Data Science 3 Project: Homework 2

Group 7: Aindrila, Suchandra, Chirag, Paritosh, Vanshika

12 Feb, 2024

1 Question 1

Two Sample Multivariate Data:

Two-sample multivariate data is a dataset that consists of observations from two populations, where each observation is characterized by multiple variables or features. Suppose we conduct a study to compare the characteristics of two different species of plants. Measurements or observations across multiple variables such as height, leaf size, flower color, etc would represent each plant in the dataset. If we collect data on these variables for both species, you would have a two-sample multivariate dataset. Also, in clinical trials, researchers can collect data on various clinical measurements from two different treatment groups. Each participant in the study would have measurements across multiple variables, forming a multivariate dataset.

In summary, two-sample multivariate data involves comparing two groups across multiple variables simultaneously, allowing for a more comprehensive analysis of differences or relationships between the groups.

High Dimensional Data:

High-dimensional data refers to datasets where the number of variables (dimensions) is much larger than the number of observations. This scenario is common in various fields such as genomics, finance, image processing, and text analysis, among others. Analyzing high-dimensional data poses several challenges due to the "curse of dimensionality" which can lead to issues such as overfitting, computational complexity, and difficulties in visualization and interpretation.

Dataset:

In this question, we will be using the brain cancer gene dataset. It is a two-sample multivariate high dimensional data with 80 observations and 54676 genes as the features. The 1^{st} sample of Ependymoma consists of 46 observations and the 2^{nd} sample of Glioblastoma consists of 34 observations. Considering the computational complexity arising when working with such a large number of columns in R, we work with a reduced dataset consisting of 10000 columns and 46 observations.

Kernel Density Estimation:

Kernel density estimation (KDE) is a non-parametric technique used for estimating the probability density function of a continuous random variable. It is particularly useful when the underlying distribution of the data is unknown or difficult to model parametrically.

Suppose you have a set of observations $x_1, x_2, ..., x_n$ from a continuous random variable. KDE involves convolving each data point with a kernel function. The kernel function, often denoted as $K(u)$, is a symmetric probability density function centered at zero. Common choices for kernel functions include the Gaussian (normal) kernel. The bandwidth parameter, often denoted as h, controls the smoothing of the kernel density estimate. A larger bandwidth results in a smoother estimate but may oversmooth the data, while a smaller bandwidth may lead to a more jagged estimate.

The kernel density estimate $\hat{f}(x)$ at a point x is calculated by summing the kernel functions centered at each data point, weighted by the bandwidth:

$$\hat{f}(x) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right)$$

where n is the number of data points, h is the bandwidth, and $K(u)$ is the chosen kernel function.

The kernel density estimate is computed for a range of values, it can be visualized as a smooth curve, representing the estimated probability density function of the underlying data distribution.

Empirical CDF:

The empirical cumulative distribution function (ECDF) is a non-parametric estimator of the cumulative distribution function (CDF) of a random variable based on observed data. It is constructed by plotting the fraction of data points less than or equal to a certain value.

At first, we arrange the observed data points $x_1, x_2, \ldots, x_n$ in ascending order. Then we calculate the fraction of observations for each data point x_i, which is less than or equal to x_i. This fraction is given by:

$$F(x_i) = \frac{\text{number of observations} \leq x_i}{n}$$

It converges to the true cumulative distribution function as the sample size increases. The ECDF is particularly useful for visualizing the distribution of data, comparing different datasets, and assessing goodness-of-fit for theoretical distributions.

Empirical Characteristic Function:

The characteristic function is a concept from probability theory and statistics that uniquely characterizes a probability distribution. It is similar to the moment-generating function but is defined for complex-valued arguments.

Given a random variable X with probability density function (PDF) $f(x)$, the characteristic function $\phi(t)$ of X is defined as:

$$\phi(t) = \mathrm{E}[e^{itX}]$$

where t is a real-valued parameter and i is the imaginary unit. Essentially, the characteristic function is the expected value of the complex exponential function of itX.

The empirical characteristic function (ECF) is a non-parametric estimator of the characteristic function of a random variable based on observed data. It is analogous to the empirical cumulative distribution function (ECDF), providing a way to estimate the characteristic function directly from empirical observations.

Given a sample $x_1, x_2, \ldots, x_n$ of n independent and identically distributed random variables, the empirical characteristic function $\hat{\phi}(t)$ is defined as:

