Master of Science in Data Science and Networking Intelligence

Promotion 2023/2024

Internship Report

Presented by Victor de Carvalho Silva

Internship conducted from 01/03/2024 to 30/09/2024 at Renault Group, carried out within the Customer Usage team, in the Ampere Cars division.

Topic: Data Science
Subtopic: Confidential Intervals

(Confidential)

2024. Évry-Courcouronnes, Île-de-France.

Abstract

This report presents the results of an internship conducted at the Ampere Cars division of Renault Group, focused on the analysis of vehicle usage data for component optimization. The primary objective was to develop and apply methods for calculating confidence intervals in real customer usage data and simulation data, using techniques such as bootstrap, jackknife, and regression. The internship included three case studies that addressed different aspects of vehicle behavior: the number of trips per vehicle per day, the impact of speed on severe conditions imposed on the automotive differential, and the average speed profile of customers.

The results showed that confidence intervals are essential for assessing variability in upper percentiles, especially in extreme usage scenarios such as the 99th percentile, where variability was significantly higher. The internship demonstrated that analyzing the variability in upper percentiles is particularly important in scenarios where usage is highly variable, as is often the case in real-world conditions.

The developed methodology allowed precise adjustments in car components and provided an in-depth understanding of vehicle behavior in real-world conditions. The work directly contributed to the development process of more efficient, reliable, and user-aligned vehicles, strengthening Renault's leadership in electric mobility.

The internship provided a valuable opportunity in a global automotive manufacturer and practical experience in applying data science and engineering techniques to real-world problems. The findings of the project were presented in workshops and shared with key stakeholders at Renault, fostering an ecosystem of collaboration and interaction among different departments.

Overall, the results of the project demonstrate the potential of advanced statistical techniques in analyzing real-world usage data from complex systems, providing insights into usage patterns that are difficult to predict through conventional testing. The proposed methodology can be applied to a variety of industries and contexts, contributing to improved processes, enhanced product efficiency, and performance.

Contents

List of Figures

1.INTRODUCTION

This internship was completed as part of the Master's program in Data Science at Telecom SudParis – Institut Polytechnique de Paris. It was carried out at the Renault Group, specifically within the Ampere Cars division, a unit dedicated to technological innovation and development within the company. Renault is one of the largest automotive manufacturers in the world, known for its capacity for innovation and its global presence. The Ampere Cars division focuses on the development of advanced technological solutions for optimizing automotive systems and components, applying modern data science and engineering techniques to enhance product efficiency and performance.

In recent years, Data Science has become an essential tool across various industries to improve processes and optimize products. In the development of complex systems, such as engines, transportation networks, or consumer technologies, understanding how users interact with products in real-world situations is critical. This enables adjustments in design and performance, ensuring greater efficiency, durability, and reliability.

The use of large volumes of data, often derived from connected sensors or simulations, provides insights into usage patterns that are difficult to predict through conventional testing. However, the uncertainty present in these data must be accounted for to make robust and reliable predictions. Advanced statistical techniques, such as the bootstrap method and regression, are essential for quantifying this uncertainty, allowing companies to make informed estimates about the variability of user behavior.

During the internship, the focus was on the analysis of real-world usage data from complex systems, specifically automotive systems. The techniques applied included the bootstrap method and regression to calculate confidence intervals that accurately reflect data variability. These confidence intervals can be used to make robust and reliable predictions about the behavior of users in real-world situations, which is essential for the development and optimization of automotive technologies. The analysis of uncertainties in data can also be applied in a variety of contexts beyond the automotive industry, such as logistics, manufacturing, and energy, to evaluate the performance of complex systems.

In summary, the internship provided valuable experience in applying advanced statistical techniques to real-world data, enabling the team to make informed estimates about the behavior of users and the variability present in large datasets. This experience will be valuable in future work and research endeavors.

2. INTERNSHIP OBJECTIVES

2.1. Main Objectives

As part of electric vehicle development projects and in collaboration with business, data science, and simulation teams, the main objective of the internship is to work with the millions of data points from simulations or collected by connected vehicles to:

- Explore and propose methods for constructing confidence intervals applicable to the customer usage calculation workflow.
- Develop and document a tool for estimating confidence intervals.

2.2. Specific Objectives

- Document the methodology and process for constructing confidence intervals applicable to the customer usage calculation workflow.
- Conduct case studies to validate the proposed methods and demonstrate their effectiveness in improving the accuracy and reliability of customer usage models.
- Develop a Python library for estimating confidence intervals, which can be integrated into the existing workflow and facilitate the implementation of the proposed methods.

3. COMPANY OVERVIEW

3.1. Renault Group

Renault Group, established in 1899 by Louis Renault and his brothers, Marcel and Fernand, is one of the world's largest automotive conglomerates. With a presence in over 130 countries, the group sells vehicles under various brands, including Renault, Dacia, Alpine, and, since 2021, Mobilize. Renowned for its long history of innovation, Renault has consistently led advancements in the automotive industry, such as introducing turbocharging in racing cars and producing one of the first modern electric vehicles on a large scale.

The Renault-Nissan-Mitsubishi Alliance, formed in 1999, is one of the largest automotive alliances globally, enhancing the group's competitive edge in both traditional and electric vehicle markets. Renault's market spans passenger and commercial vehicles, with a growing emphasis on sustainable mobility solutions. In 2023, Renault Group reported a revenue of €46.2 billion, with significant growth in electric vehicle sales, particularly in Europe. Employing over 160,000 people globally, Renault aims to solidify its position as a leader in the energy transition within the automotive industry.

As part of its broader commitment to sustainability, Renault Group emphasizes its responsibility not only in reducing its environmental footprint but also in promoting social responsibility. The company is actively working to integrate more environmentally friendly technologies into its production processes, including reducing carbon emissions across the vehicle lifecycle. Renault's focus on the circular economy—by increasing the recyclability of materials and developing cleaner manufacturing practices—underscores its dedication to minimizing environmental impact. This aligns with the "Renaulution" plan, launched in 2021, which focuses on revitalizing existing brands, driving sustainable growth, and accelerating the transition to electric mobility through the establishment of specialized units like Ampere Cars.

In addition to its environmental goals, Renault also prioritizes fostering a positive work culture for its global workforce. With a strong emphasis on innovation and collaboration, the company promotes a work environment that encourages creativity and technological advancement. Renault offers extensive training and development opportunities, empowering employees to grow their skills and actively contribute to the company's strategic goals. By creating a workplace that values diversity and inclusion, Renault ensures that its workforce remains motivated, engaged, and aligned with the company's mission to lead the transformation of the automotive industry toward a more sustainable future.

In the Île-de-France region, Renault has a significant presence with several key facilities, including the sites in Guyancourt and Lardy, where the internship was conducted. The Guyancourt site serves as a major center for the development and design of new vehicles, focusing on innovation and advanced engineering. The Lardy facility, on the other hand, specializes in the development and testing of engines, transmissions, and other automotive components. Through these strategic locations and initiatives, Renault is able to advance its dual goals of technological innovation and environmental sustainability, all while fostering a dynamic and inclusive culture for its employees.

3.2. Ampere Cars – Center of Electric Mobility Innovation

The internship was completed at Ampere Cars, a division of Renault Group dedicated exclusively to the development of 100% electric vehicles. Established as part of the "Renaulution" strategy, Ampere Cars embodies the group's commitment to sustainability and carbon emission reduction. The primary goal of this division is to accelerate the development of innovative electric vehicles for both mass-market and premium segments while optimizing production costs through advanced industrial processes.

Ampere Cars plays a crucial role in leading Renault Group's transition to electric mobility. The division collaborates closely with engineering, design, and technological innovation teams to develop cutting-edge solutions, such as modular electric platforms, high-density batteries, and rapid charging technologies. The factory located in Douai, France, represents the group's commitment to modern industrial practices and sustainability, equipped with advanced technologies to ensure the large-scale production of electric vehicles in an efficient and environmentally friendly manner.

In addition to vehicle production, Ampere Cars focuses on enhancing user experience by developing connected services and digital solutions aimed at improving the interface between the driver and the vehicle, as well as optimizing energy usage and battery management. The company is also deeply involved in circular economy initiatives, aiming to minimize environmental impact throughout the vehicle lifecycle.

3.3. Experience in the "Customer Usage" Team

The internship was conducted within the "Customer Usage" team at Ampere Cars, a crucial department dedicated to understanding and analyzing how customers use their electric vehicles. This team plays a vital role in the engineering division, particularly focusing on powertrain optimization. The primary objective of the team is to develop a customer usage model that ensures new vehicle designs align closely with customer expectations, focusing on optimizing key aspects such as performance, comfort and range.

To achieve this, the team leverages a variety of data sources, including time series data, high-frequency data, and other aggregated data, to analyze driving patterns, mileage, external conditions, and other usage factors. These analyses provide deep insights into how customers interact with their vehicles in real-world conditions, guiding decisions that improve both the design and functionality of electric vehicles. Additionally, the team conducts technical definition extrapolations—essentially using simulation modeling of the powertrain—since at this stage of development they do not have access to physical prototypes. This method enables the team to optimize the design for volume production, particularly focusing on maximizing the performance and reliability of the vehicle.

Moreover, the "Customer Usage" team plays a significant role in developing tailored solutions for diverse markets, taking into account regional and cultural differences in user needs. This includes adapting vehicle features and services to accommodate varying mobility objectives and driving conditions across different geographical locations.

The team is essential to Ampere Cars’ broader mission. The efforts not only contribute to designing electric vehicles that meet the evolving expectations of users but also support the company’s powertrain optimization goals. By focusing on customer-centric designs and using simulation-based optimization methods, the team’s work helps to increase both vehicle performance and customer satisfaction, which ultimately drives sales and strengthens Ampere Cars’ leadership in the electric mobility market.

In summary, the "Customer Usage" team's contributions are critical to ensuring that Ampere Cars continues to develop electric vehicles that cater to customer needs, while also optimizing performance, reliability, and sustainability.

4. LITERATURE REVIEW

4.1. Confidence Intervals

4.1.1. Definition

According to W. J. Conover, the confidence interval is a pivotal tool in inferential statistical analysis, enabling precise conclusions about a population based on sample data and a predefined level of confidence. This interval is derived from a point estimate of the population parameter and incorporates a standard error that reflects the variability within the sample. The confidence level, established in advance, signifies the probability that the interval encompasses the true population parameter.

In essence, a confidence interval represents a range of values for a population parameter, defined by two numbers derived from sample data, that is expected to contain the true parameter with a specified level of confidence. Typically, confidence levels range from 90% to 99%. Levels below 90% often lack precision, while levels above 99% can result in excessively wide intervals or require large sample sizes, which can be impractical and costly.

A common misconception is interpreting the confidence interval as containing the true parameter in a single sample. In reality, the confidence level indicates the proportion of intervals that would capture the true parameter across many samples. For instance, a 95% confidence level means that if numerous random samples were taken from the population, approximately 95% of the calculated intervals would include the true population parameter.

In practice, data professionals often generate a single confidence interval from one random sample, which may or may not include the actual population mean due to the challenges and costs associated with repeated sampling. Confidence intervals thus provide a valuable method for quantifying the uncertainty inherent in this process.

Random sampling is essential because it helps ensure that the sample accurately represents the population, minimizing bias and enhancing the validity of the results. This practice strengthens the reliability of confidence intervals and supports robust statistical inferences.

4.1.2. Relevance

Calculating confidence intervals is crucial for any company that seeks to make informed decisions based on data. In a business environment, where strategic and operational decisions are frequently based on sample data, the ability to quantify the uncertainty associated with estimates is fundamental. Confidence intervals provide a range within which the true value of a population parameter is expected to lie, with a predefined level of confidence. This not only enhances the accuracy of forecasts but also aids in risk management and strategic decision-making.

For companies working with small sample sizes, confidence intervals are particularly important. Small sample sizes, which can be common in startups or highly specialized sectors, exhibit greater variability and therefore higher uncertainty regarding the obtained estimates. Without proper calculation of confidence intervals, decisions may be based on inaccurate estimates, leading to financial and operational risks. Confidence intervals help mitigate these risks by providing a plausible range of values for the parameter of interest, allowing for a better understanding of the precision of the estimates.

4.1.3. Main Applications of Confidence Intervals

Confidence intervals have several practical applications in a business environment:

a. Product and Service Performance Evaluation: When launching new products or services, companies use market research surveys to gauge acceptance and satisfaction. Confidence intervals help interpret these results, providing a clearer picture of the product's potential success by reflecting the variability and reliability of consumer feedback.

b. Quality Control: In manufacturing, confidence intervals monitor product quality and consistency. They ensure that products meet standards and highlight areas for improvement, crucial for maintaining high quality and addressing performance deviations.

c. Financial Analysis: In finance, confidence intervals estimate investment risk and return, quantifying uncertainty in financial projections. This helps in making informed decisions about resource allocation and optimizing investment strategies.

d. Resource and Demand Planning: For forecasting demand, confidence intervals estimate the variability in sales forecasts. This aids in adjusting production and inventory strategies to avoid overproduction or stockouts, leading to more accurate and responsive planning.

e. Employee Satisfaction Analysis: Confidence intervals are used to analyze small sample surveys of employee satisfaction, helping to understand organizational climate and implement improvements to enhance engagement and productivity.

f. Simulation Data Representativeness: In simulations, confidence intervals assess how well simulated outcomes represent real-world scenarios. This evaluation ensures that decisions based on simulations are reliable and applicable to actual business conditions.

