A dissertation submitted to the **University of Greenwich**
in partial fulfilment of the requirements for the Degree of

Master of Science
in
Management of Business Information Technology

Assessing Data Quality Management Issues in Open-Sourced Data Sets with Statistical Methodology

Name: Omer Haluk Demirhan

Supervisor: Dr Solomon Ebenuwa
Submission Date: December, 2024
Word count: 14257

Assessing Data Quality Management Issues in
Open-Sourced Data Sets with Statistical Methodology

Omer Haluk Demirhan

Computing & Mathematical Sciences, University of Greenwich, 30 Park Row, Greenwich, UK.

(Submitted 20 December 2024)

ABSTRACT. Open-source datasets are critical resources in the data-driven era, driving innovation across fields like artificial intelligence, machine learning, and business intelligence. However, these datasets often suffer from inconsistent quality, hindering their effective application. This dissertation addresses this challenge by developing a comprehensive R package, DataQualityPackage, for assessing and improving the quality of open-source datasets. The framework evaluates key data quality dimensions, including completeness, accuracy, consistency, and timeliness, while incorporating ethical considerations such as data privacy and bias detection.

A design science methodology guided the package's development, including features like missing data imputation, outlier detection, correlation analysis, and distribution assessment. The package's effectiveness was validated using datasets from platforms like Kaggle, demonstrating its capability to identify and mitigate quality issues. Visual and statistical results confirm the framework's utility in enhancing data reliability.

This research provides a standardised tool for data quality assessment that contributes to academic and industry practices. By addressing existing gaps, the framework sets a foundation for scalable and ethical data validation methods. Future work will focus on expanding the package's features and exploring its integration with advanced data analytics pipelines.

Keywords: Open-source data quality; Data validation framework; R programming; Data completeness; Data accuracy; Data consistency; Data timeliness; Outlier detection; Missing data imputation;

Acknowledgements

First and foremost, I would like to express my deepest gratitude to **Dr Solomon Ebenuwa**, my supervisor, for his unwavering support, insightful feedback, and invaluable guidance throughout the lifecycle of this MSc project. His mentorship and encouragement have been instrumental in shaping the direction and quality of this work, and I am deeply appreciative of the time and effort he dedicated to my success.

A special and heartfelt thank you goes to my mother, whose constant love, encouragement, and belief in me have been my foundation throughout this journey. Her support and sacrifices have been a source of strength, and I am profoundly grateful for all she has done to help me reach this milestone.

I would also like to extend my gratitude to the **University of Greenwich**, particularly the **Faculty of Computing & Mathematical Sciences**, for providing me with the opportunity to pursue this Master of Science in **Management of Business Information Technology**. The supportive learning environment, resources, and guidance offered throughout the programme have been pivotal to my academic and personal growth.

Finally, I extend my thanks to my family, friends, and peers who have offered their support, kind words, and motivation during this academic endeavour. To everyone who has contributed to this journey, whether directly or indirectly, your encouragement and assistance have been truly appreciated.

Table of Contents

Table of Figure

1. Introduction

Data repositories like the World Bank, Kaggle, and UCI datasets have become critical resources for both researchers and practitioners by offering open-access, cost-free data that can be easily accessed to facilitate the growth of technologies like artificial intelligence, and machine learning. However, there is a great deal of variability in the quality of many of these datasets, and this definitely raises concerns regarding the reliability of the research findings based on them.

Data quality is an important indicator of the maturity, completeness, accuracy, and stability of a dataset. Various models have been developed to define the criteria, parameters, and dimensions of data quality; these models, however, are basically tailored for internal organisational datasets and do not fully accommodate the new complexities brought by open-source datasets.

Data quality assessment models for open-source data are still being developed and tend to work as complementary tools to traditional data quality assessment methods.

These activities include, but are not limited to, data preprocessing, database configuration, data migration, and data conversion; these processes are usually tailored to a specific dataset or adapted from classic data quality management methodologies. The existing frameworks for assessing open-source data quality maturity are therefore only partial solutions and leave a big gap where comprehensive and standardized approaches should be. The abundance of data in modern society presents a huge challenge, especially in critical fields like healthcare, economics, and social sciences, where even small errors in data quality can lead to enormous and far-reaching consequences. While statisticians, econometricians, and data scientists have proposed many methods for validation, there is still no general framework for validating and comparing open-source datasets from a quality perspective.

The absence of a concerted effort to achieve this reinforces the urgent need for a strong framework that would comprehensively and systematically address the key aspects of data quality. The goal of this dissertation is to fill this gap by providing a comprehensive framework for assessing the quality of open-source datasets, which has been implemented as a package in the R programming language. The contribution of this research to the field of data quality management is of great importance, given that the systematic approach introduced for the assessment of data quality is easily scalable and applicable for researchers and practitioners.

1.1 Background and Context

In data-dependent fields like data science, machine learning, artificial intelligence, and the social sciences, it is of utmost importance that the quality of data be given attention. There exists a broadly accepted consensus that high-quality data integrity forms the basis for ensuring the validity, reliability, and generalizability of research outcomes. The ethical and professional standards that prevail in scientific research emphasise the importance of linking the quality of research to the quality of data, whether this data supports new hypotheses or adds to a consensus among researchers working together to achieve broader investigative goals.

Although open-source data has opened the doors to vast amounts of information that were previously unimaginable, there is no guarantee that such data has been stringently checked for quality. Insufficient control increases the likelihood of faulty research results, an eventuality that is exacerbated when such open-source data is in use to drive critical choices within health and policy discourse. Moreover, the fact that such data is being openly accessible via APIs or through different platforms makes evaluation difficult for users regarding quality aspects. The lack of standardised data validation policies for open-source datasets raises questions about the reliability of research based on such data. Besides, it also shows the need for tools and frameworks that can systematically evaluate data quality in order to ensure its reliability. To address these gaps, the present study strives to put data scientists and researchers in a position to make very informed decisions regarding the datasets they use.

1.2 Research Problem and Motivation

Data quality is a cornerstone of data science, machine learning, artificial intelligence, and social science research. The processes of data collection and primary or secondary data analysis have traditionally worked in tandem to achieve research objectives. Every ethical statement issued by a scientific association emphasises the inextricable link between research quality and the quality of the underlying data. Whether the data supports groundbreaking hypotheses or contributes to a broader collective understanding, its quality directly impacts the reliability and credibility of the research. Data quality, therefore, is an attribute that permeates all forms of research, from testing hypotheses to forming generally accepted conclusions or communicating facts in broader contexts. This underscores the critical importance of maintaining high standards of data quality in all research contexts.

Over recent years, the proliferation of big data and open-source, publicly available datasets has provided researchers and practitioners with unprecedented access to vast amounts of information. However, data scientists face a fundamental challenge: there is no assurance that those who compiled these datasets considered any form of quality assessment. While the availability of such data represents a treasure trove for researchers, it also introduces significant risks to the foundations of research integrity. It is imperative for data scientists to exercise caution and remain acutely aware of these risks when relying on open-source data.

The issue is exacerbated by the widespread use of open-source datasets by audiences beyond the academic community. These audiences, including practitioners, policymakers, and industry stakeholders, may not be fully cognisant of the limitations and caveats associated with the quality of such datasets. In extreme cases, technology providers offer application programming interfaces (APIs) to access these datasets, making it nearly impossible for end-users to infer any indicators of data quality. Analysts and decision-makers relying solely on open-source datasets risk

undermining the power and reliability of established scientific methods. This raises an urgent question: can we retain the integrity of scientific inquiry in this context, or must we develop new regulatory tools to ensure data quality beyond traditional academic settings?

Reliability is the bedrock of research validity. The quality of data used in any study profoundly affects its validity, reliability, and generalisability. In the social sciences, the validity of research conclusions often depends on the robustness of the data. Sociological research, for example, strives for generalisability—not by directly applying study results to an entire population, but by testing the validity of inferences and evaluating the broader applicability of findings. Achieving this requires rigorous scrutiny of the data, including its methodological reliability and the accuracy of its content.

The growing reliance on open-source datasets has amplified the need for standardised tools to assess and validate data quality. However, existing methods are fragmented and lack a comprehensive approach tailored to the complexities of open-source data. This research addresses this gap by proposing a systematic framework for evaluating the quality of open-source datasets, implemented as a package in the R programming language. By developing robust metrics and tools for assessing dimensions such as accuracy, completeness, and reliability, this project aims to equip researchers and practitioners with a scalable solution to enhance the integrity of their analyses.

1.3 Aims and Objectives

This research aims to address the growing challenges in assessing the quality of open-source datasets by developing a comprehensive framework, implemented as an R programming package. The framework will evaluate key data quality dimensions such as completeness, accuracy, consistency, and timeliness, offering a scalable and standardised tool for researchers, data scientists, and practitioners to improve data reliability and decision-making processes.

Identify Key Data Quality Dimensions: Define and elaborate on essential dimensions of data quality, including completeness, accuracy, consistency, and timeliness, to establish a robust foundation for evaluating open-source datasets.

Develop an R Package for Data Quality Assessment: Design and implement a user-friendly R package that integrates systematic metrics and methods, enabling researchers and practitioners to evaluate and enhance the quality of open-source datasets.

Validate the Framework with Real-World Data: Apply the developed framework to real-world datasets, such as those from Kaggle or UCI repositories, to demonstrate its effectiveness in identifying and addressing data quality issues.

1.4 Research Questions

This research is guided by the following key questions:

1. **What are the essential dimensions of data quality for evaluating open-source datasets?**
    - Focuses on identifying the critical aspects such as completeness, accuracy, consistency, and timeliness that determine the reliability of datasets.
2. **How can effective metrics and methods be developed to systematically assess these data quality dimensions?**
    - Explores the creation of standardised and reproducible approaches for evaluating data quality.
3. **What are the technical challenges in developing a comprehensive framework for data quality assessment in open-source datasets?**
    - Investigates the difficulties in designing and implementing a scalable and customisable solution, including the integration of statistical algorithms and usability features.
4. **How effective is the proposed R package in identifying and addressing data quality issues in real-world datasets?**
    - Evaluates the framework's performance and practical utility using real-world datasets, such as those from Kaggle or UCI repositories.

1.5 Limitations

Although the framework is designed for wide applicability, validation is confined to publicly available datasets that are free from proprietary restrictions. Hence, the study overlooks either the domain-specific requirements or the unique challenges raised by private datasets or those from very specific domains.

Furthermore, the computational methods and resources used in the R package require at least a moderate level of programming and statistical analysis expertise, which could be a barrier for non-technical users. The future direction of research is therefore devoted to making the package more user-friendly and increasing its functionalities to allow for domain-specific applications.

In summary, although the proposed framework covers many of the important problems in data quality, it cannot be exhaustive, especially those problems caused by highly complicated or fast-changing datasets. It is therefore very much encouraged to further extend this framework and adapt it to new challenges emerging in this field of data quality management.

1.6 Legal and Ethical Issues

From a legal perspective, compliance with data licensing agreements, such as those under Creative Commons or Apache licenses, is essential to avoid intellectual property infringements. Open-source datasets often come with specific terms regarding attribution, redistribution, and use, which must be adhered to during development and testing. Additionally, the potential use of the tool in high-stakes fields necessitates vigilance against perpetuating biased or discriminatory outcomes, which could violate anti-discrimination laws and ethical standards.

Ethically, the research must prioritise transparency and accountability, ensuring that users clearly understand the methods, assumptions, and limitations embedded within the framework. Given the challenges posed by open-source data, including inherent biases and inconsistencies, the package must be equipped to identify and mitigate these issues effectively to avoid skewed analyses or unjust outcomes. The promotion of open science is another key ethical consideration; releasing the package under a permissive licence would encourage collaboration and foster trust in the tool. By addressing these considerations comprehensively, the research upholds both legal and ethical standards, ensuring its contribution to open-source data quality assessment is both rigorous and responsible.

1.7 Structure of the Dissertation

Chapter 1: Introduction
This chapter sets the stage for the research, providing an overview of the background, rationale, aims, and objectives. It introduces the research problem and the questions driving the study, alongside highlighting the significance of addressing data quality issues in open-source datasets. Limitations of the research are discussed, and the chapter concludes with a breakdown of the dissertation structure.

Chapter 2: Literature Review
This chapter critically examines existing frameworks, methodologies, and tools for assessing data quality. It identifies gaps in current research, particularly the challenges unique to open-source datasets. Ethical and professional considerations related to data quality management are also explored, providing a comprehensive context for the research.

Chapter 3: Methodology
This chapter describes the research design and methodology, including the development of the DataQualityPackage in R. It elaborates on the design science approach used to identify key data quality dimensions, create innovative assessment tools, and validate the framework using real-world datasets. Statistical and computational methods are detailed to ensure transparency and reproducibility.

Chapter 4: Implementation
This chapter focuses on the technical development of the R package. It explains the package's architecture, core functionalities, and features such as missing data analysis, outlier detection, and redundancy checks. It also addresses challenges encountered during development, such as ensuring scalability, compatibility, and user-friendliness, and details how these were overcome.

Chapter 5: Evaluation and Results
This chapter presents the results of applying the framework to real-world datasets, including examples from Kaggle. It includes detailed analyses, visualisations, and comparisons with existing data quality tools. The outcomes demonstrate the framework's effectiveness in identifying and addressing common data quality issues, such as missing values, inconsistencies, and anomalies.

Chapter 6: Conclusions and Recommendations
The final chapter summarises the key findings of the research and evaluates the extent to which the objectives were achieved. It offers actionable recommendations for future development of the framework and reflects on the personal and academic journey undertaken during the research.

2. Literature Review and Background Research

2.1 Overview of Data Quality in Open-Source Datasets

2.1.1 Understanding Data Quality

Data fidelity is crucial in machine learning and artificial intelligence (AI) applications, as it directly impacts model performance and reliability. Compliance with quality standards can be assessed using various software tools designed to evaluate data quality. For instance, TensorFlow Data Validation (TFDV) is an open-source library that facilitates the exploration and validation of machine learning data, ensuring adherence to quality standards (TensorFlow Data Validation, n.d.)

Data fidelity can be assessed by examining compliance with quality standards, and various software tools have been developed over the years to assist in this process.
For instance, data reliability in machine learning models can be inferred from the consistency of results across repeated experiments and the robustness of the model under various testing conditions. In these contexts, accuracy is evaluated by verifying the consistency of the labels and the precision of feature extraction. For example, when working with image datasets, labels must correspond accurately to the visual data they represent, and features like object boundaries should be consistently detected across varying conditions. These assessments aim to reduce uncertainty in model predictions and enhance confidence in the resulting insights.

Guidelines are necessary to establish parameters that enable data scientists and AI researchers working with open-source data to rely on specific statistical properties to ensure data quality. For example, financial data in AI models must adhere to fundamental principles, avoiding errors like negative pricing, redundancy, or invalid trading volumes. Similarly, in environmental monitoring systems driven by machine learning models, data quality is critical and must be validated to ensure that the measurements of physical and chemical environmental properties are accurate. The ISO/IEC 5259-1:2024 standard provides a framework for assessing and enhancing data quality across different phases of the data life cycle, which is crucial for reliable analytics and machine learning outcomes (ISO/IEC 5259-1:2024, n.d.)

