ISYE 7406: Homework # 4 Report

Introduction:

The objective of this analysis is to compare the performance of three different local smoothing techniques, LOESS, Nadaraya-Watson Kernel Smoothing, and Spline Smoothing. My goal is to understand how these methods perform under different conditions and to identify their strengths and limitations. I applied these smoothing techniques to noisy simulated data using an additive noise model, and estimated the Mexican Hat function under two different data Structures; Equidistant data (evenly spaced points) and Non-equidistant data (unevenly spaced points).

Smoothing techniques help remove noise in data, making it easier to see underlying patterns. Some data relationships may be nonlinear or complex, and noise can hide important trends. Local smoothing methods help reveal these trends.I compared the three techniques by looking at bias, variance, and mean squared error (MSE) to see which performed best. Since there is always a tradeoff between bias and variance, I aim to find the method that provides the best balance.

In this analysis, I used the following three smoothing methods: Locally Estimated Scatterplot Smoothing (LOESS), Nadaraya-Watson (NW) kernel smoothing, and Spline Smoothing. LOESS fits small local regression models instead of a single large function for all data. It is flexible but can be computationally demanding and struggles with high dimensional data. NW uses a weighted average of nearby points to estimate values. The kernel function determines how much weight is given to each neighboring point. Different kernel choices produce different smoothing effects. NW is computationally efficient and works well in high dimensions. Spline Smoothing uses piecewise polynomials to fit data, with smoothing controlled by the number and placement of knots. It handles complex relationships well but is computationally expensive.

Methodology:

To compare these three smoothing techniques, I applied them to simulated noisy data and estimated the Mexican Hat function. I tested two different data structures, equidistant and non-equidistant designs.

Equidistant Design:
For the equidistant setup, I generated 101 equally spaced data points between -2π and 2π. Monte Carlo cross-validation was used with 1,000 iterations, where each iteration generated a new simulated dataset. The following smoothing parameters were used:

- LOESS with span = 0.75.
- NW kernel smoothing with Gaussian kernel and bandwidth = 0.2
- Spline smoothing with the default tuning parameter (smooth.spline is default).

After applying each method bias, variance, and MSE were calculated to evaluate and compare performance.

Non-Equidistant Design:
For the non-equidistant setup, I used the provided CSV file containing 101 unevenly spaced data points. The same Monte Carlo cross-validation process was applied, but with different model parameters. The model parameters given for each smoothing technique is provided below:

- LOESS with span = 0.3365
- NW kernel smoothing with bandwidth = 0.2
- Spline smoothing with spar=0.7163 (instead of default)

After running the models, predictions were stored and MSE, variance, and bias were calculated and visualized.

Results:

Equidistant Data:

In the equidistant design, Loess had the highest bias, indicating it over-smoothed the data. NW and Spline had similar performance, but Spline had the lowest bias. For variance, spline had the highest variance and Loess had the lowest variance but at the cost of high bias. NW showed moderate variance. For MSE, Loess had the highest MSE and performed the worst. NW and spline had consistently low MSE, with spline having the lowest overall.

In the Means of Smoothers vs Original Function plot, spline and NW followed the true function well, while Loess deviated significantly, smoothing out too much detail.

The optimal smoothing technique for the equidistant data, is Spline smoothing. It had the lowest bias and the lowest MSE. Spline also best followed the true function. It does have higher variance, but this is a tradeoff for better accuracy.

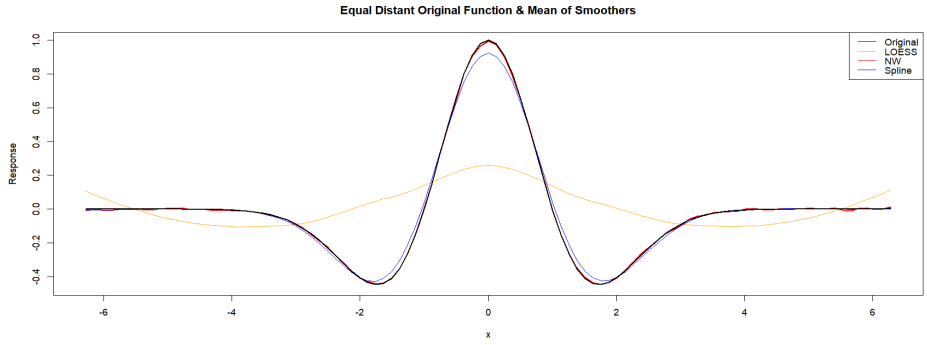

Figure 1. Equidistant design: Mean of smoothers vs. Original function comparison. This plot compares the original function (black line) to the three smoothing methods: Spline smoothing (blue) , NW kernel smoothing (red), and LOESS (orange).

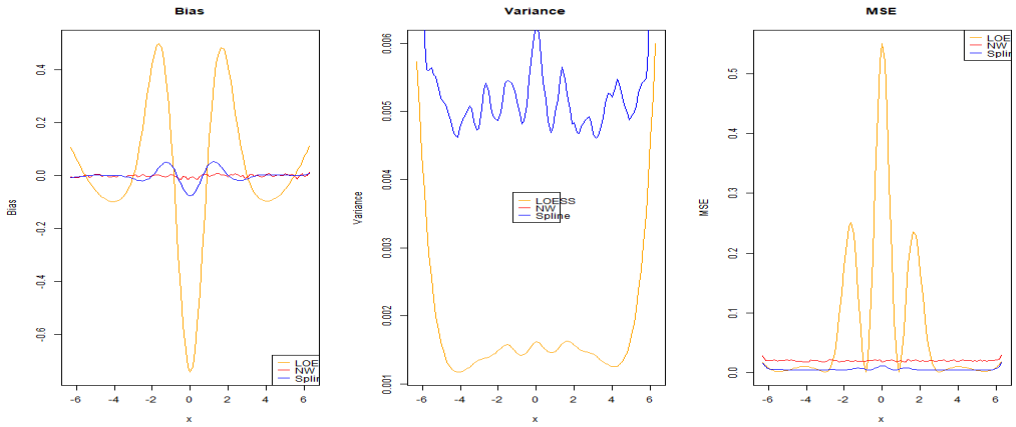

Figure 2. Equidistant design: Above is the Bias, Variance, & MSE comparison. This plot shows the bias, variance (how much each smoother fluctuates), and MSE (how accurate) for each smoother.