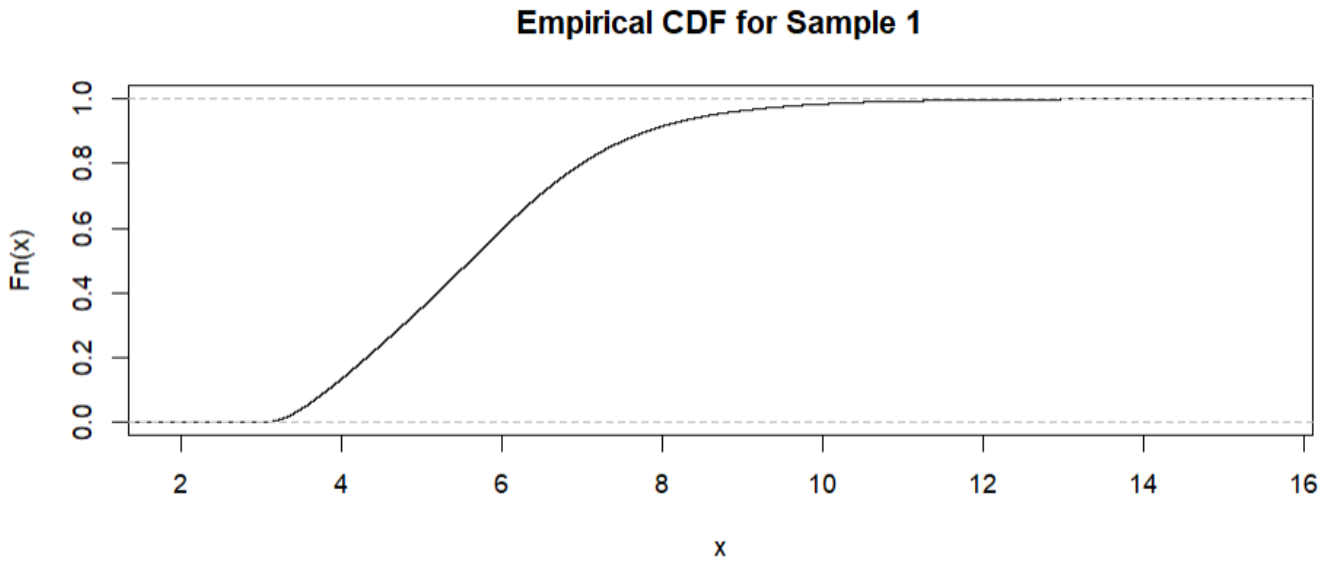
$$\hat{\phi}(t) = \frac{1}{n} \sum_{i=1}^{n} e^{itx_i}$$

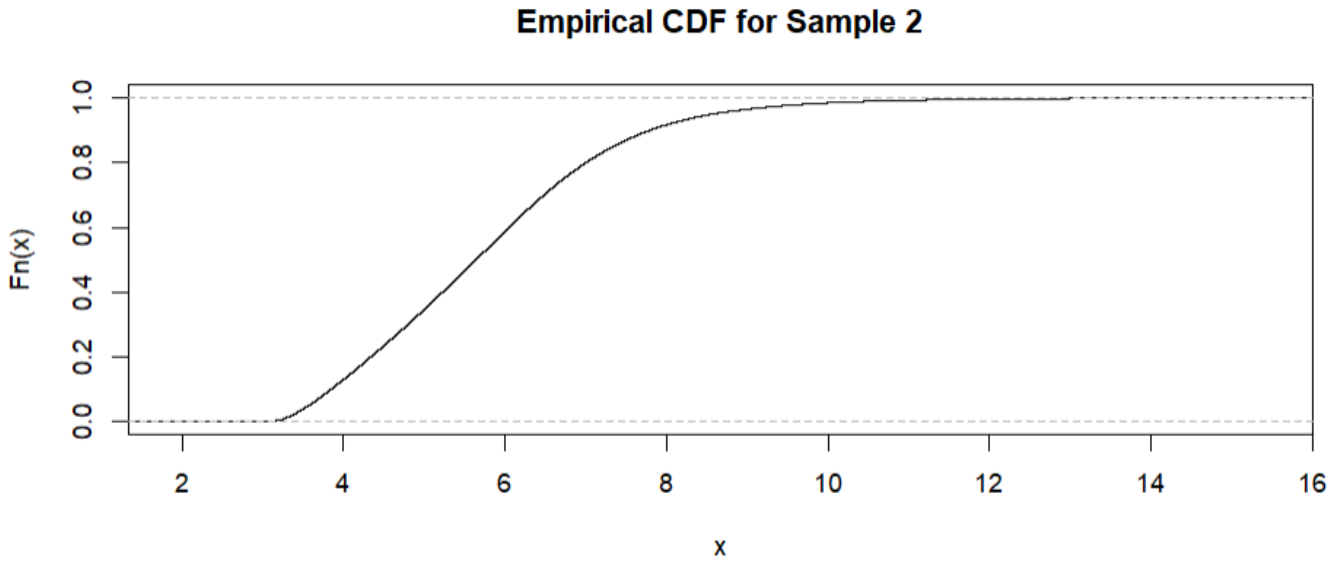
where t is a real-valued parameter and i is the imaginary unit. Overall, the empirical characteristic function provides a flexible and powerful tool for analyzing and estimating characteristics of probability distributions based on observed data.

