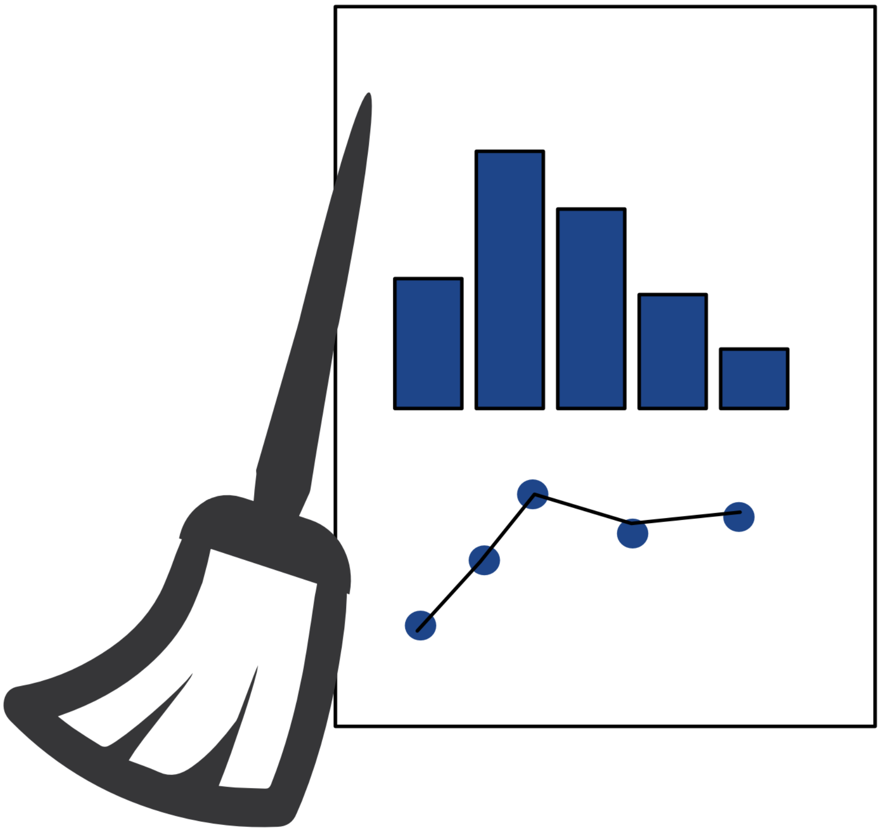
CHALMERS
UNIVERSITY OF TECHNOLOGY



Developing a Cooperative Data Cleaning Tool

Master's thesis in Engineering Mathematics and Computational Science

DEVOSMITA CHATTERJEE

Department of Engineering Mathematics and Computational Science
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2021

Master's thesis 2021

Developing a Cooperative Data Cleaning Tool

DEVOSMITA CHATTERJEE

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Mathematical Sciences
Division of Applied Mathematics and Statistics
Chalmers University of Technology
Gothenburg, Sweden 2021

Developing a Cooperative Data Cleaning Tool
DEVOSMITA CHATTERJEE

Industrial Supervisor: Sven Ahlinder, Volvo Group Trucks Technology
Academic Supervisor: Anton Johansson, Chalmers University of Technology
Examiner: Serik Sagitov, Chalmers University of Technology

Master's Thesis 2021
Department of Mathematical Sciences
Division of Applied Mathematics and Statistics
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: DataCleaningTool Application Logo.

Typeset in LaTeX
Gothenburg, Sweden 2021

Developing a Cooperative Data Cleaning Tool
DEVOSMITA CHATTERJEE
Department of Mathematical Sciences
Chalmers University of Technology

Abstract

Presently, large amount of data generated by organizations drives their business decisions. The data is usually inconsistent, inaccurate and incomplete. Poor data quality may lead to incorrect decisions for the organizations and hence, negatively affect them. Thus, high quality data is of utmost priority to draw good and valid business decisions and strategies. Data cleaning is the ultimate way to solve the data quality issues. But, data cleaning is really a time consuming task. Thus, tools which can help with the task are needed. This demands data cleaning tools for systematically examining data for errors and automatically cleaning them using algorithms. These data cleaning tools helps organizations save time and increase their efficiency.
In this thesis, we develop a cooperative, free and open source data cleaning standalone application 'DataCleaningTool' in order to achieve the task of data cleaning. This tool is able to identify the potential data problems and report results such that the users can take informed decisions to clean data effectively.

Keywords: Data Cleaning, Noisy Data, Missing Data, MissForest Method, Outliers, Data Transformation, Interactive Data Visualization.

Acknowledgements

Firstly, I would like to express my sincere gratitude to my industrial supervisor, Sven Ahlinder, for his invaluable support and encouragement throughout the project. His enthusiasm about the project motivated me a lot. I would also like to thank Lena Jansson for warmly welcoming me into her team in Volvo. Special thanks to Klara Jansson, Electromobility Group, Volvo for helpful discussions during the course of the thesis. I have thoroughly enjoyed all morning and afternoon coffee breaks, lunch talks, and interesting discussions in Volvo Powertrain department.

I would like to thank Anton Johansson, my academic supervisor, for enthusiastically supporting my work and answering my questions. He always gave me constructive feedback and helped me in setting priorities. I would also like to thank Serik Sagitov for being my examiner.

Lastly, I would like to thank my parents, my in-laws and my husband for all the support.

Devosmita Chatterjee, Gothenburg, September 2020

Contents

List of Figures

List of Tables

List of Algorithms

1 Introduction

Understanding and organizing data effectively is a crucial component for the success of modern day organizations, especially today with the advent of the what is known as the "Big Data" era. The term "Big Data" was first introduced by Roger Magoulas from O'Reilly media in 2005 [1], in order to define a large amount of data that traditional data management techniques cannot manage due to the complexity and size of the data. The organizations need to understand the four V's of big data- Volume, Velocity, Variety and Veracity [2] in order to develop tools to manage data and turn it into valuable insights.

- Volume refers to the large amount of data generated by organizations. This requires organizations to address challenges in storing and analyzing such large amount of data.
- Velocity refers to the time in which data can be processed. Data is most effective when analysed in real time rather than storing it in a database to be analyzed later. This is because ongoing analysis allows for the immediate application of findings for improvement of services.
- Variety refers to the broad range of different kinds of data being generated that come from different sources. In the present world, data comes not only from computers but also from other devices such as smartphones. Data can not only be in a structured way that fits a table but also in an unstructured way such as tweets, online comments, photos and videos in social media.
- Veracity refers to the reliability of data that is being analyzed. Data must be cleaned, current, and of high quality and reliability before it is analyzed to make right business decisions for the organizations.

The real world data is dirty and data cleaning offers a better data quality hence ensuring the aspect of data veracity.

In this thesis, we are concerned with the task of data cleaning. A tool is developed to offer cooperative support to users to clean data effortlessly. In Section 1.1, we introduce the basic background of the thesis project. In Section 1.2, we present the main objective of the data cleaning tool. Section 1.3 presents an overview of some existing data cleaning tools. The further outline of this thesis is described in Section 1.4.

1.1 Background

Engineers at "Powertrain Strategic Development" department, Volvo Group Trucks Technology develop new innovative powertrains for the trucks of the future. Data analysis is needed to correctly define and size the different components of the future powertrains. The most time consuming part is to prepare the data for analysis. The foremost approach for preparing data is to clean it which requires identification of the errors in the data. Data cleaning helps to improve the quality of the data. However, it is a daunting task to go through manually such large number of datasets for identifying the errors. Thus, tools which can help with the task are needed. This demands data cleaning tools. Nowadays, data cleaning tools have become more predominant in analytics driven organisations, that systematically examine data for errors using algorithms. These data cleaning tools help organizations save time and increase their efficiency. Such kind of tools are therefore of great interest to Volvo.

1.2 Scope

The primary idea of the thesis is to develop a cooperative tool instead of a black box. The thesis is aimed at developing a user friendly, free and open source standalone application named 'DataCleaningTool' to support data cleaning in a cooperative way. The tool motivates and illustrates its suggestions at every stage of the data cleaning process. Thereafter, the data scientists at Volvo will use the tool for data cleaning before analysing the data.
DataCleaningTool is designed to be cooperative which means

- No Black Box
  - DataCleaningTool is not a black box which means that it does not produce any result without understanding how it works.
- User cooperative
  - The primary concern is the users who take decisions at every stage of data cleaning.
- User friendly
  - DataCleaningTool is easy to install. App installation is the first thing users need to do, so it is better to be a friendly process, otherwise users are going to be afraid to use the application.
  - DataCleaningTool is a clean graphical user interface which allows users to immediately start using the application.
  - DataCleaningTool is provided with a user manual. The user manual presents an overview of the application's attributes and gives step-by-step instructions for performing a variety of tasks.
- Standalone
  - DataCleaningTool is a standalone application created from Matlab functions so that it can be used to run Matlab compiled program on computers that do not have Matlab installed.
- Freeware
  - DataCleaningTool is a freeware application so that it can be distributed, downloaded, installed and used at no monetary cost.
- Open source
  - DataCleaningTool is a open source application so that programmers have access to a computer program's source code to improve the program by adding attributes to it or fixing different parts of the program.
- Code free
  - DataCleaningTool provides a code free environment to users. This implies that the user performs tasks without writing code.
- Illustrates possible data problems.
  - DataCleaningTool displays input data in table format which represents the structural errors.
  - DataCleaningTool shows statistical information about the data.
  - DataCleaningTool contains visualization techniques for identifying noisy data, missing data and outliers.
  - DataCleaningTool contains visual methods for exploring data transformations.
- Addresses different data problems.
  - Each button aims to clean data by resolving inconsistencies, smoothing noisy data, removing outliers or filling in missing observations.
- Helps the user to take informed decisions
  - All widgets' information gets updated automatically after each activity.
  - DataCleaningTool displays both information messages and error messages.
- Provides interactive data visualizations
  - DataCleaningTool enables users to explore and manipulate various aspects of graphical representation of data by clicking on a button or moving a slider.

The general idea of DataCleaningTool is to provide the following code free assistances to users to clean data effectively. However, the user makes the final decision.

- Automated Display of Data and Statistical Information of Data
  - Display data in table format.

  - Show data properties.
  - Show descriptive statistics of numerical, text and datetime features.
- Automated Data Type Discovery
  - Discover basic statistical data types such as numerical, text and datetime.
- Removal of Unwanted Data
  - Identify irrelevant observations which do not fit the specific problem that the user is trying to solve.
  - Replace an irrelevant observation with a missing observation.
  - Drop any row with an irrelevant observation.
- Outlier Detection
  - Illustrate possible outliers.
  - Replace an outlier with a missing observation.
  - Drop any row with an outlier.
- Missing Data Handling
  - Illustrate missing observations.
  - Drop rows with missing observations.
  - Drop features with missing observations.
  - Fill in missing observations.
- Data Transformation
  - Transform numerical features.
  - Illustrate transformed numerical features.
- Data Visualization
  - Histogram for plotting a numerical feature.
  - Bar chart for plotting a categorical feature.
  - Box plot for graphing a numerical feature by categories of a categorical feature.
  - Missingness plot for visualizing missing observations.
  - Line graph for plotting the missing observations percentage of each feature.

1.3 Existing Data Cleaning Tools

Data cleaning is a process for removing incomplete, incorrect or inaccurate parts of data from a table or a database and then replacing, modifying or deleting the dirty data. Data cleaning tools help in keeping the data consistent and clean to let the users analyse data to make more informed decision visually as well as statistically. There are many data cleaning tools that provide data cleaning services such as duplicate eradication and ensuring accuracy but only few tools focus on cleaning different types of data errors or anomalies such as noisy data, missing data and outliers. Few of these tools are free, while others are priced with free trial. In this section, we give an overview of some powerful code free tools which are capable of providing user assistance for data cleaning.

OpenRefine
OpenRefine [3] formerly known as Google Refine, is an open source powerful data cleaning tool. It helps to prepare messy data by cleaning it, transforming it from one format into another and extending it with web services.

Trifacta Wrangler
Trifacta Wrangler [4] is an interactive tool for data cleaning and transformation. It is used to clean and prepare messy, real world data quickly and accurately for analysis. The data can be exported for use in Excel, R, Tableau and Protovis.

Winpure
Winpure [5] is a good data quality software. It tackles problems such as inaccurate data and duplicate data and cleans the database of duplicate data, bad entries and incorrect information.

datacleaner

datacleaner [6] is a Python package for data cleaning. It works with data in pandas DataFrames. It is used for the following tasks: drops any row with a missing observation, replaces missing observations with the mode (for categorical variables) or median (for continuous variables) on a column by column basis, encodes categorical features with numerical equivalents.

dataMaid

dataMaid [7] is a R package for data cleaning. It is used to deal with the following errors in data: incorrect class, duplicates, capitalization inconsistency, nonsensical data, extra white spaces, missing data, unique observations / categories with low count and inaccurate data.

SAS

SAS's anomaly detection system detects and excludes anomalies using the Support Vector Data Description. SAS Institute [8] is a leading American multinational developer of analytics software. Briefly, the Support Vector Data Description identifies anomalies by determining the smallest possible hypersphere using support vectors that encompasses the datapoints. The Support Vector Data Description excludes the datapoints that lie outside of the sphere.

Anodot

Anodot's automated anomaly detection system detect anomalies for time series data. Anodot [9] is an American data analytics company which uses machine learning techniques for anomaly detection. First, the system classifies the time series data and then, the system selects an optimal mathematical model which will be used to describe the normality of the data. When there is one seasonal pattern, the system uses Fourier Transform. When there are multiple seasonal patterns, the system uses its own algorithm, named "Vivaldi" based on autocorrelation function. The system determines the temporal statistical distribution of datapoints to be expected in the data. The system applies a statistical test to all datapoints based on the expected distribution. If the datapoint falls outside the distribution, it is most likely an anomaly.

Happiest Minds

Happiest Minds' automated anomaly detection system helps to detect anomalies for both categorical and numerical data using statistical, supervised and artificially intelligent algorithms. Happiest Minds [10] is an Indian IT company.

A comparison chart between different data cleaning tools is presented in table 1.1.

Table 1.1: The table represents the comparison between data cleaning tools.

Data Cleaning Tools	**Freeware**	**Handling Data Inconsistency**	**Handling Missing Data**	**Handling Outliers**	**Data Transformation**
DataCleaningTool	✓	✓	✓	✓	✓
OpenRefine	✓	✓			✓
Trifacta Wrangler	✓	✓	✓	✓	
Winpure		✓	✓	✓	
datacleaner	✓		✓		
dataMaid	✓	✓	✓	✓	
SAS				✓	
Anodot				✓	
Happiest Minds				✓	

1.4 Thesis Outline

The thesis is structured as follows: Chapter 2 demonstrates the background knowledge of data cleaning. Common data problems and corresponding data cleaning techniques are investigated. Chapter 3 explains our data cleaning approach to address common data problems which assists users to clean data in a cooperative way. In Chapter 4, the results of a performance analysis of the missForest method and the different outlier detection methods are discussed and a demo version of our data cleaning tool is presented. Lastly, Chapter 5 wraps up the thesis and presents the possible improvements for future work.

2

Data Problems and their Cleaning Approaches

This chapter provides the background theory regarding data cleaning. Section 2.1 states the concept of data cleaning. In Sections 2.2, 2.3, 2.4, 2.5 the major data problems in raw data are explored and the corresponding state-of-the art data cleaning techniques are described. Different data visualization techniques are presented in Section 2.6.

2.1 Data Cleaning

Nowadays, it is becoming easier for organizations to store and acquire large amounts of data. Machine learning can learn and make predictions on the data to facilitate improved decision making and richer analytics. However, the problem is that the real world data almost never come in a clean way and poor data quality can lead to incorrect decisions and unreliable analysis. As a result, raw data needs to be preprocessed before being able to proceed with training machine learning models. The preprocessing task which aims to deal with data problems is called data cleaning.

Data cleaning is a three-step iterative process - clean data $\longleftrightarrow$ reduce data $\longleftrightarrow$ transform data that proceeds until the data is in its most useful form to the user as shown in figure 2.1.

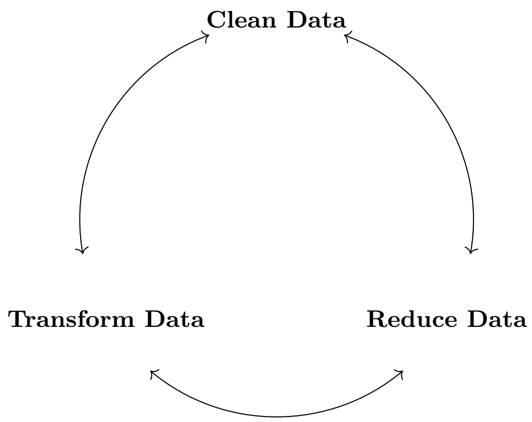


Figure 2.1: The iterative nature of the data cleaning process. Each double sided arrow indicates the relation between the different steps of the process.

The iterative steps of data cleaning are

- Clean data is the process of cleaning the data, such as noisy data and outliers.
- Reduce data is the process of reducing the data in volume, such as numerosity reduction and dimensionality reduction if the dataset is too large or high dimensional and unmanageable and the reduced data produces almost the same analytical results.
- Transform data is the process of transforming the data into useful forms, such as logarithmic transformation for data mining to statistically measure it.

We introduce the major data problems [11] and the possible approaches to fix them.

Formatting Errors
- Example: Misspellings.
- Possible Approach: Use Microsoft Word's spell checker [12].

Inconsistent feature names or columns
- Example: Feature names or columns have inconsistent capitalizations.
- Possible Approach: Use uppercase or lowercase characters.

Typographical errors
- Example: Extra white spaces.
- Possible Approach: Remove extra white spaces.

Duplicate data
- Example: Duplicate columns or rows.
- Possible Approach: Remove extra columns or rows.

Incorrect data type
- Example: Numerical instead of string entries.
- Possible Approach: Set data type constraint.