4.1.4. Calculating Confidence Intervals

The process of calculating a confidence interval involves several key steps, each critical to deriving accurate and meaningful results. Here’s a detailed guide to these steps, including formulas and considerations for different scenarios.

a. Determine the Confidence Level (CL):

The confidence level represents the probability that the confidence interval (CI) contains the true population parameter. Commonly used confidence levels include 90%, 95%, and 99%. For instance, a 95% confidence level implies that if many random samples are taken and compute a CI for each, approximately 95% of those intervals would contain the true population parameter.

b. Identify the Standard Deviation (σ) or Standard Deviation Estimate (s):

b.1. Population Standard Deviation (σ): If known, use the population standard deviation. It measures the spread of the population data.

b.2. Sample Standard Deviation (s): Use the sample standard deviation if the population standard deviation is unknown. It estimates the spread of the sample data.

c. Determine the Sample Size (n):

The sample size is the number of observations in the sample. Larger sample sizes typically lead to more accurate and reliable estimates.

d. Choosing the Appropriate Distribution and Margin of Error (ME):

d.1. For Large Samples: Z-Scores:

When dealing with large sample sizes (typically n > 30), the z-score method is employed. This approach is based on the Central Limit Theorem, that states that as the sample size increases, the distribution of the sample mean approaches a normal distribution, regardless of the shape of the original data distribution. This allows us to use the properties of the normal distribution to make inferences about the population mean for sufficiently large samples.

$$\boxed{ME = z \times \frac{\sigma}{\sqrt{n}}}$$

Where:

- z: Z-score corresponding to the desired confidence level
- σ: Population standard deviation.
- n: Sample size.

d.2. For Small Samples: T-Scores

For small sample sizes (typically n<), the t-distribution is used. This distribution accounts for the additional uncertainty in small samples with heavier tails, which helps to better reflect variability and potential outliers.

$$\boxed{ME = t \times \frac{s}{\sqrt{n}}}$$

Where:

- t: T-score from the t-distribution table, based on the confidence level and degrees of freedom (df = n−1).
- s: Sample standard deviation.
- n: Sample size.

As the sample size increases, the t-distribution approaches the normal distribution, making the choice between z-scores and t-scores less critical.

e. Construct the Confidence Interval (CI)

$$\boxed{\text{CI} = \text{Point Estimate} \pm \text{Margin of Error (ME)}}$$

Where:

- Point Estimate: Typically the sample mean ($\bar{x}$); sample proportion ($\hat{p}$).
- Margin of Error (ME): Calculated from the above steps.

4.1.5. Direct applications for computing Confidence Intervals:

a. Confidence Interval for Population Mean:

a.1. When Population Variance is Known:

$$\boxed{CI = \bar{x} \pm z \times \frac{\sigma}{\sqrt{n}}}$$

a.2. When Population Variance is Not Known:

$$\boxed{CI = \bar{x} \pm t \times \frac{s}{\sqrt{n}}}$$

b. Confidence Interval for Population Proportions:

$$\boxed{CI = \hat{p} \pm z \times \frac{\sqrt{\hat{p}(1-\hat{p})}}{\sqrt{n}}}$$

Where:

- $\hat{p}$: Sample proportion.
- n: Sample size.

4.1.6. Non-Parametric Methods:

In statistical analysis, resampling methods such as the jackknife and bootstrap provide practical tools for estimating standard errors of statistical estimators. These computational techniques offer flexibility and robustness, particularly when traditional mathematical formulas are difficult to apply. The following sections detail each method and compare their advantages and limitations.

4.1.6.1. Jackknife Method

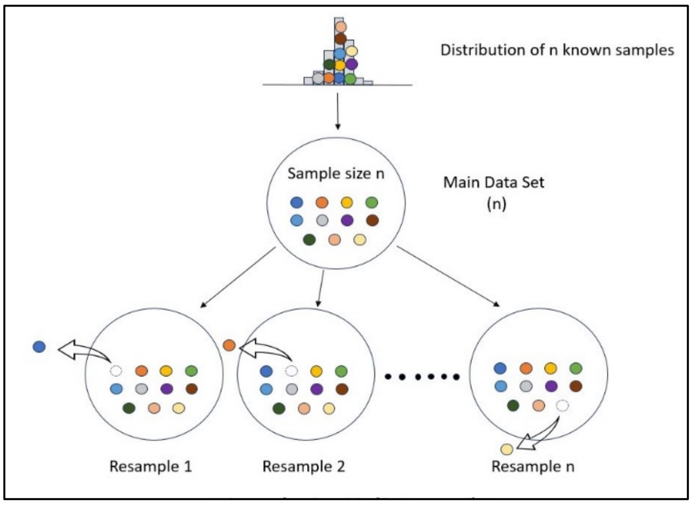


Figure 1 - Schematic of Jackknife Resampling

The jackknife method, illustrated on figure 1, involves creating multiple resampled datasets by systematically excluding one observation at a time from the original sample. This technique focuses on estimating the bias and variance of a statistical estimator through a series of steps.

Firstly, in the resampling process, jackknife samples are generated by leaving out one observation from the original dataset in each iteration. For each jackknife sample, the estimator of interest is calculated.

Next, in the bias and variance estimation phase, bias is estimated by computing the difference between the mean of the jackknife estimates and the original estimator value. Variance is estimated using the jackknife formula, which involves calculating the mean of the squared differences between jackknife estimates and their mean.

The advantages of the jackknife method include its ability to provide useful estimates of bias and variance, and its computational simplicity compared to some other methods. However, its limitations include reduced efficiency in computing confidence intervals, particularly when dealing with asymmetric distributions.

4.1.6.2. Bootstrap Method

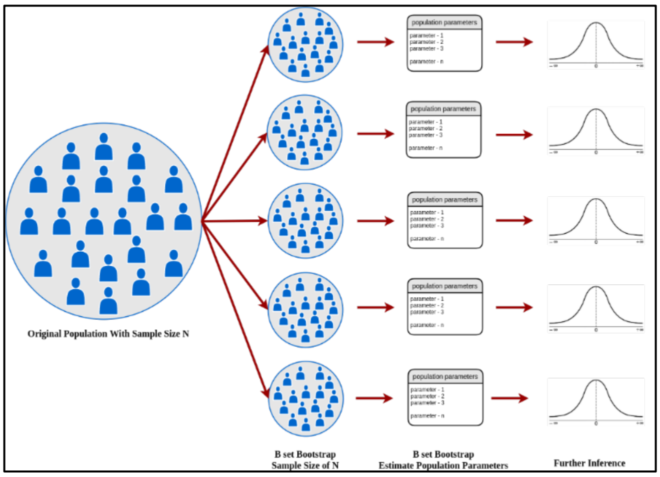


Figure 2 - Schematic of Bootstrap Resampling

The bootstrap method, illustrated on figure 2, introduced by Efron in 1979, involves generating multiple resampled datasets from the original sample by sampling with replacement. This technique is used to estimate the distribution of a statistical estimator through a structured process.

Firstly, in the resampling process, bootstrap samples are created by randomly selecting observations from the original data with replacement. This means that some observations may be selected multiple times while others may not be included at all. For each bootstrap sample, the estimator of interest is calculated.

In the distribution and confidence intervals phase, the distribution of bootstrap estimators is used to compute confidence intervals and assess variability. Typically, percentile-based methods are employed for this purpose.

The advantages of the bootstrap method include its applicability without specific assumptions about the underlying data distribution and its potential reliability over asymptotic approximations, especially in small samples or with complex data structures.

4.1.6.3. Comparison of Bootstrap and Jackknife Methods

The primary distinction between the bootstrap and jackknife methods lies in their resampling approaches:

- Bootstrap: Resamples with replacement, allowing individual observations to be selected multiple times or not at all. This method is particularly versatile and robust, often used for estimating the distribution of an estimator and constructing confidence intervals. The bootstrap's flexibility makes it suitable for a wide range of data distributions.

- Jackknife: Resamples without replacement, excluding one observation in each iteration. This method is typically employed for assessing the bias and variance of an estimator. While computationally simpler, it is generally less effective than bootstrap for constructing confidence intervals, particularly when dealing with asymmetric distributions.

4.1.6.4. Kernel Density Estimation (KDE) Method

Kernel Density Estimation (KDE) is a non-parametric method used to estimate the probability density function of a random variable, serving as an alternative to resampling techniques like bootstrap and jackknife when the data distribution is unknown. Unlike parametric methods, KDE does not assume a specific distribution form, making it particularly useful for complex or multimodal data.

KDE involves selecting a bandwidth that controls the smoothness of the density estimate, with smaller bandwidths providing more detail and larger ones producing smoother curves. The kernel function, typically a symmetric, non-negative function like the Gaussian, assigns weights to data points. By summing the contributions of all kernels centered at each data point, KDE creates a smooth density curve representing the data distribution.

The key advantages of KDE include its flexibility in modeling unknown distributions, its ability to capture features like skewness and multimodality, and its capacity to provide visual insights into data patterns. Compared to bootstrap and jackknife, KDE directly estimates the density function rather than relying on resampling, offering a more continuous and informative view of the data. This makes KDE especially effective in handling complex distributions and a robust, flexible tool for non-parametric data analysis.

4.1.7. Percentiles

Calculating confidence intervals for specific percentiles is crucial for gaining a deeper understanding of data distributions and making informed decisions. Unlike mean-based analyses, which may not fully capture the variability within a dataset, percentile-based confidence intervals provide insights into the spread and range of data. For instance, assessing the confidence interval for the 25th or 75th percentile reveals how data is distributed across different segments, which is especially valuable in skewed or non-normally distributed datasets.

This approach is particularly significant in understanding client usage patterns in the automotive industry. For example, if a company is analyzing vehicle usage data, the confidence interval for higher percentiles (such as the 90th or 95th percentile) might show a broader range, indicating greater uncertainty in extreme usage scenarios. This variability can be crucial for anticipating the needs of high-usage clients and planning for potential maintenance or support requirements.

Confidence intervals for percentiles also validate statistical results, adding credibility to analyses and ensuring more accurate interpretations. They are particularly useful in handling skewed data, where extreme values can distort the mean, by providing a clearer picture of data variability and distribution.

4.1.8. S-Curve for Confidence Intervals

One effective method for visualizing confidence intervals across different percentiles is using an S-curve. The S-curve, or the empirical cumulative distribution function (CDF) plot, illustrates how the proportion of data below each percentile changes as the percentile value increases. This curve helps in understanding the distribution of data and how confidence intervals behave across various percentiles.

The S-curve represents the cumulative probability of the data points, showing a smooth, S-shaped curve that flattens out as it approaches the extremes. The steepness of the curve in the middle reflects a dense concentration of data points, while the flattening at the ends indicates fewer data points and greater uncertainty in estimating percentiles at the extremes. By examining the S-curve, one can better understand the range and variability of confidence intervals for different percentiles, providing additional insights into the data distribution and the reliability of the statistical estimates.

Overall, using the S-curve to analyze confidence intervals for percentiles enhances the understanding of data distribution and variability, leading to more informed and reliable decision-making.

5. CUSTOMER USAGE WORKFLOW

(Confidential)

Figure 3 - Customer Usage Workflow

As mentioned earlier, the Customer Usage Team is responsible for understanding how customers use their vehicles in real-world driving scenarios. This work involves collecting and analyzing data from various sources, such as surveys, vehicle sensors, and connected car platforms. The team's primary goal is to develop a customer usage model that helps optimize powertrain systems and vehicle components based on observed usage patterns.

The team's workflow is an iterative process, with data collection and analysis occurring continuously throughout the development cycle of new systems and components. This ongoing process allows the customer usage model and the simulation tool to be continuously refined and improved based on new data and real-world driving scenarios. As a result, the team significantly contributes to the creation of vehicles that are more efficient, reliable, and tailored to real customer usage, enhancing the company's ability to develop more competitive and user-focused products.

The main databases of the Customer Usage team used during the internship will be presented below:

5.1. CDU

The CDU (Car Data Usage) database is an essential tool for collecting and analyzing data from connected vehicles, primarily focused on automotive development and engineering. CDU campaigns gather real data from vehicles equipped with connectivity, monitoring customer behavior and vehicle usage conditions. These campaigns typically last from one to two years, capturing information across different seasons.

(Confidential)

5.2. NCBS

The NCBS (New Car Buyer Survey) database is a statistically representative and essential tool for analyzing the usage behavior of new vehicles. Through interviews conducted with customers who purchased their vehicles about three months prior, the survey collects detailed data on vehicle usage, providing valuable insights for the development and adjustment of new automotive models.

(Confidential)

The survey allows for data segmentation by vehicle type or specific model, enabling targeted analyses that help manufacturers understand the usage behavior of similar vehicles and adapt their development and marketing strategies accordingly.

