Environmental Conditions and Their Impact on Maritime Modeling

Author:
Muhammad Sajjad

Supervisor:
Dr. Christopher Engström
Examiner:
Linus Carlsson

Mälardalens universitet

Division Of Mathematics and Applied Mathematics

Acknowledgements

Thank you to my supervisor, Dr.Christopher Engström, for introducing me to this topic and for all your help throughout the work on this thesis. This would not have been possible without you! Another big thank you to my family for all your support.

Contents

Abstract

Estimating fuel efficiency is critical for optimizing the performance of large sea vessels, but comparing the performance of different systems in real-world scenarios is challenging due to environmental factors, variations in input data, and associated errors. This study aims to evaluate the performance of two models, polynomial and multiplicative, used for predicting fuel efficiency in ships under varying conditions and noise types using simulated data. The polynomial model uses a second-order polynomial function to describe the relationship between the input variable and the output variable, while the multiplicative model applies a multiplicative function to the predicted values of the polynomial model. The performance of the models is evaluated based on metrics such as R-squared, root-mean-square error (RMSE), and residuals.

The research investigates the impact of different error sources, such as measurement errors, time delays, outliers, and temporary error sources, on the performance of the models. By analyzing the performance of the models under varying conditions and noise types, this study provides valuable insights into the robustness and applicability of the polynomial and multiplicative models for estimating fuel efficiency in ships. The findings can be used to inform decision-makers in the maritime industry and contribute to the development of more accurate and reliable models for fuel efficiency prediction.

Project Description

In the maritime industry, optimizing fuel efficiency is crucial for reducing operational costs and minimizing environmental impact. Accurate estimation and comparison of ship fuel efficiency are essential for informed decision-making. This project aims to analyze the performance of two distinct methods, polynomial and multiplicative models, for estimating ship fuel efficiency under varying conditions and error sources.

The project will involve generating simulated data with known models and evaluating the effectiveness of the polynomial and multiplicative models in the presence of different types of errors, such as measurement errors, time delays in model efficiency, outliers, and temporary error sources (such as acceleration). By considering these error sources and variations, the study will provide insights into the robustness and accuracy of both models.

Through statistical analysis, this project will help identify the model's strengths and weaknesses and their suitability for different scenarios. Ultimately, the findings from this study will provide valuable information for decision-makers in the maritime industry to optimize fuel efficiency and make informed choices between the two modeling approaches.

Chapter 1

Background

1.1 Literature

Maritime shipping is a vital component of global commerce, contributing to roughly 90 percent of all worldwide trade as per various studies. This sector's importance to the international economy has grown tremendously in recent years. Predictions from the International Chamber of Shipping (ICS) suggest a significant upsurge in global sea-based trade soon.

Recently, with the surge in fuel costs, the focus on fuel efficiency in shipping has become more prominent. Fuel charges largely dominate operational costs in shipping. Hence, enhancing fuel efficiency is crucial to improve the energy efficiency of vessels. The maritime sector faces two main challenges today, dealing with fluctuating fuel prices and reducing carbon emissions. Therefore, it has become imperative for ship operators to integrate energy-saving solutions to optimize vessel fuel consumption. Various techniques are being devised to scrutinize the aspects influencing the fuel efficiency of ships.

Fuel costs, often called "bunker" expenses, pose a significant financial challenge to ships. The intensity of interest and significance given to designing fuel-efficient ships correlates with the ongoing fuel prices. During the 1970s and 1980s, an observable spike in fuel costs led to the sidelining of ships with high fuel usage. However, with the onset of the 1990s, fuel costs began to decrease, which subsequently reduced the focus on fuel efficiency in the maritime sector. That being said, since the early 2000s, a fresh surge in oil costs has driven shipbuilders and designers to reconsider and innovate solutions focused on decreasing fuel usage and augmenting energy efficiency. See[2,8]

The shipping industry is especially suffering from fuel prices. However, we've got some methods to control fuel consumption. Furthermore, fuel consumption is risky for the environment. So deep interest in protecting the surroundings with the aid of using distinctive techniques. Specifically, nowadays, one of the pinnacle topics is global warming. The shipping industry contributed 4% of world CO2 in 2007, see[2] which is a truly stressful issue for maritime. So the primary intention is to reduce the emission of CO2 by way of adopting technological and operational innovations to reduce fuel consumption. See[2]

1.2 Factor Effect on Fuel Consumption

Factors affecting ship's fuel consumption

Typically a ship's power vs speed curve is prepared during the maritime delivery test. Power is a stable parameter compared to fuel consumption and, therefore, easier to measure. On the other hand, the corresponding ship's speed is measured, which is the most significant parameter that determines both the power and the fuel consumption.
In addition increases in speed, resistance, and fuel consumption increase by any of the following three parameters.

- Increased draft and displacement.
- Worsening of weather conditions.
- Worsening of hull and propeller roughness.

Theories and methods on the estimation of the contribution of each of these parameters on increased resistance and fuel consumption can be found in the literature [see 12]. However, most are based on

Environmental Impact of Maritime Shipping

experiments obtained from a series of tests on specific types of ships and hull forms. Therefore, a statistical voyage analysis [12] was carried out to investigate the influence of the ship's draft, the weather, and the hull and propeller condition to produce the fuel consumption vs. speed curve, which represents a more realistic and accurate approach for contemporary ships, as required. The approach assumes that predictions based on a previous year's performance are more accurate and reliable than those based on sea trials.

The existing power-speed curve has two drawbacks. First, when fuel efficiency and CO2 emission are of concern, fuel consumption is more important to be calculated than engine power.

Secondly, the production of a single curve during sea trials is far from adequate for the entire ship's lifetime, and such a curve is truly theoretical rather than practical. In addition, the operators do not have an analytical and systematic method to develop a more accurate, updated curve applicable for aged ships, not only for new ones. See[12]

1.3 Ship Optimization

Optimization is simply about obtaining an optimum value of a function (minimum or maximum) by selecting the values of its variables from a defined set. Optimizing the operational ship performance deals with minimizing the operational cost of the voyage, which is mainly the fuel cost. Once the best algorithm is chosen and trained, as explained previously, it is used with new data for making predictions of ship fuel oil consumption or other proportional outputs such as the propulsion power. Minimizing fuel consumption is called the objective of the optimization. Applying the prediction model to the new data variables will give the usual ship fuel consumption. However, using machine learning for modeling the ship's performance should not stop at the predictive modeling step. It always has the utmost objective of optimization because traditional analytical methods cannot optimize these complex nonlinear functions with multiple variables. To meet the optimization objective, which is minimizing the model function, the optimal input variable values should be found (see 14). The dataset for operational ship performance modeling usually contains ship data and navigational data. The input variables from navigational data are external to the ship; they describe what happens around the ship during its voyage, such as the weather forecast. Therefore, these variable values cannot be controlled for optimization. In contrast, the ship-specific input variables, such as speed or course, are manageable, and the ship operators can decide to change them depending on their schedules and targets. These variables are called Decision Variables.

Thus, optimization deals specifically with finding the ship decision variables' values to minimize the fuel consumption function of the set of variables. In many applications, these optimization results have been employed in Decision Support Systems. There are two main categories in optimization:

1.4 Unconstrained Optimization

Unconstrained optimization, as its name indicates, is finding the best variables to minimize or maximize a function without any constraint on the variable's values. Ship operators make decisions to reduce fuel consumption by reducing the speed with the specified constraint to arrive on time to avoid extra fuel usage if the ship stays in the waiting area. This is a minimization problem subject to a constraint. The decision variables are not chosen among an infinite number of values. One or more constraints rather than limit them. The new system will then choose the values that meet the defined objective while respecting the specified constraints.

1.5 Energy efficiency

The variables that influence the ship's performance during a sea voyage are numerous. These variables are ship's specific and vary according to the voyage executed seaway. Some of these variables are controllable, like the ship's speed, draught, trim, engine condition, and Hull cleanliness, while other variables are uncontrollable such as weather conditions and shallow water effect(Increased Resistance and Reduced Speed). However, the weather and route optimization techniques can improve ship performance to accommodate these uncontrollable variables. Different operators utilize various energy-saving strategies depending on their operational style. The modes of ship operation can be grouped into three types: liner, industrial, and tramp shipping. Liner shipping often follows a consistent route with set timetables, aiming to optimize revenue, much akin to a scheduled bus service. On the other hand, in industrial shipping, the operator owns both the vessel and its cargo, with the primary goal being to reduce cargo transportation costs. Tramp shipping, in contrast, operates in a manner reminiscent of on-demand taxi services; vessels navigate based on available cargo opportunities to boost their earnings. See[12]

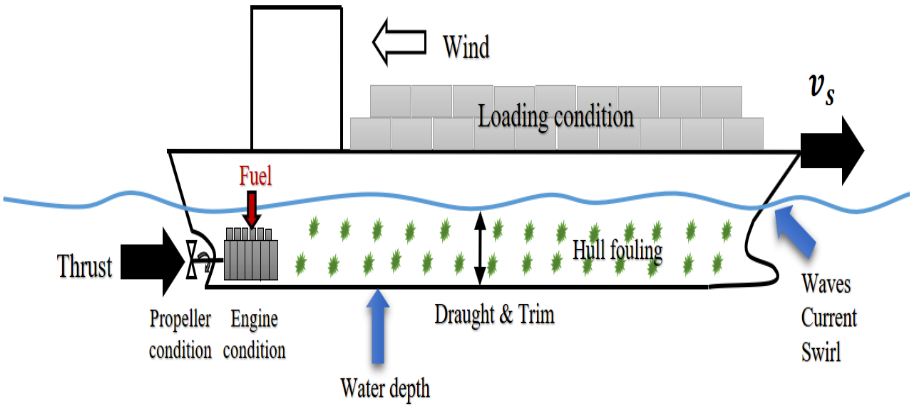

Figure 1.1: factors affecting fuel consummation

1.6 Ship Speed

The operational efficiency of maritime vessels is a critical concern in naval architecture and marine engineering, affecting both economic and environmental aspects. One key aspect of this efficiency is the relationship between a ship's speed and its fuel consumption. Understanding this relationship is essential for optimizing operational practices, designing energy-efficient vessels, and reducing environmental impacts.

1.6.1 Hydrodynamic Resistance and Power Requirement

When a ship traverses through water, it encounters resistance from various forces. These forces include frictional resistance due to the viscosity of water, wave-making resistance as the vessel generates waves, and air resistance against the ship's superstructure. The total hydrodynamic resistance (R_{total}) that a ship faces can be represented as:

$$R_{total} = R_{frictional} + R_{wave-making} + R_{air} + \dots \tag{1.1}$$

The power (P) required to propel the ship at a constant velocity (V) is proportional to this total hydrodynamic resistance:

$$P = R_{total} \times V \tag{1.2}$$

This equation underlines the direct relationship between the resistance encountered and the power needed to maintain speed. See[11]

1.6.2 The Cubic Relationship between Speed and Fuel Consumption

The fuel consumption (F) of a ship is directly related to the power required to maintain its speed, which is influenced by the specific fuel oil consumption (SFOC) of the ship's engine. The relationship between fuel consumption, power, and speed can be described as:

$$F = SFOC \times P \tag{1.3}$$

A significant aspect of the power-speed relationship is its cubic nature, especially in the context of calm water and steady cruising speeds. The required power to overcome hydrodynamic resistance increases with the cube of the speed:

$$P \propto V^3 \tag{1.4}$$

This cubic relationship indicates that even small increases in speed require disproportionately larger increases in power and, consequently, fuel consumption. See [11]

1.6.3 Empirical Evidence and Application

The cubic relationship between ship speed and fuel consumption is well-documented in naval architecture and forms the basis for operational guidelines and ship design. For a detailed theoretical foundation and empirical analysis, one may refer to classical texts such as "Principles of Naval Architecture: Volume II - Resistance, Propulsion and Vibration" by the Society of Naval Architects and Marine Engineers. This volume provides a comprehensive discussion of the factors affecting ship resistance and propulsion, offering insights into the derivation and implications of the cubic relationship.

Moreover, contemporary research in journals like the "Journal of Marine Science and Technology" and "Ocean Engineering" frequently explores the application of this relationship in the context of modern ship designs and technologies, validating its relevance through empirical studies. The cubic relationship between ship speed and fuel consumption is a fundamental principle in marine engineering, emphasizing the importance of speed management for operational efficiency. By understanding and applying this principle, maritime operations can achieve significant improvements in fuel efficiency, cost savings, and environmental sustainability.

Chapter 2

Statistical Methods and Terms

In this chapter, we explore crucial statistical methods and terms, setting the stage for the analyses conducted in Chapter 6. This foundational chapter is essential, as it provides the theoretical underpinnings needed to thoroughly evaluate and devise predictive models for fuel consumption. Through an in-depth examination of statistical distributions, regression analysis, and optimization techniques, we prepare the ground for their direct application in Chapter 6. There, these principles are employed to assess the efficacy of polynomial and multiplicative models under diverse conditions, illustrating the practical application of statistical theory in predictive modeling.

2.1 Observation

In the field of statistics, we frequently discuss 'observations.' These are essentially individual data points or reported values. In the context of this project, each observation signifies the fuel consumption of a ship. It's vital to gather a specific number of observations for accurate statistical analysis. However, the minimum required quantity of observations can vary depending on the task at hand and the granularity of the observations. Therefore, practical judgment plays a crucial role in determining the sufficient number of observations needed for robust statistical analysis.

2.2 Outliers

Before delving into the core topic of our project, it's crucial to understand the concept of outliers, as they can significantly influence the outcome of our analysis. In statistics, outliers are often referred to as extreme values or deviating values. They are observations that noticeably stand apart from the rest of the data. For instance, if we're measuring the average temperature of ten items in a room, and nine have temperatures ranging from 50 to 60 degrees Celsius, while one item is at 150 degrees Celsius, the latter is significantly different from the rest. Such distinctive data points in statistical analysis are referred to as outliers.

2.3 The Coefficient of Determination (R-square)

The term "coefficient of determination" is also referred to as the R-square value, which will be used throughout this report. This coefficient is a measure of how well a regression model represents the data being analyzed. Specifically, it quantifies the extent to which the derived regression function captures the variations present in the data set. An R-square value close to one implies that the regression equation nearly perfectly represents the variations in the data. Conversely, an R-square value near zero suggests a poor representation of the data variations by the regression equation. Notably, there can be instances where the R-square value turns out to be negative. While it might be tempting to attribute this to erroneous calculations, it could be an indication of extremely poor regression. In such situations, a negative R-square should be interpreted as a zero value. A low or negative R-square could arise if there are variations not adequately accounted for, or if a different equation might better describe the variations. Formula to calculate R^2 is given below

$$R^2 = 1 - \frac{\sum_i (y_i - f(x_i))^2}{\sum_i (y_i - \bar{y})^2}$$

In this equation, y_i represents the observed values, $f(x_i)$ represents the predicted values from the model, and $\bar{y}$ is the mean of the observed values. The numerator of the fraction inside the parentheses is the residual sum of squares, and the denominator is the total sum of squares.

2.4 Residual

A residual is the vertical distance between a data point and the regression line. In other words, the residual represents the error between the observed data and the regression model that is not explained by the regression line. Residuals can also be expressed mathematically: they are the difference between the observed values (y_i) and the predicted values $f(x_i)$ from the regression model.