Nonsensical data
- Example: Age = -1.
- Possible Approach: Set range constraint to variable - Age ≥ 0.

Extrapolation errors
- Example: A model of glacial retreat: $V = 100 - 2t$ where V = volume of ice, t = time variable, and $t = 0$ AD. If we extrapolate to earlier than $t = 0$, then ice volume becomes bigger. Mathematically, we can extrapolate back in time but then the ice volume of the glacier would exceed the total volume of the earth which is absurd.
- Possible Approach: Set range constraint to variable - $t \geq 0$.

Systematic errors
- Example: A poorly calibrated thermometer would result in measured values that are consistently too high.
- Possible Approach: No solution to the problem.

Truncation error
- Example: Difference between the actual value (2.99792458×10^8) and the truncated value up to two decimals (2.99×10^8).
- Possible Approach: Use long format [13].

Time stamp errors
- Example: The first failure time can show time prior to when the electric vehicles were produced if the vehicle clock has not been correctly set.
- Possible Approach: Set cross-field validation constraint to variable - first failure time of a vehicle > time when the vehicle was produced.

Fault code count
- Example: Fault codes are codes stored by the on-board computer diagnostic system that notify about a particular problem area found in the car. Fault code count starts only when a problem is detected in the car. Sometimes although an issue is notified, fault code count = 0.
- Possible Approach: Set range constraint to variable - fault code count > 0.

Missing data
- Example: NaN.
- Possible Approach: Imputation using MissForest method. [14].

Sparse data
- Example: Columns that are infrequently populated.
- Possible Approach: Non negative matrix factorization for non-negative sparse data [15].

Spurious correlations
- Example: US spending on science, space, and technology highly correlates with suicides by hanging, strangulation, and suffocation in US.
- Possible Approach: Additive noise method, information geometric causal inference [16].

Seasonality

- Example: A sudden surge in order volume at an eCommerce company if the high order volume occurs outside of a promotional discount or high order volume period like Black Friday. This could be due to a pricing glitch which is allowing customers to pay substantially less money for a product. Recently, on Amazon Prime Day, a pricing glitch allowed customers to buy a $13,000 camera lens for just $94.
- Possible Approach: Fourier transform for single seasonal pattern [17], autocorrelation function for multiple seasonal patterns [18].

Measurement errors

- Example: Self-reported energy intake used to estimate actual energy intake.
- Possible Approach: Leverage statistics [19].

Outliers

- Example: Fraudulent credit card transactions.
- Possible Approach: Local outlier factor [20].

In our data cleaning, we are dealing with errors such as inconsistent feature names, duplicate data, incorrect data type, nonsensical data, extrapolation errors, truncation error, time stamp errors, fault code count, missing data and outliers. Common data problems faced by Volvo analysts are truncation errors, time stamp errors and fault code count.

2.2 Data Type Discovery

One of the first step in data cleaning is to discover the different data types of all features. Not all methods are applicable for all different data types and data type discovery is therefore a vital first step in order to proceed with the analysis.

2.2.1 Data Types

Data type of a feature can be either numerical/quantitative data or categorical/qualitative data. Further, numerical/quantitative data can be classified as continuous (interval or ratio) and discrete whereas categorical/qualitative data can be classified as nominal and ordinal [21]. Figure 2.2 shows the different useful data types in machine learning and the relation between them.

Numerical/quantitative data

1. Continuous data is a type of numerical data which takes values within a range. For example, average weights for 5 women are 63 kg, 70.1 kg, 53.7 kg, 68.5 kg and 69 kg. Continuous data can be either interval or ratio [22].
   (a) Interval data have constant distances between values. It never assumes absolute zero. For example, zero on the Celsius temperature scale does not imply that there is an absence of temperature or kinetic energy rather, it indicates the temperature at which water freezes.
   (b) Ratio data assumes zero where there is no measurement. For example, the number of comments on a social media post because the case includes an absolute zero.
2. Discrete data is a type of numerical data which takes only certain fixed values. For example, number of students present in class per weekday are 25, 23, 24, 24 and 25. Number of students can not be 23.5.

Categorical/qualitative data

1. Nominal data is a type of categorical data which contains variables with no ranking order. For example, languages such as English, French, German and Spanish.
2. Ordinal data is a type of categorical data which contains variables in a finite ordered set. For this kind of data, there is a natural order among categories. For example, different sizes such as large, medium and small.
3. Binary data is a type of categorical data which contains variables with only two states. For example, two possible options such as pass or fail.

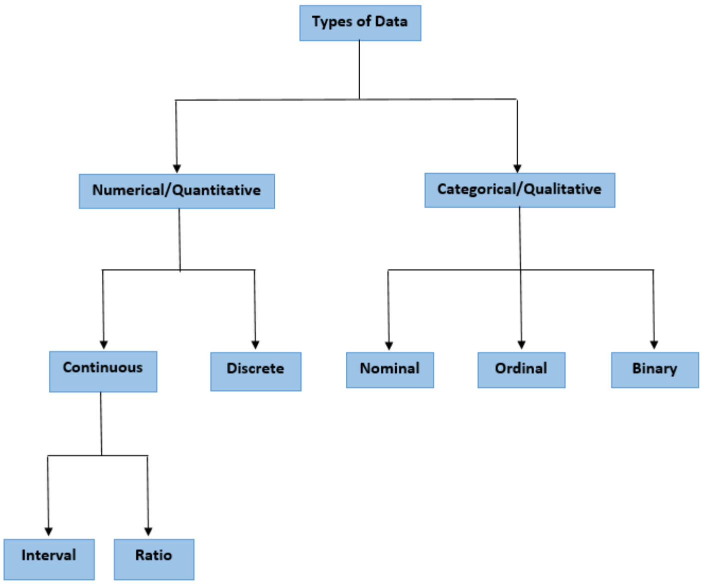


Figure 2.2: The hierarchical structure of the data types.

2.2.2 Data Type Conversion Methods

Label encoding
This is an encoding technique which convert the categorical ordinal data into model understandable numerical data. In label encoding, each category is assigned a value from 0 to $n-1$ where n is the number of categories. For example, let's say we have an ordinal data column 'safety' as seen in figure 2.3 that has labels 'low', 'medium', 'high' and 'very high'. When we apply label encoding to the 'safety' column, the label 'low' is converted to '0', the label 'medium' is converted to '1', the label 'high' is converted to '2', and the label 'very high' is converted to '3'.

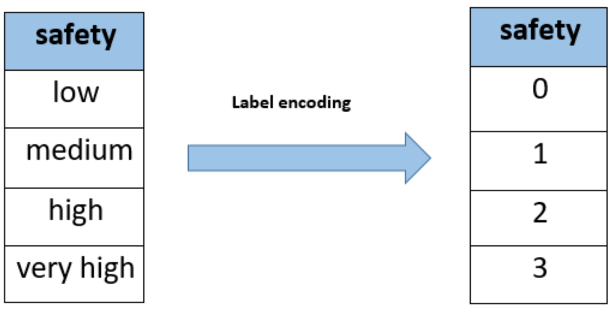


Figure 2.3: Label encoding of categorical data. After applying label encoding to 'safety' feature, the four categories of the feature - 'low', 'medium', 'high' and 'very high' are assigned values from 0 to 3.

The label encoding method has the following advantages:

- We usually apply label encoding when the categorical feature is ordinal in order to preserve the natural order that existed in the original feature.
- Label encoding preserves the natural order of the data.

The label encoding method has the following disadvantage:

- If label encoding is applied on nominal data, the numeric values can be misinterpreted by algorithms as having some kind of hierarchy or order in them.

One-hot encoding

This is an encoding approach which splits the categorical nominal data into multiple dummy variables [23]. If a categorical feature has n values, then one-hot encoding splits it into n dummy variable columns which takes only two quantitative values 1 and 0 in the presence and absence of the respective value. For example, let's say we have a nominal data column 'language' as seen in figure 2.4 that has labels 'English', 'French', 'German' and 'Spanish'. When one-hot encoding is done, the 'language' column is split into four new columns, one for each language. If the first column value of the 'language' column is 'English', then after one-hot encoding, the first column value of the 'English' column is '1' and that of the 'French', the 'German' and the 'Spanish' columns are '0'.

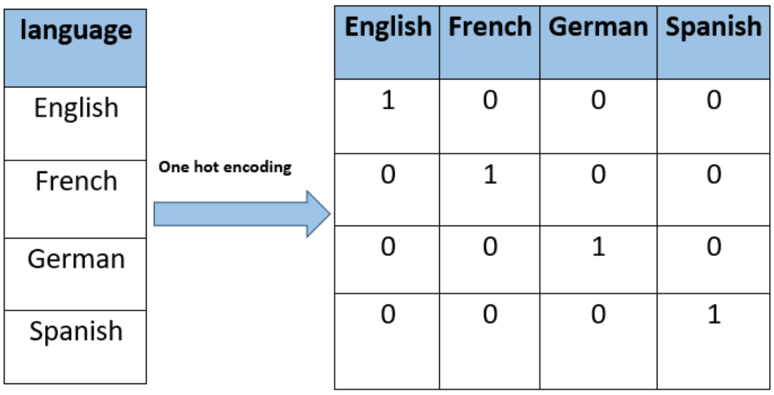


Figure 2.4: One-hot encoding of categorical data. After applying one-hot encoding to 'language' feature, the feature is split into four dummy variable columns, one for each category. If the first observation of the 'language' feature is 'English', then after one-hot encoding, the first observation of the 'English' feature is '1' and that of the 'French', the 'German' and the 'Spanish' features are '0'.

One-hot encoding results in dummy variable trap. Dummy variable trap is a scenario where the independent variables are highly correlated and one variable can be predicted from the remaining variables. Thus, dummy variable trap leads to the problem of perfect multicollinearity. Multicollinearity is a phenomenon in which two or more independent variables are highly correlated with one another in a multiple regression model. Perfect multicollinearity means that the correlation between two independent variables is equal to 1 or -1. In case of perfect multicollinearity, ordinary least squares can not calculate regression coefficients. So the recommendation is to use $n-1$ columns for multiple linear regression and logistic regression, and n columns for all kinds of subspace regression such as singular value decomposition.

Let X be a categorical feature with n categories $\{X_1, X_2, \cdots, X_{n-1}, X_n\}$. After one-hot encoding of X, the following holds

$$X_1 + X_2 + \cdots + X_{n-1} + X_n = 1. \tag{2.1}$$

Then the multivariate regression model

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_{n-1} X_{n-1} + \beta_n X_n \tag{2.2}$$

can be written as

$$\begin{aligned} Y &= \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_{n-1} X_{n-1} + \beta_n (1 - X_1 - X_2 - \cdots - X_{n-1}) \\ \implies Y &= (\beta_0 + \beta_n) + (\beta_1 - \beta_n) X_1 + (\beta_2 - \beta_n) X_2 + \cdots + (\beta_{n-1} - \beta_n) X_{n-1} \\ \implies Y &= C_0 + C_1 X_1 + C_2 X_2 + \cdots + C_{n-1} X_{n-1} \end{aligned} \tag{2.3}$$

where $C_0 = \beta_0 + \beta_n$, $C_1 = \beta_1 - \beta_n$, $C_2 = \beta_2 - \beta_n$ and $C_{n-1} = \beta_{n-1} - \beta_n$.

Thus, categorical feature with n categories is transformed to $n-1$ dummy features to avoid multicollinearity.

The one-hot encoding method has the following advantages:

- We usually apply one-hot encoding when the categorical feature is nominal.
- The result of one-hot encoding is binary rather than ordinal that lies in an orthogonal vector space.

The one-hot encoding method has the following disadvantages:

- One-hot encoding can be effectively applied only when the number of categorical features is few.
- One-hot encoding can lead to high memory consumption if the number of categorical features in the dataset is huge or the number of categories of a categorical feature is large.

2.3 Missing Data Handling

Missing data means that one or more observations are missing generally denoted by NaN, NaT or ' '. This often occurs due to improper data collection, lack of data, or data entry errors. This can lead to drastic conclusions which can affect negatively the decisions.

2.3.1 Missing Data Mechanisms

There are two important types of missing data known as ignorable and non-ignorable [24]. Ignorable missing data is where the probability that a datapoint will be missing is independent of its value whereas non-ignorable missing data is where the probability that a datapoint will be missing is dependent on its value.
Missing Data Mechanism [25] describes the relationship between the missing data and the values of the variables of the data that is integrated with missing data. Let X be a $n \times p$ data matrix where $X_i = \{X_{i,1}, \cdots, X_{i,p}\}$ is the ith row of X. Let X_{obs} and X_{mis} denote the observed and the missing parts of the complete data $X = \{X_{obs}, X_{mis}\}$, respectively. Let M be the missingness matrix which indicates whether the corresponding location in X is missing (1) or observed (0) such that

$$M_{ij} = \begin{cases} 1 & \text{if } X_{ij} \text{ is missing,} \\ 0 & \text{otherwise.} \end{cases} \tag{2.4}$$

The missing data mechanism is characterized by the probability distribution of M given X [26], $P(M \mid X, \phi)$, where ϕ is a vector of unknown parameters describing the relationship between missingness matrix, M and the complete data, X. Missing data mechanisms can be classified into three kinds - Missing Completely at Random (MCAR), Missing at Random (MAR) and Missing Not at Random (MNAR). Figure 2.5 shows the dataset of house sparrow population that contains information on badge size (Badge) and age (Age) of 10 male sparrows, and on the three missing data mechanisms in the context of the specific data [25].

Missing Completely at Random
Missing Completely at Random is a random process such that there is no relationship between the propensity of a value to be missing and the values of the variables (observed and missing). Mathematically, the probability that a variable value is missing does not depend on the missing data or the observed data and is given by

$$P(M \mid X, \phi) = P(M \mid \phi) \ \forall X, \phi. \tag{2.5}$$

For example, the variable $\text{Age}_{(MCAR)}$ in figure 2.5 is missing completely at random because the missing data on Age is not related to the observed variable, Badge.

Missing at Random
Missing at Random is a predictable process such that there is a relationship between the propensity of a value to be missing and the observed data, but not the missing data. Mathematically, the probability that a variable value is missing depends on the observed data but not on the missing data and is given by

$$P(M \mid X, \phi) = P(M \mid X_{obs}, \phi) \ \forall X_{mis}, \phi. \tag{2.6}$$

For example, the variable $\text{Age}_{(MAR)}$ in figure 2.5 is missing at random because the missing values are associated with the smallest three values of the observed variable, Badge. Thus the probability of a value being missing increases with lower observed badge sizes.

Missing Not at Random
Missing Not at Random is an unpredictable process such that there is a relationship between the propensity of a value to be missing and the missing data. Mathematically, the probability that a variable value is missing depends on the missing data and is given by

$$P(M \mid X, \phi) = P(M \mid X_{obs}, X_{mis}, \phi) \ \forall \phi. \tag{2.7}$$

For example, the variable $\text{Age}_{(MNAR)}$ in figure 2.5 is missing not at random because the three missing values are 4-year old birds and older sparrows tend to have larger badge sizes. Such a scenario is possible if a study on this sparrow population started 3 years ago, and we do not know the exact age of older birds.

Bird	Badge (Complete)	Age (Complete)	Age (MCAR)	Age (MAR)	Age (MNAR)
1	31.5	1	1	_	1
2	33.5	2	_	_	2
3	34.4	3	3	_	3
4	35.1	1	_	1	1
5	35.4	2	2	2	2
6	36.7	4	4	4	_
7	37.8	2	2	2	2
8	38.8	4	4	4	_
9	40.3	3	3	3	3
10	41.5	4	_	4	_

Figure 2.5: An example dataset explaining three missing data mechanisms - MCAR, MAR and MNAR obtained from [25]. The data shows house sparrow population that contains information on badge size 'Badge' and age 'Age' of 10 male sparrows.

The missing data mechanism should be identified since it is important for choosing the approach to deal with missing data. Ignorability is an important concept in missing data mechanism which refers to whether we can ignore the way in which data is missing when we delete or impute missing data. MCAR and MAR are ignorable while MNAR is non-ignorable. In case of MCAR, deletion and in case of MAR, imputation do not require that we make assumptions about how the data is missing. On the other hand, MNAR missingness requires such assumptions to build a model to fill in missing values such as in maximum likelihood estimation method [27]. The different missing data types are illustrated in figure 2.6.

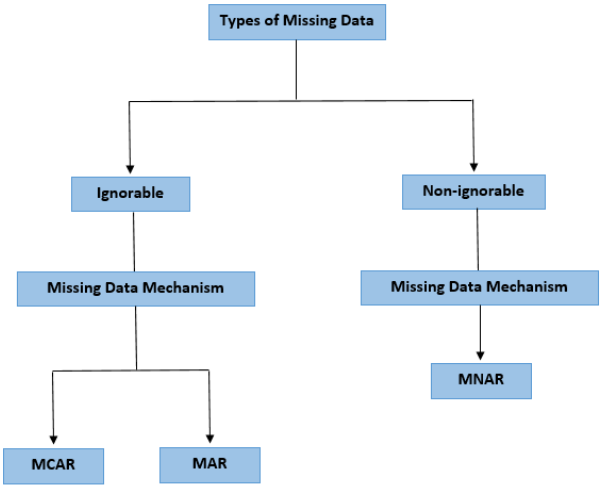


Figure 2.6: Types of missing data and the corresponding missing data mechanisms.

2.3.2 Missing Data Handling Techniques

The following techniques for dealing with missing data are investigated.

Deletion
Deletion method is typically used in case of missing completely at random. Deletion is of two types- listwise and pairwise.

1. Listwise deletion delete rows when any of the observation is missing. For example, the student with id 2 is missing data for science marks and the student with id 4 is missing data for gender as seen in figure 2.7, therefore, the students with id 2 and id 4 will be completely removed from the data because the students do not have complete data for all the variables.