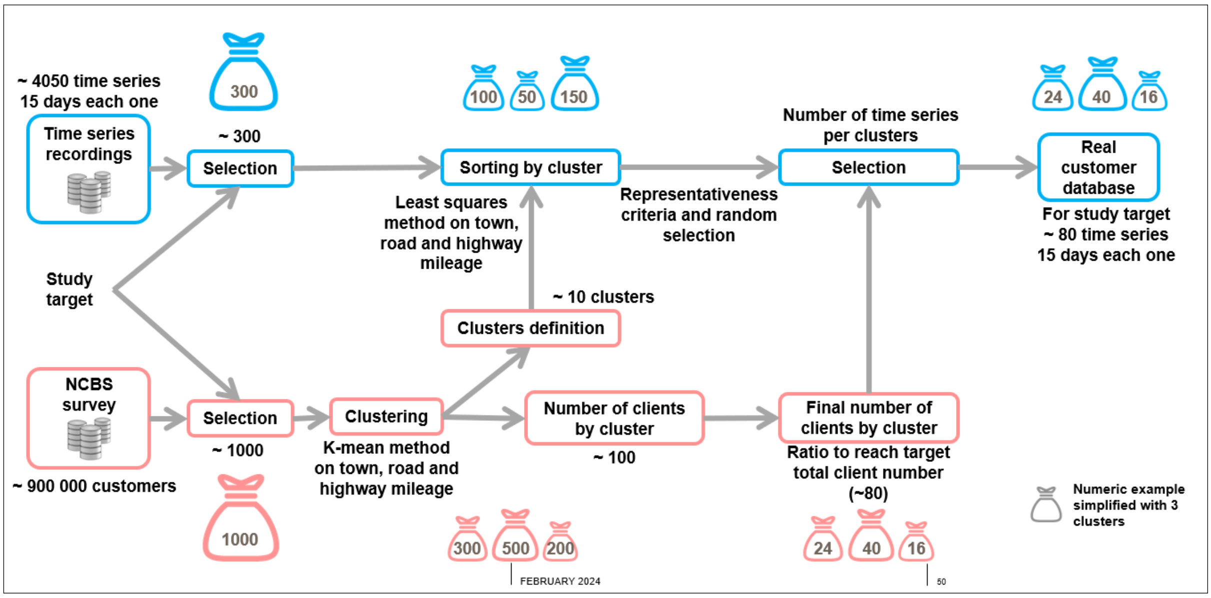
Among the information collected are annual mileage, the distribution of vehicles use in cities, on roads, and highways, as well as charging profiles, which are particularly relevant for hybrid and electric vehicles. These data are essential for understanding the real behavior of consumers and adjusting products to meet their needs.

The NCBS database is integrated with other data sources, such as specific information about new vehicles under development (mass, power, sales region) and free-drive recordings from real vehicles. This combination allows for a detailed and customized analysis, aligning simulations with consumer behavior. By using data science techniques like clustering, customer profiles are grouped and adjusted to the free-drive recordings, resulting in usage and charging profiles that are representative for simulations and performance predictions under real-world conditions.

(Confidential)

5.3. Real Customers Database (Base Client)

The Real Customer Model (Base Client) is a sophisticated method designed to simulate vehicle usage based on real customer data. This model is constructed from multiple data sources and involves several stages of processing to ensure its accuracy and representativeness.



~ 4050 time series
15 days each one
Time series recordings
300
~ 300
Selection
100
50
150
Sorting by cluster
Least squares method on town, road and highway mileage
Representativeness criteria and random selection
Number of time series per clusters
Selection
24
40
16
Real customer database
For study target
~ 80 time series
15 days each one
Study target
~ 10 clusters
Clusters definition
NCBS survey
~ 900 000 customers
Selection
~ 1000
1000
Clustering
K-mean method on town, road and highway mileage
300
500
200
Number of clients by cluster
~ 100
Final number of clients by cluster
Ratio to reach target total client number (~80)
24
40
16
Numeric example simplified with 3 clusters
FEBRUARY 2024
50

Figure 4 - Real Customer Database – Construction Workflow

(Confidential)

Data science techniques are then employed to analyze and cluster the customer data. This clustering process helps to create a representative subset of customer profiles, typically consisting of 100 to 200 profiles. These profiles are carefully chosen to reflect a statistically significant sample of the overall customer base.

6. METHODOLOGY

6.1. Tools

During the internship, a range of powerful tools and technologies were utilized to enhance data analysis and modeling.

BigQuery and Google Cloud Storage were key for managing and analyzing large datasets. Google Cloud's Vertex AI was used for building and deploying machine learning models, with Jupyter Notebooks integrated into the Vertex AI environment for interactive development and analysis. For data manipulation, SQL and Python were employed, leveraging libraries like Pandas and NumPy for data handling, as well as Matplotlib and Seaborn for visualization.

Various machine learning algorithms were implemented using Scikit-learn, and deep learning was explored with TensorFlow and the Keras API. Additionally, PrettyTable was used for data presentation, and SciPy facilitated statistical analysis.

Overall, these tools provided a robust framework for effective data analysis and predictive modeling throughout the internship experience.

6.2. Networking and Renault Ecosystem

The internship involved collaboration across multiple areas, including data science, statistics, and quality teams, to validate the methodology employed. This multidisciplinary approach ensured a comprehensive evaluation of the processes and results.

The findings were presented in workshops and shared with key stakeholders at Renault, including the director of the electric components design department, the Car Data Committee and durability commitee. Validation was conducted with various teams, fostering an ecosystem of collaboration and interaction among different departments. This emphasis on teamwork and communication enriched the overall project and strengthened the insights gained throughout the internship experience.

6.3. Python Library Deliverables

During the internship, a Python library was developed for general applications, combining the bootstrap method with the plotting of confidence intervals for percentiles. This tool is now available for diverse use within Renault.

The library includes a Python script that leverages the NumPy, Matplotlib, and SciPy libraries to calculate confidence intervals for specified percentiles of data using the bootstrap technique. It also visualizes these intervals alongside the cumulative distribution function (CDF) of the data.

The main components of the code include the following functions:

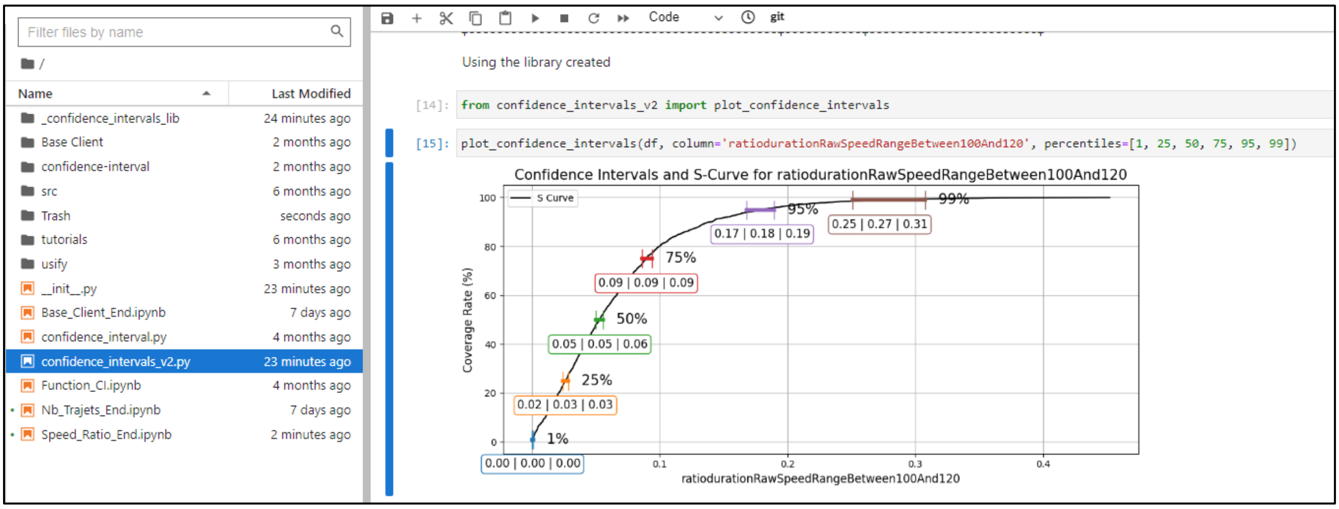
a. Calculate CDF: The calculate_cdf function computes the cumulative distribution function for the given data. It sorts the data and returns both the sorted data and its CDF, which is essential for understanding the distribution of the dataset.

b. Bootstrap Confidence Interval: The bootstrap_confidence_interval function calculates confidence intervals for specified percentiles using the bootstrap method. This function takes the dataset, the desired percentiles, the confidence level, and the number of bootstrap samples as parameters. It generates bootstrap samples, computes the specified percentiles for each sample, and returns the confidence intervals, allowing for robust statistical analysis.

c. Plotting Function: The plot_confidence_intervals function visualizes the S-curves and confidence intervals for one or all columns of a DataFrame. It handles NaN values, calculates the necessary statistical measures, and plots the results, including annotations for clarity. This visualization aids in the interpretation of the data and the understanding of the confidence intervals in relation to the distribution.

These functions collectively provide a comprehensive tool for statistical analysis and visualization, making it easier for users to understand and interpret complex data sets.

Below (figure 5), an example of its functionality is presented:



Name	Last Modified
_confidence_intervals_lib	24 minutes ago
Base Client	2 months ago
confidence-interval	2 months ago
src	6 months ago
Trash	seconds ago
tutorials	6 months ago
usify	3 months ago
__init__.py	23 minutes ago
Base_Client_End.ipynb	7 days ago
confidence_interval.py	4 months ago
confidence_intervals_v2.py	23 minutes ago
Function_CI.ipynb	4 months ago
Nb_Trajets_End.ipynb	7 days ago
Speed_Ratio_End.ipynb	2 minutes ago

Using the library created

```
[14]: from confidence_intervals_v2 import plot_confidence_intervals

[15]: plot_confidence_intervals(df, column='ratiodurationRawSpeedRangeBetween100And120', percentiles=[1, 25, 50, 75, 95, 99])
```

Figure 5 - Example of Confidence Interval Function

The full code, along with detailed commentary on each part, is included in the annex (appendices) for further reference.

6.4. Study Cases

6.4.1. CDU: Number of trips per vehicles per day

The first case study was conducted using aggregated CDU data, containing the valid trips of one model vehicles at Renault, as specified below. The objective was to calculate the confidence interval for the variable "number of trips per vehicle per day," segmented by various countries.

Valid Trip Filters	
Model	Confidential
Data Interval	03/2019 – 07/2020
Time of trips	> 60 s
Tripe mileage	> 0.5 m

Table 1. Valid Trip Filters

It was found that there are thousands unique vehicles (VIN) for the model, considering the selected filters, out of a total of millions valid trips. Below (figure 6), you can observe the data distribution by country. The left graph represents the total number of trips, while the right graph shows the number of unique vehicles. It is important to note that certain vehicles had 0 trips, which will be considered in the confidence interval calculations.

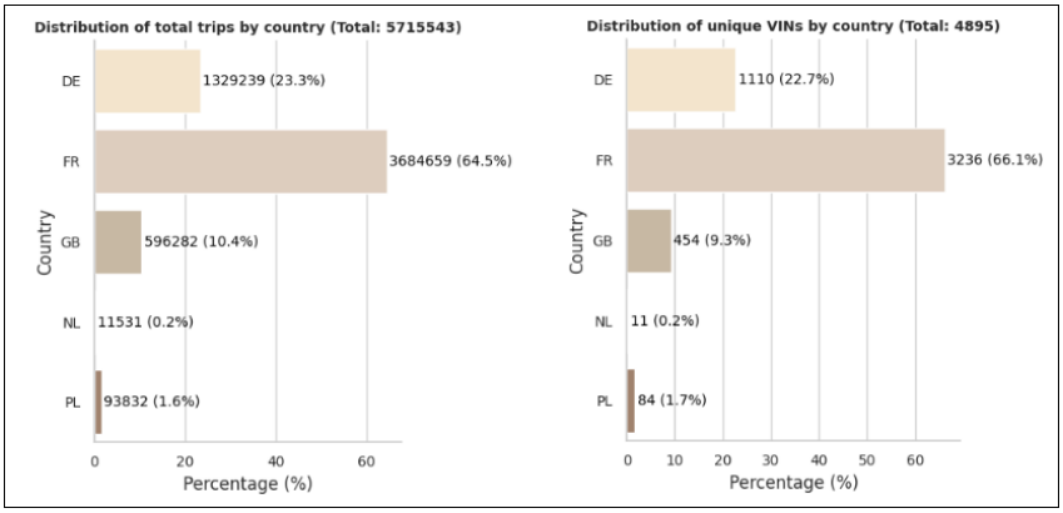


Figure 6 - Distributions of trips and unique VINs by country

6.4.1.1. Validating the CI methods

Initially, the confidence interval for the mean was calculated using three methods: the formula, bootstrap, and jackknife. The sample was randomly selected, ensuring representativeness.

A Shapiro-Wilk test revealed that the data are not normally distributed (Statistic=0.92, p-value=0.00). This deviation affects the applicability of methods assuming normality.

The formula method assumes normality but is reasonably accurate here due to the sample size being larger than 30, which supports the Central Limit Theorem. In contrast, the bootstrap and jackknife methods do not rely on normality assumptions and use resampling to estimate confidence intervals.

Method	Confidence Interval for the mean
Formula	[3.24 – 3.31]
Bootstrap	[3.23 – 3.32]
Jackknife	[3.23 – 3.32]

Table 2. Comparison among the three methods

The intervals from all three methods were similar, indicating consistency. Given the non-normality of the data, the non-parametric bootstrap method was chosen for its robustness and adaptability to complex distributions, making it the most reliable for these calculations.

6.4.1.2. Confidence Interval with the sample size decrease

After calculating the confidence interval for the entire sample of 5000 VINS, an analysis was performed for decreasing sampling values. According to the results obtained in the graph bellow (figure 7), it can be observed that the smaller the sample size, the larger the confidence interval. As sample size decreases, the confidence interval tends to widen due to increased uncertainty associated with estimating the population mean. This widening occurs primarily because smaller samples are more likely to inaccurately represent the true population mean, leading to greater variability in sample estimates. Additionally, smaller samples are more prone to containing extreme values, which can exert a larger influence on the sample mean and contribute to less stable estimates. With fewer data points available, there is greater uncertainty regarding the accuracy of the population mean estimate, resulting in a larger margin of error and, consequently, a wider confidence interval to accommodate this increased uncertainty.