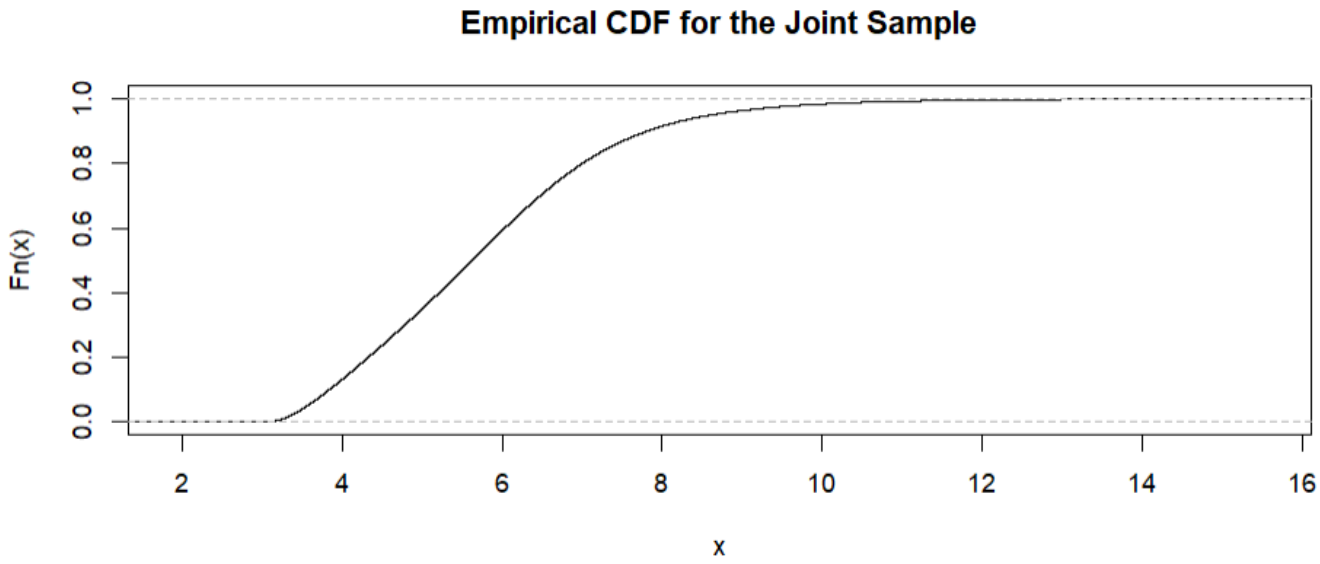
Methodology: We have used both the empirical distribution function and empirical characteristic function to check for the independence of sample 1 and sample 2.

To calculate the empirical distribution function, we have used the "ecdf" from the library "empichar" in R and calculated the "ecdf" for sample 1, sample 2, and the joint sample consisting of both these samples. Two events or distributions are defined as independent if their joint cdf

equal the product of their individual cdf. Hence, we find the product of the empirical distribution function from sample 1 and sample 2 and then find the absolute value of the difference between the empirical distribution function for the joint sample from the above product. If these absolute values for all different choices of "t" are less than a very small quantity, almost negligible, then the distributions can be considered as independent. Taking the tolerance level to be 10^4, we find that all the calculated values are greater than the tolerance value. Hence, the distributions associated with the two samples are not independent.







To calculate the empirical characteristic function in R, we have taken 10^4 vectors of t of length 10^4 in the range {-5,5}. Then, we calculate the "ecf" from the library "empichar" in R. We get 10^8 values of the ecf for the chosen t's. This process is repeated for sample 1, sample 2, and the joint sample consisting of both these samples.

Two events or distributions are defined as independent if their joint characteristic function equal the product of their individual characteristic functions. Hence, we find the product of the empirical characteristic function from sample 1 and sample 2 and then find the absolute value of the difference between the empirical characteristic function for the distribution of the joint sample from the above product. If these absolute values for all different choices of "t" are less than a very small quantity, almost negligible, then the distributions can be considered as independent. Taking the tolerance level to be 10^4, we find that all the calculated values are greater than the tolerance value. Hence, the distributions associated with the two samples are not independent.

All relevant codes are given in the .R file.

2 Question 2

We have used the IRIS dataset. As we all know, this dataset has a total of 5 variables, namely "Sepal Length", "Sepal Width", "Petal Length", "Petal Width", and "Species".

Nonparametric Regression:

Non-parametric regression is a type of regression analysis that does not make explicit assumptions about the functional form of the relationship between the dependent and independent variables. Instead of assuming a specific parametric model (e.g., linear, quadratic), non-parametric regression methods estimate the relationship between variables directly from the data. Non-parametric regression is used for several reasons. Non-parametric regression methods can capture complex and nonlinear relationships between variables without assuming a specific functional form. This flexibility allows them to model a wide range of data patterns effectively. Unlike parametric regression methods, non-parametric regression does not require assumptions about the distribution of the data or the relationship between variables. This makes them more robust when dealing with data that may not meet the assumptions of parametric models. Non-parametric regression methods can handle small datasets or datasets with irregularly spaced observations more effectively than parametric models, which may struggle with such data. Non-parametric regression methods

are often more robust to outliers in the data compared to parametric models, as they do not heavily rely on specific assumptions about the data distribution. Non-parametric regression methods are useful for exploratory data analysis when the underlying relationship between variables is unknown or complex. They can provide insights into the data structure without imposing restrictive assumptions. Non-parametric regression methods, such as kernel regression, can capture local data features, making them suitable for situations where the relationship between variables varies across different regions of the input space.