Student ID	Gender	English Marks	Science Marks
1	Female	93	85
2	Male	91	_
3	Male	95	80
4	_	90	83
5	Male	94	87

Listwise deletion

Student ID	Gender	English Marks	Science Marks
1	Female	93	85
3	Male	95	80
5	Male	94	87

Figure 2.7: Listwise deletion of missing data. The students with id 2 and id 4 are completely removed from the data because the students do not have complete data for all the features.

The listwise deletion method has the following advantage:
- It is simple to implement.

The listwise deletion method has the following disadvantage:
- It reduces the power of the model since it reduces the sample size.

2. Pairwise deletion do not delete a row completely rather, it omits rows based on the features included in the analysis. For example, the student with id 2 will be omitted from any analyses using science marks and the student with id 4 will be omitted from any analyses using gender, but they will not be omitted from analyses for which the student has complete data.

Student ID	Gender	English Marks	Science Marks
1	Female	93	85
2	Male	91	_
3	Male	95	80
4	_	90	83
5	Male	94	87

Pairwise deletion

Student ID	Gender	English Marks	Science Marks
1	Female	93	85
2	Male	91	
3	Male	95	80
4		90	83
5	Male	94	87

Figure 2.8: Pairwise deletion of missing data. The student with id 2 is omitted from any analyses using 'Science Marks' and the student with id 4 is omitted from any analyses using 'Gender', but they are not omitted from analyses for which the student has complete data.

The pairwise deletion method has the following advantage:

- It keeps all cases available for analysis thus increasing the statistical power in the analysis.

The pairwise deletion method has the following disadvantage:

- It uses different sample sizes for different variables.

Dropping Features

If a large amount of observations is missing in a feature, then we can delete the feature from the data. It needs to be checked if there is an improvement of the model performance after deletion of feature. This should be the last option. For example, 4 out of 5 observations as seen in figure 2.9 are missing in English marks feature so we need to delete the English marks feature.

Student ID	Gender	English Marks	Science Marks
1	Female	_	85
2	Male	_	84
3	Male	95	80
4	Female	_	83
5	Male	_	87

Drop variable

Student ID	Gender	Science Marks
1	Female	85
2	Male	84
3	Male	80
4	Female	83
5	Male	87

Figure 2.9: Dropping feature of missing data. The 'English Marks' feature is deleted since majority of the observations is missing in 'English Marks' feature.

The dropping features method has the following advantage:

- It is easy to use.

The dropping features method has the following disadvantage:

- The deleted feature is not anymore available for analysis.

Imputation
In an ideal scenario, data is perfect without any missing data. But perfect datasets are rarely found in scientific, engineering, medical and other fields. Methods used for analysis of big data often depend on the whole dataset. Missing data imputation is a solution to the problem. Missing data imputation is a method of replacing the missing values with estimated ones. Imputation method is typically used when the nature of missing data is missing at random. Most of the missing data imputation handling methods are restricted to coping with only one data type either continuous or categorical. Some methods can also handle mixed data types. Most commonly used imputation methods include mean, median, mode and missForest imputation methods.

1. Mean imputation is a method in which the missing value of a certain variable is replaced by the mean of the available values of the variable. If the size of the available values of a variable is n, then the missing value of the variable is replaced by the value

$$\overline{x} = \frac{\sum_{i=1}^{n} x_i}{n}. \tag{2.8}$$

For example, the missing value (third value) of 'English Marks' column as seen in figure 2.10 is replaced by the mean of the remaining values that is 92. Again, the missing values (second and fourth values) of 'Science Marks' column as seen in figure 2.10 are replaced by the mean of the remaining values that is 84.

Student ID	Gender	English Marks	Science Marks
1	Female	93	85
2	Male	91	–
3	Male	–	80
4	Female	90	–
5	Male	94	87

Mean imputation

Student ID	Gender	English Marks	Science Marks
1	Female	93	85
2	Male	91	84
3	Male	92	80
4	Female	90	84
5	Male	94	87

Figure 2.10: Mean imputation of missing data. The missing value (third value) of 'English Marks' feature is replaced by the mean of the observed values that is 92. Again, the missing values (second and fourth values) of 'Science Marks' feature are replaced by the mean of the observed values that is 84.

The mean imputation method has the following advantage:
- It is fast.
- It works well with small numerical data.
- It is generally used when the variable is normally distributed or in particular does not have any skewness.

The mean imputation method has the following disadvantage:
- It reduces the original variance of the data.
- The co-variance with the remaining variables is distorted within the data.

2. Median imputation is a method in which the missing value of a certain variable is replaced by the median of the available values of the variable. If the size of the available values of a variable n is odd, then the missing value of the variable is replaced by the value at position $\frac{n+1}{2}$

$$median(x) = x_{\frac{n+1}{2}}. \tag{2.9}$$

If the size of the available values of a variable n is even, then the missing value of the variable is replaced by the average of values at positions $\frac{n}{2}$ and $\frac{n}{2}+1$

$$median(x) = \frac{x_{\frac{n}{2}} + x_{\frac{n}{2}+1}}{2} \tag{2.10}$$

For example, the missing value (third value) of 'English Marks' column as seen in figure 2.11 is replaced by the median of the remaining values that is 92. Again, the missing values (second and fourth values) of 'Science Marks' column as seen in figure 2.11 are replaced by the median of the remaining values that is 85.

Student ID	Gender	English Marks	Science Marks
1	Female	93	85
2	Male	91	_
3	Male	_	80
4	Female	90	_
5	Male	94	87

Median imputation

Student ID	Gender	English Marks	Science Marks
1	Female	93	85
2	Male	91	85
3	Male	92	80
4	Female	90	85
5	Male	94	87

Figure 2.11: Median imputation of missing data. The missing value (third value) of 'English Marks' feature is replaced by the median of the observed values that is 92. Again, the missing values (second and fourth values) of 'Science Marks' feature are replaced by the median of the observed values that is 85.

The median imputation method has the following advantage:
- It is fast.
- It works well with small numerical data.
- It is used when dealing with skewed data or heteroscedasticity.

The median imputation method has the following disadvantage:
- It reduces the original variance of the data.

3. Mode imputation is a method in which the missing value of a certain variable is replaced by the most frequent value of the variable. For example, the missing value (fourth value) of 'Gender' column as seen in figure 2.12 is replaced by the most frequently occurring value that is 'Male'.

Student ID	Gender	English Marks	Science Marks
1	Female	93	85
2	Male	91	84
3	Male	92	80
4	_	90	84
5	Male	94	87

Mode imputation

Student ID	Gender	English Marks	Science Marks
1	Female	93	85
2	Male	91	84
3	Male	92	80
4	Male	90	84
5	Male	94	87

Figure 2.12: Mode imputation of missing data. The missing value (fourth value) of 'Gender' column is replaced by the most frequently occurring value that is 'Male'.

The mode imputation method has the following advantage:
- It is fast.
- It works well with categorical data.
- It is used when dealing with skewed data or heteroscedasticity.

The mode imputation method has the following disadvantage:
- It reduces the original variance of the data.

4. MissForest Method is a missing data imputation method with random forests [14]. Random forest is one of the best predictive models proposed by Breiman [28]. Random forests is an ensemble learning method that comprises of large number of decision trees and makes predictions over categorical or numerical response variables by outputting the class that is the mode of the predicted classes (classification) or mean prediction (regression) of the individual trees [29]. For training data $D = \{(x_1, y_1), \cdots, (x_n, y_n)\}$ where $x_i = \{x_{i,1}, \cdots, x_{i,p}\}$ denotes the p predictors and y_i denotes the response, the j^{th} fitted tree at a new point x is denoted by $\hat{h}_j(x; D)$. First with bagging, each tree j is fit to a bootstrap sample D_j of size N from the training set D. Second when splitting a node into two descendant nodes, the best split is found over a randomly selected subset of m predictor variables from available p predictors. Prediction at a new point x is given by

$$\hat{f}(x) = \frac{1}{J}\sum_{j=1}^{J}\hat{h}_j(x) \tag{2.11}$$

for regression and

$$\hat{f}(x) = \text{arg max}_y \sum_{j=1}^{J} I(\hat{h}_j(x) = y) \tag{2.12}$$

for classification [30] where $\hat{h}_j(x)$ is the j^{th} prediction at x. The mechanism of random forests is shown in 2.13.

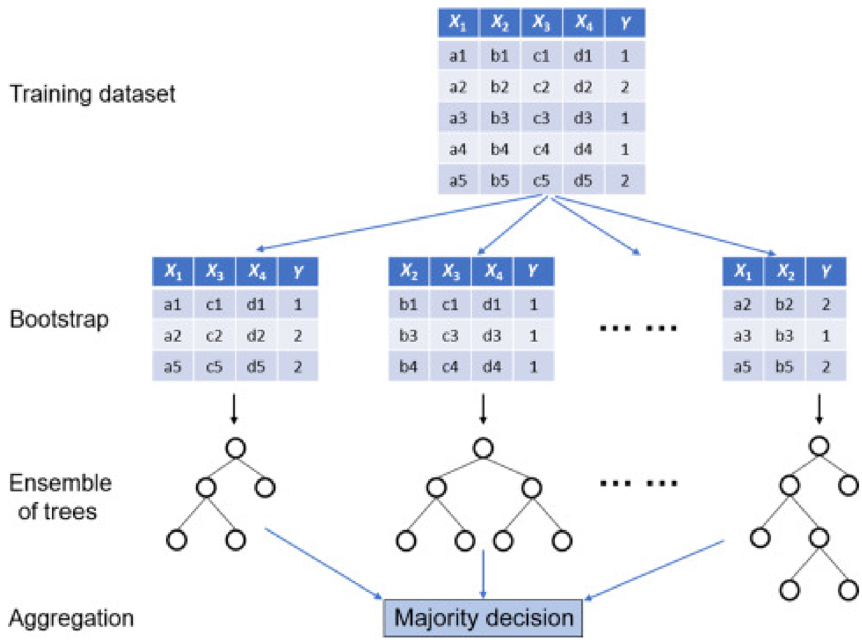


Figure 2.13: Random Forests. From [31]. Adapted with permission.

MissForest method is a non parametric method which can handle any type of input data without any assumptions regarding the distributional aspect of data. It is an iterative imputation approach which trains random forests on observed data, followed by predicting the missing data. Let X=(X_1,X_2,...,X_p) be a $n \times p$ data matrix where n is the number of observations and p is the number of features. Let X_s be an arbitrary variable containing missing values at indices $i^{(s)}_{mis}$. Then the data can be divided into four parts:

1. $y^{(s)}_{obs}$, the observed values of variable X_s.
2. $y^{(s)}_{mis}$, the missing values of variable X_s.
3. $x^{(s)}_{obs}$, the variables other than X_s with observations $\{1,...,n\} \setminus i^{(s)}_{mis}$.
4. $x^{(s)}_{mis}$, the variables other than X_s with observations $i^{(s)}_{mis}$.

MissForest imputes missing values as follows: in the beginning, make an initial guess for the missing values in X using some imputation method. Then, sort the features X_s, $s = 1, \cdots, p$ in ascending order with respect to the amount of missing values. Starting with the feature that has the least missing values, for each variable X_s, the missing values are imputed by first training an RF with response $y^{(s)}_{obs}$ and predictors $x^{(s)}_{obs}$ and then, predicting the missing values $y^{(s)}_{mis}$ by applying the trained RF to $x^{(s)}_{mis}$. The imputation procedure is repeated until a stopping criterion is met. The stopping criterion is fulfilled when the difference between the present imputed data matrix and the previous data matrix increases for the first time with respect to both numerical and categorical variable types. The difference for the set of numerical variables C is defined as

$$\Delta_C = \frac{\sum_{j \in C} (X^{imp}_{new,j} - X^{imp}_{old,j})^2}{\sum_{j \in C} (X^{imp}_{new,j})^2} \tag{2.13}$$

and for the set of categorical variables S as

$$\Delta_S = \frac{\sum_{j \in S} \sum_{i=1}^{n} I_{X^{imp}_{new,j} \neq X^{imp}_{old,j}}}{T_{mis}} \tag{2.14}$$

where X^{imp}_{old} is the previously imputed matrix, X^{imp}_{new} is the new imputed matrix and T_{mis} is the number of missing values in the categorical variables. The missForest algorithm is summarized in Algorithm 1. A flowchart of the MissForest method is shown in figure 2.14.

Algorithm 1: MissForest algorithm

Purpose: Impute missing numerical and categorical data with random forests.
Input: X, and stopping criterion
Output: Imputed matrix X^{imp}
Initialize imputation of missing values using some imputation method;
Sort indices s of columns in X w.r.t increasing amount of missing values;
while *not stopping criterion* **do**
    Store previously imputed matrix in X^{imp}_{old};
    `/* k represents the vector of sorted indices of columns in X w.r.t. increasing amount of missing values. */`
    for *s in k* **do**
        if *column s contains missing values* **then**
            Fit a random forest: $y^{(s)}_{obs} \sim x^{(s)}_{obs}$;
            Predict $y^{(s)}_{mis}$ using $x^{(s)}_{mis}$;
            Update imputed matrix X^{imp}_{new}, using predicted $y^{(s)}_{mis}$;
    Update stopping criterion;
return *The imputed matrix* X^{imp}

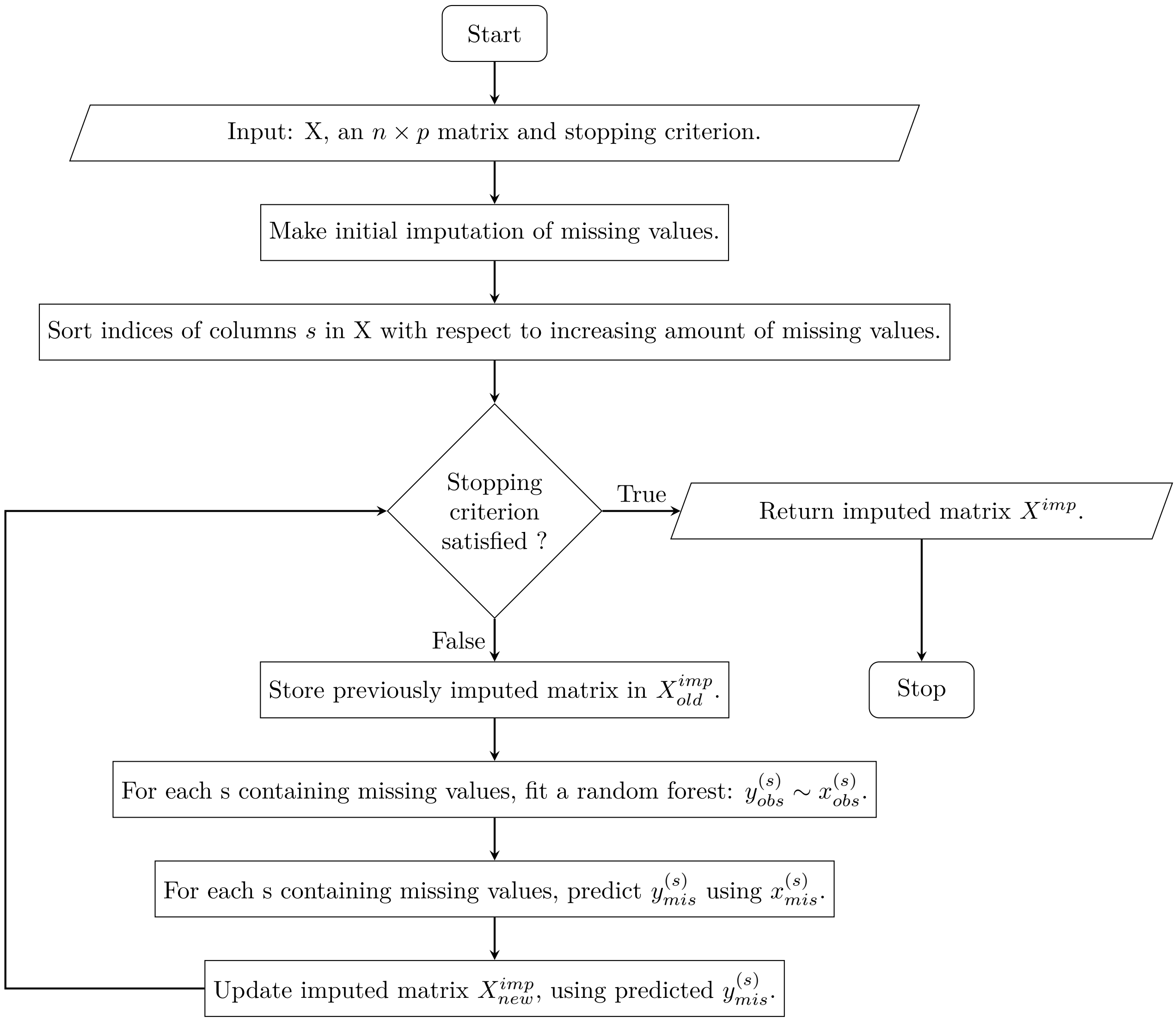


Figure 2.14: A schematic flowchart of the MissForest method.

The performance of missing data imputation is evaluated using the normalized root mean squared error for continuous variables and the percentage of erroneous categorical entries for categorical variables.
Normalized Root Squared Mean Error (NRSME) is an error measure for continuous variables given by the formula

$$NRSME = \sqrt{\frac{mean((X^{true} - X^{imp})^2)}{var(X^{true})}} \tag{2.15}$$

where X^{true} is the true matrix and X^{imp} is the imputed matrix. NRMSE is always non-negative, value near 0 is considered good. Lower values of NRSME means less residual variance and a lower NRMSE is generally considered better than a higher one.
Percentage of erroneous categorical entries (PEC) over the categorical missing values is an error measure for categorical variables given by the formula

$$PEC = \frac{\sum_{j \in S} \sum_{i=1}^{n} I_{X^{true}_{i,j} \neq X^{imp}_{i,j}}}{T} \tag{2.16}$$

where X^{true} is the true matrix, X^{imp} is the imputed matrix and T is the total number of categorical variables.