The validation step is essential to extract the advertised potential value of data before transforming it into actionable information and knowledge. Unfortunately, current practices for data validation are often specific to particular domains or tools, and beyond a few domain-specific standards, there is little protocol-level guidance on how to realise data validation and its implications, particularly for open-source data in AI and ML applications. A survey on data quality dimensions and tools for machine learning highlights the challenges and emerging trends in this area, emphasizing the need for a comprehensive understanding of data quality in machine learning to drive progress in data-centric AI (Zhou et al., 2024)

The importance of validating data has been emphasised in multiple scientific domains, as well as in commercial contexts. Domain-specific concepts such as data quality assurance, data integrity, and database validation are central to maintaining data accuracy in machine learning models. Although these terms have evolved with new data models and integration frameworks, their core meaning remains the same. As the field progresses, a clear definition of data validation is needed to provide future direction for AI and data science research.

2.1.2 Conceptual Framework

In our line of work, we meticulously assess the quality of datasets through a thorough examination of various qualitative aspects, both visually and computationally. These dimensions encompass data relevance, consistency, completeness, accuracy and correctness, timeliness, integrity, confidentiality, and innovations. We equip you with an array of effective visualisation tools and offer valuable recommendations for conducting tests and checks to gauge the quality of the dataset. (Pipino, Lee, & Wang, 2002) Additionally, we delve into the specifics of a specialised dataset structure that can prove useful in scenarios where the properties of the dataset may lend themselves to predicting certain well-known fractions. To facilitate automated evaluation, we present a highly adaptable framework designed for the comprehensive evaluation of datasets. This framework is tailored to meet the needs of dataset producers, such as data agencies or dataset owners who seek to exercise control over their data prior to its publication. Furthermore, it is designed to handle open data, proprietary and private data, as well as data accessible at other known levels. Importantly, this package is not confined to relational, geospatial, or semantic datasets, but rather, it is particularly advantageous when dealing with unruly data. The framework is available separately and as part of an R package, thereby allowing for practical testing and real-life application within the R statistical software system. (Wickham, 2019)

2.2 Key Dimensions of Data Quality

Considering features with lower quality values tend to be associated with trade-offs. Thus, it is necessary to prioritise and focus on these features for increased visibility and priority. Addressing these issues will help to establish the value of open data investments, thus enhancing the quality of data and getting a better appreciation (Batini & Scannapieco, 2016). This approach is designed to guide municipalities in creating processes and performance measurements to help evaluate how well data meets the requirements for decision-making and create measures for tracking the quality of data and improvement over time. (ISO/IEC 25024:2015)

In this paper, the proposed internal data quality assessment provides a measurement of the overall quality of data in several key dimensions, such as completeness, uniqueness, timeliness, consistency, and accuracy, amongst others. Using a comprehensive data quality scorecard and a risk-based approach may help departments identify the most important and useful metrics and data required to expedite internal processing. (Pipino, Lee, & Wang, 2002)

The data quality assessment framework is based on the five main dimensions of quality: accuracy, completeness, reliability, consistency, and unbiasedness. Broadly adapted from Kuno and Juvan (1995). These dimensions were originally developed for databases and are generally adopted for open-source data sets. The assessment framework involves testing and analysing data to manage and improve data quality through the identification and resolution of inconsistencies in the data set. The framework recognises that key dimensions of quality (accuracy, completeness, reliability, consistency, and unbiasedness) deserve different levels of attention depending on each particular feature and each data set in public data sets. For some data sets or features, the framework focuses on a certain quality dimension more than another. In the process, trade-offs will likely appear, and the quality values of the feature will be attained.

2.2.1 Completeness

Completeness refers to the extent to which a dataset contains all necessary information. Incomplete or partially missing elements in datasets may severely damage the validity of analyses and decision-making processes. Achieving completeness with open-source datasets is difficult because of several inefficiencies in data collection practices or inconsistency in methodologies adopted for data aggregation (Batini & Scannapieco, 2016). For example, such datasets may contain missing required attributes, lack specific values, or even omit entire records, so they would not be sufficient for particular research purposes, unless they undergo preprocessing (Pipino, Lee & Wang, 2002).

Completeness in a holistic manner also involves identifying missing values, calculating the ratio of available data to required data and scoring how these gaps have an impact on the usability of the dataset. Techniques to treat this dimension are imputation of missing values, reviewing null values, and completing completeness scoring techniques. Completion is an important aspect that ensures datasets are representative and reliable for their intended use (ISO/IEC 25024:2015).

2.2.2 Accuracy

Accuracy is defined as the closeness of data with the realities of occurrences or the events the data seeks to reflect. Inaccurate data would lead to wrong analysis, biased models, and questionable conclusions. Common causes of inaccuracy in open-source datasets include error in entry of data, outdated information, and incongruity between conflicting sources of data (Batini & Scannapieco, 2016).

At some time, getting the measure of accuracy may require laying the dataset against established benchmarks or reference standards, if available. In the absence of benchmarking, indirect approaches like anomaly detection, error distribution analysis, and expert domain knowledge can be quite handy to establish the accuracy of the dataset (Pipino, Lee, & Wang, 2002). High accuracy, therefore, is critical to avoid having research and the decision-making framework to become invalid by allowing for an increase in trust towards the conclusion derived from the data (ISO/IEC 25024:2015).

2.2.3 Consistency

Consistency implies that data should remain uniform within a single dataset and across other datasets over time. Inconsistencies often arise due to variations in data frequencies, aggregations, or sources. For example, differences in variable naming conventions, units of measurement, or timestamp formats can hinder data analysis and reduce its utility (Batini & Scannapieco, 2016).

Ensuring consistency involves identifying and resolving issues such as duplicate entries, mismatched formats, or conflicting values. This process requires schema validation, integrity checks, and standardisation procedures to align data formats and structures (ISO/IEC 25024:2015). Consistent data are better suited for integration, analysis, and interpretation, enabling their effective use across a wide range of applications (Pipino, Lee, & Wang, 2002).

2.2.4 Timeliness

Timeliness is the extent to which the data are up-to-date and relevant at the time of access or use. Data that are outdated may lead to unimportant or misleading analyses. For instance, in rapidly

changing fields such as health care, economic series, and the social sciences, data lag may impair policy decisions seriously (Batini & Scannapieco, 2016).

Timeliness is rated by the comparison of the most recent update timestamp of the dataset with the date or time-related requirements of the study. Datasets with higher frequencies of updates, including real-time updates, score higher in this dimension. Methods that enhance timeliness include having standards on the update frequency and having time-related metadata added, which are both requirements to maintain data validity and relevance over time (ISO/IEC 25024:2015; Pipino, Lee, & Wang, 2002).

2.3 Open-Source Data Sets

Most communities in the context of data provide a package or a software library, and for good reason. If researchers use a community-desired tool, it is because it is generally fast, consistent, and easy to visualise and run with simplicity. Most of the time, users do not think too much about what is under the use of such a package or what could be something that can be avoided (Batini & Scannapieco, 2016). This R framework package design should be implemented with a model in mind. Nonetheless, one should approach the source of the data with good practices and input testing that go beyond curated labels. There should be consistency in data projection, better handling of fixed-level errors in certain tools, promotion of efficient coding practices, and even better formatting and layout preservation. Surprisingly, given the large emphasis in the data community on proper curation of data at the source level, there needs to be a publication package advice on practical, easy-to-use tests. This includes ensuring consistency in data projection, effective handling of fixed-level errors in certain tools, promoting efficient coding practices, and preserving formatting and layout (ISO/IEC 25024:2015).

The quality of collected data is an important topic that has been raised in several domains, such as medicine, economics, civil engineering, and statistics. Collecting data is a challenging task since it is time-consuming, costly, and usually needs the assistance of a domain expert. The emergence of open-source projects helped to tackle such discussed issues. These open-source projects academically aim to provide high-quality data sets that are maintained and updated as needed and supported by the respective communities. Open-source data sets have to be collaboratively designed, curated, and maintained by the community. In other words, for a data set to be successful, individuals need to contribute continuously and responsibly in order to share best practices which ensure data quality. Such contribution to the community is usually in the form of metadata. These data could also contain links or point towards other data that is relevant or dependent on the open-source data set available (Wickham, 2019).

2.3.1 Definition and Characteristics

A review of recent literature and conference proceedings highlights the importance of open data while also exposing the risks of polluted, incorrect, and misinterpreted datasets, particularly in high-stakes settings. High-quality, open-source datasets are critical for informed decision-making across various sectors, including public health, security, environmental management, policymaking, and business processes (Batini & Scannapieco, 2016). This paper introduces a data quality framework, implemented as an R package, which enables a holistic assessment of open-source datasets. Additionally, the framework incorporates two novel quality dimensions: transparency and complexity.

Transparency pertains to how easily users can trace and understand the underlying data. Transparency is particularly crucial when datasets are used to inform persuasive arguments, as reliance on deficient, contaminated, or distorted information can lead to incorrect or misleading inferences. Without transparency, users may make unreliable decisions based on such data, undermining confidence in their conclusions (ISO/IEC 25024:2015).

Complexity refers to the intricate nature of certain datasets, which may involve advanced analytics, multiple levels of aggregation, or frequent updates. While complex datasets often lie at the forefront of analytical advancements, they also present unique challenges for quality assessment. Traditional data quality research has explored dimensions such as completeness, accuracy, validity, consistency, timeliness, reliability, relevance, precision, and accessibility (Pipino, Lee, & Wang, 2002). However, open-source datasets introduce additional complexities due to their varied structures and applications.

Open-source datasets empower stakeholders by providing access to vast volumes of data, yet these datasets are challenging and costly to compile. Datasets can range from simple ones with few rows and columns to complex datasets requiring sophisticated manipulations, regular updates, and enhanced visualisations. Complex datasets, in particular, often demand advanced algorithms, frequent updates, and high levels of aggregation. However, these processes can inadvertently mask important information, introduce inaccuracies, or omit critical data quality considerations. For example, routine updates may suffer from low reliability due to insufficient key-checking mechanisms or a lack of subject-area expertise.

The absence of a comprehensive framework for characterising and measuring these aspects of dataset quality remains a significant gap. This gap underscores the need for a unified framework that addresses both traditional quality metrics and the specific challenges posed by open-source data. Poor-quality data have frequently impaired critical decision-making in data science research and applications, spurring efforts to analyse open data more effectively. This study proposes an all-encompassing framework that integrates traditional data quality dimensions with new considerations for transparency and complexity in modern open-source datasets.

2.3.2 Advantages and Challenges

The availability of open datasets enables significant opportunities for analysing complex systems, yet challenges such as inconsistent sampling, overlapping variables, and biases due to diverse sources remain prevalent. These issues often make manual maintenance labour-intensive and limit the scalability of analyses (Pure Ulster, n.d.).

To ensure sustainability, open-source data must meet quality standards and maintain production continuity, as gaps or partial losses can hinder usability. Strong mechanisms for standardisation and timely updates are critical (COST, n.d.). Despite the challenges, open-source datasets—when transformed into labelled, annotated forms—offer invaluable resources, especially in fields like climate studies, where longitudinal data is essential (SciFormat, n.d.).

Datasets curated by domain experts adhere to standard formats, allowing researchers to focus on deriving insights rather than preprocessing. These datasets significantly contribute to fields such as healthcare and social policy (GeeksForGeeks, n.d.). However, addressing quality issues, reducing biases, and maintaining temporal consistency are key to enhancing reliability and usability (BigDataFramework, n.d.).

2.4 Current Approaches to Assessing Data Quality

Data quality assessment requires both providers and consumers to collaborate to ensure datasets are functional and reliable. While tools exist to support these efforts, many fall short of meeting the needs of data consumers—such as researchers and policymakers—who require accuracy, completeness, and reliability. Most frameworks are provider-focused, creating significant gaps in addressing consumer-specific concerns (Beyond Accuracy, MIT, n.d.). Additionally, current methods often rely on fragmented or manual tools that are insufficient for managing the complexity and scale of open-source data environments (Challenges of Data Quality, Data Science Journal, 2015). Concerns like noise, outliers, and representation faithfulness need to be addressed to bridge the gap between providers and consumers, fostering trust and improving usability across applications (Beyond Accuracy, JSTOR, 1996).

2.4.1 Manual Investigation and Testing

Manual inspection has traditionally been the initial step in data quality assessment, enabling researchers to identify pitfalls, outliers, and anomalies that could compromise data analysis validity. However, this approach is time-consuming and labour-intensive, making it impractical for large or complex datasets. Human observation is also limited by cognitive constraints, hindering comprehensive comparisons and statistical extractions. Despite these limitations, manual examination remains valuable for detecting obvious inconsistencies, such as missing values or formatting errors, before conducting advanced analyses.

In the context of big data, manual data quality assessment faces significant challenges due to the volume, velocity, and variety of data. Cai and Zhu (2015) discuss these challenges and propose a hierarchical data quality framework to address them. Additionally, the UK Government's Data Quality Framework provides guidance on assessing and improving data quality, emphasising the importance of understanding data strengths and limitations to determine fitness for purpose (UK Government, 2020). Furthermore, the World Health Organization's Data Quality Review framework highlights the necessity of comprehensive and holistic reviews of data quality, incorporating routine and regular checks to ensure data reliability (WHO, 2017).

2.4.2 Instruments and Methodologies with Automation

The shift towards automated methods for data quality assessment has introduced tools that efficiently process large datasets, identifying trends, outliers, and inconsistencies beyond manual capabilities. Web-based data acquisition often employs scripts in languages like Ruby or Python to automate data scraping and processing, facilitating the aggregation of data from diverse sources into consistent formats. This standardisation enhances data homogeneity and interoperability, which are crucial for effective analysis (Cai & Zhu, 2015).

However, automated methods present their own challenges. Despite the availability of advanced tools and open API services, the representation and storage of knowledge in online datasets can be unreliable, leading to potential inaccuracies. Automated data quality assessments typically focus on representational accuracy, evaluating data's fitness for use. Achieving this accuracy is complex, and challenges arise that necessitate further research to develop reliable automated assessment techniques (Cai & Zhu, 2015).

2.5 Proposed Framework Package

2.5.1 Design Principles

The drafting of the proposed bundle is to proceed with the following principles governing:

Alignment to Recognized Data Quality Dimensions: The proposed framework embeds commonly acknowledged standards of accuracy, completeness, and consistency, further refining these dimensions with metadata relevant to specific applications.

Flexibility and Scalability: The package is designed to process any size and complexity of the dataset for wide applicability in different domains.

Transparency of Uncertainty Management: The framework considers missing data and uncertainties regarding the typical characteristics of openly available datasets.
It has an open-source codebase, which means users can make modifications and contribute toward its development.

These principles will ensure that the package will provide a user-friendly but sound solution to data quality assessment.

2.5.2 Key Ingredients

Property Test (PT): This module validates the input data by checking for missing values, format mismatches, and compliance with specified guidelines. PT forms the foundation of the framework, ensuring the dataset meets basic quality standards.
Source Reliability Test: This module evaluates the reliability of the dataset's origin, verifying the credibility of the data provider and ensuring the dataset undergoes a reliable processing pipeline.
Consistency Check: This module compares datasets within the same category, identifying discrepancies and ensuring uniformity across observations.
Data Type Validation: These tests verify whether the dataset maintains consistent data types, enabling seamless integration into analytical models.