2.5 Confidence Interval

In statistics, a confidence interval (CI) for the median is a range of values that is likely to include the population median with a certain level of confidence. Confidence intervals provide bounds around the estimate to express the degree of uncertainty associated with it. Unlike the mean, which is influenced by outliers and skewed distributions, the median gives the central tendency of a dataset and is robust to such issues. Hence, a CI for the median can be especially useful when dealing with non-normally distributed data.

To calculate a confidence interval for the median, non-parametric methods are typically used. One common approach is to use the binomial distribution to determine the ranks in the ordered dataset within which the median will fall, given a certain level of confidence (usually 95%).

The formula for a non-parametric confidence interval for the median is often based on the order statistics. For a sample size n and a desired confidence level $1 - \alpha$, the $\frac{\alpha}{2}$ and $1 - \frac{\alpha}{2}$ quantiles of the binomial distribution give the ranks of the sample that bound the median.

For example, a 95% confidence interval for the median can be calculated using the following steps:

1. Arrange the data in ascending order.
2. Determine the $\frac{\alpha}{2}$ and $1 - \frac{\alpha}{2}$ quantiles of the binomial distribution with parameters n (the sample size) and probability $p = 0.5$. These quantiles give the lower and upper bounds of the ranks for the median.
3. Find the sample observations that correspond to these ranks. These observations are the endpoints of the confidence interval for the median. See[10,16]

2.6 Curve Fitting

The process of constructing a mathematical curve so that it has the best possible fit to some series of data points is usually referred to as curve fitting. Exactly what fit means and what constraints are put on the constructed curve varies depending on context. In this section, we will discuss a few different scenarios and methods that are related to curve fittings and estimation. We will give an introduction to a few different interpolation methods that give a curve that passes exactly through a finite set of points. If we cannot make a curve that passes through the points exactly, we will need to choose how to measure the distance between the curve and points to determine what curve fits the data points best.

2.6.1 Least Square Method

The least squares method is a statistical technique used to determine the best-fitting line or curve for a set of data points. It achieves this by minimizing the sum of the squares of the residuals, which are the distances between the observed data points and the values predicted by the model. Least squares regression is used to predict the behavior of dependent variables.

The method of least squares is a standard approach in regression analysis to approximate the solution of overdetermined systems, i.e., sets of equations in which there are more equations than unknowns. "Least squares" means that the overall solution minimizes the sum of the squares of the residuals made in the results of every single equation. The most important application is in data fitting. The best fit in the least-squares sense minimizes the sum of squared residuals (a residual being the difference between an observed value and the fitted value provided by a model). When the problem has substantial uncertainties in the independent variable (the x variable), then simple regression and least-squares methods have problems; in such cases, the methodology required for fitting errors-in-variables models may be considered instead of that for least squares.

Least-squares problems fall into two categories: linear or ordinary least squares and nonlinear least squares, depending on whether or not the residuals are linear in all unknowns. The linear least-squares problem occurs in statistical regression analysis; it has a closed-form solution. The nonlinear problem is usually solved by iterative refinement; at each iteration, the system is approximated by a linear one, and thus the core calculation is similar in both cases.

Polynomial least squares describe the variance in a prediction of the dependent variable as a function of the independent variable and the deviations from the fitted curve. See[13,17]

2.6.2 Least Square Curve Fitting

Curve fitting involves establishing mathematical relationships between dependent and independent variables through an equation for a given data set. Suppose the data points are expected to align on a straight line. If exact reproduction of the data points is not necessary, an alternative to interpolation is the least squares fitting method. This involves selecting parameters of the model, denoted as β, to minimize the sum of the squares of the residuals:

$$S(\beta) = \sum_{i=1}^{n} (y_i - f(\beta; x_i))^2 \tag{2.1}$$

This approach is suitable when the data series is influenced by independent and normally distributed noise. The most common form of least squares fitting is linear, where $f(\beta; x)$ depends linearly on β, often yielding a unique and simple-to-find solution. This is known as the least squares method. For a non-linear $f(\beta; x)$, finding the least squares fittings can be challenging, often requiring numerical methods, such as the Marquardt least squares method. See[17]

Two primary reasons for employing this method include experimental error and the fact that the underlying relationship may not be strictly linear but only approximately so. The Method of Least Squares, leveraging calculus and linear algebra, aims to find the 'best fit' line by minimizing the sum of the squares of the residuals - the differences between observed values and model predictions. If the observed data is represented as $y(x)$ and the error, or residual, at each point as $E(x)$, the goal is to find a function $f(x)$ that minimizes the sum of $E(x)^2$ over all data points:

$$f(x) = y(x) + E(x) \tag{2.2}$$

2.6.3 Non-Linear Least Squares Method

Introduction

Non-linear least squares is a statistical method used for estimating the parameters of a non-linear model. It is widely used in many scientific and engineering disciplines when the relationship between variables is inherently non-linear.

Mathematical Formulation

Consider a set of m observations (x_i, y_i), where $i = 1, 2, \ldots, m$, and a model function $y = f(\mathbf{x}, \beta)$ that is non-linear in the n parameters $\beta = (\beta_1, \beta_2, \ldots, \beta_n)^T$. The objective is to find the parameter vector β that best fits the data in the least squares sense.

Residuals and Objective Function

The residual for each observation is the difference between the observed value y_i and the model prediction $f(x_i, \beta)$:

$$r_i(\beta) = y_i - f(x_i, \beta) \tag{2.3}$$

The goal is to minimize the sum of the squares of these residuals. The objective function, also known as the cost function, is defined as:

$$S(\beta) = \sum_{i=1}^{m} r_i(\beta)^2 = \sum_{i=1}^{m} [y_i - f(x_i, \beta)]^2 \tag{2.4}$$

Derivation of the Normal Equations

To find the optimal β, we need to solve the problem:

$$\min_{\beta} S(\beta) \tag{2.5}$$

This is achieved by setting the gradient of $S(\beta)$ with respect to β to zero. The gradient is a vector of partial derivatives:

$$\nabla S(\beta) = \left(\frac{\partial S}{\partial \beta_1}, \frac{\partial S}{\partial \beta_2}, \ldots, \frac{\partial S}{\partial \beta_n} \right)^T \tag{2.6}$$

Each component of the gradient can be expanded as:

$$\frac{\partial S}{\partial \beta_j} = -2 \sum_{i=1}^{m} [y_i - f(x_i, \beta)] \frac{\partial f(x_i, \beta)}{\partial \beta_j} \tag{2.7}$$

Setting $\nabla S(\beta) = \mathbf{0}$ results in a system of n non-linear equations, which are often referred to as the normal equations of the non-linear least squares problem. See[17]

Solution Methods

Solving these equations directly is generally not feasible due to their non-linearity. Instead, iterative methods such as the Newton-Raphson, Gauss-Newton, or Levenberg-Marquardt algorithms are employed.

Newton-Raphson Method

The Newton-Raphson method involves updating the parameter estimate using the formula:

$$\beta^{(\text{new})} = \beta^{(\text{old})} - [J^T J]^{-1} J^T \mathbf{r} \tag{2.8}$$

where J is the Jacobian matrix of partial derivatives of the residuals with respect to the parameters, and $\mathbf{r}$ is the vector of residuals. See[13,17]

Gauss-Newton Algorithm

The Gauss-Newton algorithm is a simplification of the Newton-Raphson method and is more commonly used in practice. It assumes that the second-order derivatives are negligible, simplifying the update rule.

Levenberg-Marquardt Algorithm

The Levenberg-Marquardt algorithm is a popular choice for non-linear least squares problems. It combines the Gauss-Newton algorithm and the method of gradient descent, providing a balance between the speed of Gauss-Newton and the stability of gradient descent.

Convergence Criteria

The iterative process continues until a convergence criterion is met. This could be a small change in the value of the objective function, a small change in the parameter estimates, or reaching a maximum number of iterations.

2.6.4 Constrained Least Square

In constrained Least squares, we apply some conditions to the parameters. To minimize the sum of squares of error, the curve fit must satisfy other criteria. Here we discuss Linear equality constrained. Suppose curve fit must pass through the fixed contour. Satisfying such constraints is a natural application of the method of Lagrange's Multiplier.

2.6.5 Analysing how well a curve fits

In this thesis, we will discuss two models. So, it is necessary to have some method for comparing the methods and choosing the most suitable one. When the model is constructed with a certain application in mind, there is often a set of required or desired properties given by the application and choosing the best. In many cases, this process is not straightforward and often there is not one model that is better than other candidate models in all aspects, a common example is the trade-off between the accuracy and complexity of the model. It is often easy to improve the model by increasing its complexity but finding the best compromise between accuracy and complexity can be difficult. In this section, we will discuss how to compare models primarily for accuracy and the number of required parameters. See[3]

Regression

Regression is similar to interpolation, except that the presence of noise in the data is taken into consideration. The typical regression problem assumes that the data points $\{(x_i, y_i), i = 1, 2, \ldots, n\}$ are sampled from a stochastic variable of the form

$$Y_i = f(\beta; x_i) + \epsilon_i \tag{2.9}$$

where $f(\beta; x)$ is a given function with a fixed number of undetermined parameters $\beta \in B$ and ϵ_i for $i = 1, 2, \ldots, n$ are samples of a random variable with expected value zero, called the errors or the noise for the data set.

There are many different classes of regression problems defined by the type of function $f(\beta_1, \ldots, \beta_m; x)$ and the distribution of errors.

Here we would only consider the situation when the ϵ_i variable is independent and normally distributed with identical variance and that the parameter space B is a compact subset of $\mathbb{R}^k$ and that for all x_i the function $f(\beta; x_i)$ is a continuous function of $\beta \in B$.

Suppose we want to choose the appropriate set of parameters for f based on some set of observed data points. A common approach to this is called maximum likelihood estimation.

Prediction Performance Analysis

Mean Square Error

To measure the effectiveness of machine learning techniques in predicting ship fuel consumption, five prediction performance indices were constructed: the mean square error (MSE), root mean square error (RMSE), mean absolute error (MAE), determination coefficient (R^2), and machine learning runtime (T). The mean squared error (MSE) of a model is the average of the squares of the prediction errors, defined as the difference between the estimate and the true value.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

In the formula, n is the total number of data in the test set, y_i is the actual fuel consumption value, and $\hat{y}_i$ is the predicted fuel consumption value. Thus, the mean square error assesses the quality of the estimator in terms of accuracy and the degree of bias. The root mean squared error (RMSE) is the square root of the mean square error, providing a more interpretable measure of error.

Root Mean Square Error (RMSE)

RMSE is the square root of the Mean Square Error (MSE), providing a measure of the standard deviation of prediction errors. It quantifies the differences between predicted and actual values. The formula for RMSE is:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

Mean Absolute Error (MAE)

MAE measures the average magnitude of the errors in a set of predictions, without considering their direction. It's calculated as the average of the absolute differences between predicted and actual values. The formula for MAE is:

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i|$$

Coefficient of Determination (R^2)

R^2 assesses the proportion of variance in the dependent variable predictable from the independent variable(s), serving as a measure of how well-observed outcomes are replicated by the model. The formula for R^2 is:

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \overline{y})^2}$$

where $\overline{y}$ is the mean of observed values y_i.

In these formulas, n is the total number of observations, y_i represents the actual values, $\hat{y}_i$ denotes the predicted values, and $\overline{y}$ is the average of actual values. Smaller values of MSE, RMSE, and MAE indicate closer predictions to the true values, signifying higher prediction accuracy. Conversely, an R^2 value closer to unity reflects a model that accurately predicts the dependent variable. See[12]

Chapter 3

Optimization

Optimization is the process of finding the best solution to a problem by systematically exploring and selecting the optimal values of variables or parameters. In mathematics and computer science, optimization involves formulating an objective function that needs to be maximized or minimized, along with constraints that define the allowable set of solutions.
There are various types of optimization problems, including:

- Unconstrained Optimization: In unconstrained optimization, the objective function is optimized without any constraints on the variables. The goal is to find the global minimum or maximum of the objective function.

- Constrained Optimization: Constrained optimization involves optimizing an objective function while adhering to a set of constraints. Constraints can be equality constraints, where the variables must satisfy specific equations, or inequality constraints, where the variables must satisfy certain inequalities.

- Linear Programming: Linear programming is a specific type of constrained optimization where the objective function and constraints are linear. The goal is to find the optimal values for the variables that maximize or minimize the objective function while satisfying the linear constraints.

- Nonlinear Programming: Nonlinear programming deals with optimization problems where the objective function or constraints are nonlinear. Nonlinear programming problems are generally more complex and may require specialized optimization algorithms.

- Global Optimization: Global optimization aims to find the global minimum or maximum of an objective function over a given domain, considering all possible solutions. Global optimization methods explore a wide range of solutions to ensure the best possible result, but they can be computationally expensive.

- Convex Optimization: Convex optimization involves optimizing a convex objective function over a convex feasible set. Convex optimization problems have well-defined properties, and efficient algorithms exist to find the optimal solution.

To solve optimization problems, various algorithms and techniques are employed, such as gradient-based methods (e.g., gradient descent), Newton's method, genetic algorithms, simulated annealing, interior-point methods, and more. The choice of algorithm depends on the problem characteristics, such as the nature of the objective function, the presence of constraints, and the desired solution accuracy.

Optimization has applications in diverse fields, including engineering, economics, finance, operations research, machine learning, and many others. It plays a crucial role in decision-making, resource allocation, system design, and parameter estimation, allowing for the identification of optimal solutions and improving efficiency and performance.

There are numerous algorithms available to solve optimization problems, and the choice of algorithm depends on the characteristics of the problem, such as the type of objective function, the presence of constraints, the size of the problem, and the desired solution accuracy. We will use the Trust Region Reflective algorithm to optimize the problem. Trust region methods are popular in optimization because they offer several advantages and are well-suited for certain types of optimization problems. Here are some reasons why trust-region methods are commonly used:

- Flexibility in Handling Nonlinearities: Trust region methods are effective for handling nonlinear objective functions and constraints. They do not rely on linearity assumptions like some other optimization methods, such as Newton's method, which makes them more versatile in dealing with complex nonlinear problems.
- Global Convergence: Trust region methods are known for their global convergence properties. They are designed to converge to a stationary point, which could be a local minimum, global minimum, or saddle point, depending on the problem. This makes trust region methods advantageous when seeking global solutions or when there is a lack of prior knowledge about the problem's landscape.
- Ability to Incorporate Constraints: Trust region methods can handle both equality and inequality constraints in the optimization problem. They can be extended to handle constrained optimization by integrating the constraints into the trust region framework. This allows for the efficient handling of constrained optimization problems without the need for additional techniques.

3.1 Trust Region Algoritm

Let's consider a scenario where we aim to minimize an objective function that relies on real-valued variables without any constraints or limitations on their potential values. Mathematically, let $x \in R^n$ be a real vector with $n \geq 1$ components and let

$$f : R^n \Longrightarrow R$$

be a smooth function. Then unconstrained optimization problem is,

$$\min f(x)$$

Unconstrained optimization problems can emerge both independently in certain applications and indirectly through reformulations of constrained optimization problems. In some cases, it is feasible to replace the constraints of an optimization problem with penalty terms incorporated into the objective function, thereby transforming the problem into an unconstrained form. The trust region reflective algorithm is an optimization algorithm used for solving nonlinear least squares problems. It combines the concepts of trust-region methods and reflective methods to efficiently handle nonlinear constraints and bounds on variables.