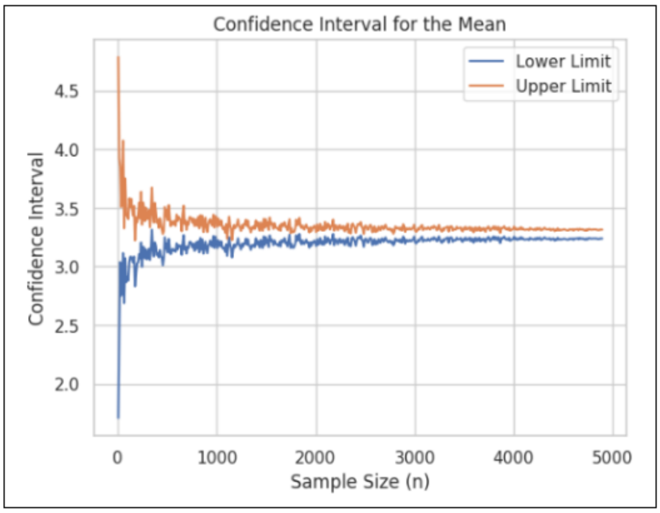


Figure 7 - Variation of the Confidence Interval with the sample size decrease

6.4.1.3. Percentiles

The primary objective was to observe the confidence interval of the variable percentiles. Therefore, the intervals were calculated with bootstrap and plotted on the S-curve for all countries (figure 8). However, I have selected the worst-case scenario to illustrate here, which corresponds to the Great Britain (GB). The worst-case scenario refers to the largest confidence interval for the 99th percentile, commonly chosen for team simulations.

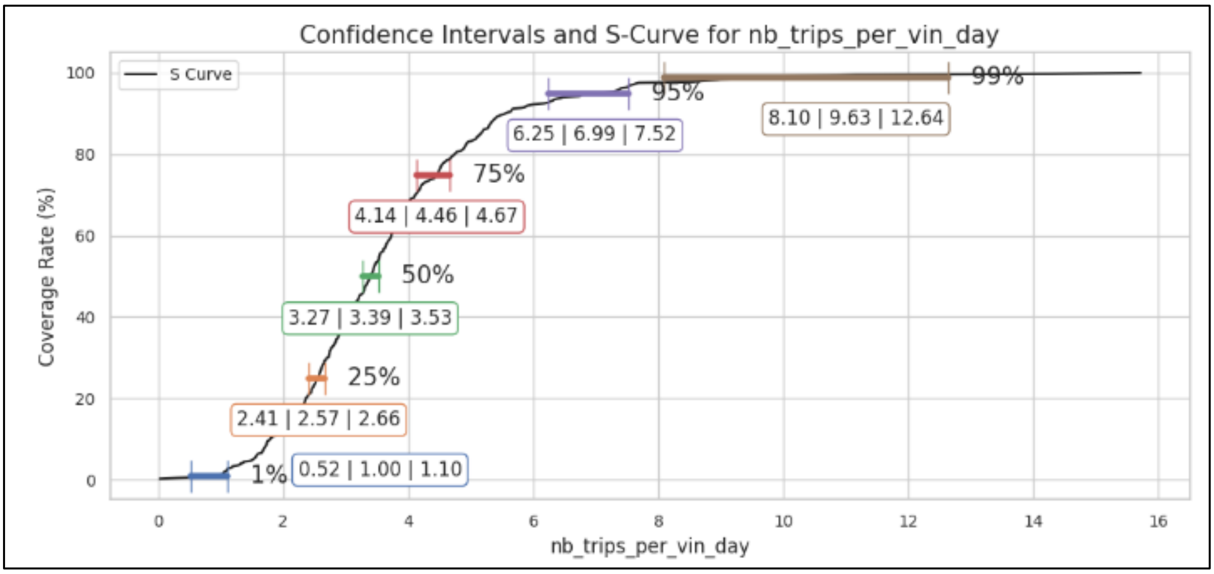


Figure 8 - Confidence Intervals and S-Curve for “Number of trips per vehicles and per day”

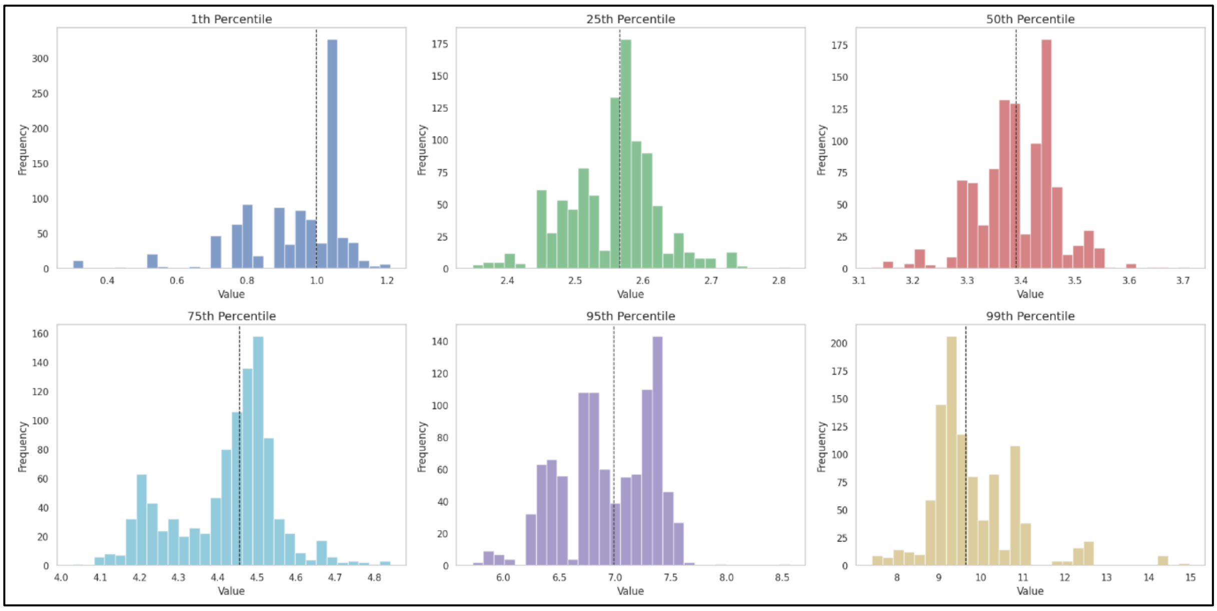


Figure 9 - Data distribution for the percentiles of the S-Curve: Study Case 1

The confidence interval for the 99th percentile was notably large, emphasizing the need to consider this factor. As percentiles increase, so do their confidence intervals, due to greater variability and uncertainty in extreme values. This means that while analyzing higher percentiles, the broader confidence intervals reflect increased uncertainty, which is important for making informed decisions.

In contrast, lower percentiles (1st, 25th, 50th) are more stable and predictable, indicating that these portions of the distribution are less sensitive to resampling. Higher percentiles (75th, 95th, 99th) exhibit greater variability, revealing challenges in estimating the tail end of the distribution. This variability (illustrated on figure 9) suggests that the tail is more affected by extreme values or distribution skewness, making it less predictable and more sensitive to sampling.

6.4.2. CDU: Ratio duration Speed Range

The second case study focused on analyzing the usage behavior of the automotive differential, specifically targeting customers who impose severe conditions on this component, leading to increased damage. The study monitored the speed variable, with particular attention to the fact that low speeds are more detrimental to the differential. The main database used has a total of **(Confidential)** unique vehicles and trips associated with one electric model in the European market.

The automotive differential is a critical component in the drivetrain system, allowing the wheels to rotate at different speeds, which is essential when the vehicle turns. This mechanism improves stability, traction, and overall handling, particularly during cornering. However, when vehicles operate at low speeds, the differential endures higher levels of stress and friction due to the increased torque demands. This can lead to accelerated wear, overheating, and potential failure of the differential over time, especially under severe usage conditions. Monitoring and understanding these operating behaviors help identify critical usage patterns that contribute to component degradation, enabling better maintenance strategies and design improvements.

As this case study was a continuation of previous work by the team, the variables were already appropriately filtered and processed, facilitating the application of previously studied confidence interval methods. The variable “ratiodurationRawSpeedRangeBetweenXAndY” was selected for the study. This variable represents the proportion of time the vehicle's speed was within a specified range, between X and Y. It provides insight into how often and for how long the vehicle operates within that speed range, which is crucial for assessing the differential's exposure to potentially damaging conditions.

6.4.2.1. Percentiles

The confidence interval techniques, including both normal approximation and bootstrap, were applied once more to validate the approach. Given the confirmation of the methods' robustness and the reasons previously discussed, the bootstrap technique was selected for the analysis. Consequently, S-curves were plotted (figure 10), and the worst speed profile, in terms of confidence interval, was identified as “ratiodurationRawSpeedRangeBetween90And100”.

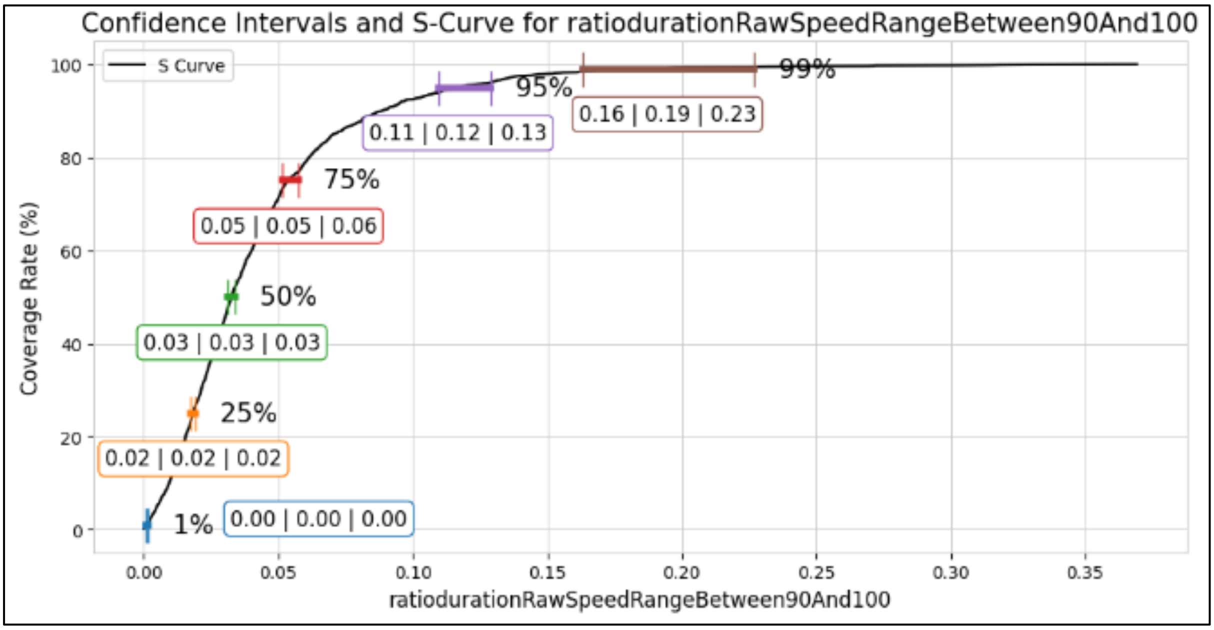


Figure 10 - Confidence Intervals and S-Curve for “ratiodurationRawSpeedRangeBetween90And100”

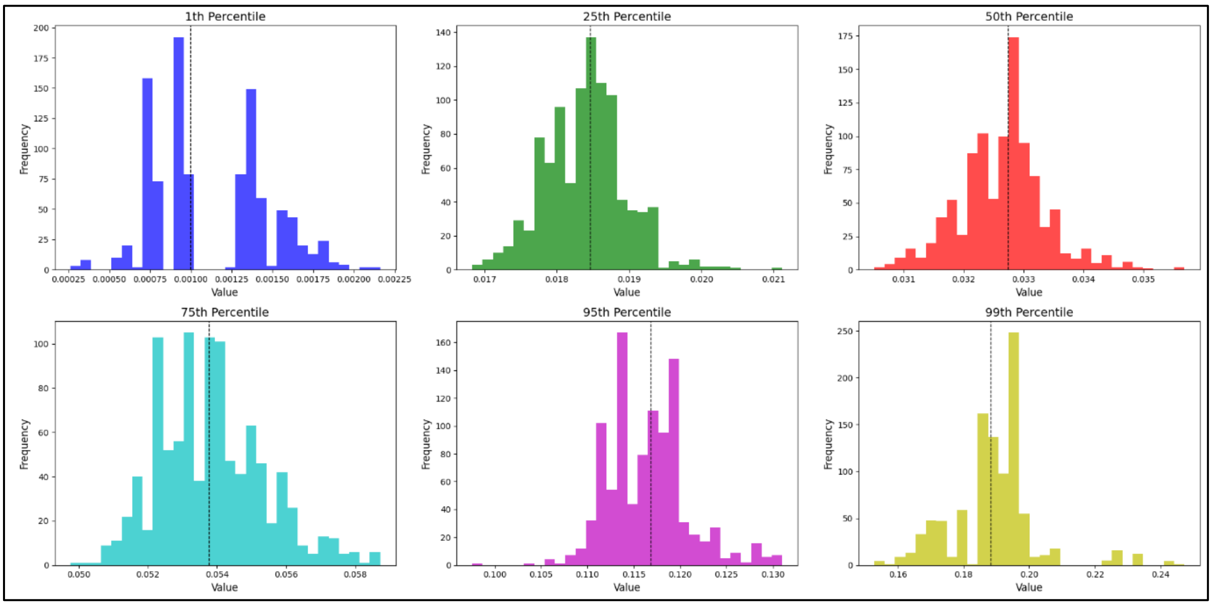


Figure 11 - Data distribution for the percentiles of the S-Curve: Study Case 2

The analysis of percentiles (figure 11) revealed clear patterns between lower and higher percentiles, with significant implications for interpretation and decision-making. Lower percentiles, such as the 1st, 25th, and 50th, exhibited stable and predictable behavior, with low variability among bootstrap samples. This consistency suggests that these percentiles are more reliable and provide a solid view of the central distribution.