2.6 Evaluation of Data Quality

Ensuring data quality is vital in data-driven processes, as poor quality leads to false conclusions and resource misallocation. Tools like Deequ, an open-source library, assess data quality by measuring metrics, verifying constraints, and monitoring data distribution (Deequ, n.d.). Similarly, the IMF's Data Quality Assessment Framework (DQAF) provides a structured approach to evaluate reliability and relevance, enabling informed decision-making (IMF, n.d.).

2.6.1 Indicators and Measures

The framework employs metrics such as vector and raster error rates to assess data accuracy, alongside indicators that define relevance and reliability. These metrics provide a structured approach for evaluating the suitability of datasets for specific applications, ensuring they meet user needs and organisational criteria (Liao & Bai, 2009; Rass, 2019). Notably, the framework's portability and accessibility represent significant advancements in the field of data quality assessment (IMF, 2003).

2.7 Gaps in Current Literature

While there has been significant development in the arena of data quality management, the existing literature mentions a few major drawbacks, especially regarding open-source datasets. These deficiencies, on the other hand, identify requirements for broader, more standardized frameworks that could effectively tackle the special challenges posed by open-source data. The following key issues explain the shortcomings of the existing research:

Lack of Standardisation: Current frameworks of data quality are all fitted for an internal organizational dataset, often not standardized in open-source data. Because of this, different methodologies and disjointed approaches result, which, consequently, do not consider the variability and complexity of open-source data (Data Ladder, n.d.).

Narrow Focus on Data Recipients: Current research primarily captures the perspective of data providers alone and misses unique needs and concerns of data consumers. Data consumers-for instance, researchers, policy makers, and practitioners-often require tools to deal with specific issues such as noise levels, outliers, and the precision of representation, which are not captured by existing frameworks (Rass, 2019).

Shortcomings in Automation and Scalability: Manual inspection and traditional methods for data quality assessment are very labour-intensive and cannot scale to big datasets. There are some partial automation tools, but the applicability is limited by incomplete methodologies with no integrated solutions (Cai & Zhu, 2015).

Clarity and Understandability: In the literature, there has been very little discussion on disclosure requirements concerning data quality, particularly when these are heavily processed or summarized. By implication, limited knowledge about the underlying data, and its reliability, may then imply a higher risk of misinterpretation and faulty decision-making (Open Knowledge Foundation, 2017).

Lack of standardised structures: While different studies have advocated for individual measures or approaches, there remains an evident gap regarding integrated models that consolidate various dimensions of data quality—such as completeness, accuracy, consistency, and timeliness—into a single, scalable instrument. These deficiencies highlight the urgent need for a comprehensive framework that standardises data quality assessment practices, incorporates ethical and consumer-focused aspects, and provides scalable, automated solutions. The aim of this research is to address these challenges by developing an open-source R package designed to evaluate critical dimensions of data quality, ensuring transparency, usability, and adaptability across diverse datasets and fields.

2.8 Summary of Findings

This study developed and validated a comprehensive framework for assessing the quality of open-source datasets, implemented as an R package. The framework systematically evaluates key data quality dimensions—completeness, accuracy, consistency, and timeliness—addressing critical gaps in existing data quality assessment methodologies. Through its application to real-world datasets, the study yielded the following key findings:

1. **Effectiveness of the Framework:** The proposed framework successfully identified data quality issues, including missing values, inconsistencies, and outliers, across various datasets. It demonstrated robustness in handling datasets of varying sizes and complexities, validating its scalability and adaptability.

2. **Insights into Data Quality Dimensions:**
    - **Completeness:** The framework highlighted significant gaps in data coverage, such as missing values and incomplete records, particularly in publicly available datasets.
    - **Accuracy:** It identified inaccuracies through automated anomaly detection and manual validation techniques, confirming the need for benchmarking data against reliable sources.
    - **Consistency:** The framework effectively resolved inconsistencies in data formatting, naming conventions, and integration across multiple sources.
    - **Timeliness:** The framework revealed challenges in datasets with infrequent updates or delayed availability, underscoring the importance of real-time data management.
3. **Utility of Automated Tools:**
   The integration of automated tools within the R package significantly reduced the time and effort required for data quality assessment compared to manual methods. The package's ability to process large-scale datasets and generate detailed insights further highlights its practical utility.

In conclusion, this study demonstrates that the proposed framework not only addresses key challenges in open-source data quality assessment but also provides a scalable, ethical, and user-friendly tool for researchers and practitioners. These findings underscore the importance of systematic data quality assessment and highlight the potential for further development and application of the framework.

3. Methodology

This chapter describes the research design, the development process of the framework, the main features of the developed R package, and the datasets used for the validation. This methodology follows a structured approach to ensure that the proposed framework addresses the peculiar challenges related to the assessment of open-source datasets in a manner.

3.1 Research Design and Approach

3.1.1 Design Science Methodology

This research employs a **design science methodology** to develop and validate the proposed data quality framework. Design science focuses on creating and evaluating artefacts that solve identified problems within a domain. This methodology is particularly suitable for developing the R package, as it enables iterative design, implementation, and validation to address the gaps in existing data quality assessment frameworks.

The methodology is structured into the following phases:

1. **Problem Identification and Motivation:**
   Understanding the challenges of assessing open-source datasets and identifying the need for a comprehensive framework.
2. **Design and Development:**
   Creating the R package to integrate key data quality dimensions, automation, and user-friendly features.
3. **Demonstration and Evaluation:**
   Applying the framework to real-world, open-sourced datasets to demonstrate its effectiveness in identifying and resolving data quality issues.

This iterative approach ensures the framework is both practically relevant and theoretically grounded.

3.2 Framework Development Process

The development of the framework centred on combining traditional data quality dimensions with innovative tools for automation and transparency. This process began with an in-depth review of existing literature and frameworks to identify the most critical dimensions of data quality, such as completeness, accuracy, consistency, and timeliness. The architecture of the R package was then designed to enable modular and scalable analysis, ensuring flexibility and ease of use. Core functionalities, including data import, type detection, missing data analysis, and outlier detection, were implemented to address common data quality issues effectively. The framework underwent iterative testing and refinement using real-world datasets to ensure both usability and accuracy in diverse scenarios. Its modular design not only supports current features but also allows for the integration of additional functionalities, making it adaptable to various domains and data structures as new needs arise. This approach ensures the framework remains a versatile and robust tool for data quality assessment.

3.3 Features of the R Package

1. Data Import: Includes functionality to input datasets directly within R for seamless integration with RStudio and R.

---

2. Data Type Detection: Automatically detects column types (e.g., numeric, text, date, ID, categorical).

---

3. Text and Date Validation: Text Cleaning (Removes special characters, trims white spaces, and converts text to lowercase.), and Date Validation (Checks for invalid dates, including dates in the future.)

---

4. Missing Data Analysis: Identifies patterns of missing data, summarising the extent of missing values in rows and columns.

---

5. Imputation for Missing Data: Provides two methods for imputation; **Mean** and **Median.**, Allows group-wise imputation for specific categories within the data.

---

6. Duplicate Detection: Identifies duplicate rows within the dataset and provides a list for review. Ensures data records are unique, improving data reliability.

---

7. Outlier Detection: Detects outliers in numeric columns using **Z-score** and **IQR.** Flags values with a Z-score above a configurable threshold (default: >3). Also Visualises outliers with boxplots for better interpretability.

---

8. Redundancy Detection: Identifies highly correlated columns based on a configurable correlation threshold (default: >0.5). Flags redundant variables, allowing users to eliminate unnecessary data.

---

9. Consistency Checks: Uses Variance Inflation Factor (VIF) to identify multicollinearity among numeric variables. Computes a "Consistency Score" to indicate the dataset's overall consistency level.

---

10. Correlation Analysis: Generates a correlation matrix for numeric columns. Supports subgroup correlation analysis by grouping data based on a specified category column. Visualises correlations with heatmaps for intuitive understanding.

---

11. Distribution Analysis: Evaluates the normality of numeric columns using **Skewness** (Measures asymmetry in the distribution.), **Kurtosis** (Assesses the "tailedness" of the distribution.), and **Shapiro-Wilk Test** (Tests for statistical normality). Provides histogram plots for visual representation.

---

12. Unique Identifier Detection: Identifies columns that may act as unique identifiers for dataset rows. Ensures that each record is distinct, improving data reliability.

---

13. Anomaly Detection: Employs a random forest model to detect anomalies in relationships between variables. Flags significant deviations based on residual errors and provides insights into variable interactions.

---

14. Quality Assessment Reports: Generates a comprehensive data quality report that includes ;
Missing values analysis, Outlier detection results, Redundancy and consistency scores, Correlation heatmaps and statistical summaries, visualisations for enhanced interpretation

3.4 Methodology of the Functions

3.4.1 Completeness Score

The **completeness score** calculates the percentage of non-missing values in the dataset:

1. **Formula**:

$$\text{Completeness Score} = 10 - (\text{Average Missing Rate per Column} \times 10)$$

Where:

$$\text{Average Missing Rate per Column} = \frac{1}{n}\sum_{i=1}^{n}\frac{\text{Total Missing Values in Column}_i}{\text{Total Values in Column}_i}$$

Figure 1 Formula for Calculating Completeness Score

Here, Total Values refers to the total number of values in each column, while Total Missing Values represents the count of missing values within each column. The average missing rate per column is calculated by taking the mean proportion of missing values across all columns in the dataset. This average missing rate is then scaled to a range of 0 to 10 by subtracting the product of the average missing rate and 10 from the maximum score of 10. As a result, if there are no missing values, the Completeness Score will be 10, indicating full completeness. Finally, the Completeness Percentage is obtained by scaling the Completeness Score to a range of 0 to 100%, providing an overall measure of the dataset's completeness.

3.4.2 Outlier Detection

Two methods are available for **outlier detection**:

1. Z-score Method

- Outliers are identified based on their **z-score**.
- **Formula**:

$$z_i = \frac{X_i - \mu}{\sigma}$$

Figure 2 Formula for Calculating Z-Score

Here, the Z-score Method is used to identify outliers by determining how far a data point deviates from the mean of the column in terms of standard deviations. Where Xi is a data point, μ is the mean of the column, and σ is the standard deviation of the column. A data point is flagged as an outlier if its absolute Z-score (|zi|) is greater than 3, which indicates that the point lies more than three standard deviations away from the mean. This approach is particularly effective for identifying outliers in datasets that follow a normal distribution.

2. IQR Method

- The **Interquartile Range (IQR)** method uses quartiles to detect outliers.
- **Formulas**:
  - First Quartile (Q1): Value at the 25th percentile.
  - Third Quartile (Q3): Value at the 75th percentile.
  - Interquartile Range (IQR): IQR=Q3−Q1
- Data points are considered outliers if they fall outside:

$$[Q1 - 1.5 \times \text{IQR}, Q3 + 1.5 \times \text{IQR}]$$

Figure 3 Formula for Identifying Outliers Using the IQR Method

Here, the **Interquartile Range (IQR) Method** is used to detect outliers by identifying values that lie outside a specific range based on the distribution of the data. Any data point is considered an outlier if it falls outside the range defined. This method effectively captures extreme values without assuming a normal distribution, making it particularly robust for skewed datasets or datasets with non-normal distributions.

3.4.3 Correlation Matrix Calculation

The **correlation matrix** measures the linear relationship between pairs of numeric variables.

1. **Formula**:

$$\text{Corr}(X, Y) = \frac{\text{Cov}(X, Y)}{\sigma_X \sigma_Y}$$

Figure 4 Formula for Calculating Pearson Correlation Coefficient

where Cov(X,Y) is the covariance of X and Y, and σX and σY are the standard deviations of X and Y.

2. **Redundant Columns**:

   1. **Correlation Matrix**:
      - A correlation matrix is computed to measure the pairwise linear relationships between numeric columns. Each cell in this matrix represents the correlation coefficient between a pair of columns.

2. **Flagging Redundant Columns**:
    - After calculating the correlation matrix, column pairs with a correlation coefficient that exceeds the specified **correlation threshold** are flagged as redundant.
    - For example, if the correlation threshold is set at 0.5, any column pairs with |Corr(X,Y)|>0.5| are considered redundant.

3. **Outcome**:
    - The function returns a list of columns flagged as redundant. Analysts can then decide to drop one of each pair of redundant columns to reduce redundancy and multicollinearity in the dataset.

3.4.4 Consistency Check: Variance Inflation Factor (VIF)

The **Variance Inflation Factor (VIF)** detects multicollinearity among predictor variables, indicating how much a variable's variance is inflated due to its correlation with other variables.

1. **Formula**:

$$\mathrm{VIF}_j = \frac{1}{1 - R_j^2}$$

Figure 5 Formula for Variance Inflation Factor (VIF)

where Rj2 is the R-squared value obtained by regressing the predictor variable Xj on all other predictor variables in the model. A higher VIF value indicates a stronger correlation and, therefore, a higher level of multicollinearity. Typically, a VIF value greater than 5 or 10 is considered problematic, suggesting that the variable's variance is significantly inflated due to multicollinearity with other variables.

Threshold:

- **Outlier Detection Threshold**:
    1. **Z-score Method**: A z-score threshold of 3 is commonly used. Data points with an absolute z-score greater than 3 are flagged as outliers. |zi|>3
    2. **IQR Method**: For the Interquartile Range (IQR) method, data points that fall outside [Q1−1.5×IQR,Q3+1.5×IQR] are considered outliers.

- **Correlation Threshold**:
    1. In redundancy detection, a **correlation threshold** is applied to the correlation matrix to identify highly correlated columns. Columns with a correlation coefficient above this threshold are flagged as redundant (highly correlated) columns.
    2. **Default Threshold**: This threshold is set by the user (e.g., correlation_threshold = 0.5). A threshold of 0.5, for example, flags any

pair of columns with an absolute correlation coefficient greater than 0.5 as redundant. |Corr(X,Y)|>0.5

- **VIF Threshold (for Multicollinearity)**:
    1. Variance Inflation Factor (VIF) is calculated to detect multicollinearity among numeric predictors. A **VIF threshold** is used to identify variables that are excessively collinear.
    2. **Default Threshold**: In your code, the default VIF threshold is 5. If a variable's VIF is greater than 5, it indicates a strong correlation with other variables, which may lead to instability in regression models. VIFj>5

3.4.5 Distribution Analysis

Skewness and Kurtosis

Skewness and **kurtosis** are used to assess the normality of the distribution.

1. **Skewness**:

Skewness is a statistical measure used to assess the symmetry of a distribution around its mean. It quantifies the extent to which a dataset deviates from a perfectly symmetrical distribution.

$$\text{Skewness} = \frac{\frac{1}{n}\sum_{i=1}^{n}(X_i - \mu)^3}{\sigma^3}$$

Figure 6 Formula for Calculating Skewness

where Xi represents each data point, μ is the mean of the data, σ is the standard deviation, and n is the total number of data points. Indicates symmetry of the distribution around the mean. Positive values indicate right skew, negative values indicate left skew.

2. **Kurtosis**:

$$\text{Kurtosis} = \frac{\frac{1}{n}\sum_{i=1}^{n}(X_i - \mu)^4}{\sigma^4} - 3$$

Figure 7 Formula for Calculating Kurtosis

where Xi represents each data point, μ is the mean of the data, σ is the standard deviation, and n is the total number of data points. The term −3−3 is subtracted to standardize the result so that a normal distribution has a kurtosis of 0. Kurtosis describes the "tailedness" of the distribution. A kurtosis > 0 indicates a heavy-tailed distribution, while a kurtosis < 0 indicates a light-tailed distribution.