Here is a brief overview of the trust region reflective algorithm:

- initialization
  - Choose an initial guess for the parameters.
  - Define the trust region radius, which represents the region around the current iterate where the local model is considered accurate.
- Iteration:
  - Evaluate the objective function and its Jacobian matrix at the current iterate.
  - Compute the step direction by solving a trust region subproblem. This subproblem aims to find the step that minimizes a local quadratic model of the objective function within the trust region.
  - Determine whether to accept or reject the step based on the reduction in the objective function and the agreement with the bounds and constraints.
  - Adjust the trust region radius based on the step quality and convergence criteria.
  - Update the iterate with the accepted step.
  - Repeat the iteration until convergence criteria are met or a maximum number of iterations is reached.
- Convergence:
  - The algorithm converges if the step size becomes small or if the objective function and constraint violation decrease to a desired tolerance level.

The trust region reflective algorithm incorporates the concept of reflective boundaries, which helps handle constraints and bounds on the variables. When a step violates the bounds or constraints, it is reflected back into the feasible region by flipping the sign of the corresponding components.

By combining trust-region methods with reflective techniques, the algorithm efficiently navigates the parameter space, making progress toward the optimum while respecting the constraints and bounds.

The specific mathematical details and equations involved in the trust region reflective algorithm can be quite involved and are beyond the scope of this overview. However, this algorithm builds upon the foundation of trust-region methods and incorporates additional techniques to handle constraints and bounds, making it a powerful tool for solving nonlinear least squares problems. See[19,20]

Chapter 4

Introduction to Statistical Distributions

Statistical distributions are mathematical models that describe how observations of a random variable are dispersed or spread across possible values. These models are essential in statistical analysis, enabling researchers and analysts to understand data patterns, make predictions, and infer population characteristics from sample data. This chapter introduces the fundamental concepts of statistical distributions, emphasizing their importance and the key principles underlying their application.

4.1 Understanding Statistical Distributions

A statistical distribution represents the possible values a random variable can take and the probability of these values occurring. This concept is pivotal in statistics as it provides a framework for modeling uncertainties and variability in real-world phenomena. Distributions can be broadly categorized into two types based on the nature of the random variable they represent:

- **Discrete distributions** are used when the random variable takes on countable values, such as the number of defective items in a batch or the number of heads in a series of coin tosses.
- **Continuous distributions** apply to random variables that can assume any value within an interval or range, such as the height of individuals in a population or the time required for a chemical reaction to complete.

These categories encompass a wide range of distributions, each suited to modeling specific types of data and phenomena.

4.2 The Significance of Statistical Distributions in Data Analysis

Statistical distributions are foundational to various aspects of data analysis and statistical inference. They enable researchers to:

1. Model the distribution of data points within a dataset, facilitating the understanding of data characteristics and behaviors.
2. Calculate probabilities associated with specific outcomes, which is crucial for risk assessment, decision-making, and prediction models.
3. Conduct hypothesis testing to assess the validity of assumptions about population parameters based on sample data.
4. Estimate confidence intervals for population parameters, providing a range of plausible values for these parameters based on sample observations.

Understanding the appropriate distribution to apply in a given scenario is critical for accurate analysis and reliable conclusions.

4.3 Key Concepts in Statistical Distributions

Several fundamental concepts are integral to working with statistical distributions:

4.3.1 Probability Density Function (PDF) and Probability Mass Function (PMF)

The PDF and PMF are functions that describe the likelihood of a random variable assuming specific values. The PDF applies to continuous distributions and gives the probability of the variable falling within a particular interval, whereas the PMF is associated with discrete distributions and specifies the probability of the variable taking on an exact value. See[1]

4.3.2 Cumulative Distribution Function (CDF)

The CDF is a function that indicates the probability of a random variable being less than or equal to a certain value. It is a fundamental tool for understanding the distribution of a variable and is applicable to both discrete and continuous distributions.

4.3.3 Expected Value, Variance, Skewness, and Kurtosis

These statistical measures provide insight into the characteristics of a distribution:

- The **expected value** (mean) represents the central tendency of the distribution.
- **Variance** measures the spread or dispersion of the distribution around the mean.
- **Skewness** quantifies the asymmetry of the distribution around the mean.
- **Kurtosis** indicates the "tailedness" of the distribution, reflecting the likelihood of extreme values.

4.4 Common Types of Statistical Distributions

This section provides a detailed overview of the statistical distributions utilized in Chapter 6 to analyze various effects. We will highlight the unique properties and applications of each distribution, setting the stage for a deeper understanding of their role in our analysis.

4.4.1 Extreme Value Distribution

Definition The probability density function for the extreme value distribution with location parameter μ and scale parameter σ is

$$f(x|\mu,\sigma) = \frac{1}{\sigma}\exp\left(-\frac{x-\mu}{\sigma} - \exp\left(-\frac{x-\mu}{\sigma}\right)\right) \tag{4.1}$$

This form of the probability density function is suitable for modeling the minimum value. To model the maximum value, use the negative of the original values. If T has a Weibull distribution with parameters a and b, then log T has an extreme value distribution with parameters μ = log a and σ = 1/b. See [1]

4.4.2 Background

Extreme value distributions are often used to model the smallest or largest value among a large set of independent, identically distributed random values representing measurements or observations. The extreme value distribution is appropriate for modeling the smallest value from a distribution whose tails decay exponentially fast, such as the normal distribution. It can also model the largest value from a distribution, such as the normal or exponential distributions, using the original values' negative. For example, figure 4.1 fits an extreme value distribution to minimum values taken over 1000 sets of 500 observations from a normal distribution.

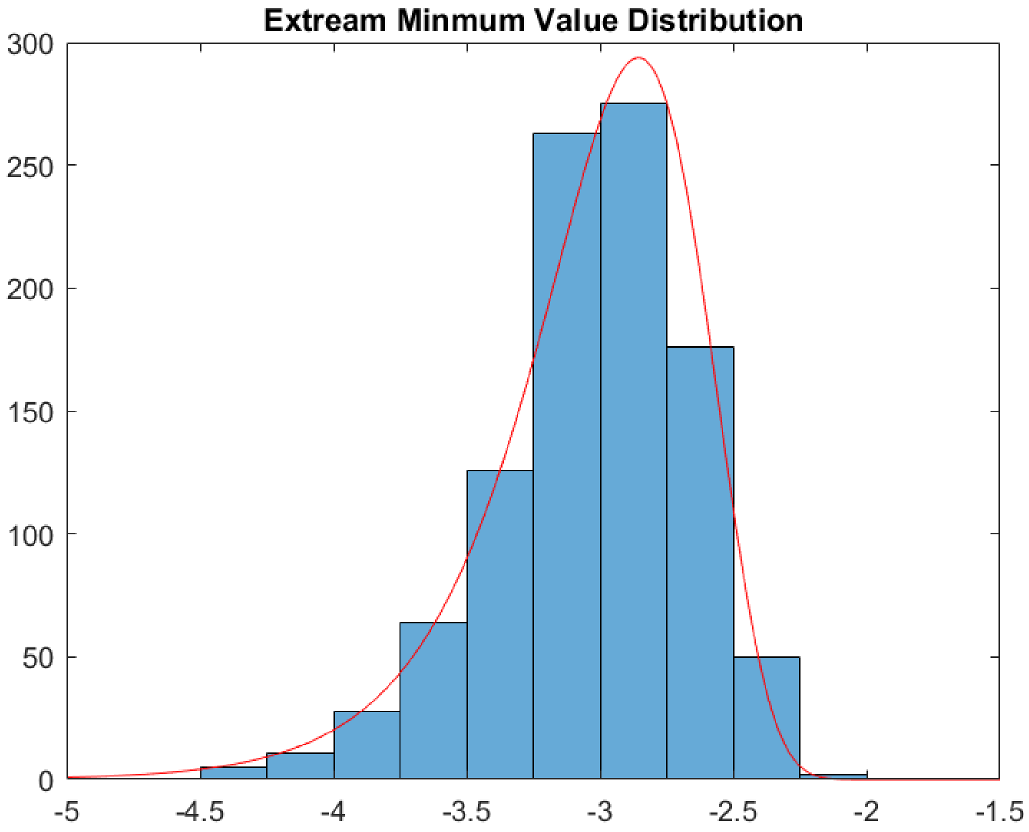

Figure 4.1: Extream Min Value

Figure 4.2 fits an extreme value distribution to the maximum values in each set of observations. Although the extreme value distribution is often used as a model for extreme values, you can also use it for other types of continuous data. For example, extreme value distributions are closely related to the Wei-bull distribution. If T has a Wei-bull distribution, then log(T) has a type 1 extreme value distribution.

The extreme value distribution is skewed to the left, and its general shape remains the same for all parameter values. The location parameter, mu, shifts the distribution along the real line, and the scale parameter, sigma, expands or contracts the distribution. Figure 4.3 represents the probability function for different combinations of μ and σ. See[1]

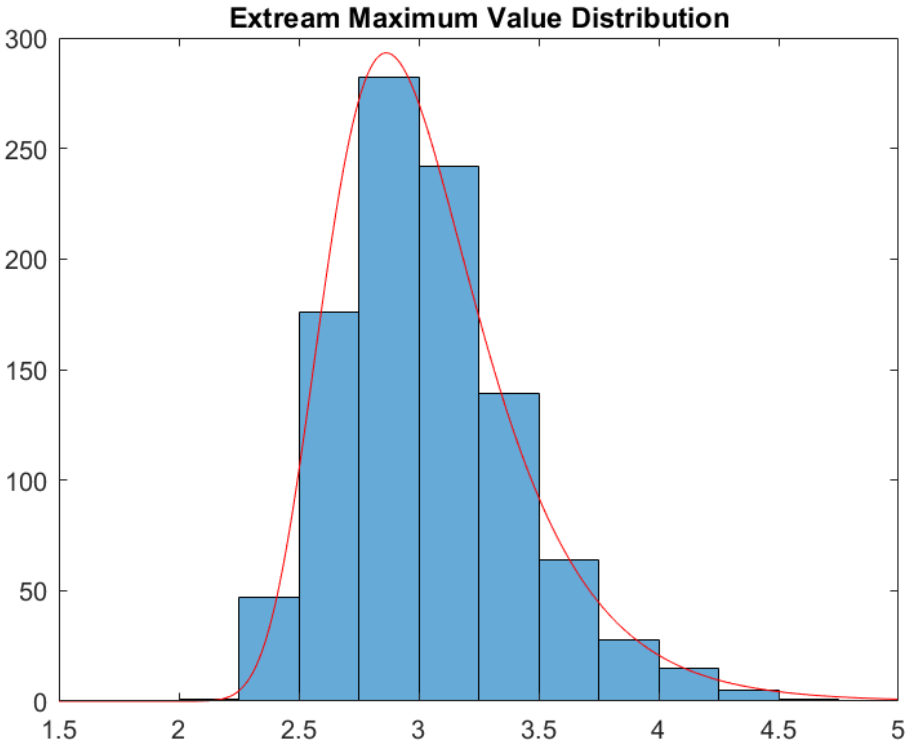

Figure 4.2: Extream Max Value

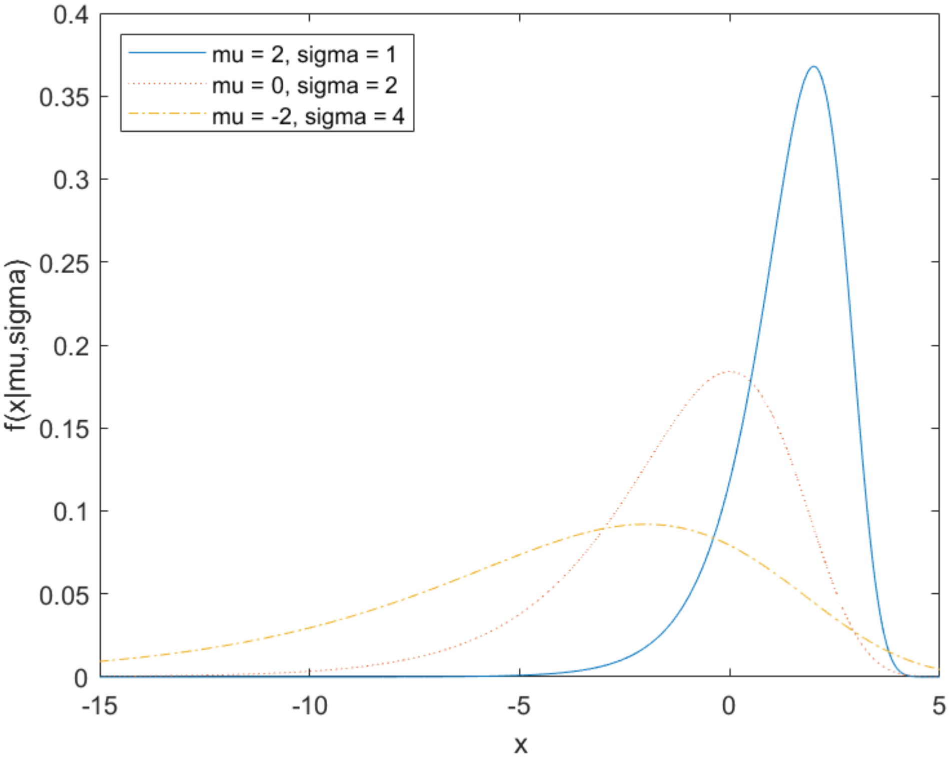

Figure 4.3: Extream Distribution of different combinations of mu and sigma.

4.4.3 Cauchy Distribution

The Cauchy distribution, named after the French mathematician Augustin Cauchy, is a probability distribution that is characterized by its heavy tails and lack of finite moments. It is a continuous probability distribution that often arises in various areas of statistics and physics.
The probability density function (PDF) of the Cauchy distribution is given by:
The probability density function (PDF) of the Cauchy distribution is given by:

$$f(x; x_0, \gamma) = \frac{1}{\pi\gamma} \cdot \left[\frac{\gamma^2}{(x - x_0)^2 + \gamma^2} \right]$$

where x is a real-valued variable, x_0 is the location parameter representing the median of the distribution, and γ is the scale parameter determining the spread of the distribution.

- Heavy Tails: The Cauchy distribution has tails that extend infinitely. This means that extreme values are more likely to occur compared to distributions with finite tails, such as the normal distribution.
- Lack of Finite Moments: Unlike many other probability distributions, the Cauchy distribution does not possess finite moments. This means that its mean, variance, and higher-order moments are undefined.
- Symmetry: The Cauchy distribution is symmetric around its median (x_0). This symmetry is evident in its PDF and impacts various statistical properties of the distribution.
- Central Limit Theorem: The Cauchy distribution does not obey the Central Limit Theorem. This implies that the sum of independent Cauchy-distributed random variables does not converge to a normal distribution as the sample size increases.

The Cauchy distribution has applications in fields such as physics, finance, and Bayesian statistics. It is particularly useful in modeling scenarios involving heavy-tailed data or situations where outliers play a significant role. However, it is important to note that the Cauchy distribution can be challenging to work with due to its unique properties, and its infinite tails require careful consideration when applying statistical techniques. See [1]

4.4.4 Normal Distribution

The normal distribution, also known as the Gaussian distribution, is a fundamental probability distribution in statistics that is symmetric around the mean, indicating that data near the mean are more frequent in occurrence than data far from the mean. When plotted, the normal distribution takes the shape of a bell curve.