In contrast, higher percentiles, such as the 75th, 95th, and 99th, showed greater variability and uncertainty. The 99th percentile, similar to what was observed in the first case study, stood out due to its broad range of values and elevated standard deviation. This increase in uncertainty is expected for extreme percentiles, reflecting the difficulty of accurately estimating these values due to the higher dispersion in the data.

The comparison between studies confirms the consistency of these patterns. While lower percentiles remain stable and predictable, higher percentiles reveal significant variability. This variability, characteristic of extreme percentiles, underscores the need for caution in data interpretation.

6.4.3. Base client: Mean Speed Profile

In the preceding case studies, confidence interval calculations were performed using aggregated data from the CDU. The validity of confidence interval methods was assessed using formula-based, bootstrap, and jackknife approaches. Despite the data's deviation from normal distribution, the robustness of the bootstrap method, which does not rely on normality assumptions, was determined to be the most reliable for these calculations.

The forthcoming third case study will focus on confidence interval calculations utilizing the Real Customer Database. This database, constructed for one vehicle model, from a real-world customer usage model, incorporates time series data from approximately 100 to 200 customers. Unlike the aggregated CDU data, which was randomly sampled, the Real Customer Database is generated through specific filtering and clustering.

The primary challenge with this dataset arises from its non-random nature, complicating confidence interval calculations. The selective curation of data based on criteria such as vehicle type, geographical region, and usage patterns means that traditional methods, which assume random sampling and normality, may not be applicable. Consequently, more advanced statistical techniques are required to account for the dataset's structured and potentially biased nature.

To address these challenges, this study will employ a regression model developed using the Real Customer Database and a non-parametric model using the NCBS, as illustrated bellow (figure 12):

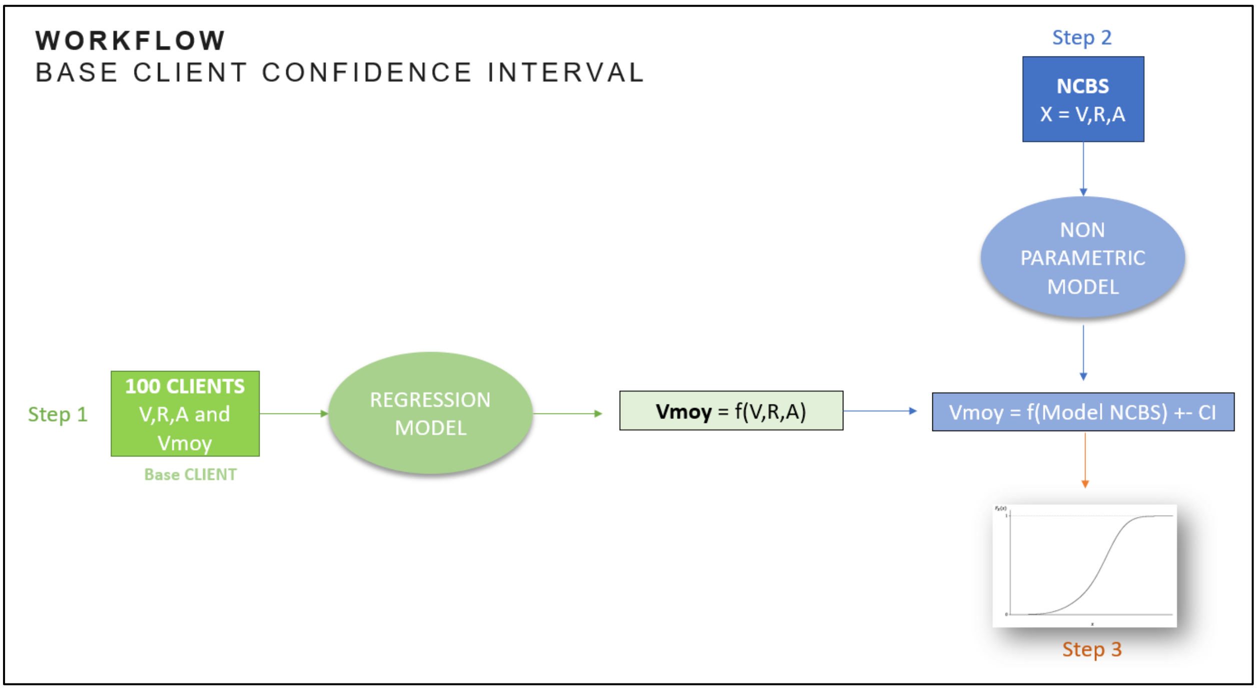


Figure 12 - Confidence Interval for Base Client - Workflow

6.4.3.1. Step 1: Regression Models

The use of regression models is motivated by the need to quantify the relationship between quantitative variables in the dataset and the variable of interest, Vmoy (Average Speed). These models allow us to identify and measure the influence of explanatory variables (such as the proportions of distances traveled on different types of roads) on the response variable, enabling robust predictions and inferences.

The model will relate "Mean Speed" (Vmoy) to the distance ratio variables collected from the dataset, specified bellow.

Features	Specification	Type
Vmoy	Mean Speed	Response
V	(Distance traveled in the City / Total Distance)*100	Explanatory
R	(Distance traveled on Route / Total Distance)*100	Explanatory
A	(Distance traveled on Highway / Total Distance)*100	Explanatory

Table 3. Features of the Regression Model

a. Correlation

To evaluate the relationships between the variables, a correlation matrix was constructed (figure 13), which identified the degree of association between the explanatory variables and Vmoy. The variable A (the proportion of the distance traveled on highways relative to the total distance) showed a strong correlation with Vmoy, indicating that the proportion of travel on highways is a significant factor in determining the average speed of the vehicle.

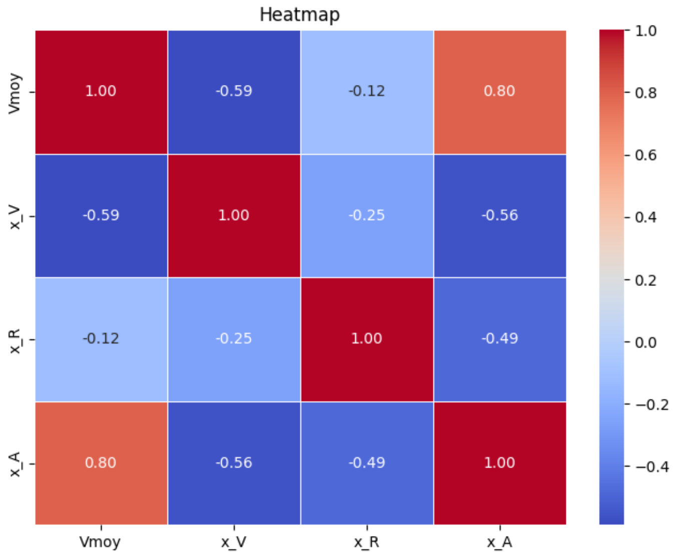


Figure 13 - Correlation Matrix

This strong correlation suggests that variable A could be a good predictor of average speed, enhancing the accuracy of the estimates. Furthermore, identifying significant correlations provides a better understanding of behavioral patterns in the data, allowing adjustments to the model to reflect vehicle usage under different road conditions.

b. Standarization

The Standard Scaler is a normalization technique that transforms data into a distribution with a mean of zero and a standard deviation of one by subtracting the mean of each variable and dividing by its standard deviation. This standardization is essential in machine learning models that are sensitive to variable scales, such as regressions, neural networks, and distance-based algorithms (e.g., K-Nearest Neighbors).

In the case of the variables V, R and A, there are significant differences in the scales and dispersions of the data, which can be clearly observed by analyzing the boxplot of the variables. The boxplot (figure 14) highlights the distinct ranges, medians, and the presence of possible outliers, emphasizing the need for standardization to balance the impact of each variable on the models.

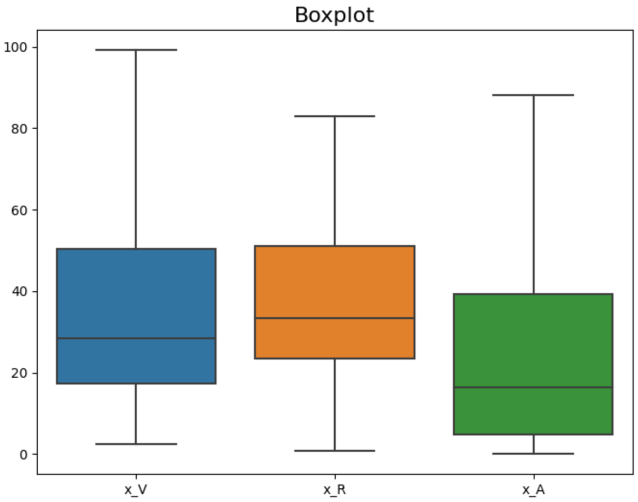


Figure 14 - Boxplot

Even though they vary between 0 and 100, the variables V, R, and A have different standard deviations (22.45, 19.13, and 24.41, respectively). Standardization balances these dispersions, making the variables comparable in terms of variation and relative magnitude.

Without standardization, these differences can compromise the performance and interpretability of the results, as variables with greater dispersion or extreme values can disproportionately influence the models. Therefore, the use of the method is applicable and recommended, ensuring that all variables contribute equally to the analysis and improving the effectiveness of the models.

c. Models

The problem addressed in this project is a regression problem, which involves predicting a continuous outcome based on input features. There are various methods available for performing regression, each with its strengths and limitations. However, given the constraint of having only 100 data points, certain methods may become overly complex and less suitable for this dataset.

The objective of this section is not to delve into a detailed explanation of each model, as they are well-established in the literature, but a brief overview of the regression methods tested includes:

c.1. Linear Regression: A straightforward approach that assumes a linear relationship between input features and the target variable.

c.2. Ridge Regression: An extension of linear regression that incorporates L2 regularization to prevent overfitting by penalizing large coefficients.

c.3. Lasso Regression: Similar to Ridge, but utilizes L1 regularization, which can lead to sparse solutions by driving some coefficients to zero.

c.4. Decision Tree: A non-linear method that creates branches based on feature values, allowing for complex relationships but prone to overfitting with small datasets.

c.5. Random Forest: An ensemble method that combines multiple decision trees to enhance accuracy and robustness, although it can also be complex and susceptible to overfitting with limited data.

c.6. Support Vector Machine (SVM): A powerful method aimed at finding the optimal hyperplane for regression, requiring careful tuning and can be computationally intensive.

c.7. XGBoost: An efficient implementation of gradient boosting that can manage missing values and mitigate overfitting but may be too complex for smaller datasets.

c.8. Neural Networks: Flexible models capable of capturing non-linear relationships but necessitate larger datasets to avoid overfitting.

d. Metrics

In regression analysis, several key metrics are used to evaluate model performance:

d.1. Mean Squared Error (MSE): Measures the average of the squared differences between predicted and actual values. Lower MSE indicates better accuracy but is sensitive to larger errors.

d.2. Mean Absolute Error (MAE): Calculates the average absolute differences between predictions and actual values. It’s easier to interpret since it's in the same unit as the target variable, with lower values indicating better performance.

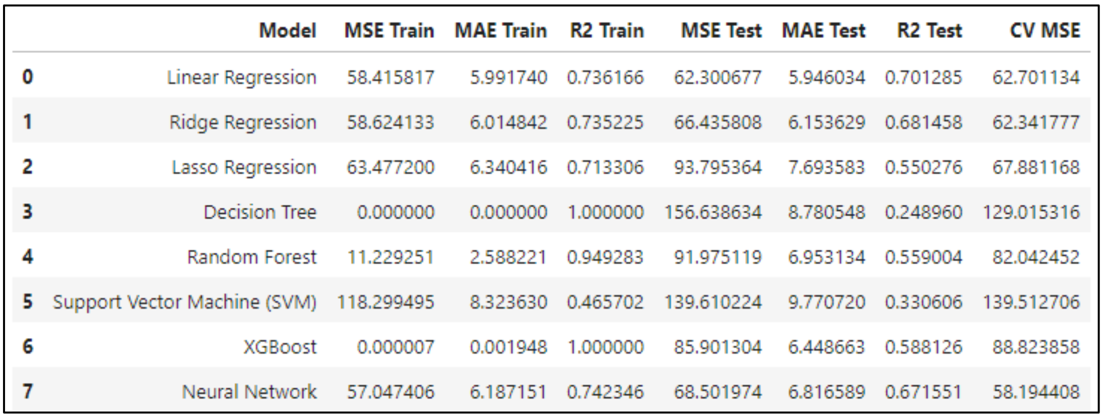
d.3. R-squared (R²): Represents the proportion of variance in the dependent variable explained by the independent variables, ranging from 0 to 1. Higher values indicate better explanatory power, but it can be misleading if overfitting occurs.

d.4. Cross-Validation Mean Squared Error (CV MSE): Derived from cross-validation, this metric averages the MSE across multiple training and validation sets, providing a more reliable estimate of model performance on unseen data.