Shapiro-Wilk Test

The Shapiro-Wilk Test is a test of normality in statistics. It is truly powerful for sample sizes ranging from a few to medium. This test calculates a certain W statistic that summarizes the correspondence of observed data to the values which should be expected from normality distribution. The smaller the W, the further the data set is from normality. The test gives a p-value that can then be interpreted. If the **p-value** is less than the significance level (usually 0.05), the distribution significantly deviates from normality.

3.5 Summary of Methods

This section outlines the key methods and techniques used for data quality assessment, as implemented in the R package:

- **Completeness Analysis**:
  - Measures the proportion of non-missing values in the dataset.
  - A completeness score is calculated by assessing missing data across columns and scaling the result to a range of 0–10.
- **Outlier Detection**:
  - **Z-score Method**: Identifies outliers by measuring how far a data point deviates from the mean, with a threshold of $|Z|>3$.
  - **IQR Method**: Uses the interquartile range (IQR) to detect outliers that fall outside $[Q1-1.5\times IQR, Q3+1.5\times IQR]$.
- **Redundancy Detection (Correlation)**:
  - Identifies highly correlated variables by calculating a correlation matrix.
  - Variables with correlation coefficients exceeding a user-defined threshold (e.g., 0.5) are flagged as redundant.
- **Consistency Analysis (VIF)**:
  - Assesses multicollinearity among predictor variables using the Variance Inflation Factor (VIF).
  - A VIF value exceeding a predefined threshold (e.g., 5) indicates multicollinearity.
- **Distribution Analysis**:
  - **Skewness**: Measures the asymmetry of the data distribution. Positive values indicate right skew, and negative values indicate left skew.
  - **Kurtosis**: Assesses the "tailedness" of the distribution. Positive kurtosis indicates heavy tails, while negative kurtosis indicates light tails.
  - **Shapiro-Wilk Test**: Evaluates normality of the data distribution. A p-value less than 0.05 suggests significant deviation from normality.
- **Missing Data Imputation**:
  - Missing values are imputed using user-specified methods such as the column **mean** or **median**.
  - Group-wise imputation can also be performed for categorical subsets.
- **Anomaly Detection**:
  - Flags anomalies in relationships between variables using machine learning models such as Random Forest.

- **Visualization Tools**:
    - Includes visual representations such as boxplots for outliers, correlation heatmaps for redundancy, and histograms for distribution analysis.

3.6 Datasets for Validation

The dataset (Mall Customers) used for the validation of the proposed framework has been taken from Kaggle, one of the most acknowledged open sources for different domain-based datasets. Most of these datasets are repeatedly reused in several data science, machine learning, and analytics projects simply because of easy access and diversity. In this paper, the particular dataset was selected with the aim of fulfilling the required analysis and testing the ability of the framework in managing common data quality issues like missing values, inconsistencies, and outliers.

3.7 Ethical Considerations

The dataset is licensed under the Apache License 2.0 (Apache Software Foundation, 2004). This licence allows for the use, modification, and redistribution of the dataset, provided proper attribution is maintained. By ensuring adherence to the licensing terms, the study upholds the principles of responsible and ethical use of open-source data while promoting transparency and reproducibility in research.

4. Implementation

The implementation phase of this research involved developing a comprehensive R package—**DataQualityPackage**—to evaluate and enhance the quality of open-source datasets. This section provides an overview of the development process, core functions, challenges faced, and the integration of the package with existing tools for data analysis.

4.1 Overview of the R Package Development

The **DataQualityPackage** was developed to be able to respond to challenges likely to assess the important problems of data quality in open-sourced dataset's missing values, inconsistencies, redundancy, outliers-fully in an automated, flexible, modular fashion, thereby effectively supporting both the researcher and practitioner in their respective evaluation tasks on datasets.

The primary objectives in the development process were:

- Providing a complete data quality assessment pipeline for open-source datasets.
- Offering customisable parameters for adaptability across different datasets and domains.
- Ensuring user-friendliness, scalability, and performance through structured functions and modular architecture.

4.2 Core Functions and Their Roles

4.2.1 Main Function: data_quality_assessment

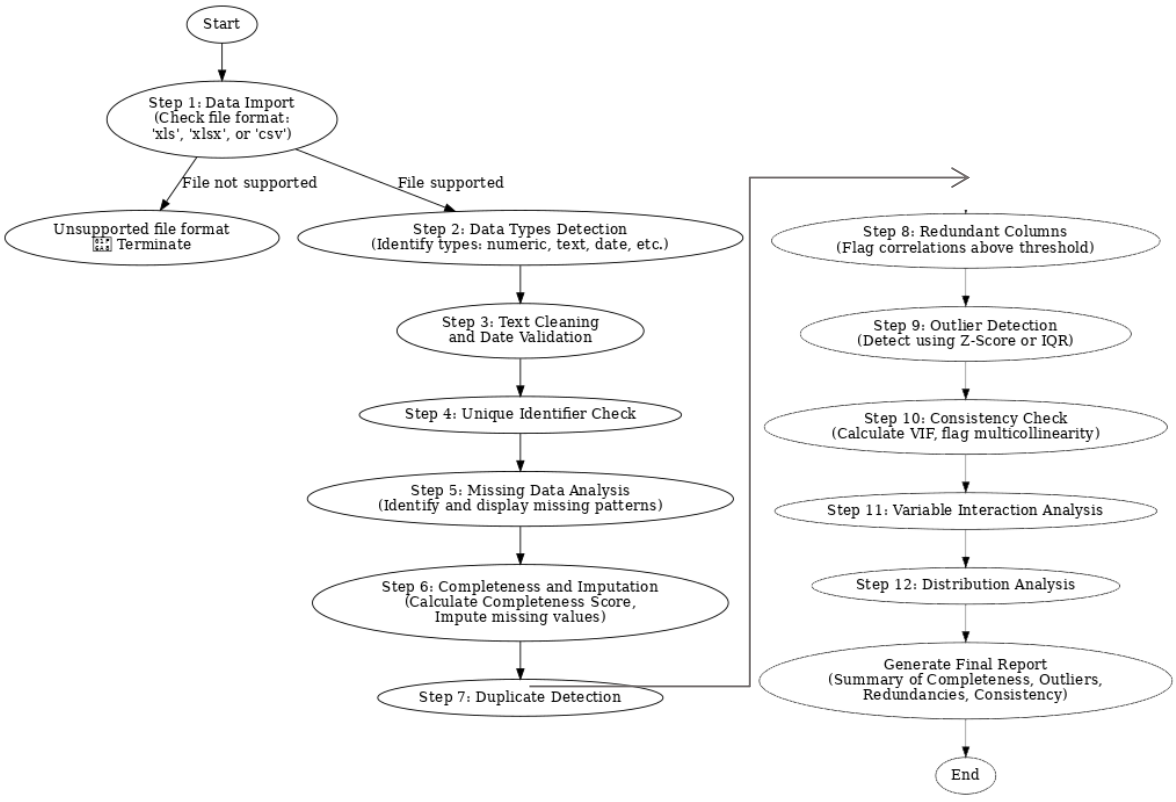


Figure 8 4.2.1 Main Function

The data_quality_assessment function is the core function in the package, conducting a complete data quality assessment on a dataset. Users can control various analysis techniques, test parameters, and thresholds, allowing them to adapt the assessment to specific data and quality requirements.

Features

1. **Data Import**: Supports reading .csv, .xls, and .xlsx files.
2. **Data Type Detection**: Identifies column types, including numeric, text, date, ID, and categorical columns.
3. **Text and Date Cleaning**: Cleans text columns (removing special characters) and checks date columns for invalid or future dates.
4. **Unique Identifier Check**: Detects columns that may act as unique identifiers.
5. **Missing Data Analysis**: Analyses missing data patterns, identifying columns or rows with high levels of missing values.
6. **Imputation**: Provides customizable imputation methods (mean or median) to fill in missing values.
7. **Duplicate Detection**: Identifies duplicate rows in the dataset.
8. **Redundant Column Detection**: Flags columns with high correlation to other columns based on a configurable threshold.
9. **Outlier Detection**: Detects outliers using configurable methods (z_score or iqr).
10. **Consistency Check**: Assesses multicollinearity using VIF and calculates a consistency score.
11. **Correlation Analysis**: Generates correlation heatmaps and allows for category-specific correlation analysis.
12. **Distribution Analysis**: Analyses distribution for normality through skewness, kurtosis, and the Shapiro-Wilk test.
13. **Anomaly Detection**: Uses a machine learning model (random forest) to flag anomalies in variable relationships.

Arguments

- data: Use when providing a dataset directly
- path: File path for the dataset. Supported formats: .xls, .xlsx, .csv.
- outlier_method: Outlier detection method. Options:
    - "z_score": Flags values with a z-score > 3.
    - "iqr": Flags values outside the interquartile range (IQR) ±1.5 IQR.
- imputation_method: Method for imputing missing values. Options:
    - "mean": Imputes with the column mean.
    - "median": Imputes with the column median.
- correlation_threshold: Threshold for flagging redundant columns (default = 0.5). Columns with correlation above this value are considered redundant.
- vif_threshold: Variance Inflation Factor (VIF) threshold for multicollinearity detection (default = 5). Columns with VIF above this value indicate potential multicollinearity.
- visualize_outliers: Logical. If TRUE, visualizes outliers using boxplots.
- category_column: Optional column name to categorize data. This enables grouped analyses for outliers and correlations, enhancing insights within specific groups.

Returns

Key quantitative measures include the **Completeness Score**, which evaluates non-missing values, the **Outlier Count**, identifying anomalies in numeric data, and the **Consistency Score**, determined through VIF analysis to detect multicollinearity. Additionally, it summarises missing data patterns, flags duplicate rows and redundant columns, and evaluates data distribution using skewness, kurtosis, and normality tests.

Visual outputs such as **boxplots** for outliers, **heatmaps** for correlation analysis, and **histograms** for distribution assessment provide intuitive diagnostics, while an anomaly report highlights variable relationship irregularities through machine learning-based methods. Designed for both novice and expert users, the function's outputs deliver actionable insights that facilitate rigorous data evaluation and enhance the quality of open-source datasets.

Console Outputs

Upon execution, the package produces a detailed report summarising the key findings from each quality check. For instance, using the **Iris dataset** as a demonstration, the package outputs the following:

- **Completeness Score**: 100%
- **Total Outliers Detected**: 1
- **Consistency Score**: 2.5%
- **Duplicate Rows**: 1
- **Redundant Columns**: "Petal.Length," "Petal.Width," "Sepal.Length"

Additionally, the output includes:

- **Distribution Analysis**: Statistical measures for key columns (e.g., skewness, kurtosis, and Shapiro-Wilk p-values).
- **Outlier Listings**: A structured count and details of outlier values identified in the dataset.

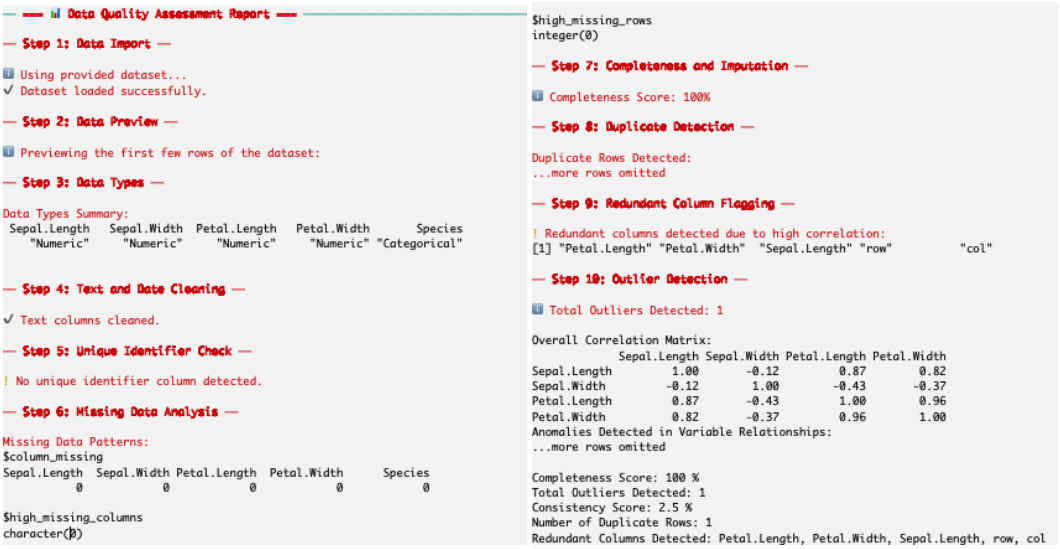


Figure 9 Console Outputs

Further outputs include:

- **Missing Data Patterns**: Summarises the occurrence and distribution of missing values, flagging columns or rows with excessive null values.
- **Duplicate Detection**: Identifies redundant rows within the dataset, offering clarity on dataset integrity.
- **Redundant Columns**: Flags highly correlated features based on user-defined correlation thresholds, thereby reducing dimensional redundancy.
- **Distribution Analysis**: Examines data normality through statistical measures such as **skewness**, **kurtosis**, and the **Shapiro-Wilk test**. Histograms are generated to facilitate visual assessment.
- **Anomaly Detection**: Machine learning-based methods (e.g., **Random Forest residuals**) identify irregularities in variable relationships, supporting advanced diagnostics.

Distribution analysis values

```
── [1m [1mStep 13: Distribution Analysis [1m [22m ──
Analysis for Sepal.Length:
- Skewness: 0.31
- Kurtosis: 2.43
- Shapiro-Wilk p-value: 0.0102 (Significant deviation from normality)
Analysis for Sepal.Width:
- Skewness: 0.32
- Kurtosis: 3.18
- Shapiro-Wilk p-value: 0.1012 (Normal distribution)
Analysis for Petal.Length:
- Skewness: -0.27
- Kurtosis: 1.6
- Shapiro-Wilk p-value: 0 (Significant deviation from normality)
Analysis for Petal.Width:
- Skewness: -0.1
- Kurtosis: 1.66
- Shapiro-Wilk p-value: 0 (Significant deviation from normality)
```

Figure 10 Distribution analysis values

Outliers list

Description: df [1 × 5]

	Sepal.Length <dbl>	Sepal.Width <dbl>	Petal.Length <dbl>	Petal.Width <dbl>	Species <fctr>
143	5.8	2.7	5.1	1.9	virginica

1 row

Figure 11 List of Identified Outliers

Visual Outputs

The package generates an array of intuitive visual outputs to aid in diagnostics and facilitate a comprehensive understanding of data quality:

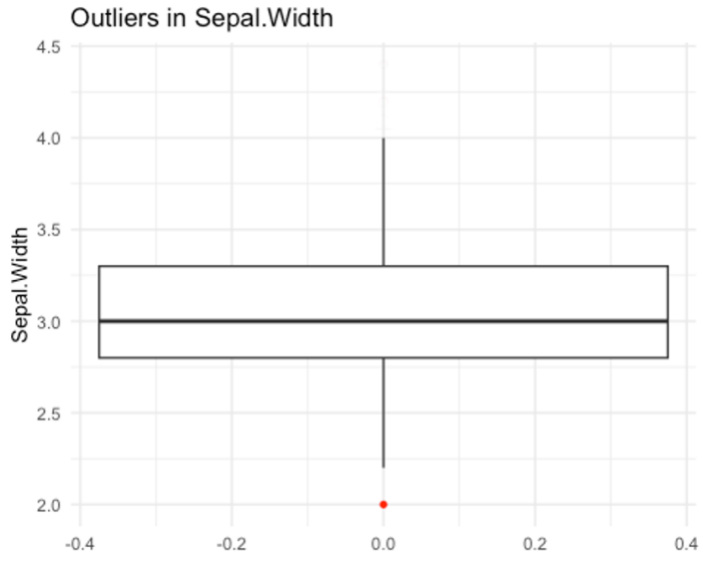
1. **Boxplots**: Visualise outliers in numerical variables, providing clear insights into data anomalies.
2. 