The missForest imputation method has the following advantages:

- MissForest method allows missing value imputation on any type of data.
- MissForest method do not require tuning of parameters such as standardization of the data or dummy coding of categorical variables.
- MissForest method can be applied to high dimensional datasets.
- MissForest method can handle large amount of missing data.

The missForest imputation method has the following disadvantages:

- It is computationally complex due to the aggregation of large number of decision trees.
- Due to the complexity of the MissForest method, it is more time consuming than other imputation methods like k nearest neighbours. The runtimes of different imputation methods on datasets of different dimensions are compared in figure 2.15.

Dataset	n	P	KNN	MissPALasso	MICE	missForest
Isoprenoid	118	39	0.8	170	–	5.8
Parkinson's	195	22	0.7	120	–	6.1
Musk (cont.)	476	166	13	1400	–	250
Insulin	110	12626	1800	NA	–	6200
SPECT	267	22	1.3	–	37	5.5
Promoter	106	57	14	–	4400	38
Lymphography	148	19	1.1	–	93	7.0
Musk (mixed)	476	167	27	–	2800	500
Gaucher's	40	590	1.3	–	130	29
GFOP	595	18	2.7	–	1400	40
Children	55	124	2.7	–	4000	110

Runtimes are averaged over the amount of missing values since this has a negligible effect on computing time. NA, not available.

Figure 2.15: Comparison of runtimes between different imputation methods. From [14]. Adapted with permission.

2.4 Outlier Detection

2.4.1 Outliers

Datapoints which are significantly different from the rest of the data are called outliers. Outliers can be categorized into following three types.

Global Outlier

Global outlier is a datapoint which is significantly different from the rest of the data. Global outlier is shown in figure 2.16. For example, sales performance scores of employees has a linear dependence on sales target achieved by the respective employees of an organization. Figure 2.16 shows the scatterplot of sales target achieved versus sales performance score. An employee is considered to be a global outlier marked in red color as seen in figure 2.16 since the employee does not follow the general trend of the rest of the data and gets a score of only 40 out of 100 after achieving sales target of more than 80,000 SEK. This is possibly due to the employee's bad attitude in the workplace.

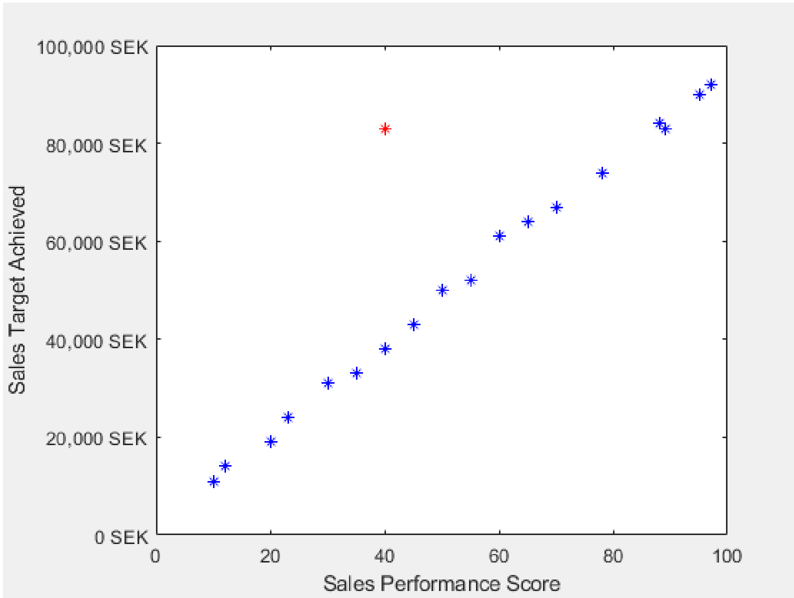


Figure 2.16: Global outlier. This is an example which shows the evaluation of sales performance scores based on sales target achieved of employees of an organization. An employee is a global outlier marked in red color if the employee gets a low score even after achieving a high sales target.

Contextual Outlier.
Contextual Outlier is a datapoint which is significantly different in a specific context. Contextual outlier is shown in figure 2.17. For example, normal systolic blood pressure is 120. During exercise in the morning 08:00-12:00, systolic blood pressure usually increases to 140. But if the sudden increase in systolic blood pressure occurs outside of a high blood pressure period such as exercise session or running, especially during night 20:00-24:00, then it is considered to be a contextual outlier marked in red color as seen in figure 2.17. Here the context is high blood pressure period. This could be due to serious health problems such as heart attack and stroke.

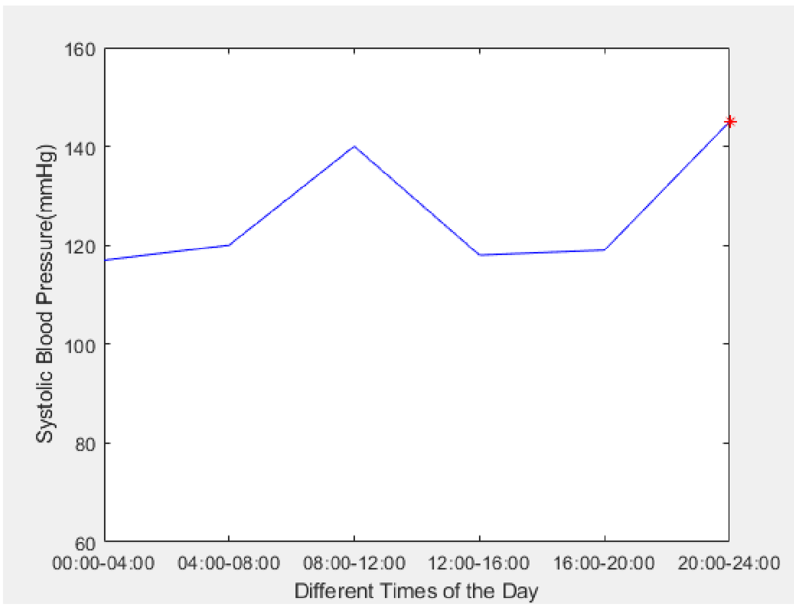


Figure 2.17: Contextual outlier. This is an example of contextual outlier which shows the sudden increase in systolic blood pressure marked in red color arising outside of a high blood pressure period such as exercise session or running.

Collective Outliers

Collective Outliers is a collection of datapoints which is significantly different from the rest of the data. Collective outliers are shown in figure 2.18. For example, a human electrocardiogram output. The red region denotes collective outliers because the low values exist for an abnormally long time corresponding to an Atrial Premature Contraction. The low value itself is not an outlier but its successive occurrence for long time is an outlier.

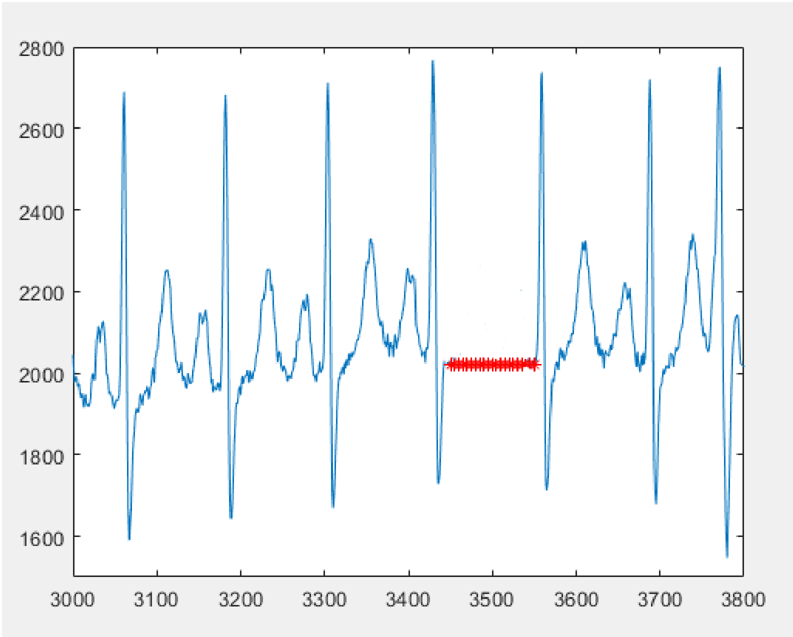


Figure 2.18: Collective outliers.This is an example which shows collective outliers marked in red color in an human electrocardiogram output corresponding to an Atrial Premature Contraction.

2.4.2 Outlier Detection Methods

Based on the extent to which the labels are available in a dataset, outlier detection methods can operate in one of the following three modes [32].

Supervised outlier detection

Supervised Anomaly Detection describes a setup which comprises of both fully labeled training and test datasets and involves training a classifier. This scenario is very similar to traditional supervised classification algorithms except that classes in supervised anomaly detection are highly unbalanced.

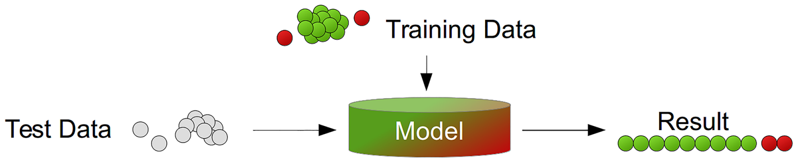
Semi-supervised outlier detection

Semi-supervised anomaly detection constructs a model from outlier-free normal training dataset and then deviations in the test data from the normal model are used to detect outliers.

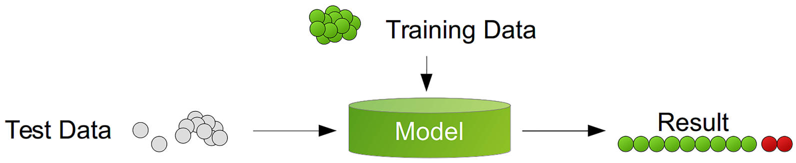
Unsupervised outlier detection

Unsupervised anomaly detection is the most adaptable setup which does not require any labels. The idea is that unsupervised outlier detection methods score the data entirely based on the intrinsic properties of the dataset such as distance and density.

Different outlier detection modes are shown in figure 2.19.



(a) Supervised anomaly detection



(b) Semi-supervised anomaly detection

(c) Unsupervised anomaly detection

Figure 2.19: Different outlier detection modes depending on the availability of labels in a dataset. From [33]. CC-BY.

When given a random raw dataset, we hardly have any information about the data. The assumptions of supervised anomaly detection that the data is normally distributed and outliers are labeled correctly are rarely satisfied. Again, data almost never come in a clean way, which also restricts the use of semi-supervised anomaly detection. Therefore, unsupervised anomaly detection algorithms seem to be the more reasonable choice. The output of an outlier detection algorithm [34] can be of two types:

1. Outlier Scores: Scoring techniques assign an outlier score to each instance in the test data depending on the degree to which that instance is considered an outlier. Thus the output of such techniques is a ranked list of outliers. It allows an analyst to choose a domain specific threshold to select the most relevant anomalies. For example, local outlier factor (LOF) and local distance-based outlier detection approach (LDOF) are scoring techniques.
2. Binary Labels: Labeling techniques assign a binary label (normal or anomalous) to each instance in the test data. It do not directly allow the analysts to make a choice, although this can be controlled indirectly through parameter choices within each technique. For example, z-score and Tukeys Method (box plot) are labeling techniques.

We will discuss the most used outlier detection methods.

Z-score

Z-score can quantify the abnormal behaviour of a datapoint when the data distribution is gaussian. Z-score is a numerical measurement which indicates how far the value of the datapoint is from its mean for a specific feature. Z-score is expressed as

$$Z = \frac{X - \mu}{\sigma} \tag{2.17}$$

where μ is the mean and σ is the standard deviation of feature X. In particular, z-score measures exactly how many standard deviations below or above the population mean a datapoint is. If a datapoint is a certain number of standard deviations away from the mean, then the datapoint is considered an outlier. Default threshold value for finding outliers are z-scores of ± 3 from zero. For the normal distribution as seen from figure 2.20, one standard deviation from the mean (dark blue region) accounts for about 68% of data , two standard deviations from the mean (medium and dark blue region) account for about 95% of data, while three standard deviations (light, medium, and dark blue region) account for about 99.7% of data. Datapoints outside the three standard deviations are identified as outliers. However, z-score can fail to detect outliers if the outliers are extreme because the extreme outliers increase the standard deviation.

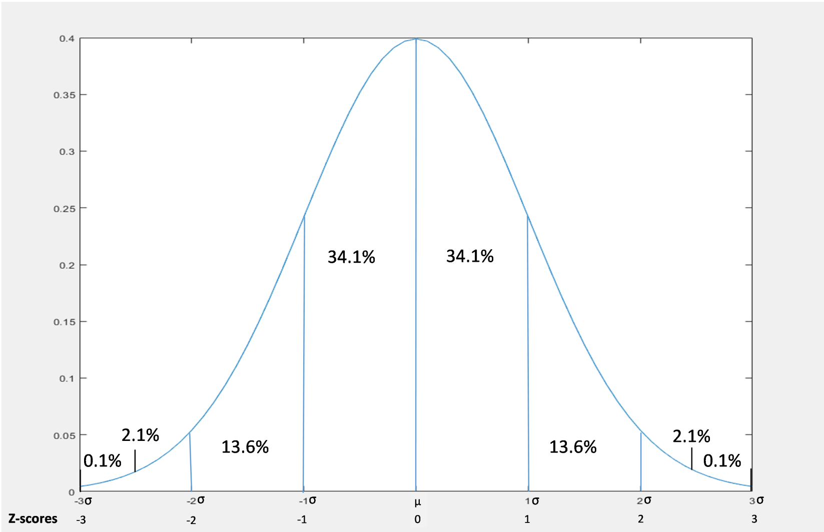


Figure 2.20: Z-score.

The z-score has the following advantages:

- Z-score takes into account both the mean value and the variability in a set of scores.
- Z-score can be used to compare scores that are from different normal distributions.

The z-score has the following disadvantage:

- Z-score always assumes normal data distribution. If this assumption is not met, then the scores cannot be interpreted as a standard proportion of the distribution. Let's say if the data distribution is skewed, then the area within one standard deviation to the left of mean is not equal to the area within one standard deviation to the right of mean.
- It is only suitable to use in a low dimensional feature space, in a small to medium sized dataset.

Leverage

Leverage statistics is an outlier detection method for linear regression model. Leverage statistics is a regression diagnostic on how far the datapoint is from the remaining datapoints.
Let $y = \{y_1, y_2, \cdots, y_n\}$ be a $n \times 1$ vector of dependent variables, $\beta = \{\beta_0, \beta_1\}$ be the 2×1 vector of regression parameters and, $\epsilon = \{\epsilon_1, \epsilon_2, \cdots, \epsilon_n\}$ be the $n \times 1$ vector of errors.

We construct a $n \times 2$ design matrix X as $\begin{pmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_n \end{pmatrix}$. Then the simple linear regression is written as

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i, i = 1, \cdots, n \tag{2.18}$$

$$\Rightarrow Y = X\beta + \epsilon. \tag{2.19}$$

The above formulation can be generalized to multiple linear regression with predictor variables $x_1, \cdots, x_{p-1}$. We construct a $n \times p$ design matrix X as $\begin{pmatrix} 1 & x_{11} & x_{12} & \cdots & x_{1p-1} \\ 1 & x_{21} & x_{22} & \cdots & x_{2p-1} \\ \vdots & \vdots & & & \\ 1 & x_{n1} & x_{n2} & \cdots & x_{np-1} \end{pmatrix}$. Then the multiple linear regression is written as

$$y_i = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_p x_{ip-1} + \epsilon_i, i = 1, \cdots, n \tag{2.20}$$

$$\Rightarrow Y = X\beta + \epsilon. \tag{2.21}$$

We use Least Squares to fit a model to the data $\{x_i, y_i\}_{i=1}^n$ where $x_i = \{x_{i1}, \cdots, x_{ip-1}\}$. We define the cost function or modelling criterion as

$$Q(\beta) = (y - X\beta)'(y - X\beta). \tag{2.22}$$

Our aim is to find the regression parameters by minimizing the criterion.
Taking derivatives with respect to β, and setting these to zero, we get

$$\frac{dQ}{d\beta} = -2X'(y - X\beta) \tag{2.23}$$

$$\Rightarrow (X'X)\beta = X'y \tag{2.24}$$

$$\Rightarrow \hat{\beta} = (X'X)^{-1}X'y. \tag{2.25}$$

The fitted values can be written as

$$\hat{y} = X\hat{\beta}. \tag{2.26}$$

$$\hat{y} = X(X'X)^{-1}X'y. \tag{2.27}$$

The $n \times n$ matrix $X(X'X)^{-1}X'$ is called the Hat matrix. The Hat matrix is usually denoted by H. H is also called the projection matrix since it inputs data y and projects it in a plane spanned by X such that

$$\hat{y} = Hy \tag{2.28}$$

$$\Rightarrow \begin{pmatrix} \hat{y_1} \\ \hat{y_2} \\ \vdots \\ \hat{y_n} \end{pmatrix} = \begin{pmatrix} h_{11} & h_{12} & \cdots & h_{1p-1} \\ h_{21} & h_{22} & \cdots & h_{2p-1} \\ \vdots & \vdots & & \\ h_{n1} & h_{n2} & \cdots & h_{np-1} \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix} \tag{2.29}$$

The amount an observation contributes to its own fitted value, is called the leverage. The leverage values [19] are the diagonal elements of the Hat matrix H defined by

$$h_{ii} = x_i(X'X)^{-1}x_i', i = 1, \cdots, n \tag{2.30}$$

where x_i is the i-th row in X.
Since H is symmetric and idempotent ($H^2 = H$), we get

$$h_{ii} = h_{ii}^2 + \sum_{j \neq i} h_{ij}^2 \tag{2.31}$$

$$\Rightarrow 0 \leq h_{ii} \leq 1. \tag{2.32}$$

Also, we show that eigenvalues of H are either 0 or 1. Let v be an eigenvector of H associated with eigenvalue λ. Then

$$Hv = \lambda v. \tag{2.33}$$

Multiplying the equation by H, we obtain

$$H^2 v = \lambda H v. \tag{2.34}$$

Since $H^2 = H$ and $Hv = \lambda v$,

$$Hv = \lambda^2 v. \tag{2.35}$$

Then,

$$\lambda^2 = \lambda \tag{2.36}$$

$$\Rightarrow \lambda = 0, 1. \tag{2.37}$$

Since eigenvalues of H are either 0 or 1 and the number of non-zero eigenvalues is equal to the rank of the matrix. Then, $rank(H) = rank(X) = p$ and hence $trace(H) = p$. Therefore, average size of hat diagonal $\bar{h}$ is given by

$$\bar{h} = \frac{\sum h_{ii}}{n} = \frac{trace(H)}{n} = \frac{p}{n}. \tag{2.38}$$

Leverage threshold is the threshold where, if a datapoint has a larger leverage, we consider it as an outlier. Leverage threshold is generally considered to be greater than $2\bar{h}$ that is, $h_{ii} > 2\bar{h} = 2\frac{p}{n}$. The threshold is not applicable when $2\frac{p}{n} > 1$.