Non-Equidistant Data:
In the non-equidistant design, the same three smoothing techniques were applied. Loess again showed high bias , under-smoothing important peaks. NW had relatively constant bias for the most part. NW had the highest variance, making it less stable. Spline and Loess had similar lower variance, with spline having the lowest. For MSE, Loess and spline consistently maintained low MSE except for a peak near x=0. NW had a slightly higher MSE but remained more stable and low throughout.

In the Mean of smoothers vs original function plot, NW most closely followed the true function, confirming that it works well for unevenly spaced data.

The optimal smoothing technique for non-equidistant data is NW Kernel Smoothing. It balanced bias and variance the best and it also followed the true function very closely. Spline also followed the true function closely, but was less stable in sparse areas.

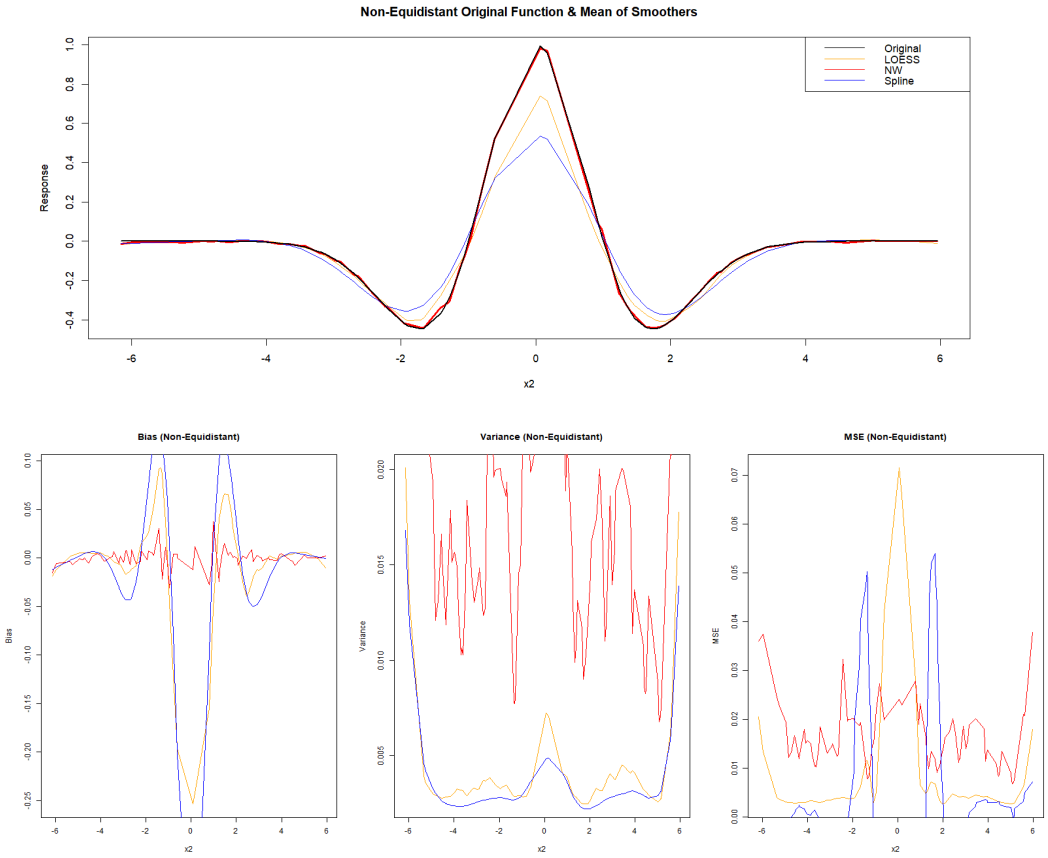

Figure 3. Non-Equidistant design: Bias, Variance, & MSE comparison.

Conclusion:

This analysis shows that different smoothing techniques work better for different data structures. For evenly spaced data, spline smoothing is the best choice. It minimizes overall error, has low bias, and follows the true function well. Its higher variance is a tradeoff for better accuracy. For unevenly spaced data, NW Kernel

smoothing performs the best. It balances bias and variance well and adapts effectively when data points are unevenly spaced.

Parameter tuning plays a big role in how well these models work. The tuning parameters were provided for this analysis, but in real world applications the optimal parameters will not be given and finding the best parameters is necessary. This often requires trial and error or methods like cross-validation, which can be computationally expensive for selecting optimal hyperparameters. Choosing the wrong parameters can lead to under-smoothing or over-smoothing, leading to overall poor accuracy.

There were also statistical challenges. The empirical bias, variance, and MSE were high near $x = 0$, meaning it was difficult to estimate the function in this region and the smoothers struggle here. This suggests that the Mexican hat function is harder to predict in areas with fewer data points or sharp changes. The bias-variance tradeoff was also observed, as one improves the other declines. Methods with lower bias tended to have higher variance and vice versa. For example, Spline smoothing had low bias but high variance, while Loess had high bias but low variance. Finding the right balance between bias and variance is key to getting the best model.

The findings highlight the importance of selecting the right smoothing technique based on the data structure, and how much computational effort is required for tuning parameters.