Figure 12 Visual Output of Outliers

3. **Correlation Heatmaps**: Present relationships between numeric variables, highlighting redundant features and dependencies.

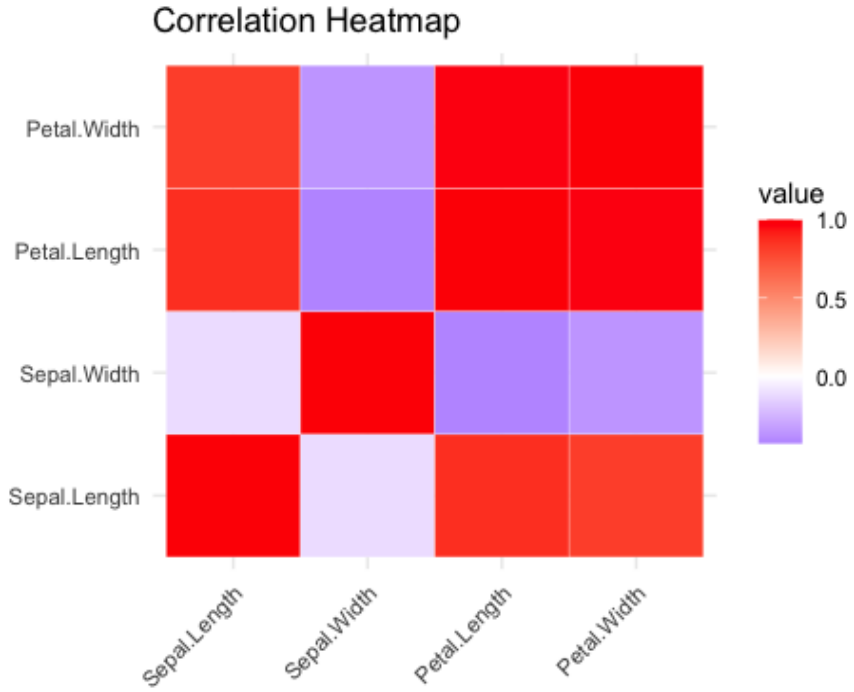


Figure 13 Visual Output of Correlation Heatmap

4. **Histograms**: Assess data distribution for each column, enabling users to identify skewness or deviations from normality.

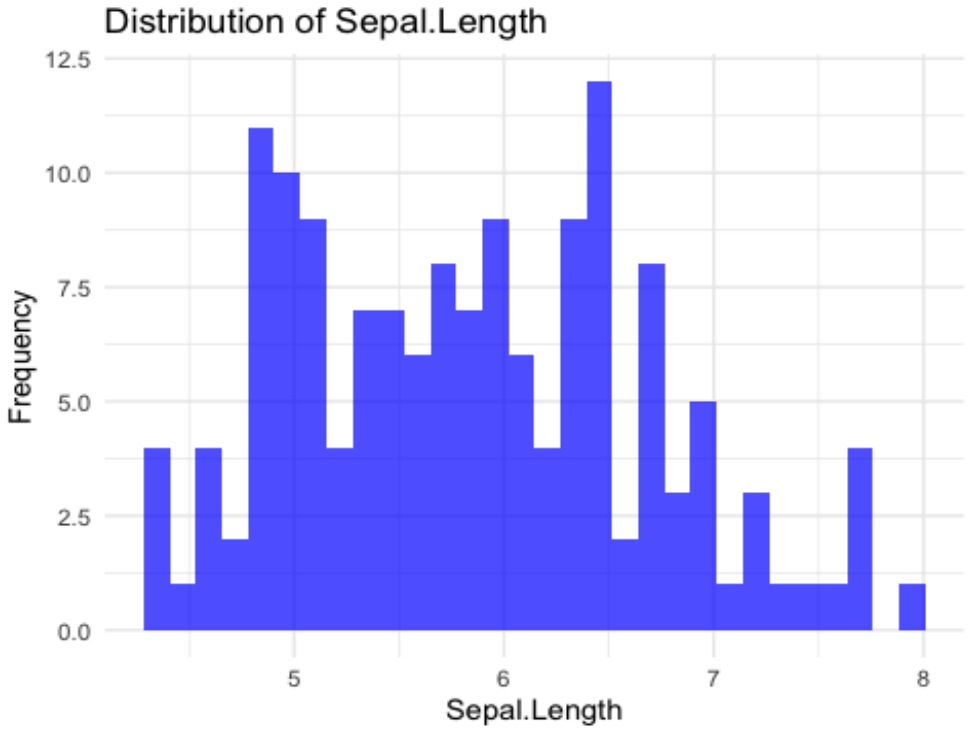


Figure 14 Visual Output for Distribution

Example Usages: *Basic Usage for General Analysis*

data_quality_assessment(path = "data.csv") or data_quality_assessment(data_name)

path: Specifies the location of a CSV file named data.csv to be imported and analysed.

Default parameters include:

- Outlier detection using the "z_score" method.
- Missing data imputed using the column mean.
- Redundant columns flagged at a correlation threshold of 0.5.
- Multicollinearity issues detected at a VIF threshold of 5.
- No visualisation of outliers, as visualise_outliers is FALSE by default.

Advanced Analysis with Visualisation

```
data_quality_assessment(
  data = my_data_frame,
```

- **data**: Uses a preloaded data frame my_data_frame instead of reading from a file.

```
  outlier_method = "iqr",
```

- **outlier_method**: Sets outlier detection to use the Interquartile Range (IQR) method, flagging data points outside 1.5 times the IQR.

```
  imputation_method = "median",
```

- **imputation_method**: Fills missing values with the column median.

```
  correlation_threshold = 0.7,
```
- **correlation_threshold**: Flags redundant columns with a correlation coefficient above 0.7 (less sensitive than the default).

```
  vif_threshold = 3,
```
- **vif_threshold**: Lowers the threshold for Variance Inflation Factor (VIF) to 3, tightening the multicollinearity check.

```
  visualise_outliers = TRUE
```
- **visualise_outliers**: Enables visualisation of outliers using boxplots for better interpretation.

```
)
```

Figure 15 Advanced Analysis with Visualisation

Grouped Analysis for Categorical Data

```
data_quality_assessment(
  path = "survey_data.xlsx",
```
- **path**: Imports an Excel file named survey_data.xlsx.

```
  outlier_method = "z_score",
```
- **outlier_method**: Uses z-scores to detect outliers.

```
  imputation_method = "mean",
```
- **imputation_method**: Imputes missing values with the column mean.

```
  category_column = "Region"
```
- **category_column**: Performs outlier and correlation analyses separately for each group within the Region column, allowing insights tailored to different geographical areas.

```
)
```

Figure 16 Grouped Analysis for Categorical Data

4.2.2 Helper Functions

The core function (data_quality_assessment) operates by sequentially invoking 13 helper functions. Additionally, these helper functions can also work independently. The pseudocode for the helper functions can be found in **Appendix A: Helper Functions Pseudocode**. These include:

1. **Data Type Detection** (detect_data_types)

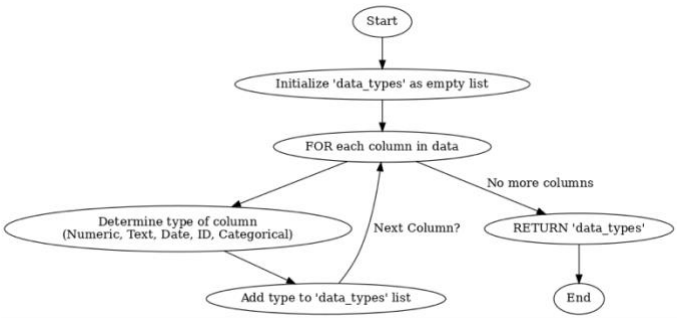


Figure 17 Workflow for Data Type Detection

The process begins by **initialising an empty list**. The system then iterates through each column, identifies its type (e.g., numeric, text, date, ID, or categorical), and adds the determined type to the list.

Once all columns are processed, the final list, summarising the type of each column, is returned. This process ensures a clear understanding of the dataset structure, supporting subsequent analysis or data cleaning tasks.

2. **Text Cleaning** (clean_text_columns)

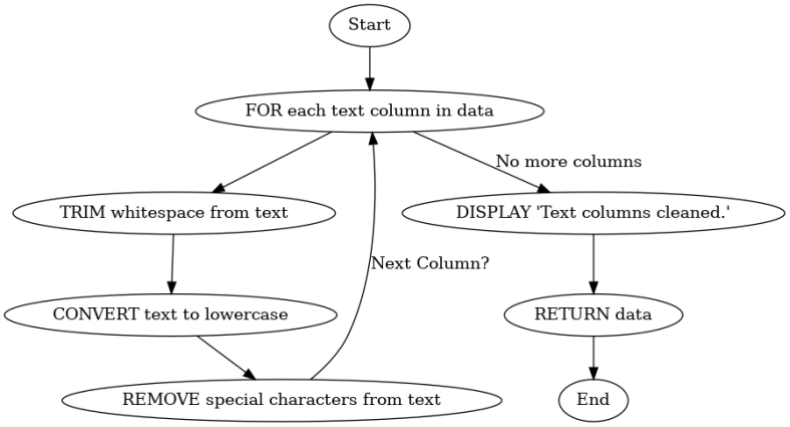


Figure 18 Workflow for Text Cleaning

For each text column, the system performs three operations sequentially: **trimming whitespace**, **converting text to lowercase**, and **removing special characters**. Once all text columns are processed, a confirmation message ("Text columns cleaned") is displayed, and the cleaned dataset is returned.

This ensures the text data is standardised, improving consistency and preparing it for further analysis or processing. The output is a **cleaned dataset** with uniformly formatted text columns.

3. **Date Validation** (validate_dates)

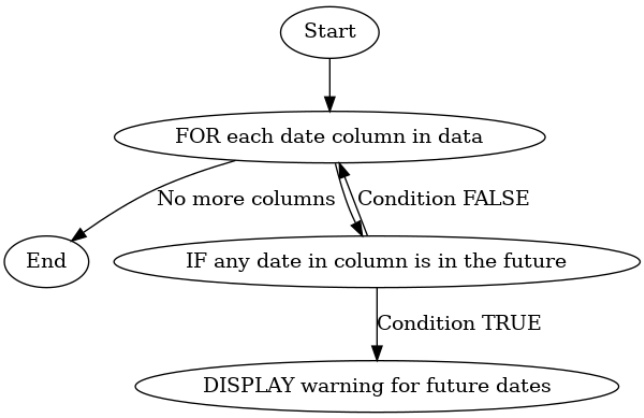


Figure 19 Workflow for Date Validation

The system iterates through each date column in the data. If any dates in a column are found to be in the future, a **warning is displayed** to highlight these anomalies. If no such dates exist, or once all columns are checked, the process concludes.

This method ensures that invalid or unexpected future dates are flagged, helping maintain the integrity and reliability of date-related data within the dataset. The output is a clear warning for any future dates detected.

4. **Unique Identifier Check** (check_unique_identifiers)

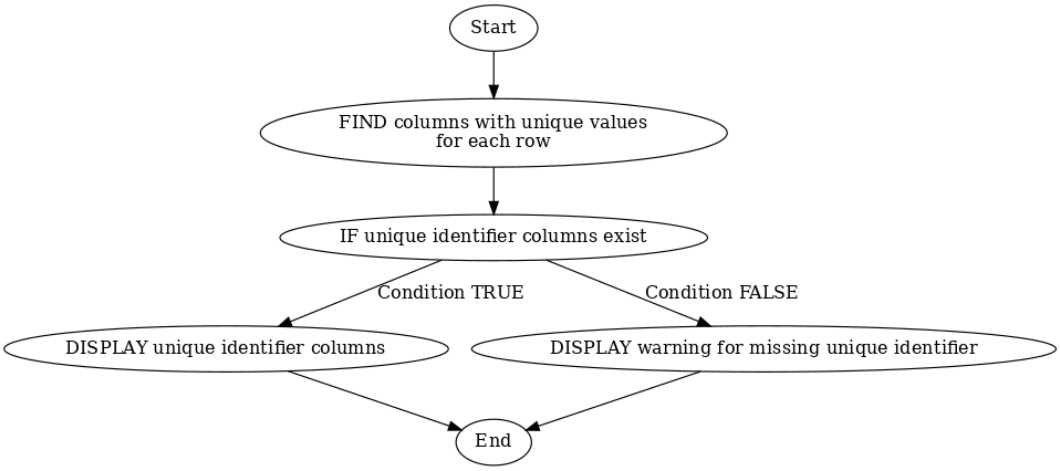


Figure 20 Workflow for Unique Identifier Check

The process begins by analysing the data to **find columns** containing unique values for each row. If such columns exist, they are **displayed** as potential unique identifiers. If no unique identifier columns are found, a **warning** is displayed to alert users about the absence of a proper unique identifier.

The process ensures that datasets have reliable identifiers, which are essential for tasks such as indexing, deduplication, and relational database operations. The output is a **list of potential unique identifier columns** or a notification when none are detected.

5. **Duplicate Detection** (detect_duplicates)

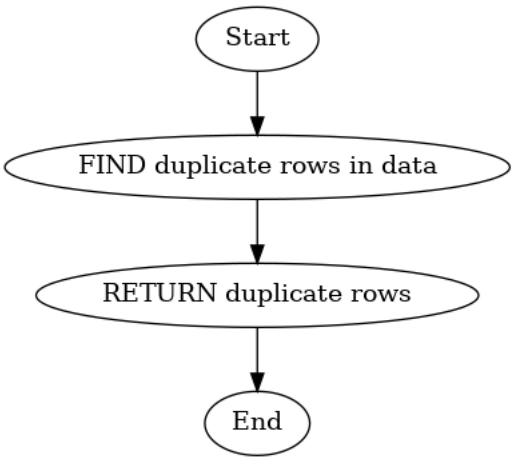


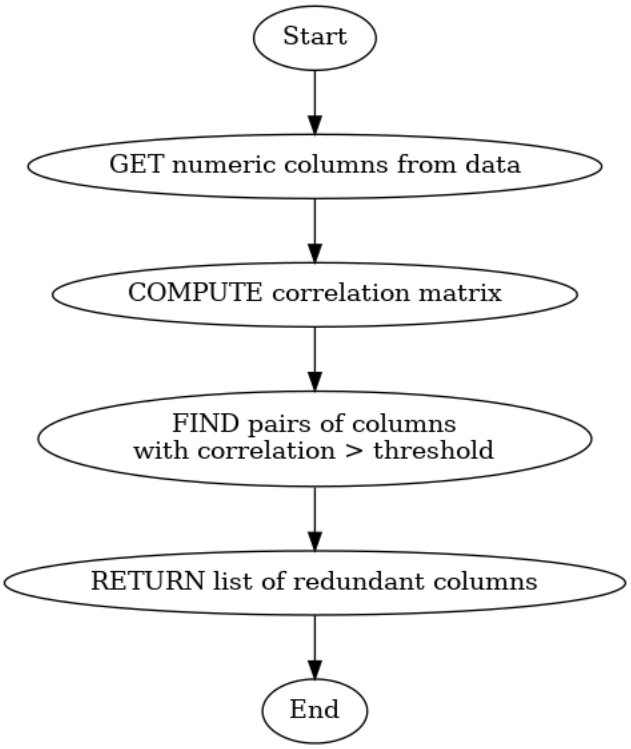
Figure 21 Workflow for Duplicate Detection

The process begins by scanning the data to **find duplicate rows** based on all columns. Once duplicates are detected, the system **returns a data frame** containing the identified duplicate rows.