Together, these metrics offer a comprehensive evaluation of a model's accuracy, robustness, and generalization ability.

e. Results

The performance of these models was evaluated, as documented in the Python code and the results table below (figure 15). This approach enables a balance between complexity and the need for reliable predictions, ensuring effective utilization of the available data.



	Model	MSE Train	MAE Train	R2 Train	MSE Test	MAE Test	R2 Test	CV MSE
0	Linear Regression	58.415817	5.991740	0.736166	62.300677	5.946034	0.701285	62.701134
1	Ridge Regression	58.624133	6.014842	0.735225	66.435808	6.153629	0.681458	62.341777
2	Lasso Regression	63.477200	6.340416	0.713306	93.795364	7.693583	0.550276	67.881168
3	Decision Tree	0.000000	0.000000	1.000000	156.638634	8.780548	0.248960	129.015316
4	Random Forest	11.229251	2.588221	0.949283	91.975119	6.953134	0.559004	82.042452
5	Support Vector Machine (SVM)	118.299495	8.323630	0.465702	139.610224	9.770720	0.330606	139.512706
6	XGBoost	0.000007	0.001948	1.000000	85.901304	6.448663	0.588126	88.823858
7	Neural Network	57.047406	6.187151	0.742346	68.501974	6.816589	0.671551	58.194408

Figure 15 - Metrics for the tested models

As its possible to see on the table, the Random Forest model stood out with a training R² of 0.95 and a low MSE of 11.23, suggesting a strong ability to fit the training data. However, its test performance deteriorated significantly, as indicated by a high MSE of 91.98 and a lower R² of 0.56. This discrepancy points to potential overfitting, where the model performs well on training data but fails to generalize effectively to unseen data.

Linear Regression showcased a more stable performance across both training and testing datasets. With an R² of 0.74 on the training set and 0.70 on the test set, it indicates that the model maintains a reasonable predictive capability without becoming overly complex. This consistency makes it a strong candidate for generalization in scenarios with limited data.

On the other hand, the Ridge and Lasso Regression models exhibited slightly worse performance compared to Linear Regression, with higher MSE values on both training and test sets. Although they incorporate regularization techniques to mitigate overfitting, their results indicate that the benefits may not be as pronounced in this specific dataset.

The Neural Network model achieved a reasonable performance as well, but like the Random Forest, it demonstrated signs of overfitting, as seen in the disparity between training and testing metrics. Its complexity did not translate into better generalization.

Despite testing various techniques, including grid search for hyperparameter tuning, no significant improvements were observed in the model performance. Therefore, Linear Regression remained the best choice for predicting Vmoy based on V, R, and A, particularly given the limited dataset size. Its balance of simplicity and effectiveness mitigates the risk of overfitting, ensuring better performance in practical applications.

6.3.3.2. Step 2: Non-parametric model

To ensure the model's robustness, non-parametric methods will be applied to the distance variables from the NCBS dataset, which are used as explanatory variables. Kernel Density Estimation (KDE) will be employed to estimate the distribution of these distance metrics non-parametrically, offering a detailed view of the data's variability without assuming a specific distribution shape.

The method will generate multiple resamples from the NCBS data to estimate the variability of the distance metrics. This variability will be incorporated into the regression model, allowing it to better account for uncertainty.

By integrating the uncertainty obtained from these sampling methods into the regression model, the study aims to provide more accurate and robust estimates of the confidence intervals for Vmoy.

First, a histogram (figure 16) of the original data distributions for V, R, and A is plotted to visualize their shapes. This histogram illustrates how the values are distributed across the dataset.

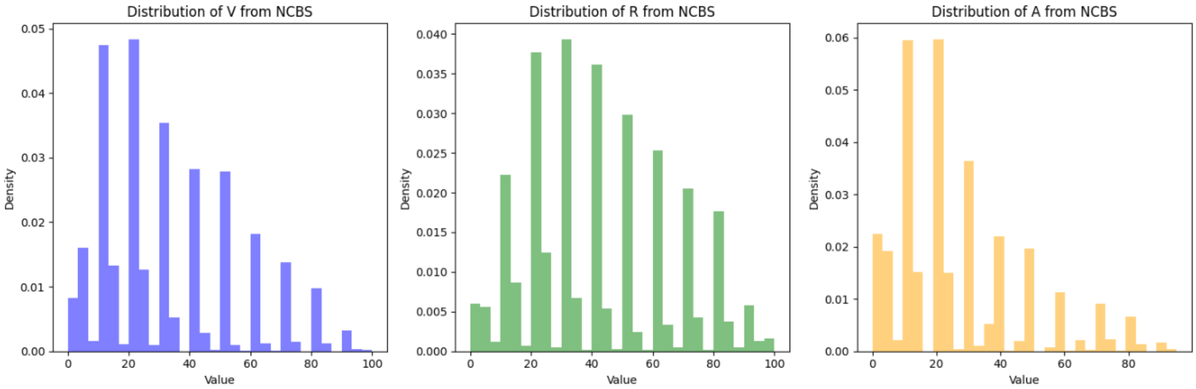


Figure 16 - Data distribution for V,R,A before the model

Additionally, kernel density estimation (KDE) with a Gaussian kernel is applied to create a smoother representation of the data distribution (figure 17). The KDE produces a continuous curve that represents the estimated density, facilitating a comparison with the original data distribution and providing insights into how well the KDE captures the underlying characteristics.

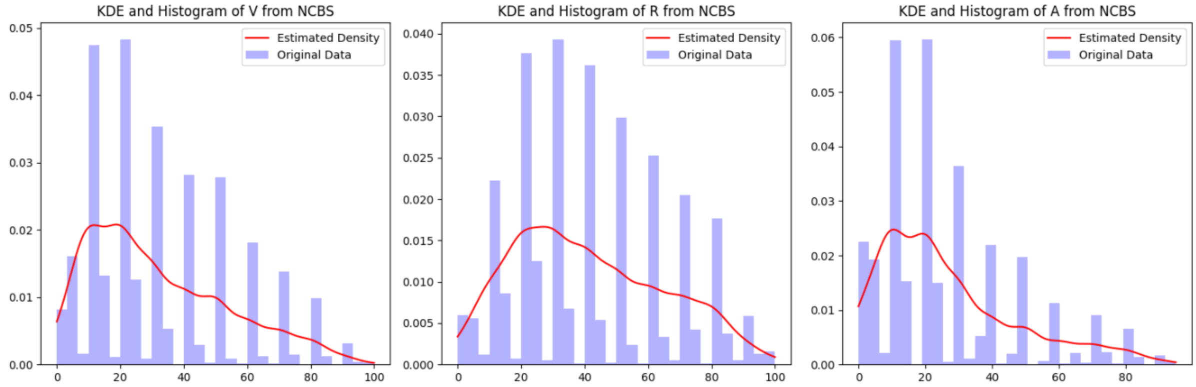


Figure 17 - Kernel Density estimation for V,R,A

Next, 10,000 samples are generated based on the estimated density. These samples help assess the behavior of the response variable under various scenarios, thereby supporting the analysis of confidence intervals. The distribution of these generated samples is then visualized (figure 18), allowing for a comparison with the original data and reinforcing the understanding of the variability expected in the response variable.

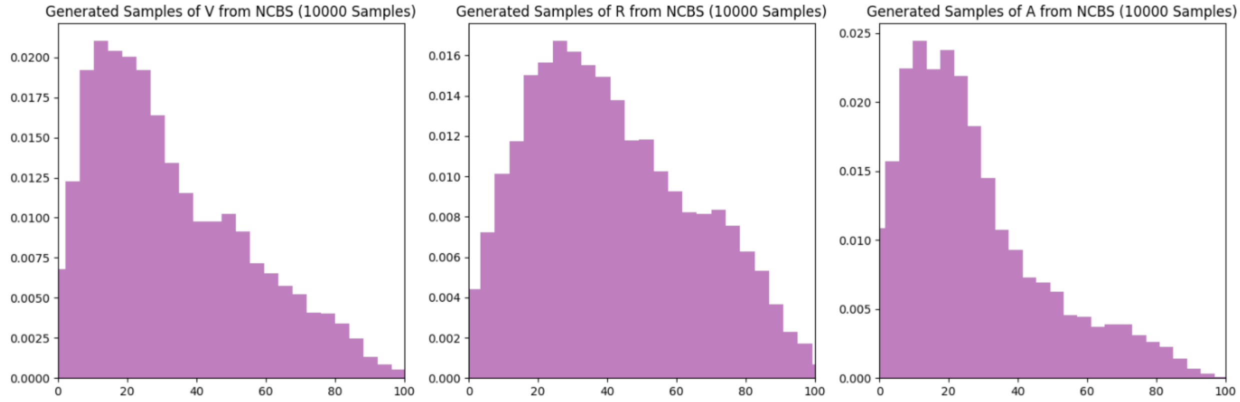


Figure 18 - Generated Samples for V,R,A

By the end these samples are concatenated to form a new dataset containing 10,000 observations for each variable. The concatenation enables a multidimensional exploration of how these variables interact, ultimately contributing to more reliable predictions and insights.

6.3.3.3. Step 3: S – Curve

Finally, a curve S will be plotted, showcasing the estimated Vmoy along with the corresponding confidence intervals, following the approach used in previous case studies. This visualization will provide a clear representation of the variable's behavior and the reliability of the predictions, ensuring that the confidence intervals reflect the true characteristics and variability of the underlying data.

This methodology not only addresses the biases inherent in the non-random Real Customer Database but also strengthens the validity of the predictions by ensuring that the confidence intervals are accurate and representative of the data's true distribution.

While the 10,000 samples will not be used for training the model, they provide a larger dataset for examining the response variable's distribution and help visualize the S-curve (figure 19), illustrating how variations in V, R, and A impact the response.

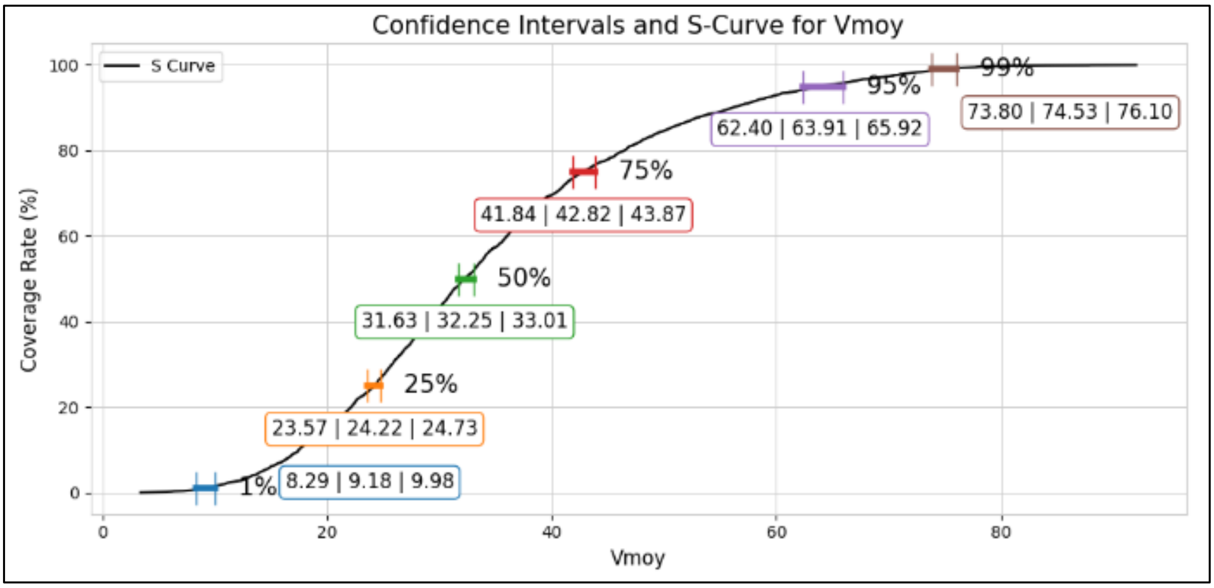


Figure 19 - Confidence Intervals and S-Curve for “Vmoy” after the models

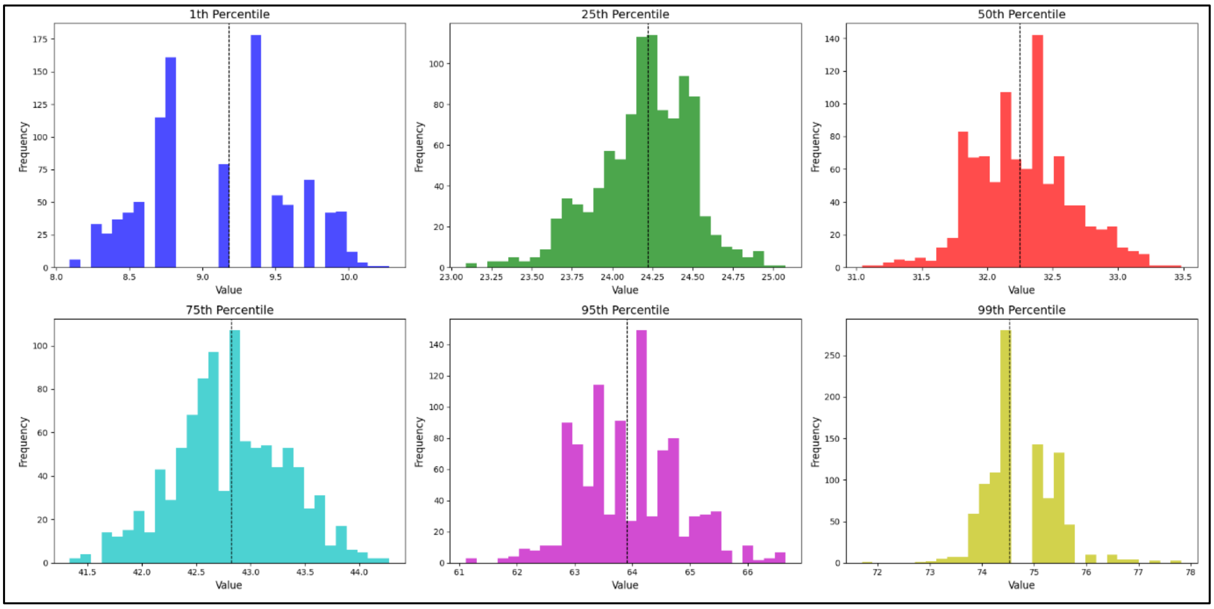


Figure 20 - Data distribution for the percentiles of the S-Curve: Study Case 3

The results of the percentiles (figure 20) indicate the distribution of the data across different ranges. For the 1st percentile, the mean is 9.12, and the variability is low (standard deviation of 0.49), showing that most values fall below this limit, although there are some outliers that raise the maximum to 10.91. In the 25th percentile, the mean increases to 24.21, with similar variability (0.29) and a concentrated range from 22.99 to 24.97, suggesting relatively homogeneous data.