4.4.5 Mathematical Explanation

The probability density function (PDF) of the normal distribution is mathematically represented as:

$$f(x|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{4.2}$$

where:

- x is the variable,
- μ is the mean of the distribution,
- σ is the standard deviation of the distribution, and
- σ^2 is the variance.

4.4.6 Properties of the Normal Distribution

Several key properties characterize the normal distribution:

1. **Symmetry**: It is perfectly symmetrical around its mean.
2. **Mean, Median, Mode Equality**: The mean, median, and mode of a normal distribution are all equal.
3. **Bell-shaped Curve**: The distribution displays a bell-shaped curve, where the spread is determined by the standard deviation (σ). A larger σ results in a wider, flatter curve, while a smaller σ leads to a narrower, more peaked curve.
4. **Asymptotic**: The tails of the distribution curve approach the horizontal axis but never touch it.
5. **Empirical Rule**: Approximately 68% of the data falls within one standard deviation of the mean, about 95% within two standard deviations, and about 99.7% within three standard deviations.

The normal distribution is widely applied across various fields, including finance, science, and engineering, to model phenomena that naturally follow or approximate this distribution. See [1]

Chapter 5

Detailed Methodology for Data Generation

In this chapter, we delve into the creation of a simulated dataset, a pivotal component of our research. The need for simulated data arises from the necessity to model complex real-world phenomena in a controlled environment, allowing for a thorough examination of the phenomena under study. Two key sections form the core of this chapter:

- **Rationale for Simulated Dataset Design:** Understanding the 'why' behind our approach is as crucial as the approach itself. This section sheds light on the motivations and objectives behind the design of our simulated dataset. It elucidates the specific characteristics we chose to include in the dataset and explains how these characteristics align with the broader goals of our research.
- **Data Generation Methodology:** With a clear understanding of the rationale, we then describe the 'how' of our data generation. This section provides a detailed account of the methodological steps, algorithms, and statistical models employed to generate the dataset. It serves as a guide to the technical processes we followed, ensuring reproducibility and transparency in our research methods.

Through these sections, the chapter aims to provide a comprehensive overview of the thought process and technical procedures involved in the generation of the simulated data, which forms the foundation for our subsequent analyses and discussions.

5.1 Rationale for Simulated Dataset Design

Before detailing the process of data generation, it is crucial to understand the reasoning behind the design of the simulated dataset. The purpose of the simulation is to create a controlled environment that mimics certain conditions or phenomena that are of interest to our study.

5.1.1 Objectives of Simulated Data

The primary objectives for generating this particular dataset are as follows:

1. **Replication of Real-World Conditions:** The dataset is designed to reflect real-world conditions where fuel consumption data exhibit abrupt changes due to various factors such as acceleration, gear shifts, or changes in terrain.
2. **Testing of Analytical Methods:** By introducing known jumps and patterns, the dataset serves as a benchmark to test the efficacy of analytical methods in detecting and quantifying sudden changes in data, which are common in many real-world applications.
3. **Algorithm Development:** The structured nature of the dataset with predefined jumps allows for the development and fine-tuning of algorithms that can later be applied to real, noisy, and unpredictable data.

4. **Understanding Data Behavior:** The simulation helps in understanding how certain data behaviors manifest and how they can be captured and analyzed. This is particularly useful in domains where data-driven decisions are critical.

5.1.2 Significance of Specific Data Characteristics

The specific characteristics of the dataset are chosen based on the following considerations:

1. **Jumps in Data:** Real-world fuel consumption data often display jumps due to abrupt operational changes. Simulating these jumps allows us to study their impact on data analysis and model performance.

2. **Segmentation of Data:** By dividing the dataset into intervals, we can analyze the data in segments, which is a common approach in time-series analysis, especially when looking for patterns or trends within specific time frames.

3. **Inclusion of Noise:** Adding noise to the data simulates the measurement error and natural variability present in actual data collection, providing a realistic challenge for data processing techniques.

5.2 Data Generation Methodology

Following the rationale outlined previously, this section delves into the specific methodologies employed to generate our simulated dataset. The focus here is on the practical application of the concepts and principles that were previously discussed, detailing the step-by-step process used to create a dataset that accurately mirrors the intended real-world scenarios and research objectives.

Key steps in this process include the random generation of jump indices, the simulation of jump heights from a log-normal distribution, the segmentation of data into intervals, and the introduction of varying levels of noise. Each of these steps is crucial for ensuring that the dataset not only reflects the complexity of real-world data but also aligns with our research goals.

In this chapter, we detail the methodology adopted for generating a simulated dataset. This dataset is specifically designed to model scenarios characterized by abrupt changes or 'jumps' in data, akin to those observed in fields such as environmental monitoring or financial markets.

5.2.1 Overview of Data Generation Process

Our data generation process aims to create a dataset of 10,000 observations, interspersed with 20 randomly placed jumps. These jumps are intended to represent sudden changes in the measured variable, which, for the purpose of our simulation, we will refer to as 'fuel consumption'. The process is executed as follows:

5.2.2 Setting the Framework

- **Number of Observations (N):** We set N to 10,000, establishing the total number of data points in our dataset.
$$N = 10000$$

- **Number of Jumps (jN):** We define jN as 20, indicating the total number of significant changes or jumps to be introduced in the data.
$$jN = 20$$

5.2.3 Random Generation of Jump Indices

- The jump indices, jumpInd, are generated to randomly determine where these jumps will occur within our dataset. This randomness is key to simulating the unpredictability inherent in real-world data.
$$\text{jumpInd} = \{i_1, i_2, \ldots, i_{jN}\}$$

The indices for these jumps denoted as jumpInd, are randomly selected from the set $\{1, 2, \ldots, N\}$ to ensure unpredictability in their locations. The randomness mimics the irregular occurrence of events in natural datasets. Mathematically, it can be represented as:

$$\text{jumpInd} = \text{sort}(\text{randperm}(N, jN))$$

where `randperm(N, jN)` generates a random permutation of `jN` unique integers from 1 to `N`.

- We sort these indices in ascending order and extend them to frame the entire range of observations, including the start and end points.

$$\text{jumpInd} = [0, \text{jumpInd}, N]$$

5.2.4 Generation of Jump Heights

- **Log-Normal Distribution:** Each jump's magnitude is determined by drawing from a log-normal distribution, reflective of many natural and economic phenomena. The distribution parameters are set to $\mu = \log(15)$ and $\sigma = 0.5$.

$$\text{jumpHeights} = \{X_1, X_2, \ldots, X_{jN+1}\}$$

Each jump height X_i is a random variable following a log-normal distribution:

$$X_i \sim \text{Lognormal}(\mu, \sigma^2)$$

$$\text{jumpHeights} \sim \text{Log-Normal}(\log(15), 0.5)$$

5.2.5 Populating the Dataset

- We initialize a data vector, f, of length N with zeros. This vector will subsequently be populated with the jump heights.

$$f = \text{zeros}(N, 1)$$

- For each interval defined by consecutive elements in jumpInd, we assign the corresponding jump height from jumpHeights to the data vector f.

$$\forall i \in \{1, 2, \ldots, jN + 1\}, \text{ set } f[\text{jumpInd}[i] + 1 : \text{jumpInd}[i + 1]] = \text{jumpHeights}[i]$$

5.2.6 Significance of the Data Generation Approach

This method of data generation allows us to simulate a scenario where the variable of interest exhibits sudden changes at irregular intervals. By employing a log-normal distribution for the jump magnitudes, we incorporate a realistic skewness into the dataset, thereby enhancing the applicability of our simulation to real-world situations.

In the subsequent chapters, we will apply various analytical techniques to this dataset, examining the effectiveness of these methods in identifying and interpreting the characteristics and implications of these jumps.

5.3 Introduction of Conditional Behavior in the Dataset

Following the generation of the primary dataset `f`, we introduce an additional layer of complexity by implementing a conditional behavior mechanism. This mechanism is realized through the creation of a logical index array, `ind`, which plays a pivotal role in segmenting the dataset and applying different conditions or models to distinct segments.

5.3.1 Formulation and Purpose of the Index Array (ind)

The index array `ind` is designed to divide the dataset into distinct segments, within which different conditions or behaviors can be simulated. This segmentation is mathematically formulated as follows:

- **Initialization:** The array `ind` is initialized as a zero vector of length N (the total number of observations), representing the default state.

  $$\texttt{ind} = \text{zeros}(N, 1)$$

- **Segmentation:** The dataset is divided into five equal segments, each representing an interval.

  $$\texttt{interval\_N} = \frac{N}{5}$$

  This division is crucial for introducing periodic changes in the dataset.

- **Assignment of Logical Values:** Within each of these intervals, the first half is assigned a value of 1 (true), while the second half remains 0 (false).

```
for i = 1:5
    ind(interval_N*(i-1)+1:interval_N*(i-1)+interval_N/2) = 1;
end
ind = logical(ind);
```

  This assignment creates a pattern where the ‘true‘ and ‘false‘ values alternate in the dataset.

  $$\text{ind}_i = \begin{cases} 1 & \text{if } i \leq \frac{\text{interval\_N}}{2} \text{ in each segment} \\ 0 & \text{otherwise} \end{cases}$$

For each observation indexed by i, the indicator variable ind_i is defined to distinguish between the first and second halves of each interval. It is set to 1 for observations within the first half of each interval and 0 for the second half. This binary classification allows for the investigation of how data characteristics might differ between two distinct segments within each interval.

5.3.2 Application in Data Generation

The logical array `ind` is then utilized to conditionally modify or influence the generated data in f. For instance, in scenarios where f represents a measurement like fuel consumption, `ind` can be used to simulate different operational modes or environmental conditions affecting consumption rates. This approach allows for the simulation of more realistic and dynamic scenarios, closely mimicking the complexities observed in real-world data.

5.3.3 Significance in the Context of the Study

The introduction of `ind` and its application to the dataset f is significant for several reasons:

- It allows for the examination of how different conditions or states affect the observed variable.
- It introduces a controlled variability into the dataset, enabling the testing and validation of analytical models under varying conditions.
- It enhances the realism of the simulated data, making it more representative of actual scenarios where periodic changes are common.

In the subsequent analysis, we leverage this conditional behavior to investigate how environmental fluctuations influence the growth rates of coral reefs.

5.4 Results

Following the comprehensive data generation methodology outlined in section [5.2,5.3], the resultant dataset was analyzed and visualized to assess the characteristics and behaviors embedded through the simulation process. This section presents the graphical representation of the dataset and provides an analysis of the observed patterns.

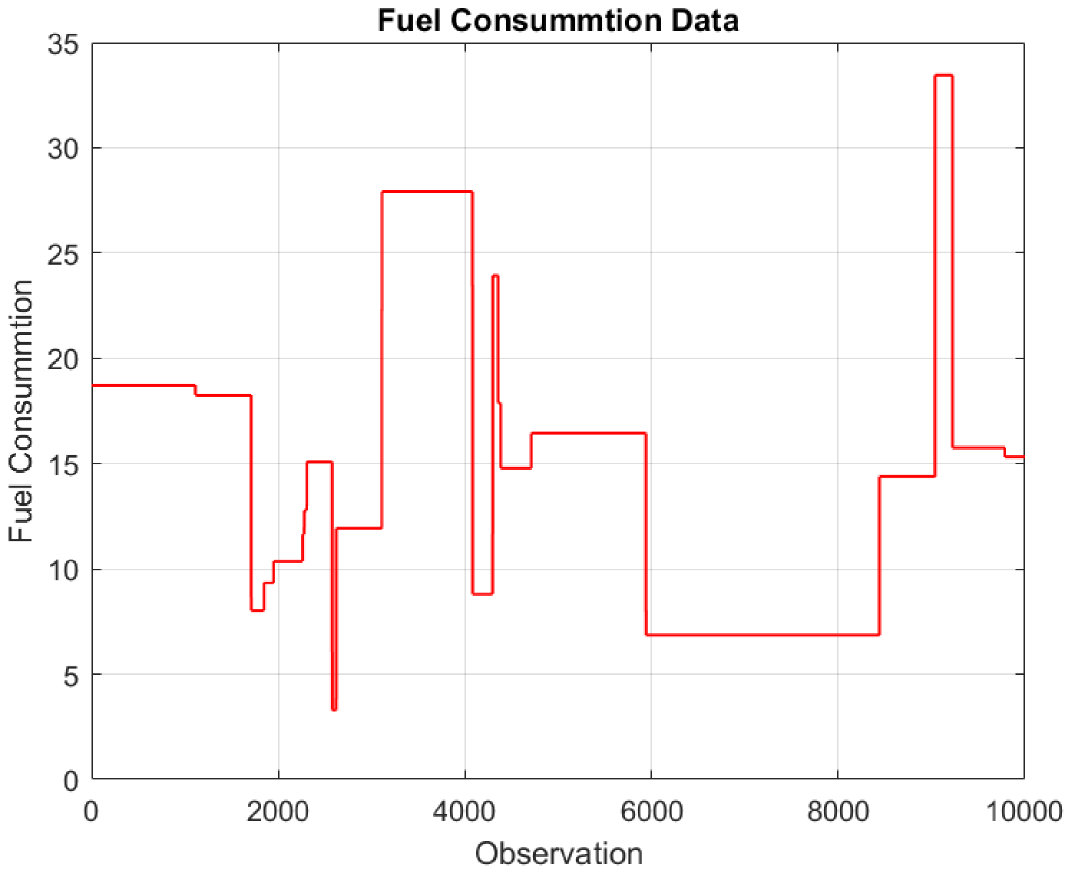

Figure 5.1: The plot illustrates the simulated fuel consumption over the number of observations.

The simulated fuel consumption data is depicted in Figure [5.1]. The x-axis represents the number of observations, while the y-axis quantifies the fuel consumption.
The graphical analysis indicates that fuel consumption remains constant for certain periods, followed by abrupt shifts to different consumption levels. These patterns were deliberately inserted into the dataset to model the behavior of a vessel engine under various conditions. The consistency of the flat regions aligns with our expectation of steady-state operation, where the vessel maintains a consistent speed and engine load, resulting in uniform fuel consumption. Meanwhile, the jumps align with the simulated anomalies, which represent scenarios such as rapid acceleration or deceleration, changes in the vessel's operational mode, or external factors like weather conditions that would impact fuel consumption

Chapter 6

Fuel Consumption Prediction Model

6.1 Case Study

Fuel consumption is generally strongly dependent on speed, meaning that increasing the speed leads to an increase in fuel consumption. To better understand this relationship, a performance curve can be derived. In this thesis, the performance curve will be approximated using a polynomial order-2 equation. This approximation allows for the identification of the most economic vessel speed, which corresponds to the minimum point on the curve. Furthermore, in this thesis, two main models will be considered: the polynomial model and the multiplicative model. These models provide different approaches to analyzing the relationship between fuel consumption and one or more input variables. The polynomial model captures non-linear relationships by employing a polynomial equation, while the multiplicative model incorporates interactions between variables through multiplication. Through the application of these models and subsequent data analysis, this thesis seeks to uncover the patterns within the models and assess the impact of various distributions on them.