This process enables efficient detection of redundant records, ensuring data integrity and helping to streamline data cleaning workflows.

6. **Redundant Column Detection** (flag_redundant_columns)

Figure 22 Workflow Redundant Column Detection



The process begins by extracting the **numeric columns** from the data, followed by computing the **correlation matrix** to measure pairwise relationships between these columns.

Next, column pairs with correlation values exceeding the given threshold are flagged as redundant. The process concludes by returning a **list of redundant columns**, enabling users to identify and address multicollinearity issues in the dataset. This approach improves data quality and optimises further analysis by eliminating highly correlated variables.

7. **Missing Data Analysis** (analyse_missing_data_patterns)

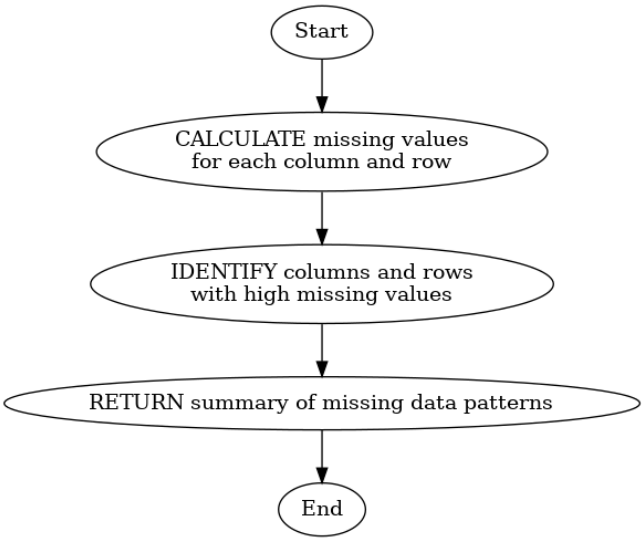


Figure 23 Workflow for Missing Data Analysis

The process begins by **calculating missing values** for each column and row. Next, the system identifies specific **columns and rows** with significant levels of missing data, helping to highlight problematic areas in the dataset.

The final step involves returning a **summary of missing data patterns**, which includes a list of columns and rows with excessive missing values. This process enables a systematic understanding of missing data, supporting informed decisions for imputation, removal, or further analysis.

8. **Imputation** (impute_missing_values)

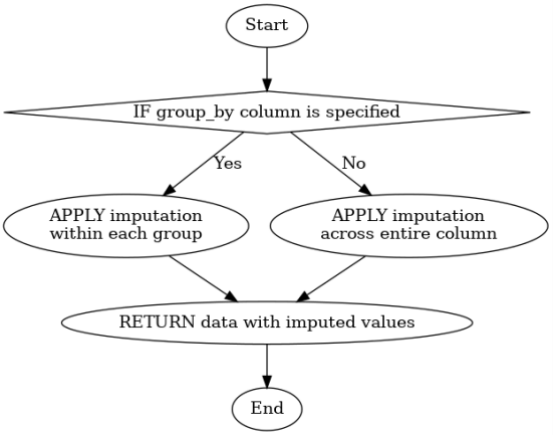


Figure 24 Workflow for Imputation

The process begins by checking if a **group by column** is provided. If specified, the system applies the imputation (mean or median) **within each group** of the column. If no group by column is specified, the imputation is applied **across the entire column**.

Once the missing values are imputed, the updated dataset with filled values is returned. This process ensures flexibility, allowing for both group-level and column-wide imputation, making it suitable for datasets with varying levels of granularity or subgroup-specific patterns.

9. **Outlier Detection** (detect_outliers)

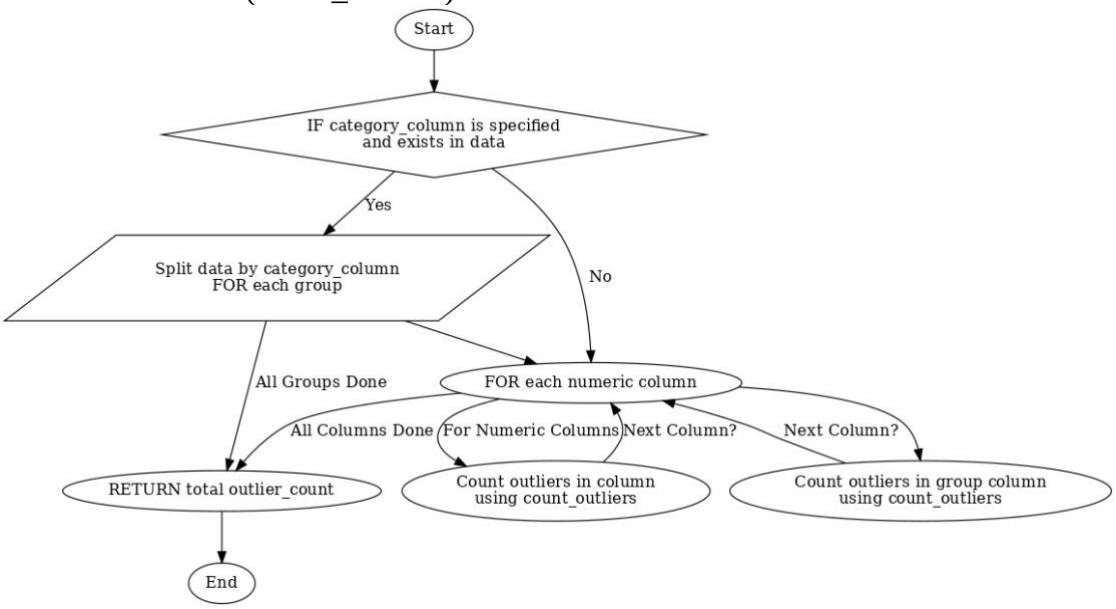


Figure 25 Workflow for Outlier Detection

The process first checks if a **category column** is specified and exists in the data. If it is provided, the data is split by the category column, and outliers are counted for each group. If no category column is specified, the system iterates through each numeric column to count the outliers directly.

Outliers are detected using the chosen method (z-score or IQR), and the counts are accumulated for each column or group. Once all columns or groups have been processed, the system returns the **total outlier count**. This approach ensures flexibility by supporting both overall and subgroup-based outlier detection, providing an effective means of identifying anomalies in numeric data.

10. **Consistency Check** (calculate_consistency_score)

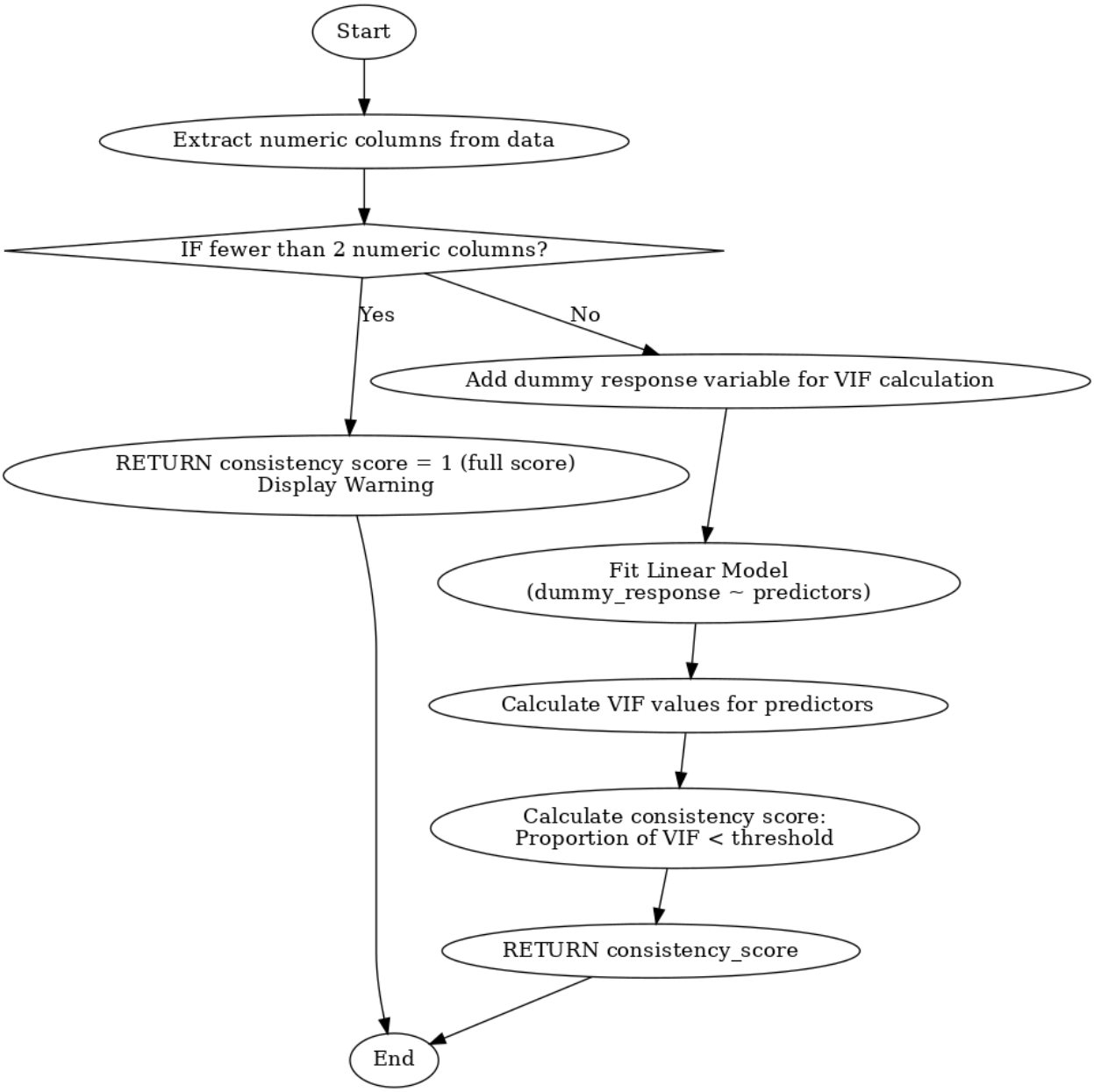


Figure 26 Workflow for Consistency Check

The system begins by extracting numeric columns from the dataset. If fewer than two numeric columns are detected, a warning is displayed, and a full consistency score of 1 is returned. Otherwise, a **dummy response variable** is introduced to fit a linear model with the predictors.

The VIF values for the predictors are calculated, and the consistency score is determined based on the **proportion of VIF values** below the specified threshold (e.g., 5). The process concludes by returning the **consistency score**, which indicates the degree of multicollinearity within the dataset. This approach ensures a quantitative assessment of predictor consistency, helping identify multicollinearity issues that could affect data quality or model reliability.

11. **Correlation Analysis** (analyse_correlation)

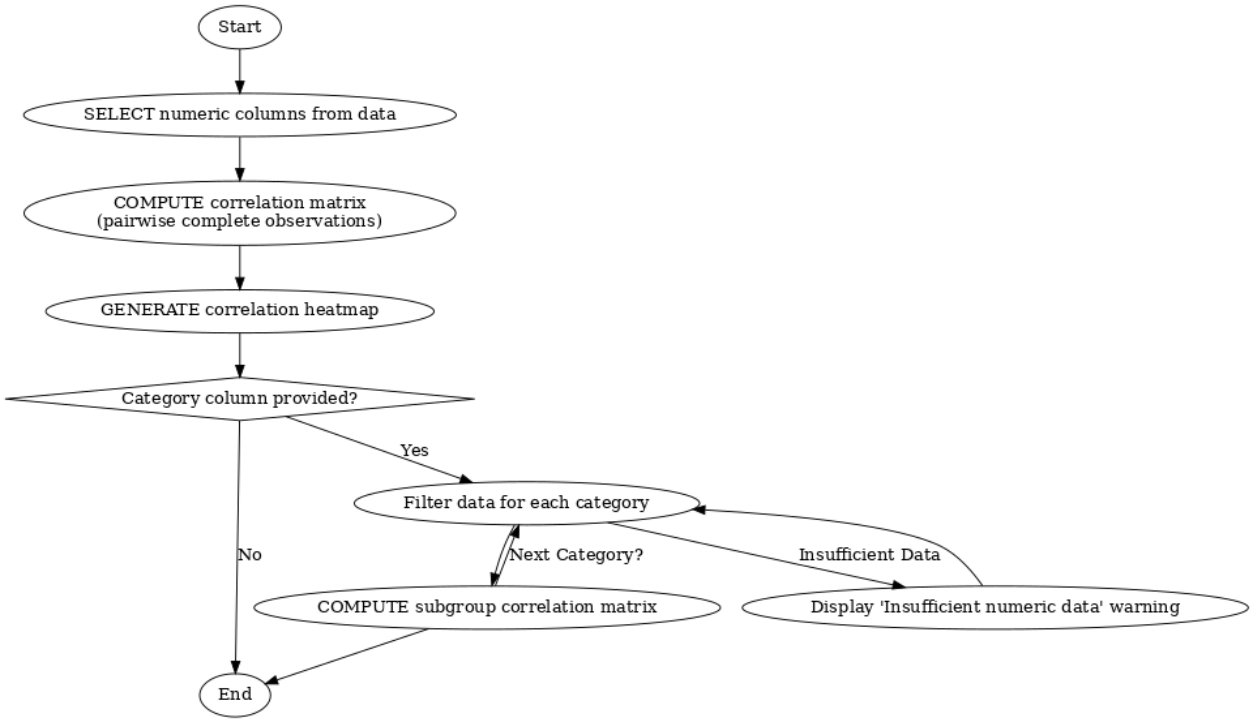


Figure 27 Workflow for Correlation Analysis

The system selects numeric columns and computes the **correlation matrix** using pairwise complete observations. A visual heatmap is generated to represent the relationships between variables. If a **category column** is provided, the data is filtered into subgroups based on each category, and a subgroup-specific correlation matrix is computed for further analysis.

The process ensures flexibility by supporting both overall and subgroup-specific correlation analyses. If insufficient numeric data is available for any subgroup, a warning message is displayed. The output includes a **correlation heatmap** for the entire dataset and subgroup correlations, providing valuable insights into variable relationships.