The median (50th percentile) presents a value of 32.27, with a slightly higher standard deviation (0.38), reflecting a mild dispersion. In the 75th percentile, the mean is 42.81, with a standard deviation of 0.51, indicating a greater concentration of higher values, while the range expands from 41.28 to 44.26.

The 95th percentile reveals a mean of 63.97 and a standard deviation of 0.90, evidencing significant dispersion among the extreme data, with values ranging from 61.38 to 67.22. Finally, the 99th percentile has a mean of 74.79 and a standard deviation of 0.69, which is relatively low compared to the 95th percentile, which may seem unexpected. The range is from 72.95 to 77.82, showing that, despite the expectation of greater variability, the distribution of extreme data may be less dispersed due to specific characteristics of the sample.

These results suggest that while the lower percentiles exhibit low variability and concentration, the higher percentiles reflect greater dispersion, highlighting the complexity of the data distribution and the influence of factors such as the presence of outliers.

For comparison purposes, an S-curve for Vmoy was plotted using the data from the 100 customers (figure 21).

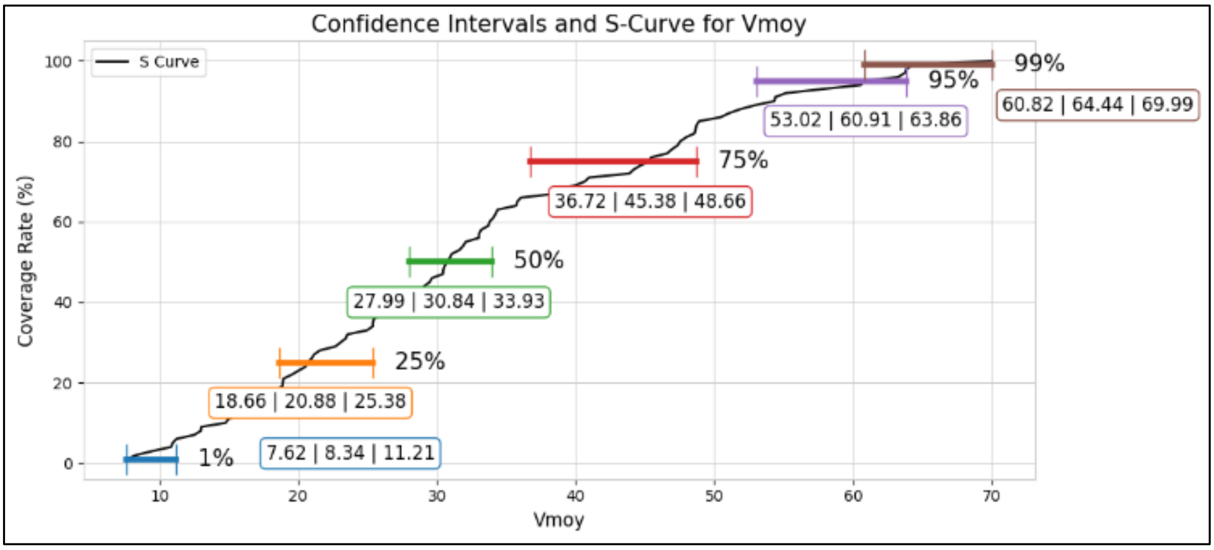


Figure 21 - Confidence Intervals and S-Curve for “Vmoy” before the models

As can be observed, the confidence intervals became narrower after modeling, which is generally considered positive. Narrower intervals indicate greater precision in the estimates, suggesting that the model is better capturing the relationship between the variables and reducing uncertainty in the predictions of Vmoy.

Additionally, the resulting curve became less distorted and more closely resembles an S-shape, which is a sign that the estimates are more robust and reliable. An S-shape suggests a more natural and continuous relationship between the variables, reinforcing the validity of the model.

In summary, the reduction in confidence intervals and the improvement in the shape of the curve indicate that the combination of the regression model with the

NCBS data has enhanced the quality of the analysis, leading to more accurate and reliable predictions.

6.5. Conclusions

The project developed during the internship at Renault Group focused on analyzing real usage data of vehicles to optimize the design and performance of components, particularly within the "Customer Usage" team of the Ampere Cars division. The methodology demonstrated a robust application of inferential statistics and predictive modeling techniques, reflecting a significant effort to capture the variability and uncertainty of data collected from real customers.

The use of different methods to calculate confidence intervals, such as bootstrap and jackknife, was essential for dealing with the non-normal nature of the data. The choice of non-parametric methods, especially bootstrap, proved robust for estimating the variability of extreme percentiles, which are critical for simulating intense usage scenarios. The comparison between methods revealed that bootstrap offers greater flexibility and reliability, especially with complex and non-normally distributed data.

Conducting three distinct case studies significantly contributed to understanding the usage patterns of electric vehicles and the challenges associated with modeling this data. The first case study focused on analyzing Car Data Usage (CDU) data, specifically the number of trips per vehicle per day, segmented by countries, helping to validate confidence interval methods and highlighting the need for representative samples. The second study analyzed automotive differential behavior, emphasizing how critical operating conditions impact component performance and reveal uncertainties associated with extreme usage. In the third study, Client Base data was explored, using real customer data to generate more representative simulations and applying regression models and non-parametric techniques, such as Kernel Density Estimation (KDE), to capture the relationship between speed variables and driving profiles on different road types. This approach allowed for more detailed modeling of real conditions, ensuring that the results reflected user behavior.

In all case studies, it was observed that higher percentiles, such as the 95th and 99th, exhibited larger confidence intervals, reflecting greater uncertainty and variability. These wide intervals indicate that in extreme usage scenarios, there is a higher data dispersion, making estimates less precise. This highlights the importance of carefully selecting these percentiles in analyses, as they can distort the perception of typical user behavior and impact strategic decisions, especially in simulations that depend on predicting high-demand scenarios. The consistency of this observation across the three studies reinforces the need for a thorough analysis of extreme data to ensure informed decisions regarding vehicle performance.

The insights gained from modeling real usage data provided valuable feedback for vehicle development, enabling adjustments to critical components and anticipating maintenance needs. Analyzing real usage patterns allowed for the identification of areas where user experience could be improved, contributing to solutions more tailored to the regional and cultural needs of consumers.

The project demonstrated the importance of a meticulous approach to analyzing real usage data of vehicles, combining advanced statistics with predictive modeling to optimize the development of electric vehicles. The methodologies employed reinforced Renault's ability to create products that meet customer expectations in a competitive market, establishing a solid foundation for future innovations in sustainable mobility.

6.6. Future Applications and Improvements

a. Incorporation of Confidence Interval Calculations in the Workflow: A future application could involve integrating the calculation of confidence intervals directly into the Customer Usage workflow. By incorporating this statistical analysis at various stages, the team can better quantify the uncertainty and variability in customer usage data, leading to more robust customer usage models. This would enhance the accuracy of simulations and optimizations for powertrain systems and vehicle components, ensuring that the recommendations are based on a clearer understanding of potential variations in real-world driving scenarios.

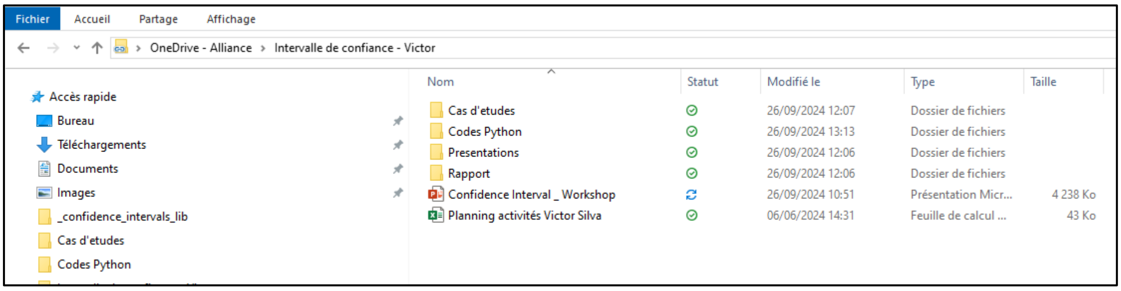
b. Integration with External Data and Real-Time Feedback: Continuous integration of real-time feedback from connected vehicles can further enrich predictive models, allowing for dynamic adjustments in product design and preventive maintenance strategies.

c. Enhancements to the Base Client Model: Improvements to the Base Client model could involve exploring advanced machine learning techniques, evaluating regression model assumptions, and expanding cross-validation methods. These enhancements would increase the model's accuracy and representativeness, allowing the team to capture more complex patterns in the data and ensure better generalization for new data scenarios.

d. Expansion of Analysis to New Segments: Broadening the analysis to include data from new vehicle models or exploring specific segments could provide additional insights and adapt the developed methodologies for an even wider range of applications.

6.7. Other Deliverables

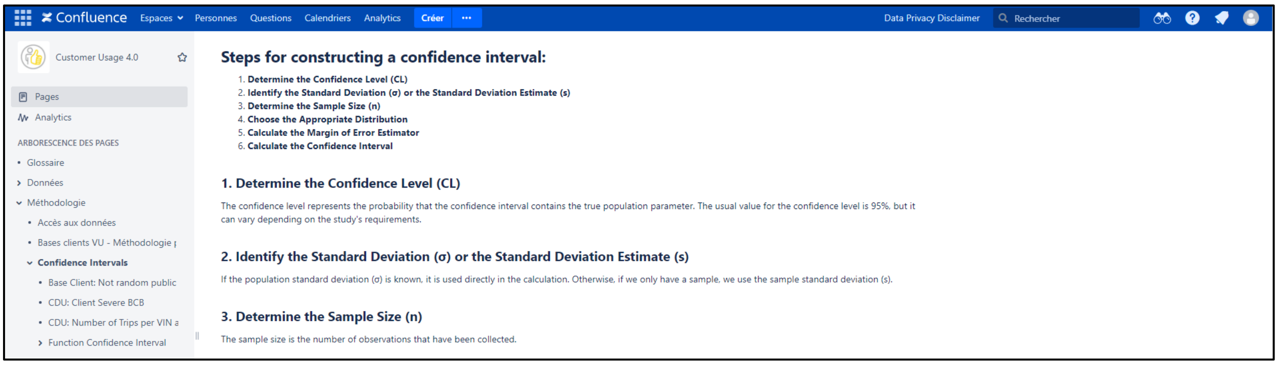
In addition to the case studies and the Python library, comprehensive documentation of the methodology was developed (figures 22, 23, 24, 25). All deliverables are organized in a OneDrive folder, including codes, presentations, organizational charts, case studies, and other relevant materials.



Nom	Statut	Modifié le	Type	Taille
Cas d'etudes	✓	26/09/2024 12:07	Dossier de fichiers	
Codes Python	✓	26/09/2024 13:13	Dossier de fichiers	
Presentations	✓	26/09/2024 12:06	Dossier de fichiers	
Rapport	✓	26/09/2024 12:06	Dossier de fichiers	
Confidence Interval _ Workshop		26/09/2024 10:51	Présentation Micr...	4 238 Ko
Planning activités Victor Silva	✓	06/06/2024 14:31	Feuille de calcul ...	43 Ko

Figure 22 - One Drive Example

The full details of the internship and the methodologies employed are available in this report.



Steps for constructing a confidence interval:

1. **Determine the Confidence Level (CL)**
2. **Identify the Standard Deviation (σ) or the Standard Deviation Estimate (s)**
3. **Determine the Sample Size (n)**
4. **Choose the Appropriate Distribution**
5. **Calculate the Margin of Error Estimator**
6. **Calculate the Confidence Interval**

1. Determine the Confidence Level (CL)

The confidence level represents the probability that the confidence interval contains the true population parameter. The usual value for the confidence level is 95%, but it can vary depending on the study's requirements.