DBSCAN

Density-based spatial clustering of applications with noise (DBSCAN) is a density based clustering method. Given a dataset, it groups together the points in clusters which are in high density regions whereas the other points are marked as noise.
Let eps represents how close the datapoints should be to each other to be a part of a cluster and minPts denotes the minimum number of datapoints to form a dense region. The larger the dataset, the larger the value of minPts should be chosen. The value for eps is chosen by using a k (= minPts)-Nearest Neighbor graph.
A point is a core point if it has at least minPts points within eps distance. A point is a border point if it has less than minPts points within eps distance but is in the neighborhood of a core point. A point is considered to be outlier if it is neither a core point nor a border point.
A point q is directly density reachable from a point p if the point q is within distance ϵ from core point p. A point q is density reachable from p if there are a set of core points leading from p to q. The DBSCAN algorithm is summarized in Algorithm 2. A flowchart of the DBSCAN method is shown in figure 2.21.

Algorithm 2: DBSCAN algorithm

```
Purpose: Groups together the datapoints in clusters which are in high density regions
  marking the other points as noise.
Input:  D, a dataset, eps, and minPts
Output: Datapoints in clusters.
for each datapoint P belonging to the dataset D do
    Retrieve all datapoints density reachable from P with respect to eps and minPts;
    if P is a core point then
        A cluster is formed;
    if P is a border point then
        No point is density reachable from P;
    if P is neither a core point nor a border point then
        Mark P as noise;
return The clusters
```

The DBSCAN has the following advantages:

- Works well when data distribution is not known.
- Effective if the feature space is multidimensional.
- It detects clusters of complex shapes.
- The number of clusters is not an input parameter.

The DBSCAN has the following disadvantages:

- The data need to be scaled accordingly. Otherwise, choosing a meaningful distance threshold is difficult.
- DBSCAN is sensitive to clustering parameters eps, minPts but selecting such optimal parameters can be difficult.

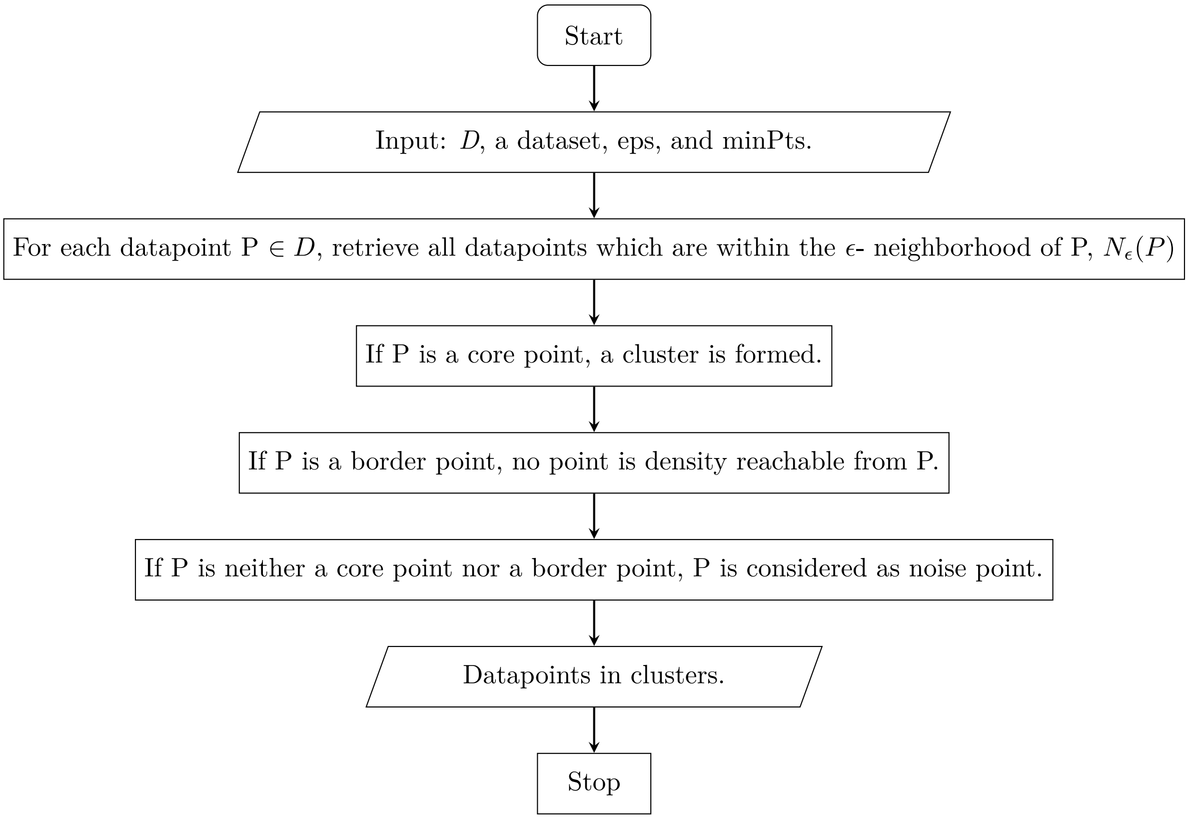


Figure 2.21: A schematic flowchart of the DBSCAN method.

Local Outlier Factor
Local outlier factor (LOF) is a powerful outlier detection method [35]. We also consider the method local outlier factor in our experiments.

2.4.3 Outlier Handling Techniques

The following techniques for dealing with outliers are examined.

Removal of Observations
If there is an outlier or few outliers that may be due to some mistake in the data, then we can treat it as a missing value and impute a new value using some imputation method.

Feature deletion
If there are many outliers in a variable or if we do not need a variable, we can simply delete the variable.

Transformation
Transformation of data is an approach to find true outliers by using a transformed data rather than the data itself. The variation caused by outliers can be reduced by taking the natural logarithm of a value or changing a value into percentile.

2.5 Data Transformation

Data transformation is a method of applying a mathematical function to the data. Transformation is done for the ease of comparison and interpretation.

2.5.1 Standardization

Standardization, also known as z-score is a scaling method which rescales each feature around mean 0 with standard deviation 1. Standardization is defined as

$$Z = \frac{X - \mu}{\sigma} \tag{2.39}$$

where μ is the mean and σ is the standard deviation of each feature X. Standardization is important when the features have different units and the method we use assumes that the data distribution is normal such as regression. The dummy features should not be standardized because after standardization, they are hard to interpret.

2.5.2 Normalization

Normalization is another scaling method which rescales each feature between values 0 and 1. Normalization is defined as

$$Z = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{2.40}$$

where X_{min} and X_{max} are the minimum and maximum of each feature X, respectively. Normalization is important when the features have different scales and the method we use does not assume anything about the data distribution such as k-nearest neighbors and neural networks.

2.5.3 Logarithm Transformation

Logarithmic transformation is a transformation method which replaces each variable by its logarithmic value. Logarithmic transformation is defined as

$$Z = \log(X) \tag{2.41}$$

where X is each variable in the data. Commonly used logarithmic transformations are logarithm base 10, logarithm base 2 and natural logarithm. Logarithmic transformation is useful when transforming highly positive skewed data into a more normalized one.

2.5.4 Exponential Transformation

Exponential transformation is a transformation method which replaces each variable by its exponential value. Exponential transformation is defined as

$$Z = \exp(X) \tag{2.42}$$

where X is each variable in the data. Exponential transformation is useful when transforming skewed distributions into symmetric normal-like distributions.

2.5.5 Square root Transformation

Square root transformation is a transformation method which replaces each variable by its square root value. Square root transformation is defined as

$$Z = \sqrt{X} \tag{2.43}$$

where X is each variable in the data. Square root transformation is useful when transforming nonnegative skewed data into a more normalized one.

2.5.6 Inverse Transformation

Inverse transformation is a transformation method which replaces each variable by its inverse value. Inverse transformation is defined as

$$Z = X^{-1} \tag{2.44}$$

where X is each variable in the data. Inverse transformation is needed when transforming extremely skewed data into less skewed data.

2.6 Data Visualization techniques

Data visualization is the graphical representation of data. Some of the most common data visualization methods or techniques are as follows.

2.6.1 Histogram

Histogram is one of the most common graphical representation of the distribution of numerical or quantitative data. Histogram is used to visualize outliers because outliers are datapoints which lie outside the overall pattern of distribution. Histogram is shown in figure 2.22.

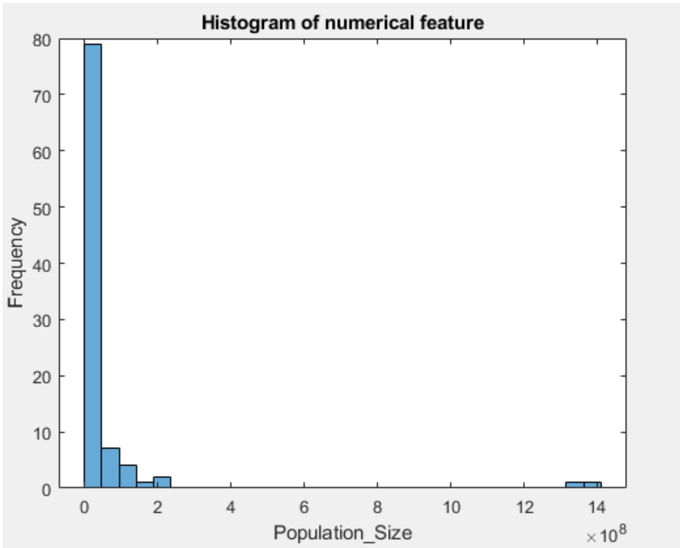


Figure 2.22: Histogram.

2.6.2 Bar Chart

Bar chart is a graphical display of categorical or qualitative data using rectangular bars with heights proportional to the values that they represent. Bar chart is used to visualize outliers because outliers are datapoints which are distant from most of the other data. Bar Chart is shown in figure 2.23.

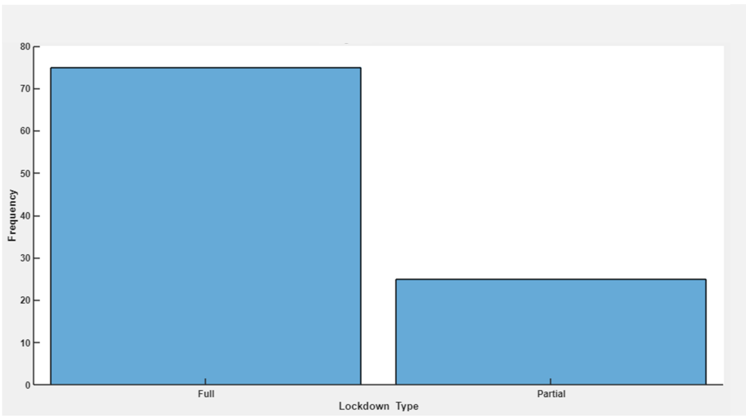


Figure 2.23: Bar Chart.

2.6.3 Box Plot

A box plot is a visual representation of the distribution of numerical data through quartiles. It displays the data distribution based on a five point summary (minimum, first quartile, median, third quartile, and maximum).

- Median (Q2/50th Percentile): The midpoint of the data.
- First quartile (Q1/25th Percentile): The datapoint below which the lower 25% of the data are contained.
- Third quartile (Q3/75th Percentile): The datapoint above which the upper 25% of the data are contained.
- InterQuartile Range(IQR = Q3-Q1): The range of datapoints between the lower (Q1) and upper (Q3) quartiles.
- Maximum (Q3+1.5*IQR): The largest datapoint excluding outliers.
- Minimum (Q1–1.5*IQR): The smallest datapoint excluding outliers.

The whisker corresponds to approximately $\pm$ 2.7 standard deviation and 99.3 percent coverage if the data is normally distributed.

Box plot can handle extremely large datasets easily. Box plot is used to visualize outliers which are marked as individual points distant from the other datapoint. If a datapoint is below minimum or above maximum, then it is identified as an outlier. The red points in figure 2.24 are marked as outliers.

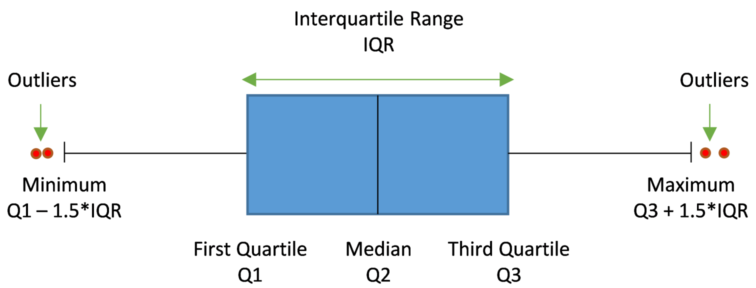


Figure 2.24: Boxplot.

Box plot can show the skewness of a dataset which is seen in figure 2.25. Box plot is used to show if a dataset is symmetrically distributed or skewed. The distribution is symmetric when the median is in the middle of the box and the whiskers are about the same on both sides of the box. The distribution is positively skewed or right skewed when the median is closer to the bottom of the box and the whisker is shorter on the lower end of the box. The distribution is negatively skewed or left skewed when the median is closer to the top of the box and the whisker is shorter on the upper end of the box.

The whisker lengths are different in skewed distributions because the distance 1.5*IQR is used in determining the threshold so as to decide if a point is an outlier or not, but then a line is drawn to the point that is closest to being an outlier, but is within distance 1.5*IQR.

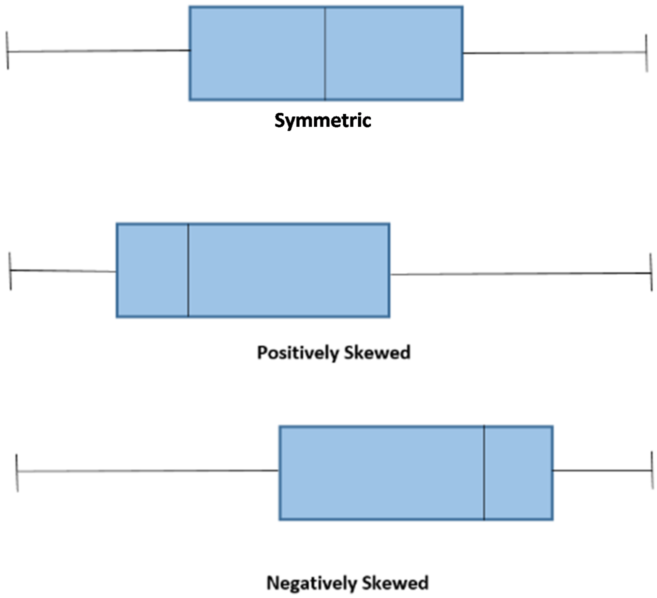


Figure 2.25: Box plot showing the skewness of a dataset.

2.6.4 Missingness Map

Missingness map is a plot showing where missingness occurs in the data. Missingness map is shown in figure 2.26.

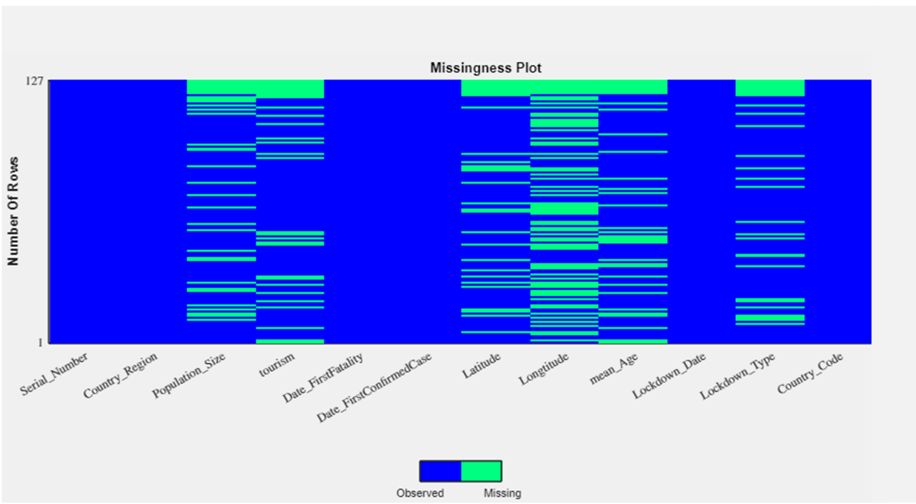


Figure 2.26: Missingness Map.

2.6.5 Line Graph

Line graph is used to plot the missing observations percentages of the variables against the variables. Line graph is shown in figure 2.27.