12. **Distribution Analysis** (analyse_distribution)

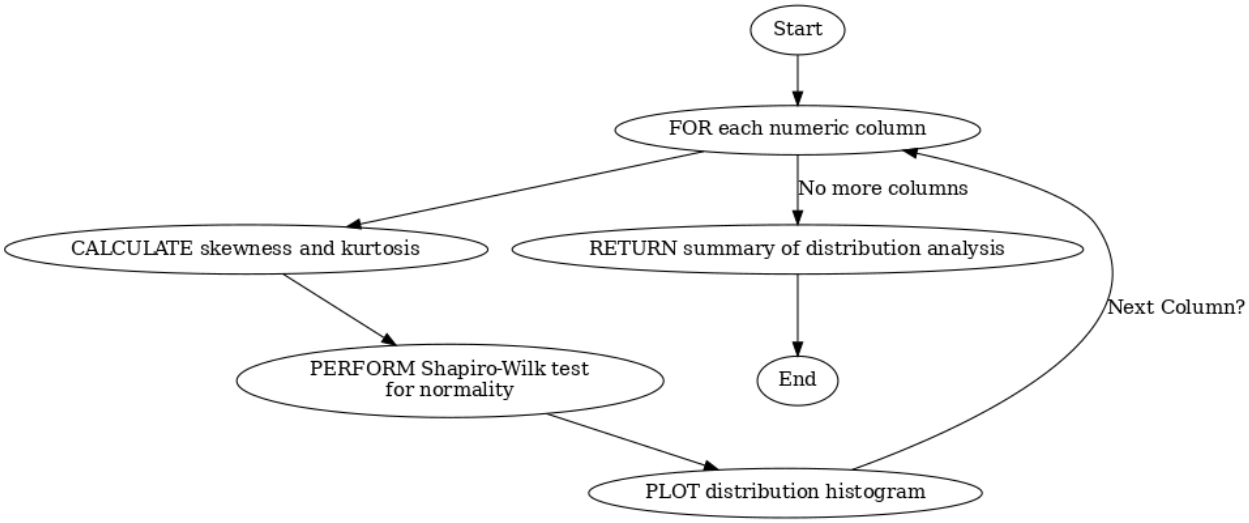


Figure 28 Workflow for Distribution Analysis

For each numeric column, the system calculates **skewness** and **kurtosis** to evaluate the symmetry and shape of the distribution. It then performs the **Shapiro-Wilk test**, a statistical method to determine if the data deviates significantly from a normal distribution. Additionally, **distribution histograms** are plotted to provide a visual representation of the data's distribution.

The process concludes by summarising the results of the analysis, highlighting any columns that deviate from normality. This approach ensures both statistical and visual evaluation, enabling users to identify and address non-normal distributions effectively. The outputs include a detailed summary report and accompanying visualisations for further interpretation.

13. **Anomaly Detection in Relationships** (flag_anomalies_in_relationships)

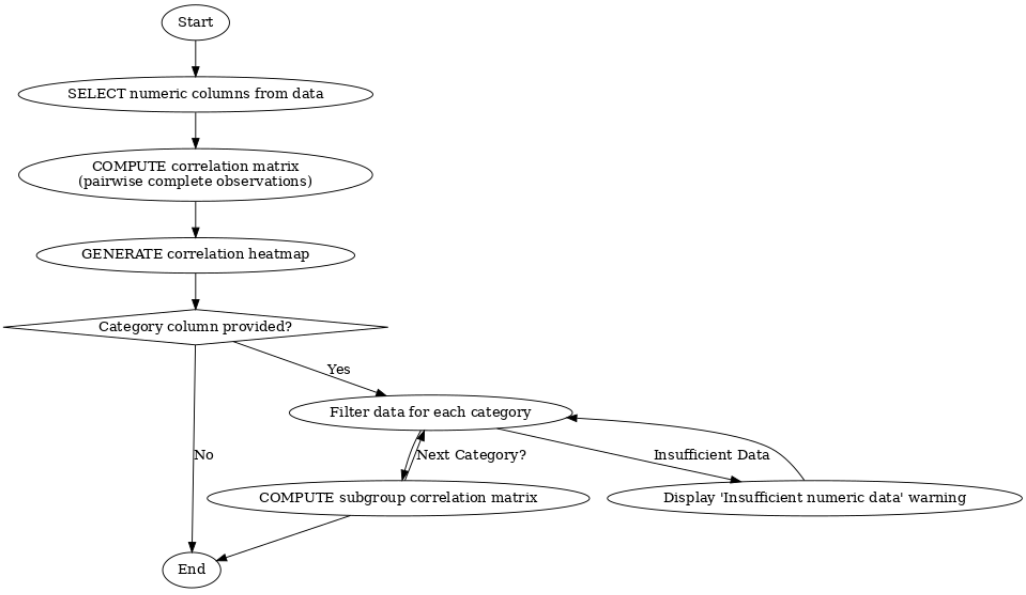


Figure 29 Workflow for Anomaly Detection in Relationships

Process of the function begins by selecting numeric columns and calculating a correlation matrix using pairwise complete observations, followed by generating a heatmap to visualise the relationships between variables. If a categorical column is provided, the data is filtered into subgroups, and a separate correlation matrix is produced for each category. Anomalies in variable relationships are identified using a random forest model, which analyses residual errors and flags significant deviations as unusual patterns.

This process ensures a thorough assessment of data quality by automating correlation analysis, subgroup-specific insights, and anomaly detection. Where there is insufficient numeric data, a warning is displayed. The outputs include correlation heatmaps, reports highlighting anomalies, and visualisations, making the analysis reliable, efficient, and scalable for large or complex datasets.

4.3 Challenges in Implementation

During the development of the R package, several challenges required innovative solutions. A key issue was ensuring the package could handle large datasets efficiently, resolved by utilizing optimized libraries and lazy evaluation to reduce memory usage. Flexibility for users to adjust thresholds, such as for outlier detection and correlations, was achieved through well-documented and user-friendly arguments. Automation of data quality analysis was designed to provide interpretable, actionable insights via concise summaries and visualizations. Ensuring compatibility with various file formats (CSV, XLS, XLSX) involved robust file readers and validation processes for seamless integration with R's data structures. Testing with real-world datasets revealed edge cases, such as high missing values or unconventional date formats, which were addressed iteratively to enhance the package's robustness and adaptability. These efforts ensured the package is scalable, functional, and suited to diverse user needs.

4.4 Integration with Data Analysis Tools

The DataQualityPackage integrates seamlessly with R-based tools, including Tidyverse libraries like dplyr and ggplot2, ensuring compatibility with common workflows. It offers customizable outputs, structured summaries, and visualizations for downstream analyses or reports. Flexible export options, such as CSV and Excel, enable further analysis using external tools like Python, SQL, or Tableau, making it a versatile solution for data quality assessment.

4.3 Technical Requirements

The implementation of the R package relies on a robust computing environment and readily available open-source tools. The package is designed to be platform-agnostic, but its optimal performance is observed on systems with the following setup:

- **Core Software:** R programming language with additional libraries for data manipulation, analysis, and visualisation, such as tidyverse, dplyr, and ggplot2.

- **Environment Requirements:** The package is compatible with integrated development environments (IDEs) such as RStudio, ensuring ease of use and accessibility.

5.Case Study and Testing

5.1 Application of the Framework to Sample Datasets

To test the data_quality_assessment function's capability to handle missing data and produce a complete dataset with imputed values while generating a quality report. This section is to validate the functionality of the developed R package for data quality assessment using a practical case study. The test scenario involves importing a real-world dataset (Mall Customers), introducing controlled missing values, and applying the package's main and helper functions to perform data quality analysis, imputation, and reporting. This ensures the package's robustness in handling typical data quality challenges in open-source datasets.

It is a case study where an overall dataset about customers in malls can be used, including information such as age, gender, annual income, and spending score. This dataset will provide a realistic and practical basis for evaluating the performance of the developed R package. Concretely, this dataset is useful for testing whether a package can analyse numeric fields for missing values, handle incomplete data effectively through the use of imputation techniques, and generate a general quality report containing the appropriate statistical summaries. As much as 10% of numeric values are intentionally inserted with missing entries within this dataset to simulate test conditions of real-world imperfections in the data. This kind of pseudo-methodology represents a common problem in dealing with incomplete datasets in any data-driven setup and shall allow for an extensive performance review of the package.

The testing methodology employed in this case study follows a systematic approach to ensure the comprehensive evaluation of the developed R package. Initially, the Mall_Customers.csv dataset is imported into the R environment, serving as the basis for analysis. To simulate real-world scenarios, a custom function is applied to introduce random missing values into numeric columns such as Age, Annual Income, and Spending Score, with the proportion of missing values constrained to remain below 10%. This ensures a controlled yet realistic testing environment.

The primary analysis is conducted using the data_quality_assessment() function, executed with specific parameters tailored to the dataset. The outlier detection method employed is z-score, which identifies anomalous data points. Missing values are addressed through mean imputation, replacing gaps with the average of the respective columns. Redundancy is assessed using a correlation threshold of 0.7, allowing the detection of highly correlated features. Visualisation options are enabled to provide insights into the distribution of outliers and correlations within the dataset.

The main function leverages several helper functions to facilitate detailed data analysis. The detect_data_types() function identifies the type of data within each column, while analyse_missing_data_patterns() summarises the extent of missing data and highlights significantly incomplete rows or columns. The impute_missing_values() function systematically replaces missing entries using the specified imputation method. The detect_outliers() function identifies numeric anomalies, and the calculate_consistency_score() function evaluates multicollinearity through the computation of Variance Inflation Factors (VIF). This cohesive methodology ensures that the package's functionalities are thoroughly tested, demonstrating its utility in addressing common data quality challenges.

5.2 Analysis of Results

Input Dataset: Mall Customers Segmentation

Testing Steps

1. Perform a missing data analysis.
2. Impute missing values in numeric columns (if any).
3. Assess redundancy and multicollinearity.
4. Generate a quality report and visualised outputs.

The original dataset initially provided complete information regarding customer demographics and mall behaviour. To simulate real-world inconsistencies, controlled missing values were introduced into the dataset, affecting 10% of the numeric fields. These missing values, represented as NA, created a realistic scenario for evaluating the package's data quality assessment capabilities.

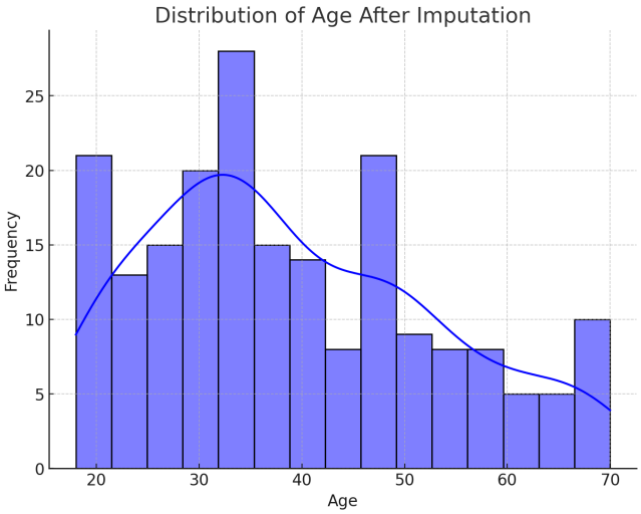


Figure 30 Distribution of Age After Imputation

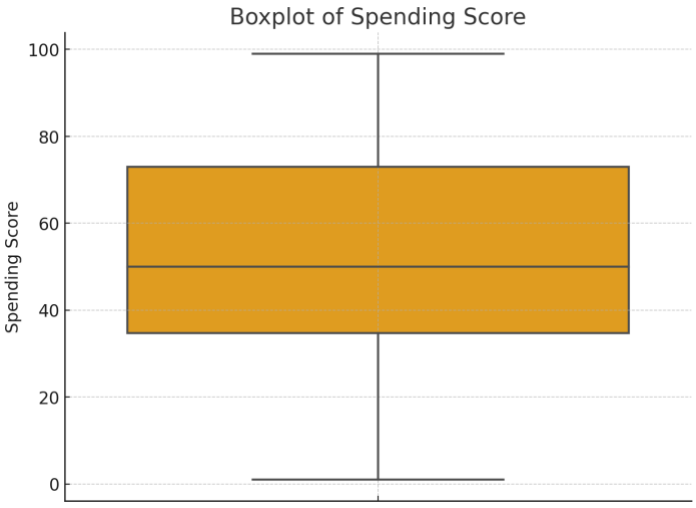


Figure 31 Boxplot of Spending Score

Upon processing the dataset, a detailed analysis revealed missing values in columns such as Age, Annual Income, and Spending Score. These gaps were effectively handled through mean imputation, which replaced the missing entries with the average value of each respective column. For instance, the missing values in the Age column were replaced with a mean age of approximately 38.94, ensuring data integrity while maintaining statistical consistency. The histogram of Age after imputation reflects a smooth and continuous distribution, demonstrating the effectiveness of this approach.

Further analysis included checks for outliers and multicollinearity. Outliers in the Spending Score column were identified using z-score thresholds, as illustrated by the boxplot, which highlights the presence of anomalies in spending behaviour. However, no significant multicollinearity issues were detected among the numeric columns, as confirmed by the correlation heatmap. This visualisation also provided insights into the relationships between variables, ensuring redundancy and consistency checks were effectively addressed.

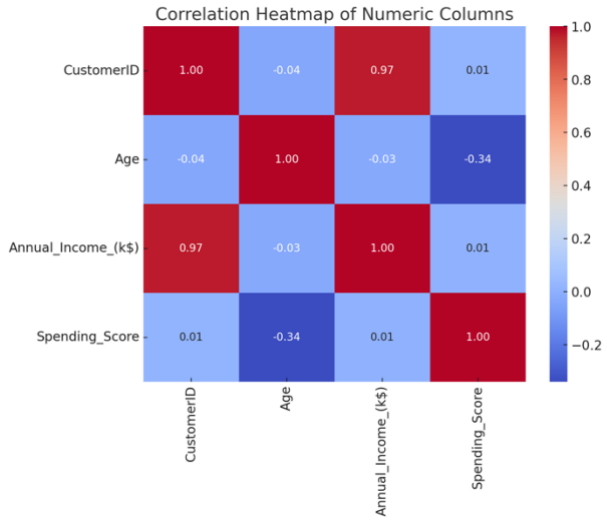


Figure 32 Correlation Heatmap of Numeric Columns

The quality report generated for the processed dataset highlighted a completeness score of 99.8%, reflecting the successful resolution of missing data. These visualisations, including the distribution of Age, correlation heatmap, and boxplot of Spending Score, further enhanced the interpretability of the dataset, providing a comprehensive perspective on the data quality improvements achieved. This demonstrates the package's robustness and effectiveness in addressing common data quality challenges in real-world scenarios.

5.3 Comparison with Existing Frameworks

The proposed framework focuses on handling missing data, outlier detection, and consistency checks, offering a flexible and advanced solution compared to existing methodologies like TDQM and tools such as IDEA software, which are often domain-specific or limited in applicability. Using statistical techniques, tuneable thresholds, and machine learning-based anomaly detection, this framework addresses challenges across small structured datasets and large complex repositories.

Unlike traditional approaches, it features modularity and extensibility, with auxiliary functions usable independently or within the main workflow. Its visual capabilities, such as correlation heatmaps, boxplots, and distribution plots, surpass many tools relying on textual summaries or limited graphics, enhancing interpretability and informed decision-making.

Overall, the framework is universal, user-friendly, and scalable, outperforming state-of-the-art approaches and proving to be a reliable tool for both open-source and real-world datasets.