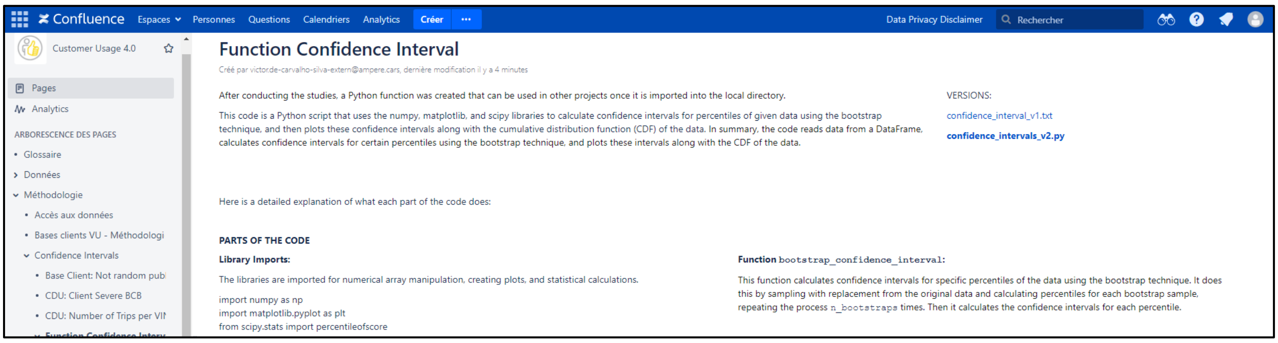
2. Identify the Standard Deviation (σ) or the Standard Deviation Estimate (s)

If the population standard deviation (σ) is known, it is used directly in the calculation. Otherwise, if we only have a sample, we use the sample standard deviation (s).

3. Determine the Sample Size (n)

The sample size is the number of observations that have been collected.

Figure 23 - Documentation of methodology – Example



Function Confidence Interval

Créé par victor.de-carvalho-silva-extern@ampere.cars, dernière modification il y a 4 minutes

After conducting the studies, a Python function was created that can be used in other projects once it is imported into the local directory.

This code is a Python script that uses the numpy, matplotlib, and scipy libraries to calculate confidence intervals for percentiles of given data using the bootstrap technique, and then plots these confidence intervals along with the cumulative distribution function (CDF) of the data. In summary, the code reads data from a DataFrame, calculates confidence intervals for certain percentiles using the bootstrap technique, and plots these intervals along with the CDF of the data.

VERSIONS:

confidence_interval_v1.txt

confidence_intervals_v2.py

Here is a detailed explanation of what each part of the code does:

PARTS OF THE CODE

Library Imports:

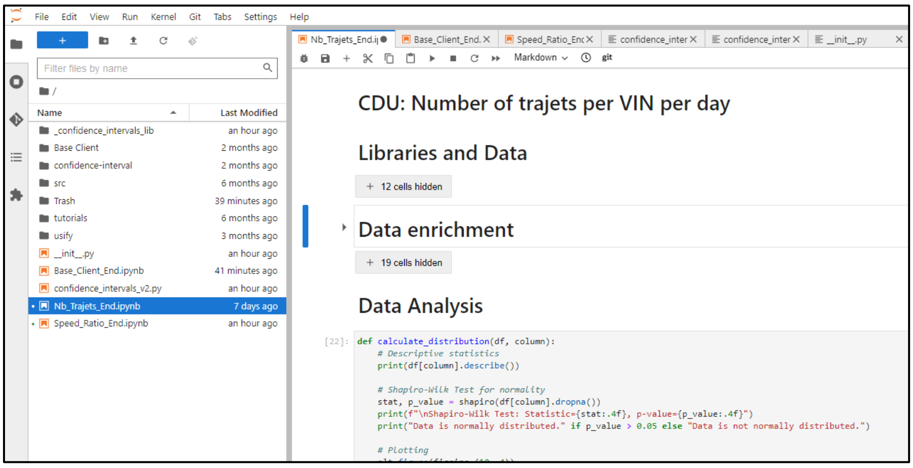
The libraries are imported for numerical array manipulation, creating plots, and statistical calculations.

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import percentileofscore
```

Function `bootstrap_confidence_interval`**:**

This function calculates confidence intervals for specific percentiles of the data using the bootstrap technique. It does this by sampling with replacement from the original data and calculating percentiles for each bootstrap sample, repeating the process `n_bootstraps` times. Then it calculates the confidence intervals for each percentile.

Figure 24 - Documentation of Python Function – Example



Name	Last Modified
_confidence_intervals_lib	an hour ago
Base Client	2 months ago
confidence-interval	2 months ago
src	6 months ago
Trash	39 minutes ago
tutorials	6 months ago
usify	3 months ago
__init__.py	an hour ago
Base_Client_End.ipynb	41 minutes ago
confidence_intervals_v2.py	an hour ago
Nb_Trajets_End.ipynb	7 days ago
Speed_Ratio_End.ipynb	an hour ago

CDU: Number of trajets per VIN per day

Libraries and Data

+ 12 cells hidden

Data enrichment

+ 19 cells hidden

Data Analysis

```
[22]: def calculate_distribution(df, column):
          # Descriptive statistics
          print(df[column].describe())

          # Shapiro-Wilk Test for normality
          stat, p_value = shapiro(df[column].dropna())
          print(f"\nShapiro-Wilk Test: Statistic={stat:.4f}, p-value={p_value:.4f}")
          print("Data is normally distributed." if p_value > 0.05 else "Data is not normally distributed.")

          # Plotting
```

Figure 25 - Study Case on Jupyter Notebook – Example

7. INTERNSHIP OVERVIEW

7.1. Strategic Benefits for the Company

The methodologies developed during the internship have the potential to bring significant strategic benefits to Renault Group, particularly in the context of electric vehicle development and the shift towards sustainable mobility. By leveraging advanced data analysis techniques, specifically the application of confidence intervals, Renault can enhance its understanding of customer usage patterns and gain crucial advantages across several areas.

One key possibility is improving the product development process. Confidence interval analysis applied to real customer data would provide more accurate insights into how vehicles are used in different conditions, identifying variations and extremes that could impact performance. This would allow engineers to fine-tune designs to better meet real-world customer needs, resulting in more reliable and efficient products. Additionally, the ability to simulate scenarios using confidence intervals would enable Renault to test new vehicle models in diverse usage contexts, accelerating development cycles and reducing time to market.

Another strategic benefit lies in regional vehicle customization. Confidence interval-based analysis, segmented by different markets and regions, would allow Renault to tailor its vehicles to specific customer demands in various parts of the world. This could significantly enhance the company's competitiveness by offering vehicles better suited to local factors such as climate, infrastructure, and driving habits.

The integration of real-time data from connected vehicles, combined with confidence interval calculations, presents an opportunity to incorporate this analysis into Renault's operational processes. This could lead to a more proactive predictive maintenance strategy, using data variability to anticipate issues before they become critical. This approach would not only improve vehicle durability and reliability but also lower operational costs and increase customer satisfaction by offering personalized, preventive maintenance services.

In summary, the application of confidence intervals in analyzing real customer usage data offers Renault the potential to enhance product efficiency, foster innovation, and better respond to customer demands in a growing market for electric mobility solutions.

7.2. Internship Contributions to Student Growth

The internship at Renault has been a significant milestone in my career as a data scientist. The experience of working in a large company has provided me with invaluable exposure to a professional environment, improving my ability to analyze and interpret real-world data and complex systems.

During the internship, I had the opportunity to develop my soft skills, such as communication, teamwork, and adaptability, which are essential in any professional setting. Working in an international company also improved my language skills, particularly in English and French, and broadened my perspective on global issues and trends.

The knowledge and skills learned through the Data Analysis and Network Intelligence (DANI) master's program were instrumental in the successful completion of the internship. The coursework and practical projects in the DANI program provided a solid foundation in statistical analysis, machine learning, and data visualization, which were essential for analyzing and interpreting the large volumes of data collected during the internship.

It was incredibly rewarding to see the theoretical concepts learned in the DANI program being applied in a real-world setting. The internship provided me with the ideal opportunity to put my skills and knowledge into practice.

Overall, the combination of the DANI program and the internship at Renault has provided me with a well-rounded education in data science, equipping me with the skills and knowledge necessary to succeed in this exciting field. I am grateful for the opportunity to work with talented and dedicated colleagues, and I look forward to applying the experience gained during the internship to future projects and endeavors.

8. BIBLIOGRAPHY

[1] Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer.

[2] Blitzstein, J. K., & Hwang, J. (2014). *Introduction to Probability*. CRC Press.

[3] Conover, W. J. (1999). *Practical Nonparametric Statistics* (3rd ed.). Wiley.

[4] Devore, J. L. (2011). *Probability and Statistics for Engineering and the Sciences* (8th ed.). Cengage Learning.

[5] Efron, B. (1979). Bootstrap methods: Another look at the jackknife. *Annals of Statistics, 7*(1), 1-26.

[6] Freedman, D., & Pisani, R. (2007). *Statistics*. Norton & Company.

[7] Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., & Rubin, D. B. (2013). *Bayesian Data Analysis* (3rd ed.). CRC Press.

[8] Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* (2nd ed.). Springer.

[9] James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An Introduction to Statistical Learning with Applications in R*. Springer.

[10] Montgomery, D. C., & Runger, G. C. (2014). *Applied Statistics and Probability for Engineers* (6th ed.). Wiley.

[11] Navidi, W. (2015). *Statistics for Engineers and Scientists* (4th ed.). McGraw-Hill.

[12] Renault Annual Report 2023. Available in the financial reports section of the Renault Group website.

[13] Renault Group Official Website. https://www.renaultgroup.com

[14] Triola, M. F. (2017). *Elementary Statistics* (13th ed.). Pearson.

[15] Wasserman, L. (2004). *All of Statistics: A Concise Course in Statistical Inference*. Springer.

Appendices

Python Function developed:

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import percentileofscore

def calculate_cdf(data):
    """
    Calculate the cumulative distribution function (CDF) for the given data.
    """
    sorted_data = np.sort(data)
    return sorted_data, [percentileofscore(sorted_data, value) for value in sorted_data]

def bootstrap_confidence_interval(data, percentiles=(1, 25, 50, 75, 95, 99), confidence=0.95, n_bootstraps=1000):
    """
    Calculate the confidence interval for percentiles using the bootstrap method.

    Parameters:
    - data: The dataset to analyze.
    - percentiles: List of percentiles to calculate (default is [1, 25, 50, 75, 95, 99]).
    - confidence: Confidence level for the interval (default is 0.95).
    - n_bootstraps: Number of bootstrap samples to generate (default is 1000).
    """
    bootstraps = [np.percentile(np.random.choice(data, size=len(data), replace=True), percentiles) for _ in range(n_bootstraps)]
    return {p: (np.percentile(bootstraps, (1-confidence)/2*100, axis=0)[i],
                np.percentile(bootstraps, (1+confidence)/2*100, axis=0)[i]) for i, p in enumerate(percentiles)}

def plot_confidence_intervals(df, column=None, percentiles=[1, 25, 50, 75, 95, 99], confidence=0.95, n_bootstraps=1000):
    """
    Plot S-curves and confidence intervals for one or all columns of a DataFrame.

    Parameters:
    - df: DataFrame containing the data.
    - column: Specific column to plot (None to plot all columns).
    - percentiles: List of percentiles to calculate (default is [1, 25, 50, 75, 95, 99]).
    - confidence: Confidence level for the interval (default is 0.95).
    - n_bootstraps: Number of bootstrap samples to generate (default is 1000).
    """
    if column:
        # Plot for a single column
        columns = [column]
    else:
        # Plot for all columns
        columns = df.columns

    n_cols = 2  # Number of columns in the subplot grid
    n_rows = (len(columns) + n_cols - 1) // n_cols  # Number of rows in the subplot grid

    fig, axs = plt.subplots(n_rows, n_cols, figsize=(20, 5 * n_rows))
    axs = axs.flatten()

    for i, col in enumerate(columns):
        data = df[col].dropna()  # Remove NaN values
        sorted_data, cdf = calculate_cdf(data)
        intervals = bootstrap_confidence_interval(data, percentiles, confidence, n_bootstraps)
        ax = axs[i]

        # Plot the S-curve
        ax.plot(sorted_data, cdf, color='k', label='S Curve')
    # Plot confidence intervals and percentiles
    for j, p in enumerate(percentiles):
        interval = intervals[p]
        percentile_value = np.percentile(data, p)
        ax.plot(interval, [p, p], '|-', markersize=20, color=f'C{j}', linewidth=4)
        ax.text(interval[1], p, f'   {p}% ', fontsize=15, verticalalignment='center')
        ax.annotate(f'{interval[0]:.2f} | {percentile_value:.2f} | {interval[1]:.2f}',
                    (percentile_value, p), textcoords="offset points", xytext=(0, -30), ha='center',
                    fontsize=12, bbox=dict(boxstyle="round,pad=0.3", edgecolor=f'C{j}', facecolor="white"))

    # Set axis limits, labels, and titles
    ax.set_xlabel(col, fontsize=12)
    ax.set_ylabel('Coverage Rate (%)', fontsize=12)
    ax.set_title(f'Confidence Intervals and S-Curve for {col}', fontsize=15)
    ax.legend(loc='best', fontsize=10)
    ax.grid(True)
    ax.tick_params(axis='both', labelsize=10)

# Turn off unused subplots
for i in range(len(columns), len(axs)):
    axs[i].axis('off')

plt.tight_layout()
plt.show()
```