In the first model present in equation (6.1), f is an independent variable, and v_1 is the dependent variable.

$$v_1 = a_1 + a_2 f + a_3 f^2 \tag{6.1}$$

The parameters a_1, a_2, and a_3 are the coefficients that regression tries to minimize such that the resulting function describes the best result as much as possible. The 2nd model is described in equation (6.2). The 2nd model is an extension of the 1st model.

$$v_2 = k(a_1 + a_2 f + a_3 f^2) \tag{6.2}$$

The main focus is on comparing the two models, Model $v1$ and Model $v2$. Model $v1$ has three parameters

- a_1 is the constant term, possibly representing the baseline performance measure when there is no fuel consumption.
- a_2 is the linear coefficient, indicating the initial rate of change in v_1 with respect to fuel consumption.
- a_3 is the quadratic coefficient, which accounts for the rate of change in the relationship's slope. This term becomes significant at higher levels of fuel consumption, potentially reflecting increased energy loss to factors like drag or inefficiencies that grow disproportionately with speed.

Model $v2$ also has three parameters, but model $v2$ are slightly different

- a_1, a_2, and a_3 has the same interpretation in model v_2 as in model $v1$. In other words, the $v2$ model is just a rescaled version of the $v1$ model, where k is a scaling constant that could represent an adjustment for different conditions, such as vehicle load, mode of operation, or environmental factors that affect performance uniformly across all levels of fuel consumption.

6.2 Integration and Application of Model $v3$ in Data Analysis

After establishing the foundational models v1 and $v2$, we introduce $v3$, a composite model, to our analytical toolkit. This model is not only a fusion of the two preceding models but also an adaptive framework tailored to the complexities and nuances evident in our dataset.

6.2.1 Contextualizing $v3$ with the Dataset

Our dataset, as visualized in [data plot section 5.4], exhibits characteristics that necessitate a flexible modeling approach. Notably, the data portrays periods of varied fuel consumption patterns, suggesting different operational conditions. $v3$ is specifically designed to address this variability.

6.2.2 Operational Mechanism of $v3$

The model $v3$ operates on a fundamental principle: selecting the appropriate model (either $v1$ or the scaled version $v2$) based on the condition at each data point. This selection is governed by the logical index array 'ind', The logical index array *ind* is derived based on specific conditions observed within the data, which dictate the switching mechanism between models $v1$ and the scaled version $v2$. This is to ensure the modeling approach accurately captures the observed phenomena. The process involves analyzing the dataset for certain characteristics or patterns that necessitate a change in the model to reflect the data's behavior accurately. .

- When 'ind(i)' is true, indicating standard operational conditions, $v3$ aligns with $v1$. This scenario is reflected in parts of the dataset where the data demonstrates stable and expected patterns without significant fluctuations or anomalies. These conditions might include regular operating environments where the system performance is within expected norms.
- Conversely, when 'ind(i)' is false, suggesting altered conditions, 'v3' shifts to the $v2$ model, scaling the output to represent scenarios such as increased load, different operational modes, or other stress conditions on the system. These altered conditions might reflect situations where the system is under unusual stress or operating in a mode that deviates from the norm, requiring the adjusted model to accurately predict performance.

6.2.3 Mathematical Expression of $v3$

Recalling the mathematical formulation:

$$v3(i) = \begin{cases} a_1 + a_2 \cdot f(i) + a_3 \cdot f(i)^2 & \text{if } ind(i) = \text{true} \\ k \cdot (a_1 + a_2 \cdot f(i) + a_3 \cdot f(i)^2) & \text{if } ind(i) = \text{false} \end{cases}$$

The parameters a_1, a_2, and a_3 are consistent with those in $v1$, while the scaling factor k is the same as in $v2$. The behavior of $v3$ at each data point directly responds to the operational conditions as signaled by 'ind'.

6.2.4 Analytical Implications of Using $v3$

Applying $v3$ to our dataset allows us to dissect the data into segments that either correspond to the standard model $v1$ or the adjusted model $v2$. This segmentation leads to a more nuanced understanding of efficiency under different operational modes in response to environmental changes.

In the following analysis, we apply $v3$ across the dataset, showcasing its effectiveness in capturing and differentiating the behaviors under varying operational conditions. This approach not only validates the underlying assumptions of our models but also enriches our understanding of the complex dynamics present in maritime fuel consumption.

6.3 Parameter Estimation via Nonlinear Least Squares

In our study, we aim to optimize the parameters of the composite model $v3$ using a nonlinear least squares method. This approach is essential for accurately fitting our model to the observed data, given the complexity and nonlinear nature of $v3$.

6.3.1 Mathematical Formulation

The model $v3$ is defined as a piecewise function, where its form depends on the logical index array 'ind'. Our objective is to find the parameter set $\{a_1, a_2, a_3, k\}$ that minimizes the sum of squared residuals between the observed data and the predictions made by $v3$.

Residual Function

The residual function for a given observation is the difference between the observed fuel consumption value and the value predicted by the model. Formally, the residual for the i-th observation is defined as:

$$r_i(a_1, a_2, a_3, k) = y_i - v3(i)$$

where y_i is the observed fuel consumption, and $v3(i)$ is the predicted value, calculated as:

$$v3(i) = \begin{cases} a_1 + a_2 \cdot f_i + a_3 \cdot f_i^2 & \text{if } ind(i) = \text{true} \\ k \cdot (a_1 + a_2 \cdot f_i + a_3 \cdot f_i^2) & \text{if } ind(i) = \text{false} \end{cases}$$

Objective Function

The objective function to be minimized, representing the sum of squared residuals over all observations, is:

$$S(a_1, a_2, a_3, k) = \sum_{i=1}^{N} r_i(a_1, a_2, a_3, k)^2$$

6.3.2 Nonlinear Least Squares Optimization

The 'lsqnonlin' function in MATLAB is used to minimize the objective function. This function iteratively adjusts the parameters $\{a_1, a_2, a_3, k\}$ to find the minimum of S. The process involves:

1. **Initial Guess:** Providing an initial guess for the parameters, such as $\{0.3, 0.4, 0.5, 1.05\}$.
2. **Iterative Algorithm:** 'lsqnonlin' employs an algorithm like trust-region-reflective to iteratively refine the parameter values.
3. **Convergence Criteria:** The process continues until a convergence criterion is met, which could be based on the change in the error metric, the change in parameter values, or the number of iterations. For more details, see appendix[6.13.1]

6.4 Selection of Parameter k

In the validation phase of our model, we subjected the parameter k to a thorough analysis by testing a range of initial values from 0.95 to 1.15. The aim was to ascertain a value for k that could be estimated by the model with a high degree of accuracy and precision, indicative of an unbiased and consistent estimation process. Figure 6.1 presents the histograms of the estimated k values over 100 simulations for each initial value of k.

Our analysis indicated that when the true value of k was set at 1.05, the distribution of estimated values centered closely around this true value, suggesting an absence of bias in the estimation process. The histogram for $k = 1.05$, in particular, showed a narrower spread in estimates in comparison to other values, denoting greater precision in estimation at this point. The alignment of the mean and median of the estimated values with the true value of k underscored the estimator's accuracy.

Furthermore, the standard deviation for $k = 1.05$ was the smallest among all tested values, indicating reduced variability in the estimates and thereby suggesting increased reliability. Notably, the 95%

confidence interval for $k = 1.05$ was the most constrained, which corroborates the consistency of the parameter estimates at this value.

The choice of $k = 1.05$ is also supported by its theoretical and empirical significance in the field of [maritime fuel consumption analysis], where it has been observed that [the distribution of estimated values centers closely around the true value, suggesting an absence of bias in the estimation process. The histogram for $k = 1.05$ shows a narrower spread in estimates compared to other values, denoting greater precision. The mean and median of the estimated values align with the true value, indicating the estimator's accuracy. Furthermore, the standard deviation for $k = 1.05$ is the smallest among all tested values, suggesting increased reliability, and the 95% confidence interval is the most constrained, corroborating the consistency of the parameter estimates at this value]. Therefore, considering both the statistical rigor and the theoretical and empirical contexts, $k = 1.05$ was chosen as the preferred value for further analysis within this study.

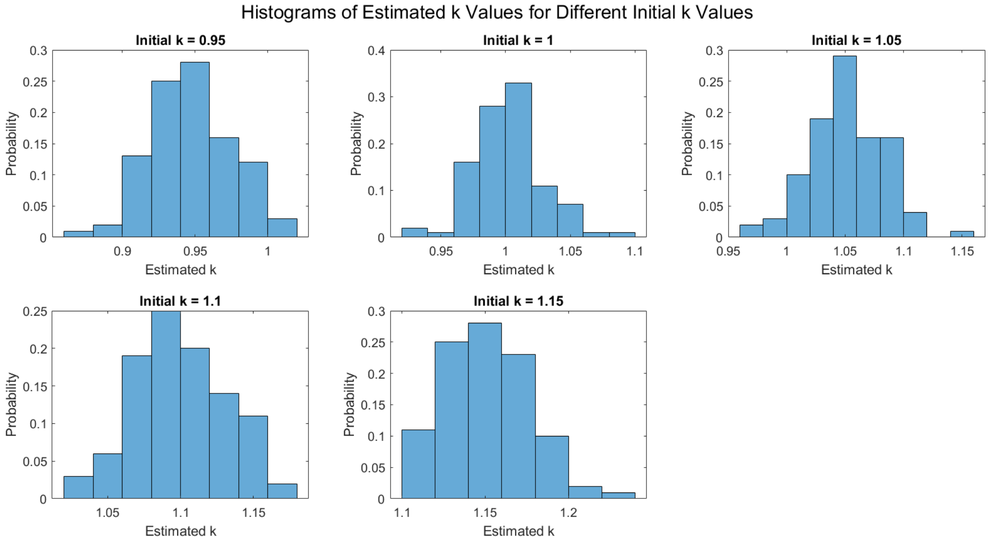

Figure 6.1: Distribution of estimated k values for various initial k values over 100 simulations.

6.5 Incorporation of Diverse Statistical Distributions into Model $v3$

Subsequent to establishing our conditional model $v3$, we enhance the realism of the simulation by integrating three distinct statistical noise distributions. This approach aims to investigate the impact of different types of variability on the model's predictions, which in this context, represent fuel consumption under varying conditions.

6.5.1 Methodology for Noise Integration

Noise is introduced to the output of model $v3$ to simulate various real-world uncertainties. We mathematically formulate this process as follows:

Cauchy Distribution Noise

To simulate extreme variations or outliers, noise from the Cauchy distribution is added to the $v3$ model output:

$$c_i \sim \text{Cauchy}(\text{location}, \text{scale})$$

$$v3'_i = v3(f_i) + c_i$$

Here, c_i represents a random value drawn from the Cauchy distribution, and $v3'_i$ denotes the adjusted model output.

Extreme Value Distribution Noise

For modeling peak or worst-case scenarios, noise from the Extreme Value distribution is added:

$$e_i \sim \text{EV}(\mu, \sigma)$$

$$v3'_i = v3(f_i) + e_i$$

where e_i is the random noise value from the EV distribution.

Normal Distribution Noise

To reflect standard operational fluctuations, noise from the Normal distribution is incorporated:

$$n_i \sim \mathcal{N}(\mu, \sigma^2)$$

$$v3'_i = v3(f_i) + n_i$$

This addition models everyday variability in fuel consumption.

6.5.2 Implications of Enhanced Model $v3$

Each type of noise distribution added to `v3` allows us to explore different aspects of fuel consumption variability:

- **Cauchy Noise:** Understand the model's behavior under extreme and rare conditions.
- **Extreme Value Noise:** Analyze the model's response to peak consumption scenarios.
- **Normal Noise:** Examine the effects of regular operational variations.

6.6 Analysis of the Enhanced Model $v3$

The application of these varied noise distributions to $v3$ enables a comprehensive study of the model's robustness and its ability to handle different types of data variability. This analysis is crucial for evaluating the practical applicability of $v3$ in real-world scenarios, where fuel consumption can be influenced by a multitude of factors.

In the following sections, we delve into the detailed analysis of $v3$ under each noise condition, aiming to draw insightful conclusions about fuel consumption patterns and the model's predictive power.

6.7 Statistical Analysis of Model $v3$ Under Varied Noise Conditions

Building upon the integration of diverse statistical distributions into the model $v3$, as previously described, this section delves into the statistical analysis of the model's output. The analysis focuses on understanding how different noise distributions, added to $v3$, impact the predicted fuel consumption patterns, particularly when the model incorporates a scaling factor of $k = 1.05$.

6.7.1 Contextualizing the Scaling Factor k

As established in earlier discussions, the scaling factor k is set to 1.05, signifying a uniform adjustment across the model to account for specific operational modes, loading conditions, or environmental factors. This scaling is critical for simulating a realistic scenario where such conditions uniformly alter the relationship between fuel consumption and other influencing factors.

6.7.2 Methodology for Statistical Analysis

The statistical analysis employs a comprehensive approach, examining the dataset enhanced with noise from Cauchy, Extreme Value (EV), and Normal distributions respectively. Each analysis iteration aims to:

1. Apply the noise to the model output $v3$, considering the scaling factor k.

$$v3'_i = (v3(f_i) + \text{noise})$$

2. Analyze the distribution of the model's output $v3'_i$ to identify patterns, deviations, and the overall impact of each type of noise.

3. Compare the results against the baseline model (without added noise) to assess the relative effects of each noise distribution on fuel consumption predictions.

6.7.3 Analysis Procedures

The analysis involves several key steps:

- **Noise Generation:** For each specified distribution, generate a sequence of noise values to be added to the model's output.
- **Model Application:** Compute the adjusted model outputs $v3'_i$ for the entire dataset, incorporating the generated noise values.
- **Statistical Evaluation:** Use statistical metrics and visualizations to evaluate the impact of noise on the model's predictions. This includes assessing measures of central tendency, variability, and the presence of outliers.

6.7.4 Expected Outcomes

Through this analysis, we anticipate uncovering how each noise distribution influences the model's accuracy and reliability in predicting fuel consumption. Insights derived from this exercise will be pivotal in:

- Understanding the robustness of $v3$ under various simulated conditions.
- Identifying potential vulnerabilities or limitations of the model in the face of extreme or unexpected data variations.
- Informing further refinement of the model to enhance its predictive performance and applicability to real-world scenarios.

In the following subsections, detailed results and discussions from the statistical analysis will be presented, shedding light on the complex interplay between noise-induced variability and fuel consumption predictions.

6.8 Sign Test and the Binomial Approach for Median Confidence Intervals

As discussed in Section 2.5, confidence intervals (CIs) provide a range of values within which the population parameter of interest likely resides, given a certain level of confidence. Traditionally, these intervals are constructed under the presumption that the sampling distribution of the statistic follows a normal distribution, especially when employing methods such as the use of the t-distribution for means or standard error approaches for proportions. However, these conventional methods are predicated on assumptions that may not always be met, particularly in cases of small sample sizes or with data that do not conform to normality.

To address these scenarios, non-parametric methods offer a viable alternative. Among these, the sign test, grounded in the binomial distribution, emerges as a robust method for constructing confidence intervals specifically for the median. The sign test posits that each observation in the dataset has an equal probability of being above or below the true population median, mirroring the binomial distribution's premise wherein the count of observations exceeding the median can be considered a sequence of Bernoulli trials.