Nadaraya-Watson Estimator:

$$\hat{m_n}(x_0) = \frac{\sum_{i=1}^{n} y_i k(\frac{x - x_0}{h_n})}{\sum_{i=1}^{n} k(\frac{x - x_0}{h_n})}$$

where h is the bandwidth and k is the kernel density function.

Overall, non-parametric regression methods offer a flexible and robust approach to modeling relationships in data, making them valuable tools in various fields such as statistics, economics, finance, and machine learning.

Kernel Density Estimation:

One common non-parametric regression method is kernel regression, which estimates the conditional expectation of the dependent variable given the independent variable(s) using a weighted average of nearby data points. A kernel function typically determines the weights.

The basic equation for kernel regression can be written as:

$$\hat{f}(x_0) = \frac{\sum_{i=1}^{n} K_h(x_0 - x_i) y_i}{\sum_{i=1}^{n} K_h(x_0 - x_i)}$$

where, $\hat{f}(x_0)$ is the estimated value of the dependent variable at the point x_0, x_0 is the value of the independent variable at which the estimation is being made, x_i are the observed values of the independent variable, y_i are the observed values of the dependent variable corresponding to x_i, K_h is the kernel function with bandwidth h, which determines the weight of each observation in the estimation. The bandwidth controls the smoothness of the estimated curve. The bandwidth parameter determines how much influence nearby data points have on the estimation. A larger bandwidth leads to smoother estimates but may over smooth the data, while a smaller bandwidth captures more detail but may result in higher variability. In summary, non-parametric regression methods like kernel regression allow for flexible modeling of the relationship between variables without assuming a specific functional form, making them suitable for situations where the relationship is complex or unknown.

Local Polynomial Mean Estimation:

Local polynomial mean (LPM) regression is a non-parametric regression method that estimates the regression function by fitting a polynomial model to local subsets of the data. Here's how to estimate the regression function using local polynomial mean regression. Initially, we decide on the order of the polynomial to be used for fitting the local regression. The choice of polynomial order depends on the complexity of the underlying relationship between the variables. Similar to other non-parametric regression methods, LPM regression requires selecting a bandwidth parameter that determines the size of the local neighborhood around each point. A larger bandwidth results in a smoother estimate, while a smaller bandwidth captures more local variation. LPM regression also uses a kernel function to assign weights to the data points within each local neighborhood. For each data point x_i, we construct a local subset of the data by selecting neighboring points within the bandwidth around x_i. Then, we fit a polynomial regression model to the local subset of data using weighted least squares. The kernel function determines the weights and reflects the influence of each

data point on the local estimate. At each data point x_i, compute the estimated regression value by evaluating the polynomial model fitted to the local subset of data. Iterate over all data points in the dataset, fitting a local polynomial regression model and estimating the regression function at each point.

Local Constant Mean Estimator:

$$\hat{m_n}(x_0) = \arg\min_{\theta \in \mathrm{R}} \frac{1}{n} \sum_{i=1}^{n} (y_i - \theta)^2 k(\frac{x - x_0}{h_n})$$

Local Polynomial Median Estimation:

Estimating the regression function using local polynomial median (LPMed) regression follows a similar process to local polynomial mean (LPM) regression, with the key difference being the use of the median instead of the mean within each local neighborhood. Here's how to estimate the regression function using local polynomial median regression. Initially, we decide on the order of the polynomial to be used for fitting the local regression. This decision depends on the complexity of the relationship between the variables and the desired level of flexibility. Determine the bandwidth parameter, which controls the size of the local neighborhood around each data point. A larger bandwidth results in smoother estimates, while a smaller bandwidth captures more local variation. Choose a kernel function to assign weights to the data points within each local neighborhood. For each data point x_i, create a local subset of the data by selecting neighboring points within the bandwidth around x_i. Fit a polynomial regression model to the local subset of data using weighted median regression. The kernel function determines the weights and reflect the influence of each data point on the local estimate. At each data point x_i, compute the estimated regression value by evaluating the polynomial model fitted to the local subset of data using the median value. Iterate over all data points in the dataset, fitting a local polynomial regression model using the median and estimating the regression function at each point.