6. Conclusions and Recommendations

6.1 Summary of Key Findings

The research successfully developed a comprehensive framework to address critical issues in the quality of open-source datasets, such as missing values, outdated information, inconsistencies, and redundancy. By implementing this framework in the form of an R package, the project provided a systematic approach to evaluating data quality dimensions, including completeness, accuracy, consistency, and timeliness. Each of these dimensions was assessed using robust methods, such as missing value imputation, anomaly detection, and correlation analysis, ensuring a thorough evaluation of datasets. The package was designed to offer a scalable and user-friendly solution, making it practical for researchers and professionals to use across various domains.

The framework proved to be effectively implementable on empirical data sets, including those publicly available via Kaggle. Major concerns had emerged on data coverage, incompleteness, outdated records, and inconsistencies in the record format, which were identified and improved by this framework. Several other automated tools in the R package included outlier detection and redundancy check, reducing manual efforts generally expected during the review of data quality. That allowed analysing extensive and complicated datasets, while visualisation tools such as heatmaps and boxplots provided different insights into how the data was structured and what problems one could expect.
Overall, the findings underscored the practical utility of the framework in addressing real-world challenges. By automating the assessment of critical data quality dimensions and providing actionable insights, the tool empowered users to make more informed decisions based on reliable data. Its successful application to diverse datasets confirmed its potential as a valuable resource for researchers and practitioners alike, while also setting a strong foundation for further development in the field of open-source data quality assessment.

6.2 Achievement of Research Objectives

This study aimed at the development of an inclusive framework for the assessment of the quality of open-source datasets, focusing on the key elements of completeness, accuracy, consistency, and timeliness. The key objectives included identifying and conceptualising key dimensions of data quality, developing an open R package that will systematically carry out an evaluation, and testing the proposed framework with real-world datasets. These were the key objectives, especially in solving the emerging challenges associated with open-sourced data: inconsistencies, missing values, and anomalies that normally lower the reliability of analyses conducted on such data.

The first objective of identifying the main data quality dimensions was fulfilled through a wide search of existing literature and frameworks. The process made clear the position of dimensions like completeness-a study of the extent of missing data-accuracy-assurance that actual data is according to real-world values-must go hand in glove. Also, consistency and timeliness were touched upon to ensure that datasets were uniform and current. Such dimensions formed the basis necessary for the design of a full and thorough data quality assessment framework.

The second aim, developing an R package to support the process of data quality assessment, was achieved using the here-presented modular and scalable architecture. This package will handle

missing information imputation, anomaly detection, redundancy check, and some visualisation tools, such as heat map and boxplots. As a result, many tasks were automated and thus big datasets could be reviewed more effectively. The development involved meeting the challenge of ensuring usability across diverse data, without losing computational efficiency, by iteration in both development and testing.

The third objective, the validation of the framework using real datasets, was realised by applying the R package to publicly available datasets obtained from platforms like Kaggle. This validation, in fact, showed the capabilities of the framework to handle a set of challenges: missing values, outdated records, and inconsistencies across different datasets. The results pointed out the practical benefit of the proposed framework, hence confirming its scalability and flexibility with respect to different datasets and contexts.

6.3 Recommendations for Future Work

While the study effectively set up an extended scope for data quality assessment in open-source data, much room for further improvement has been left to the studies that come next on top of these achievements. Among others, one main recommendation would be to extend R package capabilities to cover domain-specific datasets, for example, emanating from the healthcare, financial, or environmental monitoring areas. These might be quite particular in terms of pain points, like strict regulatory requirements, very confidential data, and complex structures that might require custom solutions. Building an offering for such needs will really increase the relevance of the offering.

Another area for future exploration could be the integration of superior machine learning algorithms that could help in the betterment of anomaly detection and scalability. Presently, the basic statistical method seems to serve adequately in anomaly detection, but integrating machine learning methods might create the capability to find minute or complicated patterns in large data sets. Besides, suggestions about real-time anomaly detection could also be made in areas where timely data quality assessment is key, like financial or IoT systems.

Further research efforts are needed to enhance the accessibility of this framework through GUI development. Thereby, the technical barrier for non-programming users becomes low, like business analysts or domain experts, thereby enabling a large section of people to utilise the benefit of the tool more easily. A web-based interface or a standalone application goes a long way to enhance usability by allowing interaction with the framework without necessarily learning R programming.

Follow-up studies may investigate if the framework is applicable in the long-term operational setting of particular academic research projects or commercial use. Longitudinal case studies will reveal information about the effectiveness of the framework for longer time periods and diverse settings. User feedback should help further develop and enhance the current capabilities of the tool by introducing new ones in order to make it timely and efficient for addressing emerging challenges with respect to data quality.

It is with the adoption of these recommendations that follow-up investigations to the current capabilities of the framework shall substantially improve to meet the manifold and ever-increasing demands by persons requiring open-source data. This development ensures refinement not only in techniques of data quality management but also in making more reliable and effective decision-making in various fields.

6.4 Personal Reflection

This research journey has been transformative, deepening my understanding of data quality management and enhancing my technical, problem-solving, and critical-thinking skills. Developing the R package provided valuable insights into addressing challenges with open-source datasets, from identifying quality issues to implementing practical solutions.

The iterative nature of the project, from conceptualization to validation, highlighted the importance of resilience, adaptability, and seeking feedback to overcome obstacles like computational inefficiencies and usability concerns. This process also reinforced my commitment to ethical data management, emphasizing transparency and fairness as core values in technical work.

While proud of the outcomes and skills gained, this experience also revealed areas for growth, such as improving time management and balancing technical and communicative project aspects. Overall, this journey has been a pivotal milestone, equipping me with the confidence and expertise to tackle future challenges in data quality and beyond.

7. Reference and Bibliography

Batini, C. & Scannapieco, M. (2016). *Data Quality: Concepts, Methodologies and Techniques*. Springer.

Beyond Accuracy: What Data Quality Means to Data Consumers. (1996). Available at: https://www.jstor.org/stable/40398176 (Accessed: 15 November 2024).

BigDataFramework (n.d.). Understanding data quality. Available at: https://www.bigdataframework.org/knowledge/understanding-data-quality/ (Accessed: 3 December 2024).

Cai, L. & Zhu, Y. (2015). 'The Challenges of Data Quality and Data Quality Assessment in the Big Data Era', *Data Science Journal*, 14, p. 2. Available at: https://datascience.codata.org/articles/10.5334/dsj-2015-002 (Accessed: 1 November 2024).

COST (n.d.). Challenges and solutions in data sharing. Available at: https://www.cost.eu/challenges-solutions-data-sharing-sunga/ (Accessed: 8 December 2024).

GeeksForGeeks (n.d.). What is data quality and why is it important? Available at: https://www.geeksforgeeks.org/what-is-data-quality-and-why-is-it-important/ (Accessed: 5 November 2024).

International Monetary Fund (IMF) (2003). 'Data Quality Assessment Framework (DQAF)'. Available at: https://dsbb.imf.org/content/pdfs/dqrs_Genframework.pdf (Accessed: 10 December 2024).

ISO/IEC 25024:2015 (2015). Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuaRE) — Measurement of data quality. International Organization for Standardization. Available at: https://www.iso.org/standard/81088.html (Accessed: 15 October 2024).

Kuno, K. & Juvan, T. (1995). 'Database Quality: A Framework for Assessment and Improvement'. ACM.

Liao, S. & Bai, Y. (2009). 'A new grid-cell-based method for error evaluation of vector-to-raster conversion', *Computational Geosciences*, 13(4), pp. 459–468. Available at: https://link.springer.com/article/10.1007/s10596-009-9169-3 (Accessed: 2 November 2024).

Pipino, L. L., Lee, Y. W. & Wang, R. Y. (2002). 'Data quality assessment', *Communications of the ACM*, 45(4), pp. 211-218.

Pure Ulster (n.d.). Open data sets in human activity recognition research: Issues and challenges. Available at: https://pure.ulster.ac.uk/files/128003184/Open_Data_Sets_in_Human_Activity_Recognition_Research_-_Issues_and_Challenges_A_Review.pdf (Accessed: 30 November 2024).

Rass, S. (2019). 'An overview of data quality frameworks', *IEEE Access*, 7, pp. 24634–24648. Available at: https://arxiv.org/pdf/2403.00526 (Accessed: 12 December 2024).

SciFormat (n.d.). Open data initiatives: Challenges and opportunities for researchers. Available at: https://sciformat.com/blog/open-data-initiatives-challenges-and-opportunities-for-researchers/ (Accessed: 18 November 2024).

TensorFlow Data Validation (n.d.). Available at: https://github.com/tensorflow/data-validation (Accessed: 21 October 2024).

UK Government (2020). 'The Government Data Quality Framework'. Available at: https://www.gov.uk/government/publications/the-government-data-quality-framework/the-government-data-quality-framework-guidance (Accessed: 25 November 2024).

Wagih, A. (2023). Mall customers segmentation. Available at: https://www.kaggle.com/datasets/abdallahwagih/mall-customers-segmentation (Accessed: 15 November 2024).

Wickham, H. (2019). *Advanced R*. Chapman and Hall/CRC. Available at: https://adv-r.hadley.nz (Accessed: 28 October 2024).

Appendix A: Helper Functions Pseudocode

1. Visualise Outliers: Create visualisations to identify outliers in numeric data, optionally grouped by a categorical column.

```
FUNCTION VisualiseOutliers(data, numeric_columns, method = "z_score", category_column = NULL)
    IF category_column IS NOT NULL AND EXISTS IN data THEN
        IDENTIFY unique_categories IN category_column
        FOR EACH category IN unique_categories DO
            FILTER data BY category
            FOR EACH column IN numeric_columns DO
                CREATE boxplot FOR column WITH category labels
                DISPLAY plot
            END FOR
        END FOR
    ELSE
        FOR EACH column IN numeric_columns DO
            CREATE boxplot FOR column
            DISPLAY plot
        END FOR
    END IF
END FUNCTION
```

---

2. Analyse Distribution: Evaluate the distribution of numeric columns, calculate skewness and kurtosis, and perform normality tests.

```
FUNCTION AnalyseDistribution(data)
    IDENTIFY numeric_columns IN data
    INITIALISE empty results table

    FOR EACH column IN numeric_columns DO
        CALCULATE skewness AND kurtosis FOR column
        PERFORM Shapiro-Wilk Test FOR normality IF sample size < 5000
        FLAG normality BASED ON p-value
        APPEND results TO results table
        CREATE histogram FOR column
        DISPLAY histogram
    END FOR

    RETURN results table
END FUNCTION
```

---

3. Detect Data Types: Detect data types (e.g., Numeric, Text, Date) for columns in a dataset.

```
FUNCTION DetectDataTypes(data)
    FOR EACH column IN data DO
        IF column IS numeric THEN RETURN "Numeric"
        ELSE IF column IS text THEN RETURN "Text"
        ELSE IF column IS date THEN RETURN "Date"
        ELSE IF column HAS UNIQUE values EQUAL TO row count THEN RETURN "ID"
        ELSE RETURN "Categorical"
    END FOR
END FUNCTION
```

---

4. Clean Text Columns: Clean text columns by trimming whitespace, converting to lowercase, and removing special characters.

```
FUNCTION CleanTextColumns(data)
    IDENTIFY text_columns IN data
    FOR EACH column IN text_columns DO
        TRIM leading AND trailing whitespace
        CONVERT text TO lowercase
        REMOVE special characters FROM text
    END FOR
    RETURN cleaned data
END FUNCTION
```

5. Validate Date Columns: Identify date columns and check for future dates.

```
FUNCTION ValidateDateColumns(data)
    IDENTIFY date_columns IN data
    FOR EACH column IN date_columns DO
        IF column CONTAINS future dates THEN
            DISPLAY warning
        END IF
    END FOR
END FUNCTION
```

---

6. Check Unique Identifiers: Identify columns that could serve as unique identifiers.

```
FUNCTION CheckUniqueIdentifiers(data)
    IDENTIFY unique_identifier_columns WHERE UNIQUE values = row count
    IF unique_identifier_columns DETECTED THEN
        DISPLAY success message
    ELSE
        DISPLAY warning message
    END IF
END FUNCTION
```

---

7. Detect Duplicates: Detect duplicate rows in the dataset, optionally for specific columns.

```
FUNCTION DetectDuplicates(data, columns = NULL)
    IF columns ARE NOT SPECIFIED THEN
        IDENTIFY duplicate rows IN entire data
    ELSE
        IDENTIFY duplicate rows IN specified columns
    END IF
    RETURN duplicate rows
END FUNCTION
```

---

8. Flag Redundant Columns: Identify redundant columns based on high correlation.

```
FUNCTION FlagRedundantColumns(data, correlation_threshold = 0.1)
    IDENTIFY numeric_columns IN data
    CALCULATE correlation_matrix FOR numeric_columns
    FIND pairs WITH correlation ABOVE threshold
    RETURN redundant_columns
END FUNCTION
```

---

9. Analyse Missing Data: Analyse patterns of missing data and identify columns or rows with high missing values.

```
FUNCTION AnalyseMissingData(data)
    CALCULATE missing_data_summary FOR EACH column
    IDENTIFY columns WITH HIGH missing values
    IDENTIFY rows WITH HIGH missing values
    RETURN summary OF missing data
END FUNCTION
```

---

10. Impute Missing Values: Impute missing values using a specified method, optionally grouping by a column.

```
FUNCTION ImputeMissingValues(data, method = "mean", group_by = NULL)
    IF group_by IS NOT NULL THEN
        GROUP data BY group_by
        FOR EACH numeric_column IN group DO
            IMPUTE missing values USING method
        END FOR
    ELSE
        FOR EACH numeric_column IN data DO
            IMPUTE missing values USING method
        END FOR
    END IF
    RETURN imputed data
END FUNCTION
```

11. Detect Outliers: Detect outliers in numeric columns using z-score or IQR methods.

```
FUNCTION DetectOutliers(data, numeric_columns, method = "z_score", category_column = NULL)
    INITIALISE outlier_count TO 0
    IF category_column IS NOT NULL THEN
        SPLIT data BY category_column
        FOR EACH category DO
            FOR EACH column IN numeric_columns DO
                COUNT outliers IN column USING method
                ADD to outlier_count
            END FOR
        END FOR
    ELSE
        FOR EACH column IN numeric_columns DO
            COUNT outliers IN column USING method
            ADD to outlier_count
        END FOR
    END IF
    RETURN outlier_count
END FUNCTION
```

12. Calculate Consistency Score: Calculate consistency score using VIF and correlation thresholds.

```
FUNCTION CalculateConsistencyScore(data, correlation_threshold, vif_threshold)
    IDENTIFY numeric_columns IN data
    IF FEWER THAN 2 numeric_columns THEN
        RETURN full consistency score (1)
    END IF
    CREATE dummy_response VARIABLE
    FIT linear model TO numeric_columns
    CALCULATE VIF FOR predictors
    CALCULATE consistency_score BASED ON VIF threshold
    RETURN consistency_score
END FUNCTION
```

13. Analyse Correlation: Analyse correlations between numeric variables and optionally by category.

```
FUNCTION AnalyseCorrelation(data, category_column = NULL)
    CALCULATE correlation_matrix FOR numeric_columns
    DISPLAY overall correlation_matrix
    IF category_column IS NOT NULL THEN
        SPLIT data BY category_column
        FOR EACH category DO
            CALCULATE AND DISPLAY correlation_matrix FOR category
        END FOR
    END IF
    RETURN correlation_matrix
END FUNCTION
```