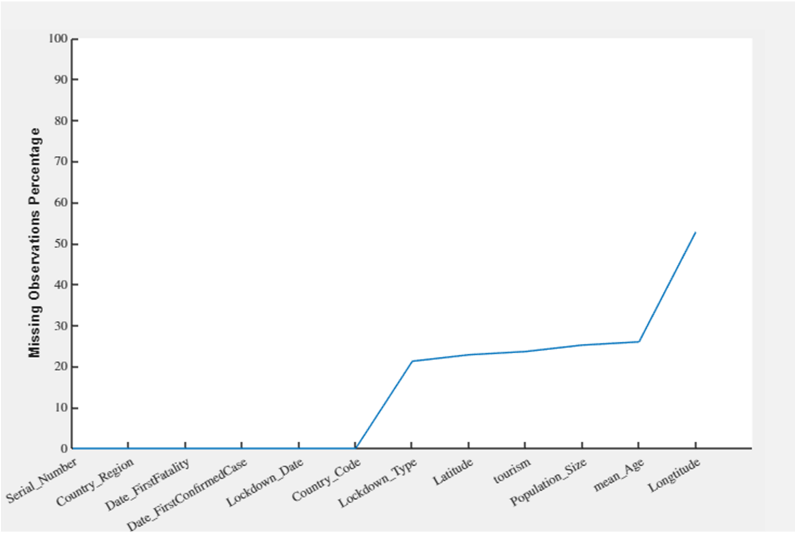


Figure 2.27: Line Graph.

Interactive data visualization is a branch of graphic visualization in the field of computer science and programming that provides users with the ability to control different aspects of visual representation of data. Data visualization is considered to be interactive if there is an aspect of human input such as clicking on a button or moving a slider. Interactive data visualizations are becoming increasingly popular in business intelligence and data analytics because of its ease of use and added value.

3

Methods

In this thesis, we developed a data cleaning application which can recommend data cleaning approaches according to the specific characteristics of the given dataset. DataCleaningTool is a user friendly open source data cleaning standalone application. DataCleaningTool is shown in figure 3.1. A few key ideas guided the construction process of the tool.

1. It identifies and solves reasonable number of data problems.
2. It should be easy and intuitive to use.
3. It should display all the information in a clear and concise manner.
4. It is code free.
5. It provides assistance to users at every stage of data cleaning.

The major data problems encountered by DataCleaningTool are as follows.

- Truncation errors such as numbers truncated to certain decimal places.
- Incorrect data type such as numerical instead of id entries.
- Structural errors such as typographical errors.
- Duplicate data such as duplicate rows and columns.
- Nonsensical data such as absurd or unusual values.
- Extrapolation errors such as extrapolating a trend back in time.
- Missing observations such as missing numerical or datetime or text values.
- Outliers such as observations that fall outside the overall pattern of a distribution.

In this chapter, we present the methodologies for designing the tool. Sections 3.1-3.9 demonstrate the various widgets and their respective powerful code free data cleaning mechanisms.

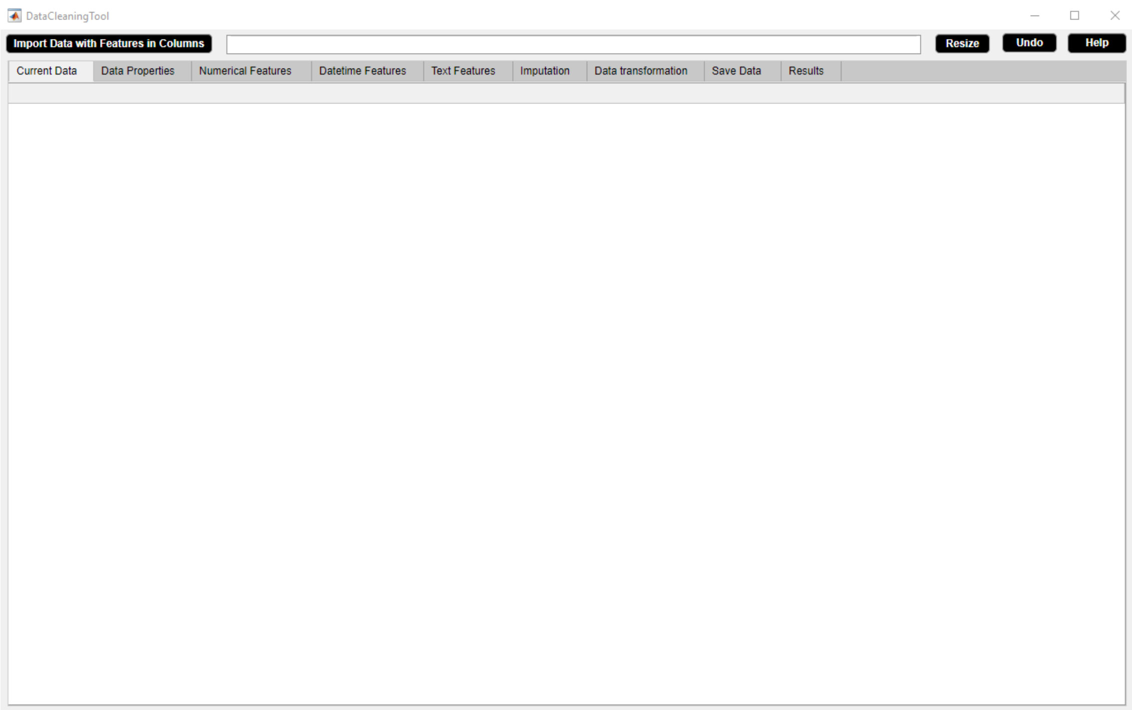
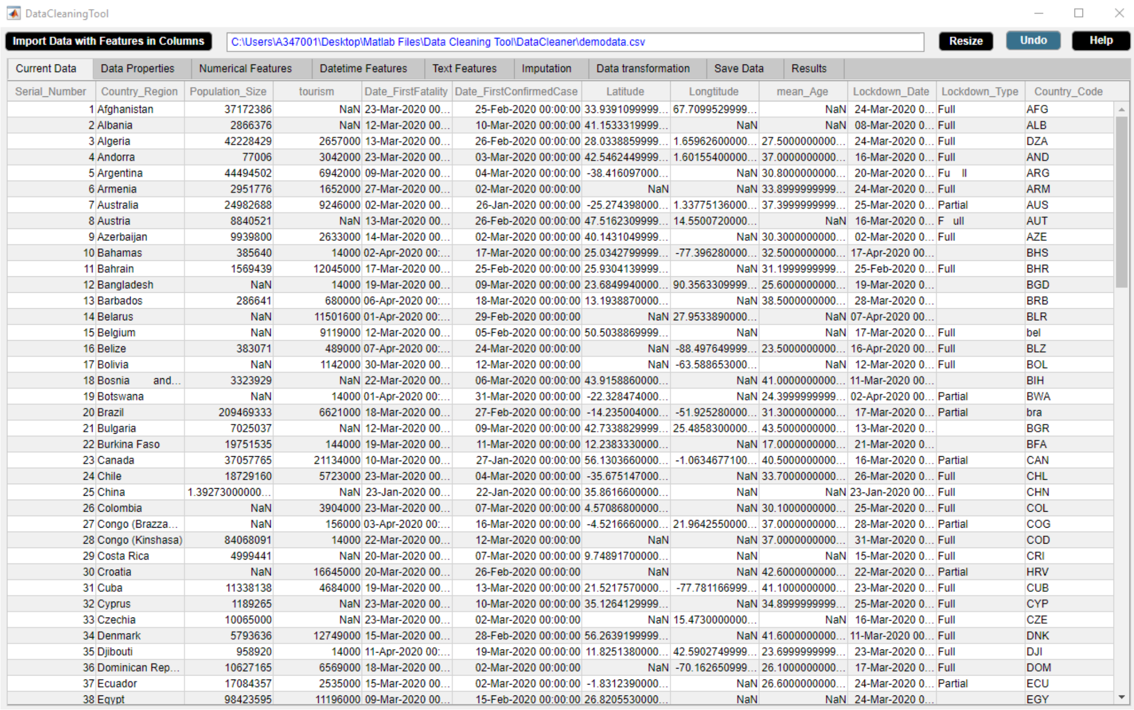


Figure 3.1: DataCleaningTool.

3.1 Current Data

The Current Data widget displays the input data in table format. The Current Data widget is shown in figure 3.2. The properties of the Current Data widget are as follows.

- The widget shows the presence of round off errors in numerical features.
- The widget shows the presence of inconsistent capitalization of feature names and features.
- The widget shows the existence of extra whitespaces in text features.
- Default datetime format is ‘dd-MMM-yyyy HH:mm:ss‘ for datetime features.
- The widget shows the presence of missing numerical observations represented by NaNs.
- The widget shows the presence of missing datetime observations represented by NaTs.
- The widget shows the presence of missing text observations represented by empty strings.
- The updated table can be visualized after each activity since the widget gets updated accordingly.



Serial_Number	Country_Region	Population_Size	tourism	Date_FirstFatality	Date_FirstConfirmedCase	Latitude	Longtitude	mean_Age	Lockdown_Date	Lockdown_Type	Country_Code
1	Afghanistan	37172386	NaN	23-Mar-2020 00...	25-Feb-2020 00:00:00	33.9391099999...	67.7099529999...	NaN	24-Mar-2020 0...	Full	AFG
2	Albania	2866376	NaN	12-Mar-2020 00...	10-Mar-2020 00:00:00	41.1533319999...	NaN	NaN	08-Mar-2020 0...	Full	ALB
3	Algeria	42228429	2657000	13-Mar-2020 00...	26-Feb-2020 00:00:00	28.0338859999...	1.65962600000...	27.5000000000...	24-Mar-2020 0...	Full	DZA
4	Andorra	77006	3042000	23-Mar-2020 00...	03-Mar-2020 00:00:00	42.5462449999...	1.60155400000...	37.0000000000...	16-Mar-2020 0...	Full	AND
5	Argentina	44494502	6942000	09-Mar-2020 00...	04-Mar-2020 00:00:00	-38.416097000...	NaN	30.8000000000...	20-Mar-2020 0...	Fu ll	ARG
6	Armenia	2951776	1652000	27-Mar-2020 00...	02-Mar-2020 00:00:00	NaN	NaN	33.8999999999...	24-Mar-2020 0...	Full	ARM
7	Australia	24982688	9246000	02-Mar-2020 00...	26-Jan-2020 00:00:00	-25.274398000...	1.33775136000...	37.3999999999...	25-Mar-2020 0...	Partial	AUS
8	Austria	8840521	NaN	13-Mar-2020 00...	26-Feb-2020 00:00:00	47.5162309999...	14.5500720000...	NaN	16-Mar-2020 0...	F ull	AUT
9	Azerbaijan	9939800	2633000	14-Mar-2020 00...	02-Mar-2020 00:00:00	40.1431049999...	NaN	30.3000000000...	02-Mar-2020 0...	Full	AZE
10	Bahamas	385640	14000	02-Apr-2020 00:...	17-Mar-2020 00:00:00	25.0342799999...	-77.396280000...	32.5000000000...	17-Apr-2020 00...		BHS
11	Bahrain	1569439	12045000	17-Mar-2020 00...	25-Feb-2020 00:00:00	25.9304139999...	NaN	31.1999999999...	25-Feb-2020 0...	Full	BHR
12	Bangladesh	NaN	14000	19-Mar-2020 00...	09-Mar-2020 00:00:00	23.6849940000...	90.3563309999...	25.6000000000...	19-Mar-2020 0...		BGD
13	Barbados	286641	680000	06-Apr-2020 00:...	18-Mar-2020 00:00:00	13.1938870000...	NaN	38.5000000000...	28-Mar-2020 0...		BRB
14	Belarus	NaN	11501600	01-Apr-2020 00:...	29-Feb-2020 00:00:00	NaN	27.9533890000...	NaN	07-Apr-2020 00...		BLR
15	Belgium	NaN	9119000	12-Mar-2020 00...	05-Feb-2020 00:00:00	50.5038869999...	NaN	NaN	17-Mar-2020 0...	Full	bel
16	Belize	383071	489000	07-Apr-2020 00:...	24-Mar-2020 00:00:00	NaN	-88.497649999...	23.5000000000...	16-Apr-2020 00...	Full	BLZ
17	Bolivia	NaN	1142000	30-Mar-2020 00...	12-Mar-2020 00:00:00	NaN	-63.588653000...	NaN	12-Mar-2020 0...	Full	BOL
18	Bosnia and...	3323929	NaN	22-Mar-2020 00...	06-Mar-2020 00:00:00	43.9158860000...	NaN	41.0000000000...	11-Mar-2020 00...		BIH
19	Botswana	NaN	14000	01-Apr-2020 00:...	31-Mar-2020 00:00:00	-22.328474000...	NaN	24.3999999999...	02-Apr-2020 00...	Partial	BWA
20	Brazil	209469333	6621000	18-Mar-2020 00...	27-Feb-2020 00:00:00	-14.235004000...	-51.925280000...	31.3000000000...	17-Mar-2020 0...	Partial	bra
21	Bulgaria	7025037	NaN	12-Mar-2020 00...	09-Mar-2020 00:00:00	42.7338829999...	25.4858300000...	43.5000000000...	13-Mar-2020 0...		BGR
22	Burkina Faso	19751535	144000	19-Mar-2020 00...	11-Mar-2020 00:00:00	12.2383330000...	NaN	17.0000000000...	21-Mar-2020 0...		BFA
23	Canada	37057765	21134000	10-Mar-2020 00...	27-Jan-2020 00:00:00	56.1303660000...	-1.0634677100...	40.5000000000...	16-Mar-2020 0...	Partial	CAN
24	Chile	18729160	5723000	23-Mar-2020 00...	04-Mar-2020 00:00:00	-35.675147000...	NaN	33.7000000000...	26-Mar-2020 0...	Full	CHL
25	China	1.39273000000...	NaN	23-Jan-2020 00...	22-Jan-2020 00:00:00	35.8616600000...	NaN	NaN	23-Jan-2020 00...	Full	CHN
26	Colombia	NaN	3904000	23-Mar-2020 00...	07-Mar-2020 00:00:00	4.57086800000...	NaN	30.1000000000...	25-Mar-2020 0...	Full	COL
27	Congo (Brazza...	NaN	156000	03-Apr-2020 00:...	16-Mar-2020 00:00:00	-4.5216660000...	21.9642550000...	37.0000000000...	28-Mar-2020 0...	Partial	COG
28	Congo (Kinshasa)	84068091	14000	22-Mar-2020 00...	12-Mar-2020 00:00:00	NaN	NaN	37.0000000000...	31-Mar-2020 0...	Full	COD
29	Costa Rica	4999441	NaN	20-Mar-2020 00...	07-Mar-2020 00:00:00	9.74891700000...	NaN	NaN	15-Mar-2020 0...	Full	CRI
30	Croatia	NaN	16645000	20-Mar-2020 00...	26-Feb-2020 00:00:00	NaN	NaN	42.6000000000...	22-Mar-2020 0...	Partial	HRV
31	Cuba	11338138	4684000	19-Mar-2020 00...	13-Mar-2020 00:00:00	21.5217570000...	-77.781166999...	41.1000000000...	23-Mar-2020 0...	Full	CUB
32	Cyprus	1189265	NaN	23-Mar-2020 00...	10-Mar-2020 00:00:00	35.1264129999...	NaN	34.8999999999...	25-Mar-2020 0...	Full	CYP
33	Czechia	10065000	NaN	23-Mar-2020 00...	02-Mar-2020 00:00:00	NaN	15.4730000000...	NaN	16-Mar-2020 0...	Full	CZE
34	Denmark	5793636	12749000	15-Mar-2020 00...	28-Feb-2020 00:00:00	56.2639199999...	NaN	41.6000000000...	11-Mar-2020 00...	Full	DNK
35	Djibouti	958920	14000	11-Apr-2020 00:...	19-Mar-2020 00:00:00	11.8251380000...	42.5902749999...	23.6999999999...	23-Mar-2020 0...	Full	DJI
36	Dominican Rep...	10627165	6569000	18-Mar-2020 00...	02-Mar-2020 00:00:00	NaN	-70.162650999...	26.1000000000...	17-Mar-2020 0...	Full	DOM
37	Ecuador	17084357	2535000	15-Mar-2020 00...	02-Mar-2020 00:00:00	-1.8312390000...	NaN	26.6000000000...	24-Mar-2020 0...	Partial	ECU
38	Egypt	98423595	11196000	09-Mar-2020 00...	15-Feb-2020 00:00:00	26.8205530000...	NaN	NaN	24-Mar-2020 0...		EGY

Figure 3.2: Current Data Widget.

3.2 Data Properties

The Data Properties widget displays several statistical aspects of the data. The Data Properties widget is shown in figure 3.3. The properties of the Data Properties widget are as follows.

- The widget automatically discovers the datatypes of features of the input dataset and shows the numerical features, the datetime features and the text features separately.
- The widget summarizes the characteristics of a dataset such as file size in megabytes, number of rows and columns, number of id, numerical, datetime and text features, number of duplicate rows and columns, and number of deleted rows and columns.
- The widget shows the percentage of missing observations in the dataset and the percentage of missing observations in each feature. The widget presents two visual methods for missing data - the missingness plot and the missing observations percentage plot. The missingness plot indicates the missing value occurrence in the data. The missing observations percentage plot indicates the percentage of missing observations in each feature. This study of missing data helps to determine the missing data mechanism and hence choose strategies like listwise deletion, pairwise deletion, dropping features, imputation which can be applied to handle missing data so that they can be used for analysis and modelling.
- The Id button is used to separate id features from numerical or datetime or text features where an id feature represents a unique identifier field in the data. This avoids the problem of overfitting during data analysis which occurs due to a unique identifier among features.
- The Feature Names button is used to change letter case of all feature names to one of the cases- lower case or upper case or capitalized case. This fixes structural errors such as unifying inconsistent capitalization of feature names.
- The Change Case button is used to change letter case of all features to one of the cases- lower case or upper case or capitalized case. This fixes structural errors such as unifying inconsistent capitalization of features.
- The Remove Extra Space button is used to remove either all spaces or to only one whitespace in a string of a feature. This fixes structural errors such as typographical errors.
- The Delete Rows button is used to delete rows that are specified by the user. For example, listwise deletion of rows containing a large number of missing observations.
- The information in the widget gets updated after each activity.

Figure 3.3: Data Properties Widget.