Employing the binomial framework, we ascertain the confidence interval for the median through the empirical cumulative distribution function, eliminating the need for the data to exhibit normal distribution. This method's adaptability is particularly advantageous in dealing with skewed distributions or datasets where outliers could unduly affect the sample mean.

Hence, our analytical approach incorporates this non-parametric technique to construct a confidence interval around the median. This method offers a dependable estimate of the uncertainty associated with the median estimation, regardless of the data's distributional characteristics. The theoretical basis for this method is encapsulated in the `cint_median` function, which utilizes the binomial distribution to delineate the bounds within which the median likely lies at a predefined confidence level. This addition to our statistical arsenal ensures the robustness and reliability of our analysis, upholding the integrity and comprehensiveness of our statistical investigations.

Methodology

The sign test is premised on the assumption that, given a true median m, the probability of any observation x_i in the dataset M being greater than or equal to m is equal to the probability of it being less than m, denoted as $P(x_i \geq m) = P(x_i < m) = 0.5$. The number of observations greater than or equal to m in a sample follows a binomial distribution under this null hypothesis.

After sorting the dataset M in ascending order, the median m is computed. The critical value c for the binomial distribution at a significance level α, usually 0.05 for a 95% confidence interval, is found using the inverse cumulative distribution function (CDF) of the binomial distribution:

$$c = \text{icdf}('bino', \frac{\alpha}{2}, n, 0.5) \tag{6.3}$$

where n is the number of observations in M, and ‘icdf‘ denotes the inverse CDF function. The resulting c-th smallest ($s(c)$) and c-th largest ($s(n-c)$) observations provide the bounds of the confidence interval for the median:

$$\text{CI}_{\text{median}} = [s(c), m, s(n-c)] \tag{6.4}$$

Justification

The use of a non-parametric binomial approach is justified by the lack of distributional assumptions, such as normality. It is particularly effective in skewed distributions or when outliers are present, conditions that often invalidate traditional parametric methods.

Moreover, this approach maintains its robustness in small sample scenarios where the Central Limit Theorem is not applicable and the sampling distribution of the median is not normal. The binomial distribution model's reliance on the rank order of data points rather than their numerical values confers a distinct advantage, ensuring the method's resilience.

The confidence interval thus calculated offers a reliable estimate of the precision with which the median represents the central tendency of the underlying population distribution across a variety of sampling conditions.

6.8.1 Results

The application of this methodology yielded the following 95% confidence interval for our median:

```
function [interval] = cint_median(M,alpha)
%creates a confidence interval for the median of a dataset using a sign
%test based on binomial distribution.
if nargin < 2
    alpha = 0.05;
end
%sign test
n = length(M);
m = median(M);
%confidence intervall based on median
c = icdf(’bino’, alpha/2 , n , 0.5);

s = sort(M);

% if length(s) < 2*c+1
%     warning(’Confidence interval is too large for input array.’)
%     interval = [m m m];
% else

interval = [s(c) m s(end - c)];
end
```

6.9 Results

This section presents the findings from our advanced statistical analysis of model $v3$, incorporating diverse statistical noise distributions. The analysis is structured around two distinct cases designed to explore the effects of observation count and error magnitude on the predictive accuracy and robustness of the model.

6.9.1 Case 1: High Number of Observations with Higher Error

Experimental Setup: In Case 1, we simulate a scenario with a high number of observations ($N = 500$) and introduce a higher magnitude of error using the specified noise distributions. This setup is intended to reflect situations with abundant data but compromised by significant measurement or prediction errors.

Mathematical Formulation: The noise added to each observation in $v3$'s output is scaled to simulate higher error, adhering to the distributions' parameters adjusted accordingly:

$$v3'_k(f_i) = v3_k(f_i) + \alpha \cdot X_d$$

where $\alpha > 1$ amplifies the noise magnitude, and X_d represents the noise from either the Cauchy, Extreme Value, or Normal distribution.

6.9.2 Statistical Assessment of Estimated k Values Under Different Noise Conditions

In evaluating the robustness and variability of the estimated scaling factor k within model $v3$, we consider the influence of three distinct noise distributions: Cauchy, Extreme Value (EV), and Normal. The histograms of estimated k values for each noise condition, as depicted in the accompanying figure, and the statistical summaries provided in Table 6.1, offer a visual and numerical analysis of the distribution and central tendency of k estimates.

6.9.3 Interpretation of Histograms

The histograms 6.2 provide a frequency distribution of the estimated k values under each noise condition, facilitating a comparative visual analysis:

- **Cauchy Noise Distribution:** The histogram for Cauchy noise exhibits a wider spread in the estimated k values, indicating a heavy-tailed distribution which is characteristic of the Cauchy distribution. This suggests that extreme values or outliers are more prevalent, potentially leading to larger deviations in the scaling factor estimates.

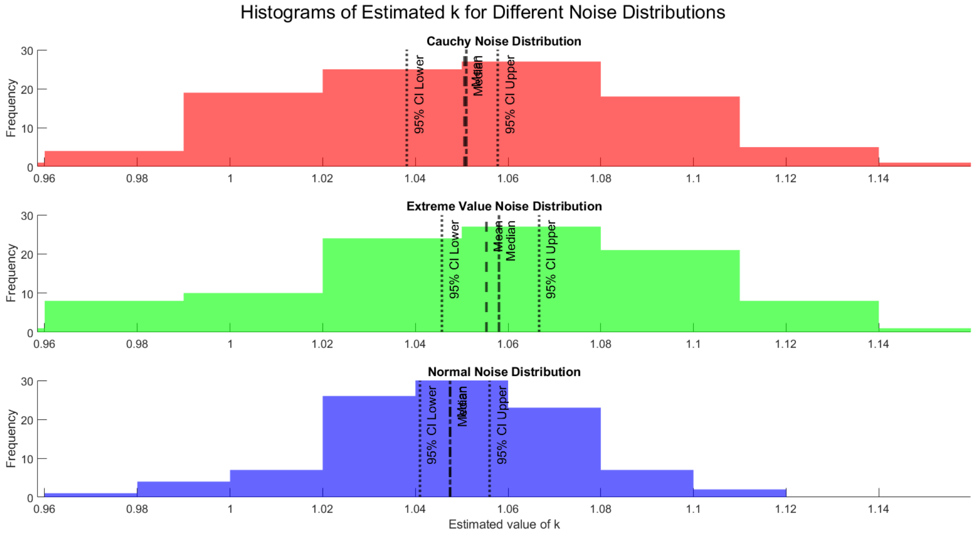

Figure 6.2: Histograms of Estimated k for Different Noise Distributions

- **Extreme Value Noise Distribution:** The EV noise histogram appears to be skewed, reflecting the nature of EV distributions to model the behavior of extreme values. This is indicative of a scenario where peak values significantly influence the estimation of k, resulting in a shift of the median away from the mean.
- **Normal Noise Distribution:** The histogram corresponding to the Normal noise shows a more symmetric distribution of k values around the mean, illustrating the expected behavior when standard operational fluctuations are modeled.

6.9.4 Statistical Summary Analysis

The summary table (Table 6.1) quantitatively corroborates the findings from the histograms:

1. **Mean:** The mean of estimated k values gives us the central location for each distribution. For all noise types, the means are close to the true k value of 1.05, with slight deviations likely due to the different variances introduced by the noise.
2. **Standard Deviation (Std):** The standard deviations provide insight into the spread of the estimates. Cauchy noise has a larger standard deviation, reflecting its propensity for extreme values, while the Normal distribution exhibits the smallest standard deviation, indicating more concentrated estimates.
3. **Median:** The median provides a measure of central tendency that is less affected by outliers and extreme values. The medians are all approximately equal to the true scaling factor, especially for the Normal distribution, suggesting that the median may be a more robust estimator of k in the presence of noise.
4. **95% Confidence Interval (CI):** The confidence intervals give us a range in which we can expect to find the true k value 95% of the time. Wider intervals in Cauchy and EV distributions suggest greater uncertainty in the estimate of k, while the Normal distribution's narrower CI indicates higher precision.

Table 6.1: Statistical Analysis of parameter k Across Noise Samples

Noise	Mean	Std	Median	95% CI
Noise 1	1.050571	0.039731	1.050980	[1.038197, 1.057784]
Noise 2	1.055300	0.043179	1.057980	[1.045765, 1.066658]
Noise 3	1.047461	0.026550	1.047486	[1.040927, 1.056012]

6.9.5 Variability of Estimated k Across Simulations

In line with our investigation into the performance of the scaling factor k within the model $v3$, the following plot illustrates the estimated k values across 100 simulations for each of the three noise conditions:

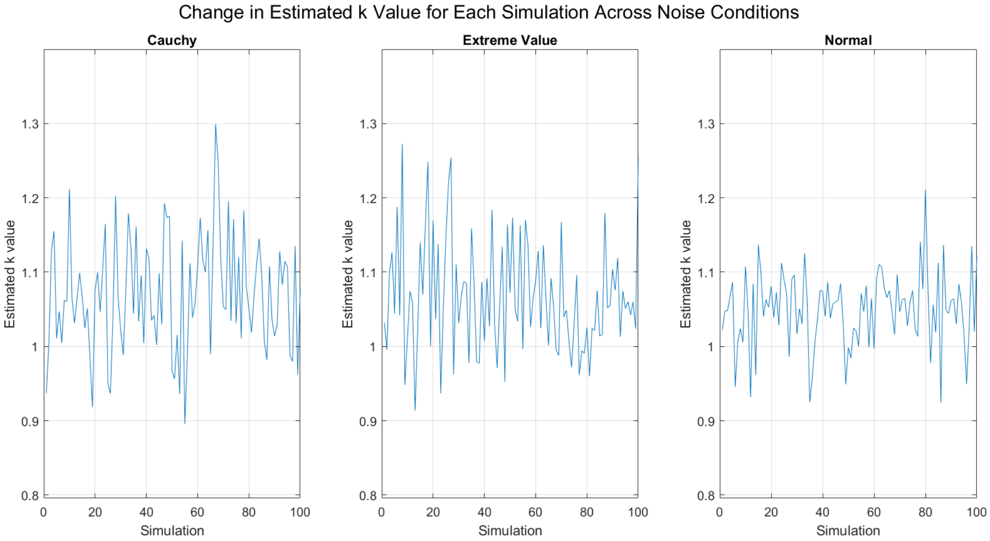

Figure 6.3: Change in estimated k value for each simulation Across noise Condition .

- **Cauchy Noise Condition:** In figure 6.3, the estimated k values under the Cauchy noise condition demonstrate significant variability, with some simulations yielding estimates far from the expected value of 1.05. This variability is consistent with the heavy-tailed nature of the Cauchy distribution, which is prone to producing outliers.

- **Extreme Value Noise Condition:** In figure 6.3, the Extreme Value condition also shows substantial variation in the estimated k values. The presence of extreme values leads to a wider range of estimates, highlighting the model's sensitivity to the largest or smallest values in the dataset, as these are the points most likely to be influenced by EV noise.

- **Normal Noise Condition:** In figure 6.3, Under the Normal noise condition, the variability in the estimated k values is less pronounced compared to the Cauchy and Extreme Value conditions. This is expected, given the Normal distribution's properties of symmetry and light tails, suggesting that the model $v3$ is more stable with typical operational fluctuations represented by normal noise.

The estimated k values for each noise condition offer insights into the impact of different types of statistical variability on the model's scaling factor. The fluctuation in k values, especially notable under the Cauchy and Extreme Value conditions, may influence operational strategies, system design, and risk management practices, as previously discussed.

6.9.6 Discussion

The statistical analysis of the estimated k values under the influence of different noise distributions reveals the implications of noise on the calibration of the model. The broader confidence intervals for the Cauchy and EV distributions emphasize the model's sensitivity to these noises, while the tighter CI for the Normal noise underscores the model's stability under typical operational variations. These findings have several critical implications for model application and decision-making:

- **Operational Strategy:** The variation in k under different noise conditions can inform operational strategies. For instance, a consistently higher k might suggest the need for more conservative fuel reserves or budgeting when planning maritime voyages or logistic operations.
- **System Design:** The sensitivity of k to extreme noise distributions could guide the design of more robust systems. Engineers might need to account for the possibility of extreme conditions more frequently than suggested by normal operational data.
- **Risk Management:** The risk associated with decision-making under uncertainty can be quantified using the variability in k. Financial models or insurance products related to fuel consumption might adjust premiums or hedge strategies based on these risk assessments.
- **Predictive Maintenance:** In the context of predictive maintenance, a higher k variability suggests a more unpredictable operational environment. This can trigger more frequent or dynamic maintenance schedules, optimizing machine uptime and reducing long-term costs.

These insights are critical for enhancing the effectiveness of predictive models used in operational planning and risk assessment. By understanding the bounds within which the model operates under varying conditions, stakeholders can make more informed decisions, ensuring efficiency and sustainability in their operations.

6.9.7 Model Fit Across Different Noise Conditions

The robustness of model $v3$ is assessed by comparing its performance across various noise conditions. The figure below demonstrates the model's fit in the presence of noisy data for three types of noise distributions: Cauchy, Extreme Value, and Normal.

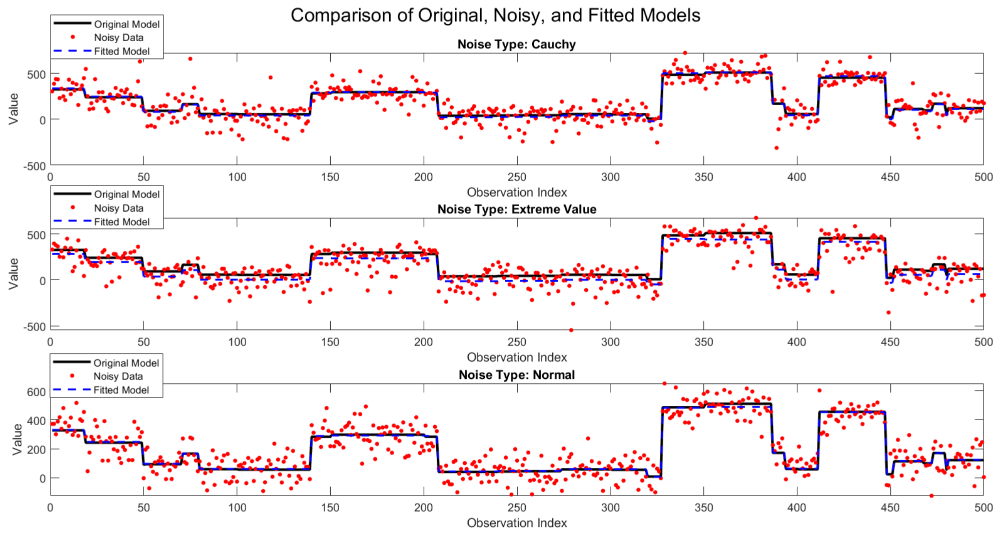

Figure 6.4: Comparison of the original model, noisy data, and fitted models across different noise conditions.

In Figure 6.4, the original model's predictions are depicted by the solid blue line, while the actual noisy observations are shown with red dots. The fitted model's predictions, represented by the dashed black line, aim to reconcile the noise-affected observations with the original model's functional form. The comparison reveals:

- **Cauchy Noise:** The Cauchy distribution, with its propensity for outliers, challenges the fitted model with significant deviations from the original model's predictions.