Local Constant Median Estimator:

$$\tilde{m_n}(x_0) = \arg\min_{\theta \in \mathrm{R}} \frac{1}{n} \sum_{i=1}^{n} |y_i - \theta| k(\frac{x - x_0}{h_n})$$

Comparison: To compare the performance of estimators of the regression function using local polynomial mean (LPM) and median approaches, you can consider several evaluation metrics and techniques:

- Bias: Compute the bias of each estimator, which measures the difference between the estimated regression function and the true regression function over many repetitions of the estimation process. Lower bias indicates better performance.
- Variance: Calculate the variance of each estimator, which measures the variability of the estimated regression function across different samples or datasets. Lower variance indicates better stability.
- Mean squared error (MSE): Compute the MSE of each estimator, which combines the bias and variance into a single metric. MSE is the average of the squared differences between the estimated regression function and the true regression function. Lower MSE indicates better overall performance.
- Median absolute deviation (MAD): Calculate the MAD of each estimator, which measures the median absolute difference between the estimated regression function and the true regression function. MAD is less sensitive to outliers compared to MSE.

By comparing these aspects of the performance of the estimators using local polynomial mean and median approaches, you can gain insights into their strengths and weaknesses and make informed decisions about which approach to use for your specific dataset and research question.

Methodology: We fit a multiple regression equation with "Sepal Length" as the predictor and "Sepal Width" as the predictor. The red line is the fitted regression line.

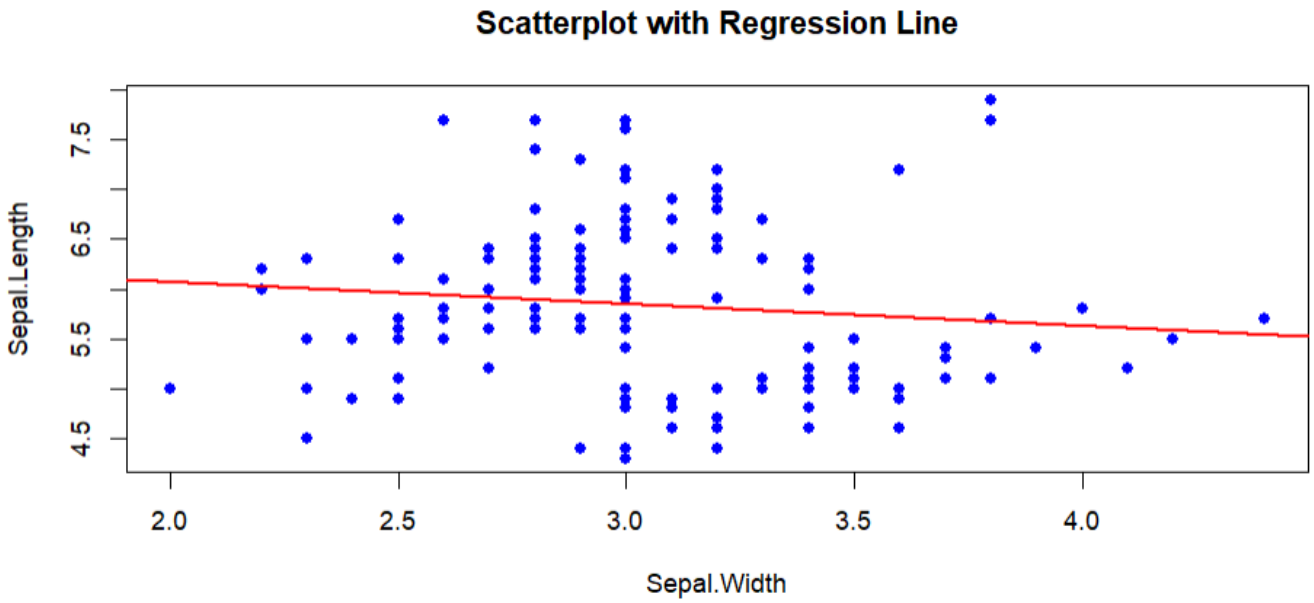


Figure 1

To calculate the local polynomial mean estimator for the regression function, the following equation has to be optimized.

$$\hat{m}_n^{(1)}(x_0), \ldots, \hat{m}_n^{(p)}(x_0) =$$

$$\arg\min_{\theta_0,\theta_1,\ldots,\theta_p \in \mathrm{R}^{p+1}} [\tfrac{1}{n} \textstyle\sum_{i=1}^{n} \{y_i - \theta_0 - \theta_1(x_i - x_0) - \frac{\theta_2}{2!}(x_i - x_0)^2 - \ldots - \frac{\theta_p}{p!}(x_i - x_0)^p\}^2 k(\frac{x - x_0}{h_n})] \quad (1)$$

Here, we only need to estimate the 1st and 2nd derivatives of the regression function. Hence, we use the following equation :

$$\hat{m}_n^{(1)}(x_0), \ldots, \hat{m}_n^{(p)}(x_0) = \arg\min_{\theta_0,\theta_1,\ldots,\theta_p \in \mathrm{R}^{p+1}} [\frac{1}{n} \sum_{i=1}^{n} \{y_i - \theta_0 - \theta_1(x_i - x_0) - \frac{\theta_2}{2!}(x_i - x_0)^2 k(\frac{x - x_0}{h_n})] \quad (2)$$

with the help of "optim" function in R.