3.3 Numerical Features

The Numerical Features widget displays statistical description of the numerical data. The Numerical Features widget is shown in figure 3.4. The properties of the Numerical Features widget are as follows.

- The widget shows the descriptive statistics of each numerical feature of the data such as minimum observation and maximum observation of the feature. Descriptive statistics of a feature gives a quantitative description of a feature.
- The widget shows the duplicate observations present in each numerical feature and the missing observations percentage of each numerical feature. Duplicate observation can be an error in the data and could possibly influence later analyses of the data.
- Cross-field validation constraint and range constraint can be set in the widget. This results in removal of unwanted numerical observations.
- The Remove Observations button replaces unwanted numerical observations by missing values.
- The Delete Rows button deletes rows with unwanted numerical observations.
- Histogram of the selected numerical feature can be visualized in the widget. This is an outlier visualization technique.
- The statistical information of the numerical data in the widget gets updated after each activity.

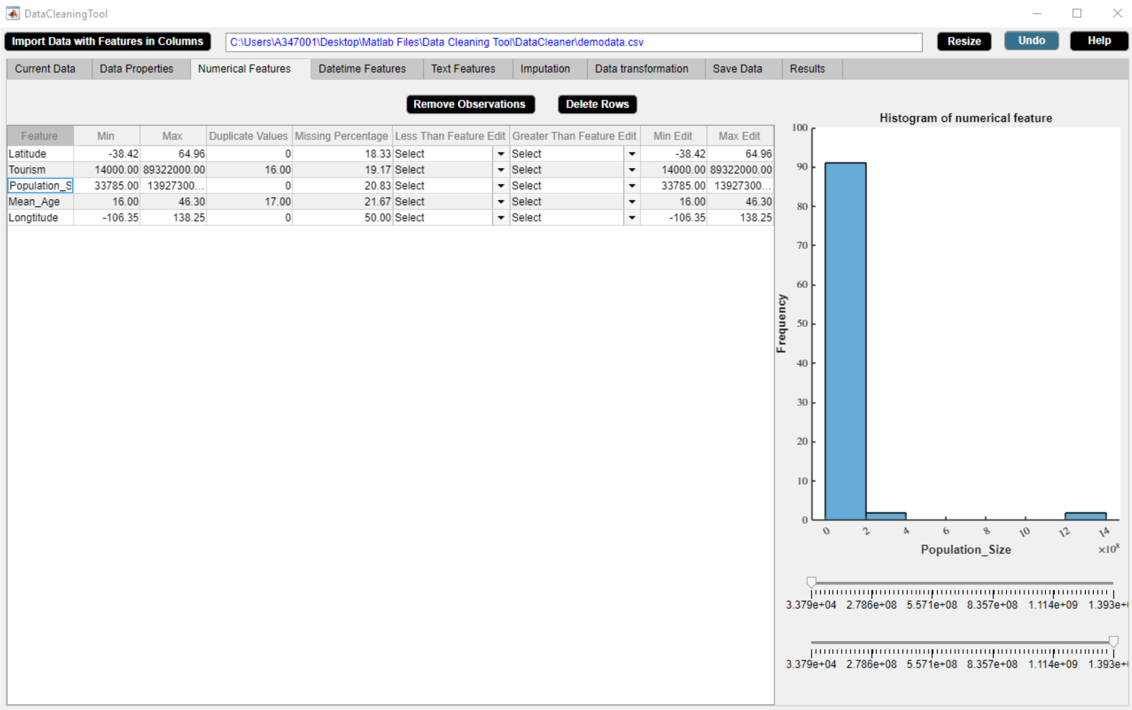


Figure 3.4: Numerical Features Widget.

3.4 Datetime Features

The Datetime Features widget displays statistical description of the datetime data. The Datetime Features widget is shown in figure 3.5. The properties of the Datetime Features widget are as follows.

- The widget shows the descriptive statistics of each datetime feature of the data such as minimum observation and maximum observation of the feature.
- The widget also shows the missing observations percentage of each datetime feature.
- The Convert To Excel DATEVALUE button converts datetime to Excel serial date number.
- Datetime format can be changed.
- Constraint and Range can be reset in the widget for each datetime feature. This will result in some unwanted datetime observations.
- The Remove Observations button replaces unwanted datetime observations by missing values.
- The Delete Rows button deletes rows with unwanted datetime observations.
- Histogram of the selected datetime feature can be visualized in the widget. This is an outlier visualization technique.
- The statistical information of the datetime data in the widget gets updated after each activity.

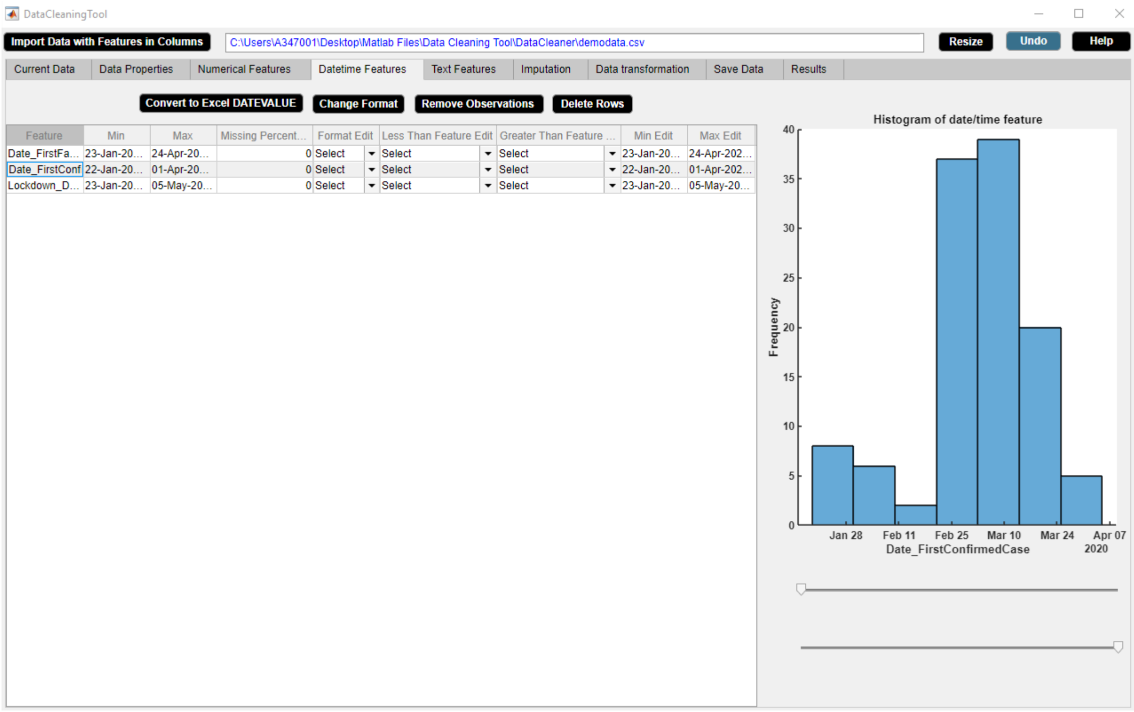


Figure 3.5: Datetime Features Widget.

3.5 Text Features

The Text Features widget displays statistical description of the text data. The Text Features widget is shown in figure 3.6. The properties of the Text Features widget are as follows.

- The widget shows the descriptive statistics of each text feature of the data such as categories and categories count of the feature.
- The widget also shows the missing observations percentage of each text feature.
- The Select Similar Categories button replaces categories with similar ones.
- The Label Encoding button assigns each category of a categorical feature a value from 0 to $n-1$ where n is the number of categories. Note that label encoding is an encoding approach usually for handling ordinal categorical features.
- The One-Hot Encoding Button button transforms n categories to either n or $n-1$ dummy variables for a categorical feature. Note that one-hot encoding is an encoding approach usually for handling nominal categorical features.
- The Remove Observations button replaces outliers by missing values.
- The Delete Rows button deletes rows with outliers.
- Histogram of the selected text feature can be visualized in the widget. This is an outlier visualization technique.
- Boxplot of the selected numerical feature versus the text feature can be visualized in the widget. This is another outlier visualization technique.
- The statistical information of the text data in the widget gets updated after each activity.

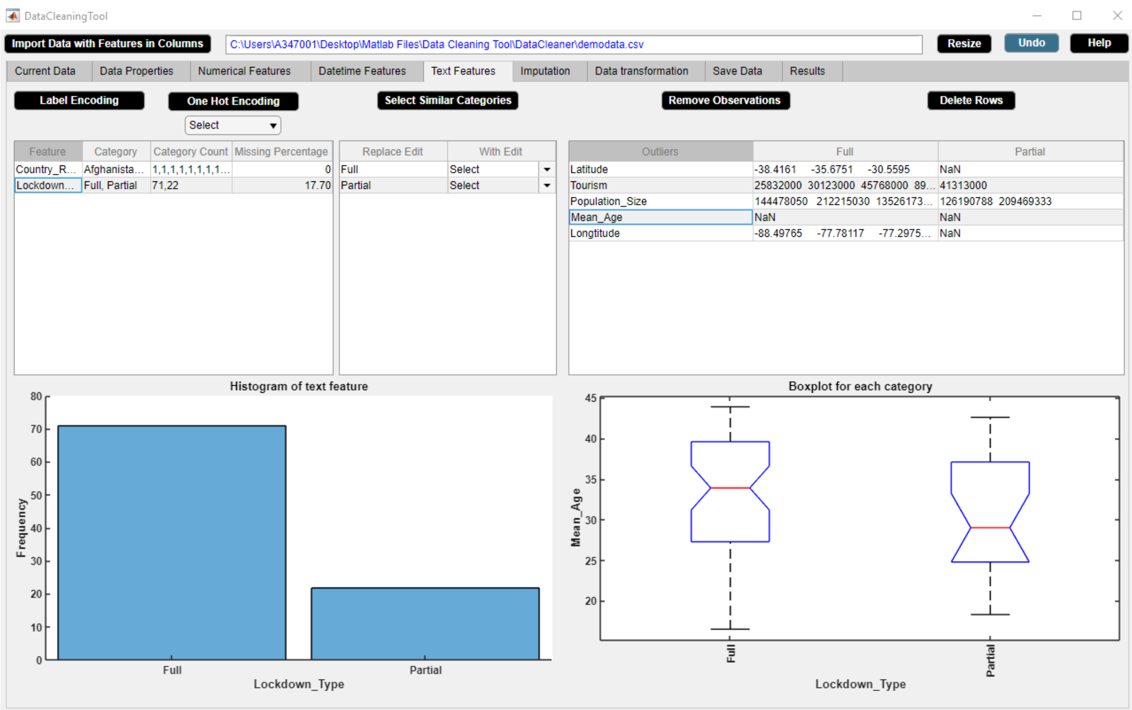


Figure 3.6: Text Features Widget.

3.6 Imputation

The Imputation widget displays information about the missing data and the expected error of imputation for numerical and categorical features. The Imputation widget is shown in figure 3.7. The properties of the Imputation widget are as follows.

- The widgets shows information about missing data such as percentage of missing data, expected error of imputation for numerical and categorical features. The performance analysis results of the missForest method discussed in chapter 4 is used to predict the expected error of imputation for numerical and categorical features for the specific ratio of data and percentage of missing data.
- The widget also presents the missing observations percentage table and the missingness plot.
- The Delete Feature button is used to delete a feature from data. This drops a feature which contains a large number of missing values.
- The Impute button is used to replace missing observations by estimated ones using missForest algorithm.
- If datetime observations are missing, a message stating that datetime imputation is not possible appears in red color in the lower side of the Imputation widget.
- The information of the missing data in the widget gets updated after each activity.

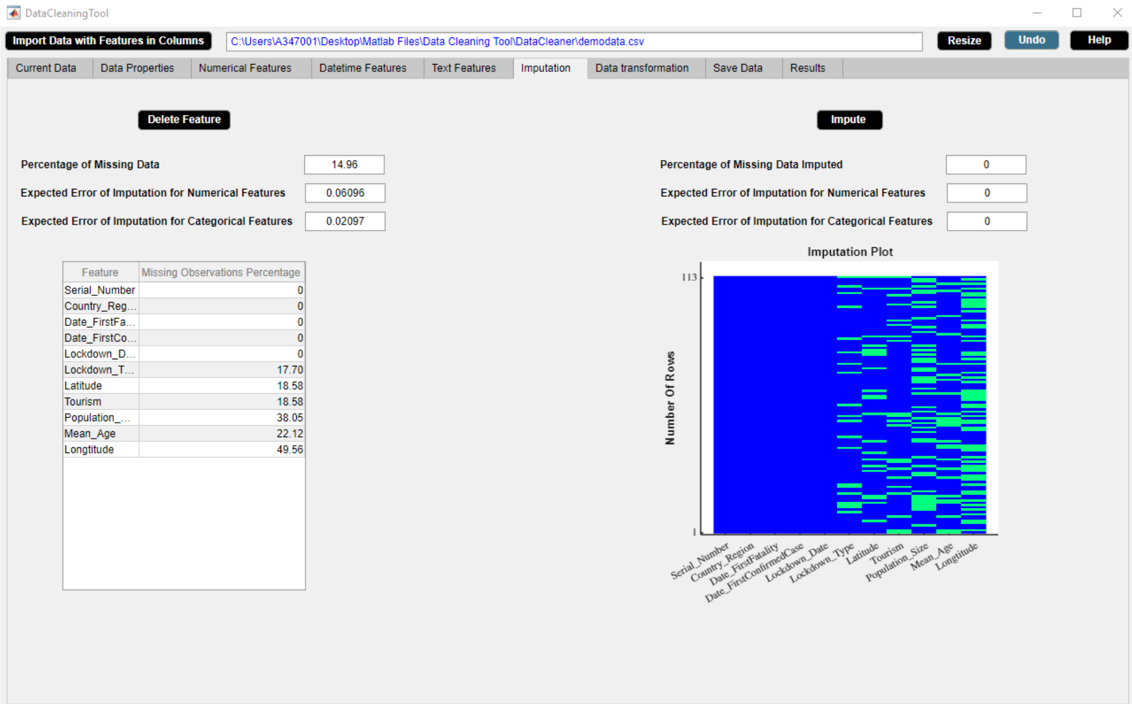


Figure 3.7: Imputation Widget.

3.7 Data Transformation

The Data Transformation widget displays the numerical features of the data on which data transformation can only be applied. The Data Transformation widget is shown in figure 3.8. The properties of the Data Transformation widget are as follows.

- The widget presents the numerical features of the data.
- The Transform button is used to standardize or normalize or logarithm or exponential or squareroot or inverse transform the selected numerical features. Here ’mean 0 and standard deviation’ represents standardize, ’between 0 and 1’ represents normalize, ’ln’ represents natural logarithm transform, ’log10’ represents logarithm base 10 transform, ’log2’ represents logarithm base 2 transform, ’exp’ represents natural exponential transform, ’sqrt’ represents squareroot transform and ’reciprocal’ represents inverse transform.
- Histogram of the transformed numerical feature can be visualized in the widget. This is an outlier visualization technique.
- A message regarding the percentage increase in missing data due to data transformation appears in red color in the lower side of the Data Transformation widget.
- The numerical features of the data in the widget gets updated after each activity.

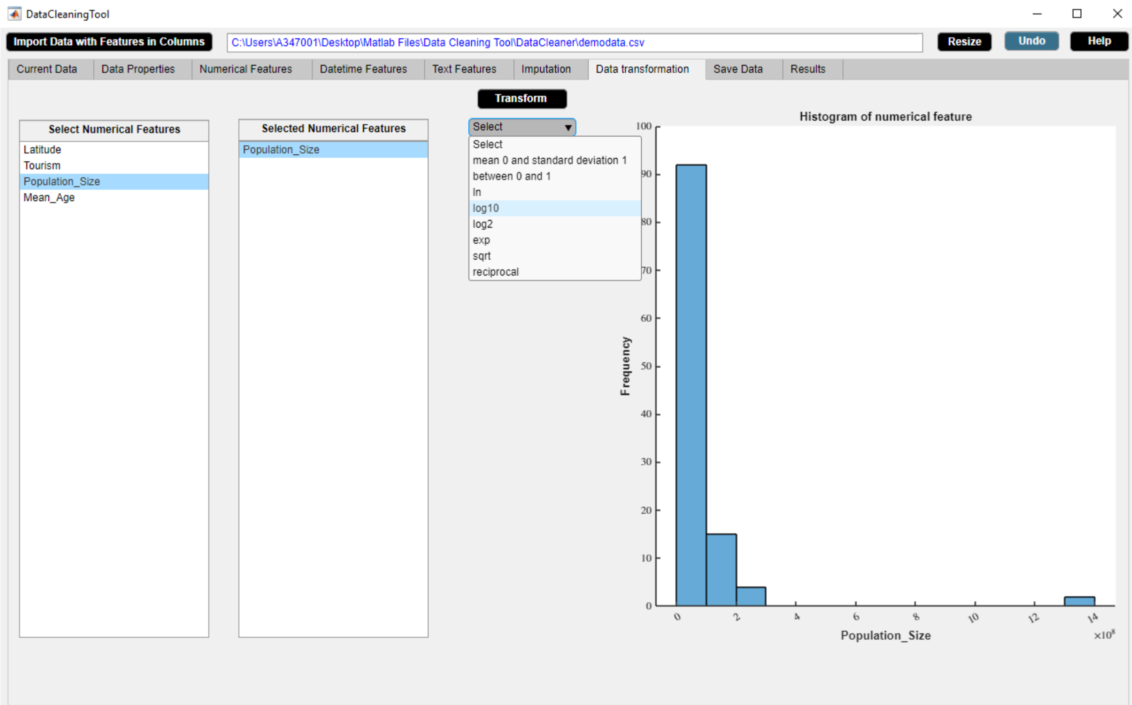


Figure 3.8: Data Transformation Widget.

3.8 Save Data

The Save Data widget displays the full paths of the saved files. The Save Data widget is shown in figure 3.9. The properties of the Save Data widget are as follows.