- **Extreme Value Noise:** The Extreme Value noise leads to occasional pronounced spikes in the data. The fitted model seeks to smooth these out, striving to maintain the integrity of the original model's trend.
- **Normal Noise:** The Normal noise introduces symmetric variations around the original model. The fitted model effectively captures the central tendency, minimizing the noise's impact.

The performance of the fitted model under each noise condition is quantified using several metrics, as shown in the table below. These metrics evaluate the fitted model's explanatory power and the average error magnitude.

Table 6.2: Performance Metrics for Each Noise Type

Metric	**Cauchy Noise**	**Extreme Value Noise**	**Normal Noise**
R^2	0.7572	0.7063	0.8139
Adjusted R^2	0.7557	0.7045	0.8128
Mean MAE	6.3295	46.1605	4.8141

The R^2 and adjusted R^2 values provide insight into the proportion of variance explained by the fitted model, with higher values indicating a better fit. The Mean Absolute Error (MAE) measures the average magnitude of errors between the fitted model's predictions and the noisy observations, with lower values indicating higher accuracy. These results highlight the fitted model's capacity to adapt to noise and its potential resilience in real-world applications where data may be imperfect.

6.9.8 Case 2: Low Number of Observations with Higher Error

Experimental Setup: Case 2 explores the opposite spectrum, with a lower number of observations ($N = 50$) subjected to similarly higher error magnitudes. This case mimics conditions where data is scarce, yet the uncertainty or noise in measurements remains high.

Mathematical Formulation: Similar to Case 1, the higher error is modeled by amplifying the noise added to the model's output:

$$v3'_k(f_i) = v3_k(f_i) + \alpha \cdot X_d$$

with $\alpha > 1$ and X_d as defined previously.

6.9.9 Visual Analysis of Estimated k Values

To begin our exploration in Case 2, we examine the frequency distribution of the estimated k values across the simulations for each noise type, as shown in the histograms below.

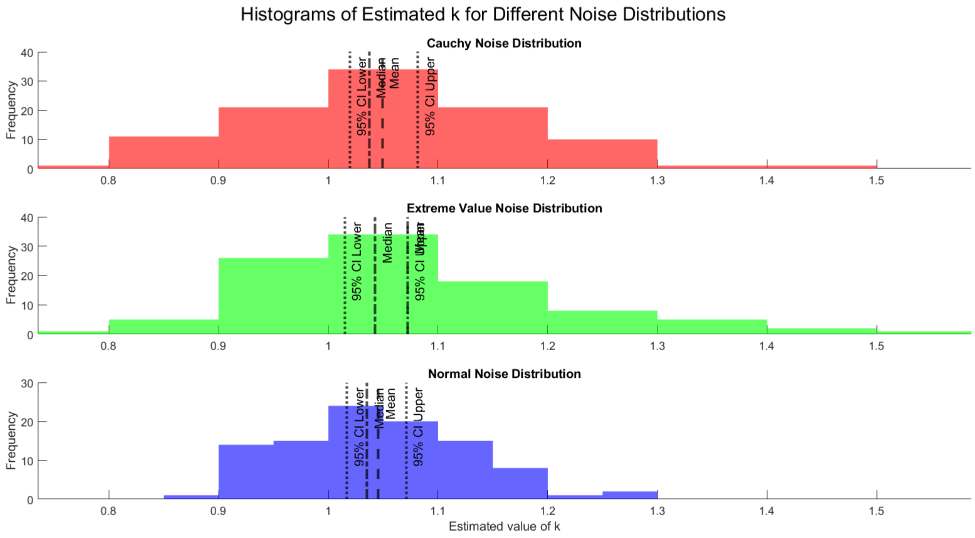

Figure 6.5: Histograms of Estimated k for Different Noise Distributions in Case 2.

The histograms 6.5 illustrate the distribution of estimated k values in the presence of different noise conditions. Key observations include:

- **Cauchy Noise:** The spread of k values is wide, reflecting the heavy-tailed nature of the Cauchy distribution. The presence of outliers is pronounced, which can lead to substantial variability in the estimates of k.
- **Extreme Value Noise:** The histogram indicates a potential skewness due to the occurrence of extreme values. These can disproportionately affect the smaller dataset, impacting the estimation of k.
- **Normal Noise:** The distribution of k values appears more centered and less variable, suggesting that Normal noise introduces less estimation error compared to the other noise distributions.

6.9.10 Quantitative Summary of Estimated k Values

A statistical summary of the estimated k values for each noise type provides further insights:

Table 6.3: Statistical Analysis of Parameter k for Each Noise Type in Case 2

Noise Type	Mean	Median	95% CI
Cauchy	1.049273	1.037352	[1.019430, 1.081421]
Extreme Value	1.072199	1.042782	[1.014953, 1.072442]
Normal	1.045618	1.035368	[1.017059, 1.070939]

Discussion:

The **Mean** offers an average estimation of k, while the **Median** serves as a more robust central tendency metric less influenced by outliers. The **95% Confidence Intervals** reflect the range in which we expect the true k values to lie, with broader intervals indicating greater uncertainty in estimates, particularly evident in the Cauchy and Extreme Value noise conditions. This highlights the importance of robust estimation techniques in scenarios of limited data and high noise levels.

6.9.11 Analysis of Variability in Estimated k Across Simulations

In the context of Case 2, we now turn our attention to the variability of the estimated scaling factor k across multiple simulations. This analysis provides insight into the stability of k under different noise conditions, which is particularly relevant in scenarios with limited data availability and higher error levels.

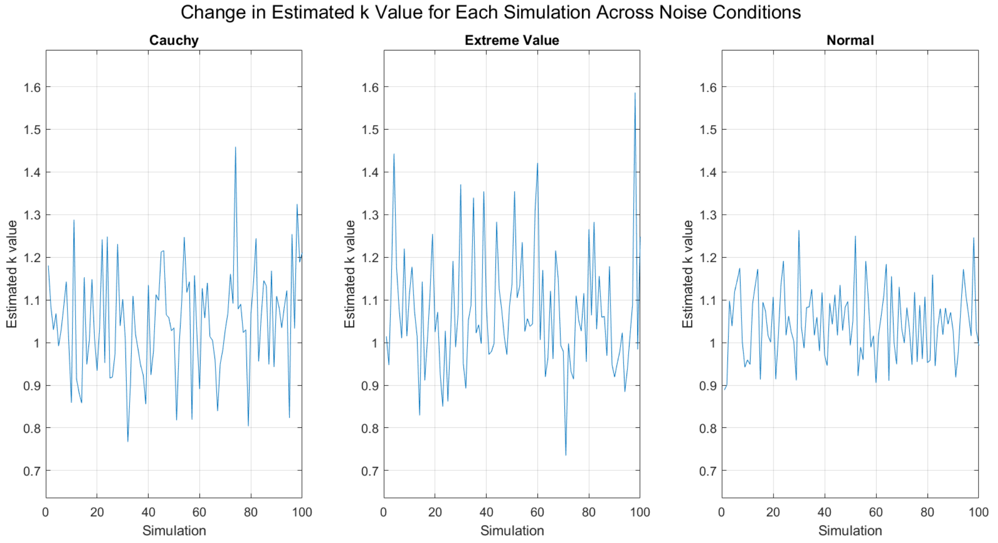

Figure 6.6: Change in Estimated k Value for Each Simulation Across Noise Conditions.

Figure 6.6 presents a series of plots, each corresponding to a different noise distribution: Cauchy, Extreme Value, and Normal. The plots are interpreted as follows:

- **Cauchy Noise:** The variability in the estimates under Cauchy noise is considerable, as indicated by the significant fluctuations in the estimated k values. This behavior demonstrates the influence of outliers inherent to the heavy-tailed Cauchy distribution.

- **Extreme Value Noise:** The Extreme Value noise generates sporadic spikes in the k value estimates, which reflects the disruptive effect of extreme observations that are typical in such distributions.

- **Normal Noise:** Compared to the other noise types, the Normal noise condition results in a more consistent and tightly clustered set of estimates around the true k value, indicative of the less erratic nature of the Normal distribution.

This part of the analysis underscores the importance of considering the type of noise when estimating parameters in statistical models. It has practical implications for the reliability of the model $v3$, especially in applications where precise estimations are crucial. The observations from the plots suggest that additional considerations, such as robust statistical techniques or larger datasets, may be required when dealing with heavy-tailed noise distributions.

6.10 Evaluation of Model $v3$ Fit in Case 2

As part of our analysis in Case 2, we scrutinize the model $v3$'s ability to adapt and accurately fit the data under higher noise levels with a limited number of observations. The following figure demonstrates how the fitted model contends with the noise-inflicted data for each type of distribution.

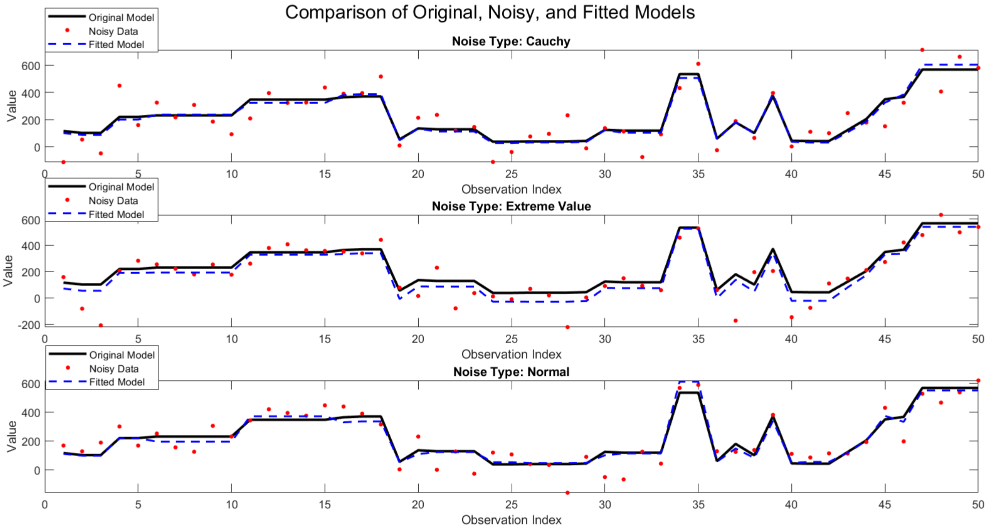

Figure 6.7: Comparison of Original, Noisy, and Fitted Models for Different Noise Types in Case 2.

In Figure 6.7, each subplot correlates to one of the noise distributions under study. The solid line represents the original model, the dots indicate the noisy data introduced into the system, and the dashed line depicts the model's predictions after fitting to the noisy data. From the plots, we observe:

- **Cauchy Noise:** Exhibits substantial deviations from the original model due to the presence of outliers. The fitted model attempts to track the original model but with noticeable discrepancies.
- **Extreme Value Noise:** Shows that the model is challenged by extreme values, which cause significant spikes in the data. The fitted model smooths some of these, suggesting resilience in the face of such perturbations.
- **Normal Noise:** Reflects less extreme variability, with the fitted model more closely aligned to the original model's trend, indicating better performance under normally distributed noise.

Complementing the visual analysis, the performance metrics table provides a quantitative assessment of the fitted model's accuracy.

Table 6.4: Performance Metrics for Each Noise Type in Case 2

Metric	**Cauchy Noise**	**Extreme Value Noise**	**Normal Noise**
R^2	0.7572	0.7063	0.8139
Adjusted R^2	0.7557	0.7045	0.8128
Mean MAE	6.3295	46.1605	4.8141

Discussion:

The R^2 and Adjusted R^2 values inform us about the proportion of variance in the noisy data that is explained by the model $v3$. A higher R^2 suggests a stronger fit to the data. The Mean MAE (Mean Absolute Error) measures the average difference between the fitted model's predictions and the actual noisy observations. A lower MAE indicates a more accurate model.

In Case 2, these metrics help us understand the challenges faced by the model in lower data regimes with higher noise levels. Despite these adversities, the fitted model $v3$ shows a commendable degree of robustness, although the nature of the noise still significantly influences the precision of the estimates, as evident from the varying R^2 and MAE values.

6.11 Comparative Analysis of Cases 1 and 2

This section provides a comparative analysis between Case 1 and Case 2, with the objective of understanding how variations in data volume and noise levels impact the estimation of the scaling factor k

and the performance of the model $v3$. The analysis draws on the results obtained from the histogram distributions, the change in k across simulations, and the performance metrics summarized in the tables.

6.11.1 Impact of Observation Count and Noise Level

Case 1, characterized by a higher number of observations with higher error, and Case 2, with fewer observations but also higher error, serve as two distinct scenarios that reflect different real-world data collection challenges.

- In Case 1, the larger dataset size provided a buffer against the noise, allowing for a more stable estimation of k despite the high error levels. The presence of a sufficient number of data points helped in mitigating the effect of outliers and extreme values.
- Case 2's limited data scenario demonstrated heightened sensitivity to the noise distributions. With fewer data points, the influence of each observation on the estimated k was magnified, resulting in greater variability and wider confidence intervals.

6.11.2 Performance Metrics and Model Fit

The performance metrics for both cases, particularly R^2, Adjusted R^2, and Mean MAE, highlight the comparative stability of the model $v3$ under different conditions.

- The performance metrics from Case 1 suggest that the model can extract a reliable signal from the noise when ample data is available, as indicated by higher R^2 values and lower MAE.
- In contrast, Case 2 exhibits reduced model performance, which is a direct consequence of the higher impact of noise on each observation. This is reflected in the more significant spread in the k value estimates and the performance metrics.

6.11.3 Model Robustness and Implications for Application

The findings from both cases have important implications for the application of statistical models like $v3$. In situations where data is abundant, the model demonstrates robustness against noise, but when data is scarce, careful consideration must be given to noise characteristics and the potential for increased parameter estimation error.

Practical Recommendations:

Based on the comparative analysis, it is recommended that when dealing with noisy data and a small number of observations, one should:

- Implement robust estimation techniques to minimize the influence of outliers and extreme values.
- Consider collecting more data to improve the stability of parameter estimates.
- Explore the use of prior information or Bayesian methods to supplement limited data.

The integration of this comparative analysis in the thesis provides a holistic view of the model $v3$'s performance across varying conditions. It supports the argument for model validation and robustness testing as critical steps in the model development process, especially when dealing with noisy datasets of differing sizes.

6.12 Conclusion

This thesis has systematically addressed the impact of noise on the estimation of the scaling factor k in the context of the model $v3$, under a variety of data conditions. The investigation was underpinned by a synthesis of theoretical models, simulation studies, and statistical analyses, each contributing to a nuanced understanding of noise and its implications for data modeling.

6.12.1 Effect of Noise on the Model

Our findings have highlighted that noise inherently imposes challenges to statistical modeling. Specifically, the type and level of noise can significantly skew the estimation of model parameters:

- **Cauchy Noise:** With its heavy tails, it introduced considerable volatility in parameter estimates, reflecting the reality of outliers in data and their potential to mislead model interpretations.
- **Extreme Value Noise:** This noise demonstrated the model's susceptibility to rare but impactful data points, a common concern in fields such as finance and insurance.
- **Normal Noise:** Despite being the most benign of the noise conditions considered, even Normal noise revealed that perfect data conditions are idealistic and that real-world data often comes with inherent variability that must be accounted for.