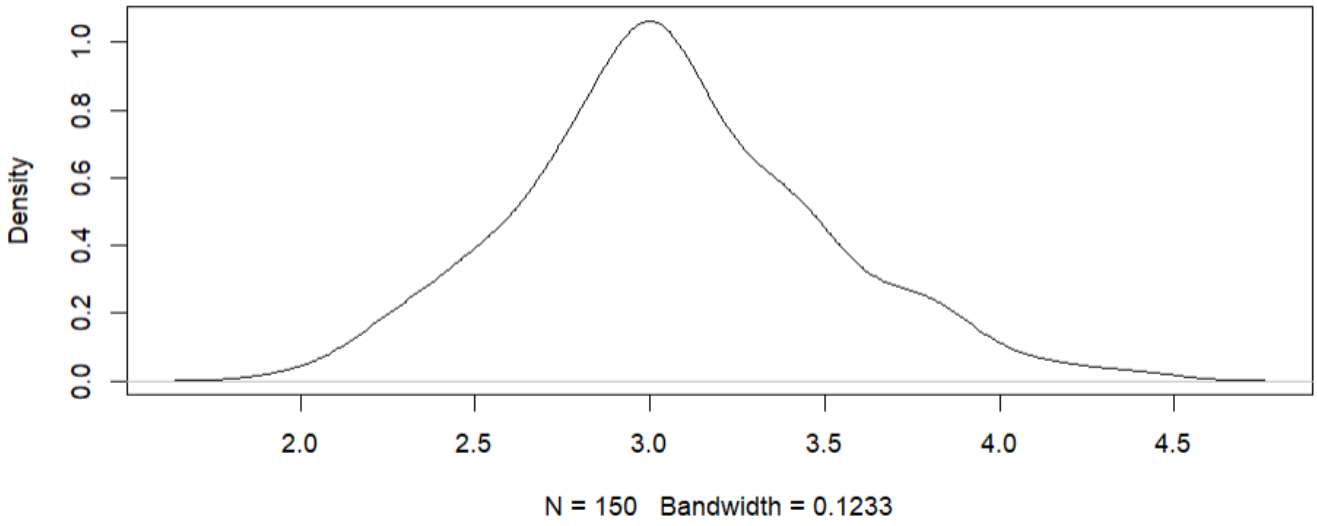


Figure 2

This fig:2 looks like a curve of normal distribution. Hence, we consider the Gaussian kernel for this problem.

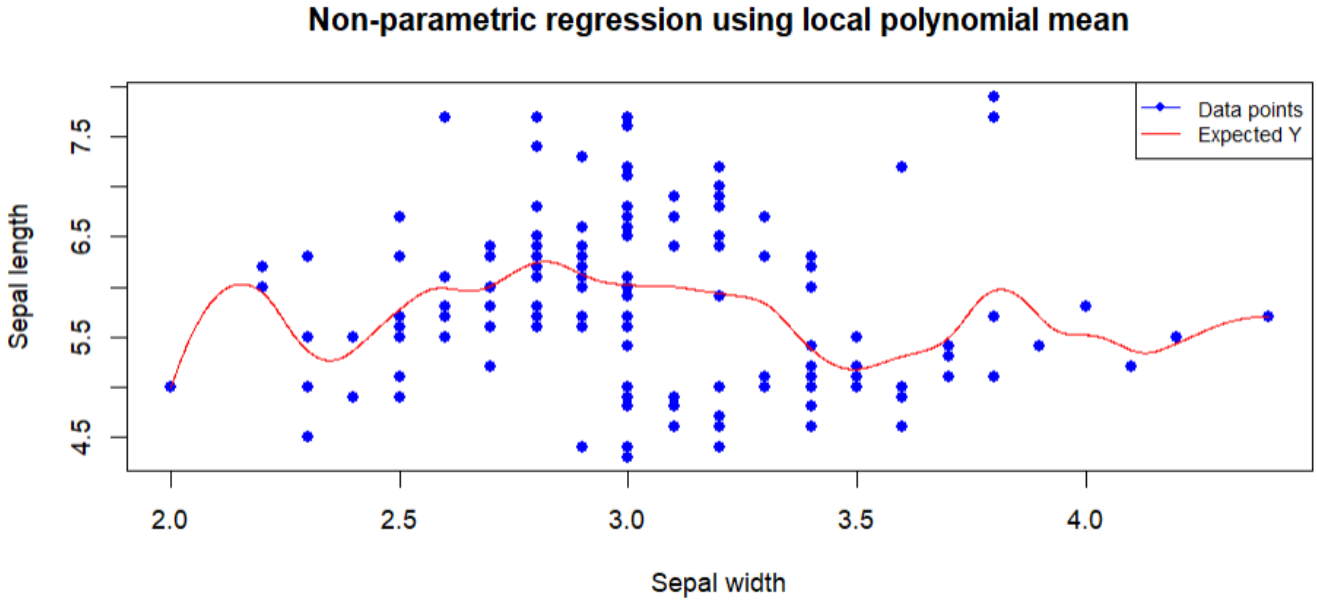


Figure 3

We estimate the regression function using the local polynomial mean. We estimate Y for x taking values in the range of sepal width data (from 2.0 to 4.4) at equal intervals (of 0.001), the red line is the line joining all such estimates and the blue dots are 150 data points (Sepal length vs Sepal width) from the iris dataset.

