook2.pdf

[3] Convergence in Distribution
https://www.probabilitycourse.com/chapter7/7_2_4_convergence_in_distribution.php

[4] Multiple Linear Regression Model
https://home.iitk.ac.in/~shalab/regression/Chapter3-Regression-MultipleLinearRegressionModel.pdf

[5] Asymptotics of OLS
https://www.bauer.uh.edu/rsusmel/phd/ec1-7.pdf

[6] legendre.polynomials: Orthogonal Legendre Polynomials Basis System
https://www.rdocumentation.org/packages/cSFM/versions/1.1/topics/legendre.polynomials

[7] Legendre Polynomials R documentation
https://search.r-project.org/CRAN/refmans/mpoly/html/legendre.html

[8] Legendre Polynomials and Applications
https://faculty.fiu.edu/~meziani/Note13.pdf

[9] Legendre Polynomials and Functions Outline
http://www.mhtlab.uwaterloo.ca/courses/me755/web_chap5.pdf