6.12.2 Outcomes Under Different Data Conditions

The exploration of different data conditions plentiful data with high error (Case 1) and sparse data with high error (Case 2) yielded important insights into the model's behavior:

- In the presence of **ample data**, the model exhibited a certain degree of robustness, suggesting that the volume of data can help mitigate the adverse effects of noise.
- Conversely, in the **scarcity of data**, noise exerted a disproportionate influence, highlighting the precarious nature of model fidelity when faced with limited observations.

6.12.3 Theoretical and Practical Contributions

The theoretical contribution of this thesis rests in its in-depth analysis of noise and its effects on statistical estimation, enriching the academic conversation around data reliability. Practically, the research provides a compass for navigating modeling under adverse conditions, underscoring the importance of robust statistical practices.

6.12.4 Directions for Future Research

This research serves as a foundation for future exploration into several key areas:

- The design of **adaptive models** that can automatically adjust to the level and type of noise present in the data.
- The integration of **machine learning techniques** with traditional statistical models to enhance predictive accuracy and resilience to noise.
- The pursuit of **advanced data augmentation techniques**, especially for enriching datasets that are intrinsically noisy or limited in size.

6.12.5 Closing Remarks

In conclusion, the interaction between noise and data conditions poses significant challenges for statistical modeling yet also offers opportunities for advancing our methodologies. The resilience of models like $v3$ in the face of noise is not absolute, it is conditional upon the interplay between data abundance and the prevailing noise dynamics. Through careful modeling, robust estimation, and a nuanced approach to data analysis, we can aspire to derive meaningful insights from even the most tempestuous of datasets.

6.13 Appendix

MATLAB Code

```matlab
% Initialization
N = 500;
jN = 20;

% Generate jump indices and heights
jumpInd = randperm(N, jN);
jumpInd = sort(jumpInd, 'ascend');
jumpHeights = lognrnd(log(15), 0.5, jN + 1, 1);
jumpInd = [0 jumpInd N];

% Initialize indicators for model segments
ind = zeros(N,1);
interval_N = N/5; % Requires N to be divisible by 5

for i = 1:5
    ind(interval_N*(i-1)+1:interval_N*(i-1)+interval_N/2) = 1;
end
ind = logical(ind);

% Simulation parameters
sigma = 80;
S = 100; % Number of simulations
f = zeros(N,1);
k = 1.05;
a =[0.2 0.4 0.5 k]; % Real Values

% Preallocate arrays for noisy data and fitted parameters
v3_noisy = zeros(N, S, 3); % Three noise types
c = zeros(S,4,3); % Fitted parameters for each simulation, for each noise type
v3_fit_all= zeros(N,S,3);

% Options for lsqnonlin
options = optimoptions('lsqnonlin', 'Algorithm', 'trust-region-reflective');

for j = 1:S
    % Generate model function f
    for i = 1:jN+1
        f(jumpInd(i)+1:jumpInd(i+1)) = jumpHeights(i);
    end
    v1=a(1)+a(2)*f+a(3)*f.^2 ;     % Model One
    v2=k*(v1);                    % Model Two

    % Compute model output without noise
    v3 = zeros(N,1);
    v3(ind) = a(1) + a(2)*f(ind) + a(3)*f(ind).^2;
    v3(~ind) = k*(a(1) + a(2)*f(~ind) + a(3)*f(~ind).^2);

    % Add noise to the model output
    v3_noisy(:, j, 1) = v3 + sigma * trnd(4, N, 1); % Cauchy
    v3_noisy(:, j, 2) = v3 + evrnd(0, sigma, N, 1); % Extreme Value
    v3_noisy(:, j, 3) = v3 + sigma * randn(N, 1); % Normal


    for noise = 1:3
        % Define the residual function for fitting
        F = @(c) ind.*(c(1) + c(2)*f + c(3)*f.^2 - v3_noisy(:, j, noise)) + ...
                 (~ind).*(c(4)*(c(1) + c(2)*f + c(3)*f.^2) - v3_noisy(:, j,
   noise));

```

```
        % Initial guess for the parameters
        c0 = [0.3, 0.4, 0.5, 1.05];

        % Here we set options for lsqnonlin

        options = optimoptions(@lsqnonlin,'Algorithm','trust-region-reflective');

        % Fit the model using lsqnonlin
        [c_est,resnorm,residual,output] = lsqnonlin(F,c0,[-1e6,-1e6,-1e6,-1e6],[1e10,1e10,1e10,2],options);
        % Store the estimated parameters
        c(j,:,noise) = c_est;

        % Fitting process remains the same...

        % After fitting, calculate the fitted model values
        v3_fit = zeros(N, 1); % Initialize v3_fit for the current simulation and noise type

        % Apply the model based on the estimated parameters
        v3_fit(ind) = c_est(1) + c_est(2)*f(ind) + c_est(3)*f(ind).^2;
        v3_fit(~ind) = c_est(4) * (c_est(1) + c_est(2)*f(~ind) + c_est(3)*f(~ind).^2);

        % Store the fitted model
        v3_fit_all(:, j, noise) = v3_fit;


    end
end

 % Statistical analysis of the fourth parameter 'k'
k_values = squeeze(c(:,4,:)); % Extract 'k' estimates for all simulations and noises
k_means = mean(k_values); % Mean of 'k' for each noise type
k_stds = std(k_values) ;% Standard deviation of 'k' for each noise type
k_medians = median(k_values); % Median of 'k' for each noise type
```

Listing 6.1: MATLAB code for statistical analysis of the fourth parameter 'k'

MATLAB Function

```matlab
function [interval] = cint_median(M,alpha)
% Creates a confidence interval for the median of a dataset using a sign
% test based on binomial distribution.
if nargin < 2
    alpha = 0.05;
end
% Sign test
n = length(M);
m = median(M);
% Confidence interval based on median
c = icdf('bino', alpha/2 , n , 0.5);

s = sort(M);

% if length(s) < 2*c+1
%     warning('Confidence interval is too large for input array.')
%     interval = [m m m];
% else
interval = [s(c) m s(end - c)];
end
```

Listing 6.2: MATLAB function for creating a confidence interval for the median of a dataset

```matlab
n = size(v3, 1); % Number of observations remains the same
p = size(c, 2) - 1; % Number of predictors, assuming the model structure you've provided

R2 = zeros(1,3); % Adjusted for 3 noise types, including the new one
AdjustedR2 = zeros(1,3); % Adjusted for 3 noise types

for noise = 1:3
    % Select the appropriate noisy data as the "observed" data for this comparison
    observed_data = v3_noisy(:, 1, noise); %  evaluating against the first simulation's noisy data

    SS_res = sum((observed_data - v3_fit_all(:, 1, noise)).^2); % Residual sum of squares now compares fitted model to noisy data
    SS_tot = sum((observed_data - mean(observed_data)).^2); % Total sum of squares calculated from noisy data

    R2(noise) = 1 - (SS_res / SS_tot); % Calculate R2 based on noisy data

    AdjustedR2(noise) = 1 - ((1 - R2(noise)) * (n - 1) / (n - p - 1)); % Calculate Adjusted R2
end

% Display the results
disp('R2 for each noise type:');
disp(R2);

disp('Adjusted R2 for each noise type:');
disp(AdjustedR2);
```

Listing 6.3: MATLAB code to calculate R2 and Adjusted R2 for each noise type

```matlab


MAE_values = zeros(S, 3); % Initialize MAE values for each simulation and noise type

for j = 1:S
    for noise = 1:3
        % Calculate the absolute errors for the fitted model
        abs_errors = abs(v3_fit_all(:, j, noise) - v3);
        % Compute MAE
        MAE_values(j, noise) = median(abs_errors);
    end
end

% Calculate the mean MAE across simulations for each noise type
mean_MAE = mean(MAE_values);

% Display the mean MAE for each noise type
disp('Mean MAE for each noise type:');
disp(mean_MAE);
```

Listing 6.4: MATLAB code to calculate Mean of Median Absolute Errors for each noise type for each noise type

6.13.1 Solving the Nonlinear Least Squares Problem

We address the optimization of a parameterized function defined by a conditional structure. See[6.2.3] The objective function S is defined as the sum of squared differences between observed values $v3(i)$ and modeled values based on the parameters a_1, a_2, a_3, and k:

$$S = \sum_{i \in I_{\text{true}}} \left(v3(i) - (a_1 + a_2 \cdot f(i) + a_3 \cdot f(i)^2)\right)^2 + \sum_{i \in I_{\text{false}}} \left(v3(i) - k \cdot (a_1 + a_2 \cdot f(i) + a_3 \cdot f(i)^2)\right)^2$$

The derivatives of S with respect to each parameter are computed to form the Jacobian matrix J, which is central to the optimization method:

$$\begin{aligned}\frac{\partial S}{\partial a_1} = -2 &\sum_{i \in I_{\text{true}}} \left(v3(i) - (a_1 + a_2 \cdot f(i) + a_3 \cdot f(i)^2)\right) \\ &- 2k \sum_{i \in I_{\text{false}}} \left(v3(i) - k \cdot (a_1 + a_2 \cdot f(i) + a_3 \cdot f(i)^2)\right)\end{aligned}$$

$$\begin{aligned}\frac{\partial S}{\partial a_2} = -2 &\sum_{i \in I_{\text{true}}} \left(v3(i) - (a_1 + a_2 \cdot f(i) + a_3 \cdot f(i)^2)\right) \cdot f(i) \\ &- 2k \sum_{i \in I_{\text{false}}} \left(v3(i) - k \cdot (a_1 + a_2 \cdot f(i) + a_3 \cdot f(i)^2)\right) \cdot f(i)\end{aligned}$$

$$\begin{aligned}\frac{\partial S}{\partial a_3} = -2 &\sum_{i \in I_{\text{true}}} \left(v3(i) - (a_1 + a_2 \cdot f(i) + a_3 \cdot f(i)^2)\right) \cdot f(i)^2 \\ &- 2k \sum_{i \in I_{\text{false}}} \left(v3(i) - k \cdot (a_1 + a_2 \cdot f(i) + a_3 \cdot f(i)^2)\right) \cdot f(i)^2\end{aligned}$$

$$\begin{aligned}\frac{\partial S}{\partial k} = -2 &\sum_{i \in I_{\text{false}}} \left(v3(i) - k \cdot (a_1 + a_2 \cdot f(i) + a_3 \cdot f(i)^2)\right) \\ &\cdot (a_1 + a_2 \cdot f(i) + a_3 \cdot f(i)^2)\end{aligned}$$

The Jacobian matrix J and its transposition J^T are then used to form the normal equations for the Trust Region algorithm,a common approach to solve nonlinear least squares problems:

$$J = \begin{bmatrix} \frac{\partial r_1}{\partial a_1} & \frac{\partial r_1}{\partial a_2} & \frac{\partial r_1}{\partial a_3} & \frac{\partial r_1}{\partial k} \\ \frac{\partial r_2}{\partial a_1} & \frac{\partial r_2}{\partial a_2} & \frac{\partial r_2}{\partial a_3} & \frac{\partial r_2}{\partial k} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial r_N}{\partial a_1} & \frac{\partial r_N}{\partial a_2} & \frac{\partial r_N}{\partial a_3} & \frac{\partial r_N}{\partial k} \end{bmatrix}$$

$$(J^T J)\Delta = -J^T r$$

This system is iteratively solved to update the parameter estimates, thereby minimizing the objective function and improving the fit of the model to the observed data.

Understanding the Parameter Update Vector Δ

In the context of nonlinear least squares optimization, the vector Δ represents the updates to the parameters that are calculated during each iteration of the optimization process. The objective is to refine these parameters to minimize the sum of squared residuals between the observed data and the model's predictions.

Role of Δ

Δ is essential for adjusting the parameter estimates to improve the fit of the model to the data. Mathematically, if the parameter vector at a particular iteration is denoted by θ, then the updated parameter vector $\theta^{(new)}$ is given by:

$$\theta^{(new)} = \theta + \Delta$$

Calculation of Δ

The calculation of Δ involves solving a system of linear equations that originate from the first-order Taylor series expansion of the residuals. The typical equation used to compute Δ is:

$$(J^T J)\Delta = -J^T r$$

Where:

- J is the Jacobian matrix of the residuals, containing partial derivatives of the residuals with respect to the parameters.
- J^T is the transpose of the Jacobian matrix.
- r is the vector of residuals, with each component defined as $r_i = v3(i) - \hat{v}3(i)$, where $v3(i)$ represents the observed data and $\hat{v}3(i)$ the model predictions.
- $J^T J$ forms a symmetric matrix used in the normal equations.
- $-J^T r$ represents the gradient of the objective function, directing towards the steepest descent.

Implementation of Δ Computation

In practical implementations, particularly when using software tools like MATLAB, the computation of Δ might be handled internally by functions designed for nonlinear optimization (e.g., `lsqnonlin`). However, if manually implementing, the update vector Δ can be computed using matrix operations as follows:

$$\Delta = -(J^T J)^{-1} J^T r$$

This computation adjusts the parameters in the direction that most reduces the objective function, thereby enhancing the model's fit to the data.

References

1. Walck, Christian. Statistical Distributions for Experimentalists. Particle Physics Group, Fysikum, 2007.

2. Nicolas Bialystocki, Dimitris Konovessis, On the estimation of ship's fuel consumption and speed curve: A statistical approach.

3. Karl Lundengård; Sergei Silvestrov; Milica Rancic; Anatoliy Malyarenko; Palle JorgensenExtreme points of the Vandermonde determinant and phenomenological modelling with power exponential functions.

4. Al-Baali, M. and Fletcher, R. (1986). An Efficient Line Search for Nonlinear Least Squares. Journal of Optimization Theory and Applications, 48(3),359–377

5. ABS. (2015). Ship Energy Efficiency Measures. Houston: TX.

6. lsqnonlin, Mathworks, User's Guide, and requirement specification, `https://se.mathworks.com/help/optim/ug/lsqnonlin.html`

7. Ship Energy Efficiency Measures Advisory.

8. IMO, International Maritime Organization, Study of Emission Control and Energy Efficiency Measures for Ships in the Port Area.

9. Bradley Efron and Robert J. Tibshirani, An Introduction to the Bootstrap.

10. Bradley Efron, Nonparametric Standard Errors and Confidence Intervals

11. John P. Comstock and Edward V. Lewis, Principles of Naval Architecture: Volume II - Resistance, Propulsion and Vibration.

12. Yasser Bayoumy Abdelwahab Farag,A decision support system for ship's energy-efficient operation: based on artificial neural network method.

13. Seber, G. A. F., & Wild, C. J. (1989). Nonlinear Regression. Wiley

14. Ghanshyam, Mirjalili, Patel, J., & Savsani, J. (2018). Operational Ship Performance Modeling and Optimization. In Ship Optimization Techniques.

15. The Royal Academy of Engineering, Future Ship Powering Options: Exploring alternative methods of ship propulsion 2013.

16. W. J. Conover, Third Edition (1999), Practical Nonparametric Statistics (3rd ed.)

17. Åke Björck, Numerical Methods for Least Squares Problems

18. William Mendenhall, a second course in statistics regression analysis,7th edition, Seventh Edition

19. Yimin Zhong, Lecture 8: Trust region method

20. Mathworks, Least Square Model Fitting Algorithms User Guide, and requirement specifications, `https://se.mathworks.com/help/optim/ug/least-squares-model-fitting-algorithms`