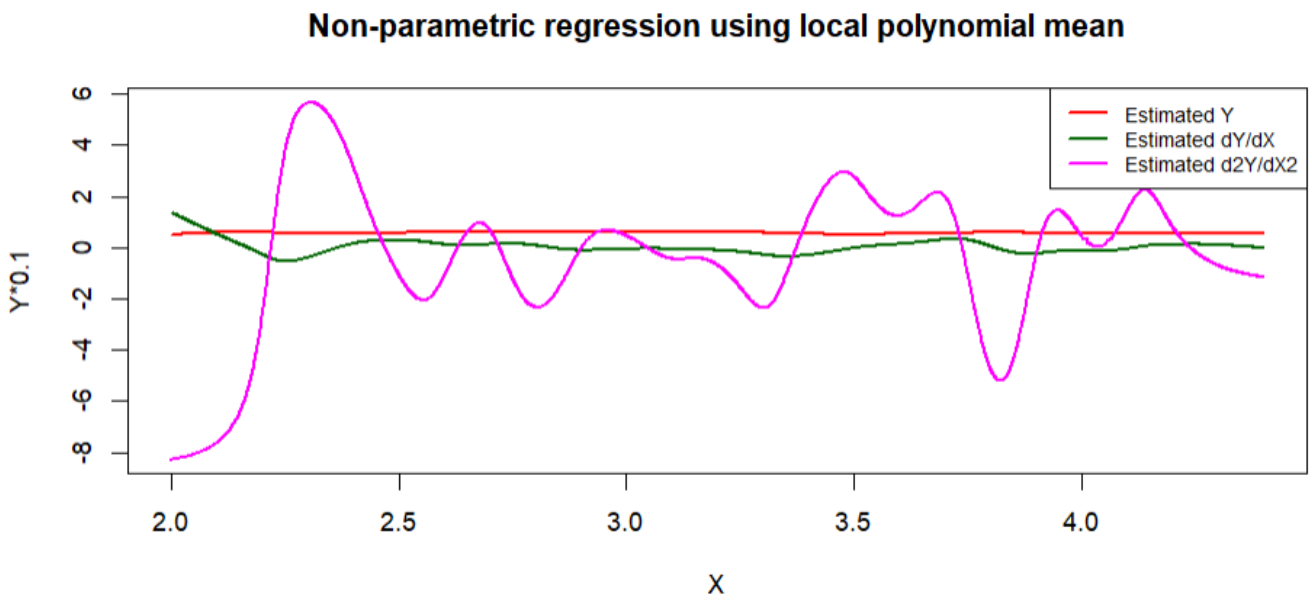


Figure 4

In fig:4 we have estimated the response value y shown in the red curve, its first derivative represents the green line and the second derivative represents the pink line using the local polynomial mean approach of non-parametric regression.

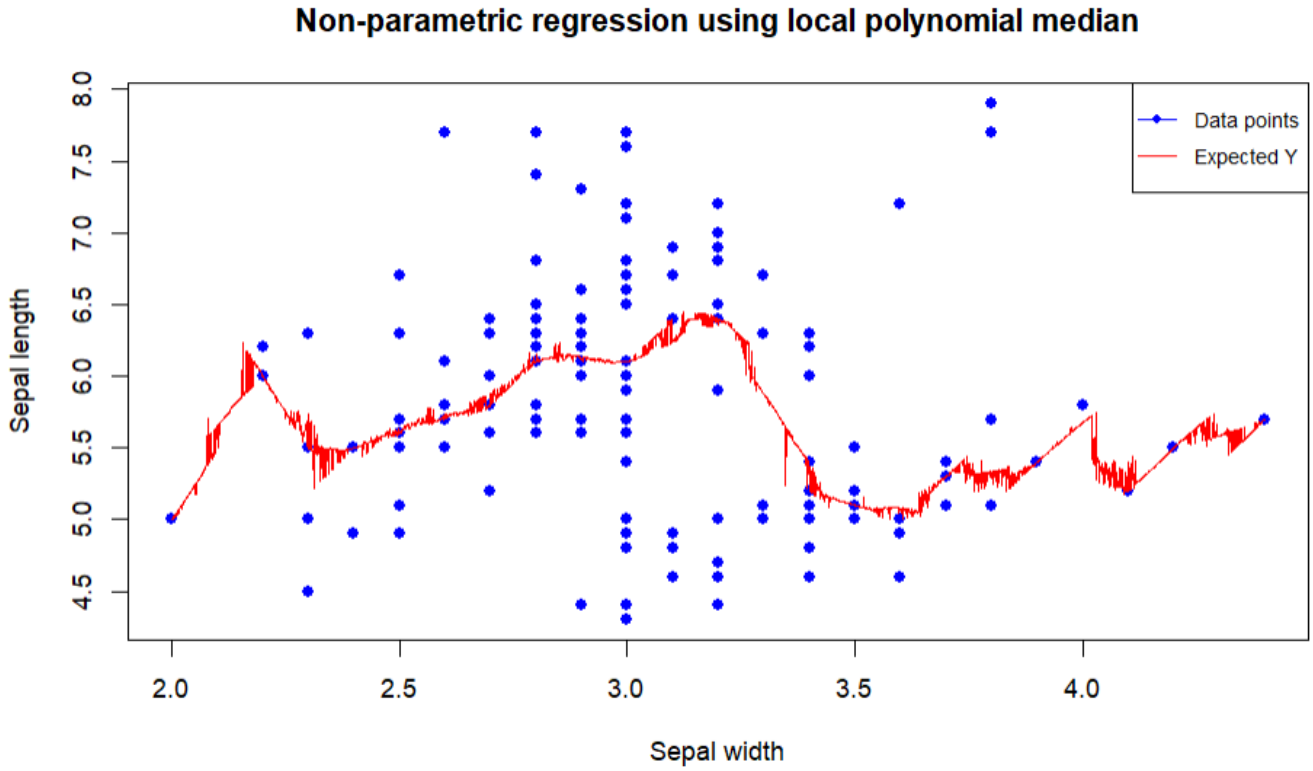


Figure 5

We estimate the regression function using the local polynomial median. We estimate Y for x taking values in the range of sepal width data (from 2.0 to 4.4) at equal intervals (of 0.001), the

red line is the line joining all such estimates and the blue dots are 150 data points (Sepal length vs Sepal width) from the iris dataset.

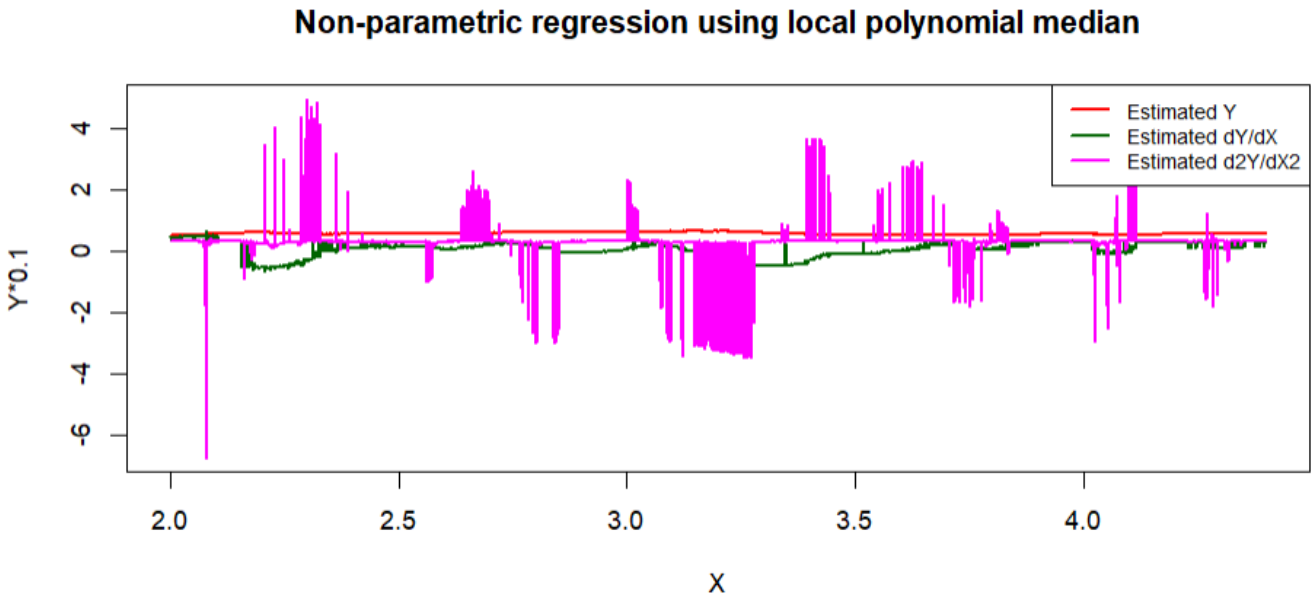


Figure 6

In fig:4 we have estimated the response value y shown in the red curve, its first derivative represents the green line and the second derivative represents the pink line using the local polynomial median approach of non-parametric regression.

Finally, We have calculated MSE for both the mean and median approaches. The values are 41.58483 and 38.97758 respectively. Hence, we can conclude that the local polynomial median approach has better performance than the local polynomial median approach in terms of MSE.