- The widget saves data in csv or xlsx format after data cleaning.
- Data can be saved for multiple times after each activity.
- The full paths of the saved files are displayed.

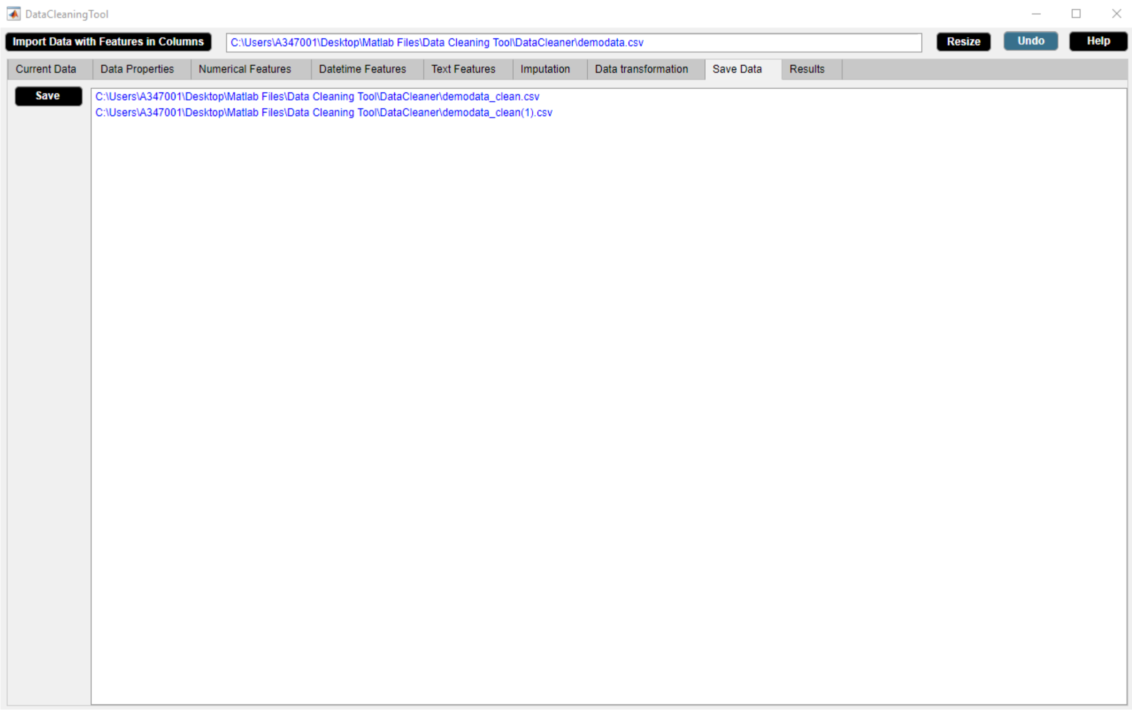


Figure 3.9: Save Data Widget.

3.9 Results

The Results widget displays information about the final report. The Results widget is shown in figure 3.10. The properties of the Results widget are as follows.

- The widget generates results in pdf format after data cleaning. The results contains a detailed report of all the changes made in DataCleaningTool.
- Results can be generated containing a detailed report of specific changes made in DataCleaningTool.
- Results can be generated for multiple times after each activity.
- The full paths of the results are displayed.

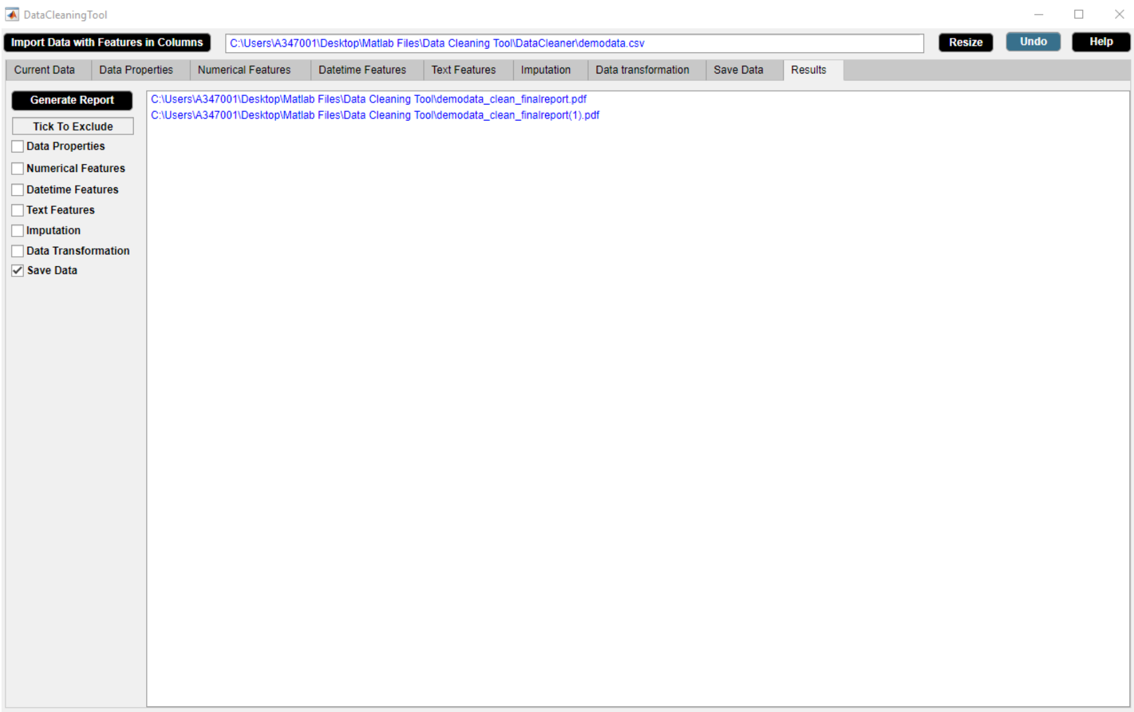


Figure 3.10: Results Widget.

4

Results and Discussion

The results are discussed in chapter 4. In Section 4.1, the performance analysis of the missForest method is studied. The analysis is done in order to get an idea of how well the method works. The results of the analysis also provide the basis for the recommendation that the user receives when they try to impute missing values. Section 4.2 presents the performance analysis of different multivariate outlier detection methods. However, none of the multivariate outlier detection methods are implemented in DataCleaningTool due to time constraint. Section 4.3 presents a demo of DataCleaningTool. This basically demonstrates the results of the methods as described in Chapter 3.

4.1 Performance Analysis of the MissForest Method

The performance of the missForest method is analysed using the automobile dataset [36]. The automobile dataset describes the relation between different car attributes and car price. The different $n \times p$ dimensional datasets used in the study are acquired by selecting random subsets of the automobile dataset. Here n is the number of observations and p is the number of features.

4.1.1 Continuous Data

In the section, we focus on continuous data only. Here all features are numeric. We examine the following three cases:

Case A: Overdetermined where number of observations is greater than number of features in the dataset, $n >> p$

- Dataset I ($n = 8p$, $n = 120$, $p = 15$): The dataset consists of 120 observations and 15 features.
- Dataset II ($n = 2p$, $n = 30$, $p = 15$): The dataset consists of 30 observations and 15 features.

Case B: Equal where number of observations is equal to number of features in the dataset, $n = p$

- Dataset III ($n = p$, $n = 15$, $p = 15$): The dataset consists of 15 observations and 15 features.

Case C: Underdetermined where number of observations is less than number of features in the dataset, $p >> n$

- Dataset IV ($n = 0.5p$, $n = 7$, $p = 15$): The dataset consists of 7 observations and 15 features.

We perform error analysis by plotting different percentages of missing data versus their respective average NRSME for the continuous datasets I-IV. The plots are shown in figure 4.1. The average NRSME values are presented in table A.1 which can be found in appendix A. The performance of the missForest method for continuous data only is discussed as follows.

– The missForest imputation method does not converge at $> 90\%$, $> 80\%$, $> 70\%$, $> 50\%$ of missing data for datasets I, II, III and IV respectively.

– The general trend as seen in figure 4.1 shows that there is a linear relationship between the average NRSME and the percentage of missing data. The average NRSME value increases with increase in percentage of missing data.
– The performance of missForest method on different datasets is compared as follows: Dataset I > Dataset II > Dataset III > Dataset IV.
– The missForest method performs best for the overdetermined case.

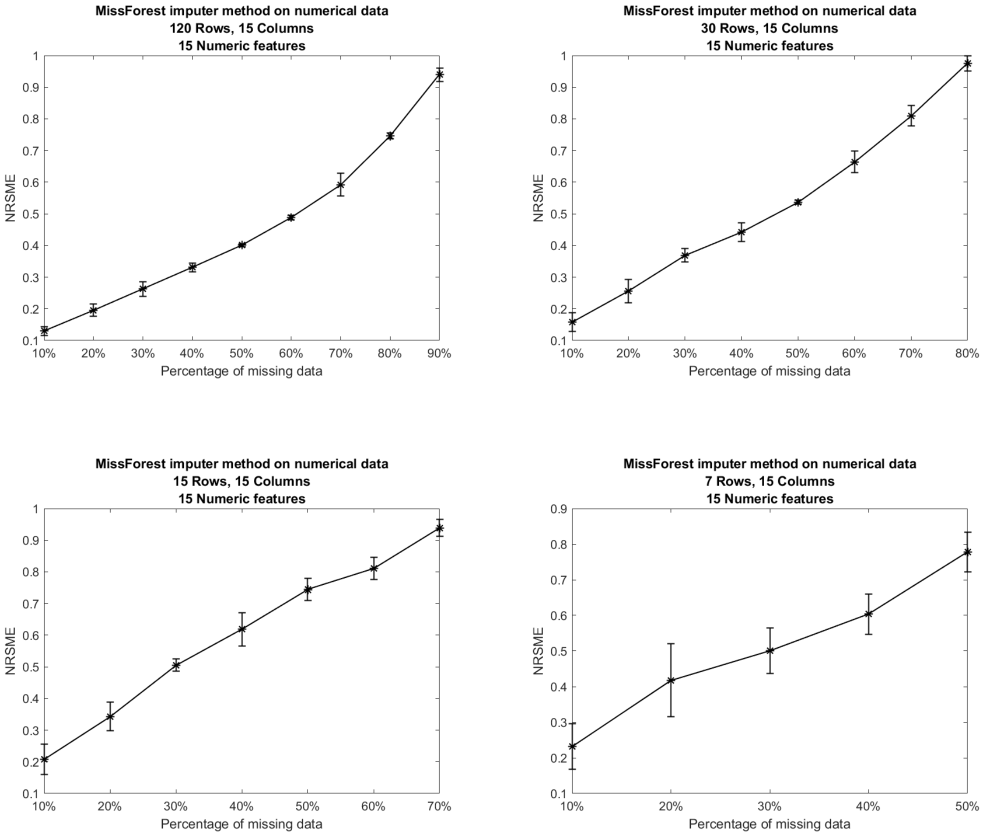


Figure 4.1: Figures show the plots of average NRSME over different percentages of missing data for continuous datasets I-IV. Asterisk represents average NRSME and vertical line represents standard deviation of average NRSME calculated for each percentage of missing data after 5 runs.

4.1.2 Categorical Data

In the section, we focus on categorical data only. Here all 9 features are categorical. We investigate the following three cases:

Case A: Overdetermined where number of observations is greater than number offeatures in the dataset, $n >> p$

- Dataset V ($n = 8p$, $n = 72$, $p = 9$): The dataset consists of 72 observations and 9 features.
- Dataset VI ($n = 2p$, $n = 18$, $p = 9$): The dataset consists of 18 observations and 9 features.

Case B: Equal where number of observations is equal to number of features in the dataset, $n = p$

- Dataset VII ($n = p$, $n = 9$, $p = 9$): The dataset consists of 9 observations and 9 features.

Case C: Underdetermined where number of observations is less than number of features in the dataset, $p >> n$

- Dataset VIII ($n = 0.5p$, $n = 4$, $p = 9$): The dataset consists of 4 observations and 9 features.

We perform error analysis by plotting different percentages of missing data versus their respective PEC for the categorical datasets V-VIII. The plots are shown in figure 4.2. The PEC values are presented in table A.2 which can be found in appendix A. The performance of the MissForest Method for categorical data only is discussed below:

– The missForest imputation method does not converge at $> 90\%$, $> 80\%$, $> 70\%$, $> 40\%$ of missing data for datasets V, VI, VII and VIII respectively.
– The general trend as seen in figure 4.2 that the PEC values has a linear relationship with different percentages of missing data. The PEC value increases with increase in percentage of missing data.
– The performance of missForest method on different datasets is compared as follows: Dataset V $>$ Dataset VI $>$ Dataset VII $>$ Dataset VIII.
– The missForest method performs best for the overdetermined case.

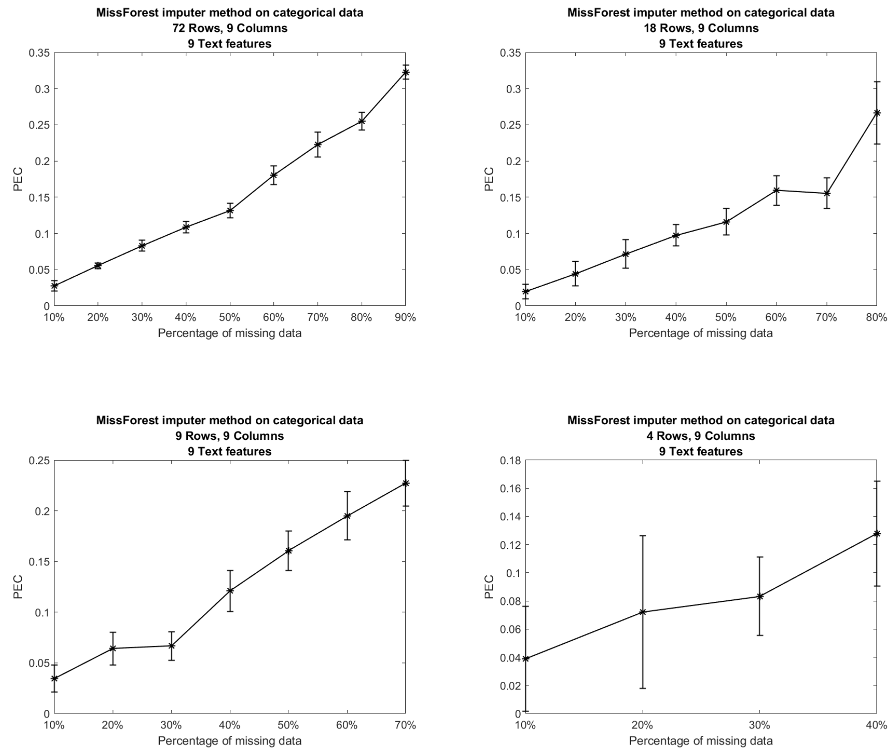


Figure 4.2: Figures show the plots of PEC over different percentages of missing data for categorical datasets V-VIII. Asterisk represents the PEC and vertical line represents the standard deviation of PEC calculated for each percentage of missing data after 5 runs.

4.1.3 Mixed-Type Data

In the section, we focus on mixed-type data. Here 15 features are numeric and 9 features are text. We study the following three cases.

Case A: Overdetermined where number of observations is greater than number of features in the dataset, $n >> p$

- Dataset IX ($n = 8p$, $n = 192$, $p = 24$): The dataset consists of 192 observations and 24 features.
- Dataset X ($n = 2p$, $n = 48$, $p = 24$): The dataset consists of 48 observations and 24 features.

Case B: Equal where number of observations is equal to number of features in the dataset, $n = p$

- Dataset XI ($n = p$, $n = 24$, $p = 24$): The dataset consists of 24 observations and 24 features.

Case C: Underdetermined where number of observations is less than number of features in the dataset, $p >> n$

- Dataset XII ($n = 0.5p$, $n = 12$, $p = 24$): The dataset consists of 12 observations and 24 features.

We perform error analysis by plotting different percentages of missing data versus their respective average NRSME and PEC for the mixed type datasets IX-XII. The plots are shown in figure 4.3. The average NRSME values and the PEC values are presented in table A.3 and table A.4, respectively which can be found in appendix A. The performance of the MissForest Method for mixed type of data is discussed below.

- The missForest imputation method does not converge at $> 90\%$, $> 90\%$, $> 80\%$, $> 70\%$ of missing data for mixed type datasets IX, X, XI and XII respectively.

- The general trend as seen in figure 4.2 that the average NRSME values and the PEC values has a linear relationship with different percentages of missing data. The average NRSME value and the PEC value increases with increase in percentage of missing data.

- The results of the comparison of different datasets are seen in figures A.3 and A.4. The missForest method performs as follows: Dataset IX > Dataset X > Dataset XI > Dataset XII.

- The missForest method performs best for the overdetermined case.

- The MissForest method works well for any type of data. Particularly, it can handle both continuous and categorical data at the same time.

- There is no need for prior scaling of data to perform the MissForest method.

- The imputation method performs well for underdetermined case ($n = 0.5p$). This implies that the MissForest method can handle high dimensional data.

- For mixed type data, the imputation method does not converge at $> 90\%$ of missing data for overdetermined system ($n = 8p$ & $n = 2p$), whereas the imputation method does not converge at $> 80\%$ of missing data for equal system ($n = p$) and $> 60\%$ of missing data for underdetermined system ($n = 0.5p$). This shows that the MissForest method can perform imputation for large amount of missing observations in the data.

- From our analysis, we see the trend that both NRMSE and PEC increases with increasing percentage of missing data. The MissForest algorithm is less biased than other imputation methods since it is based on random forests. Random forests consider multiple trees and each tree is trained on a subset of data and the final outcome depends on all the trees which reduces the biasedness of the method.

- Although the MissForest method can handle missing data very well, it is computationally complex due to the large number of decision trees joined together. Due to the complexity of the MissForest method, it is much more time consuming than other imputation methods. The comparison of runtimes of several imputation methods is given in figure 2.15.

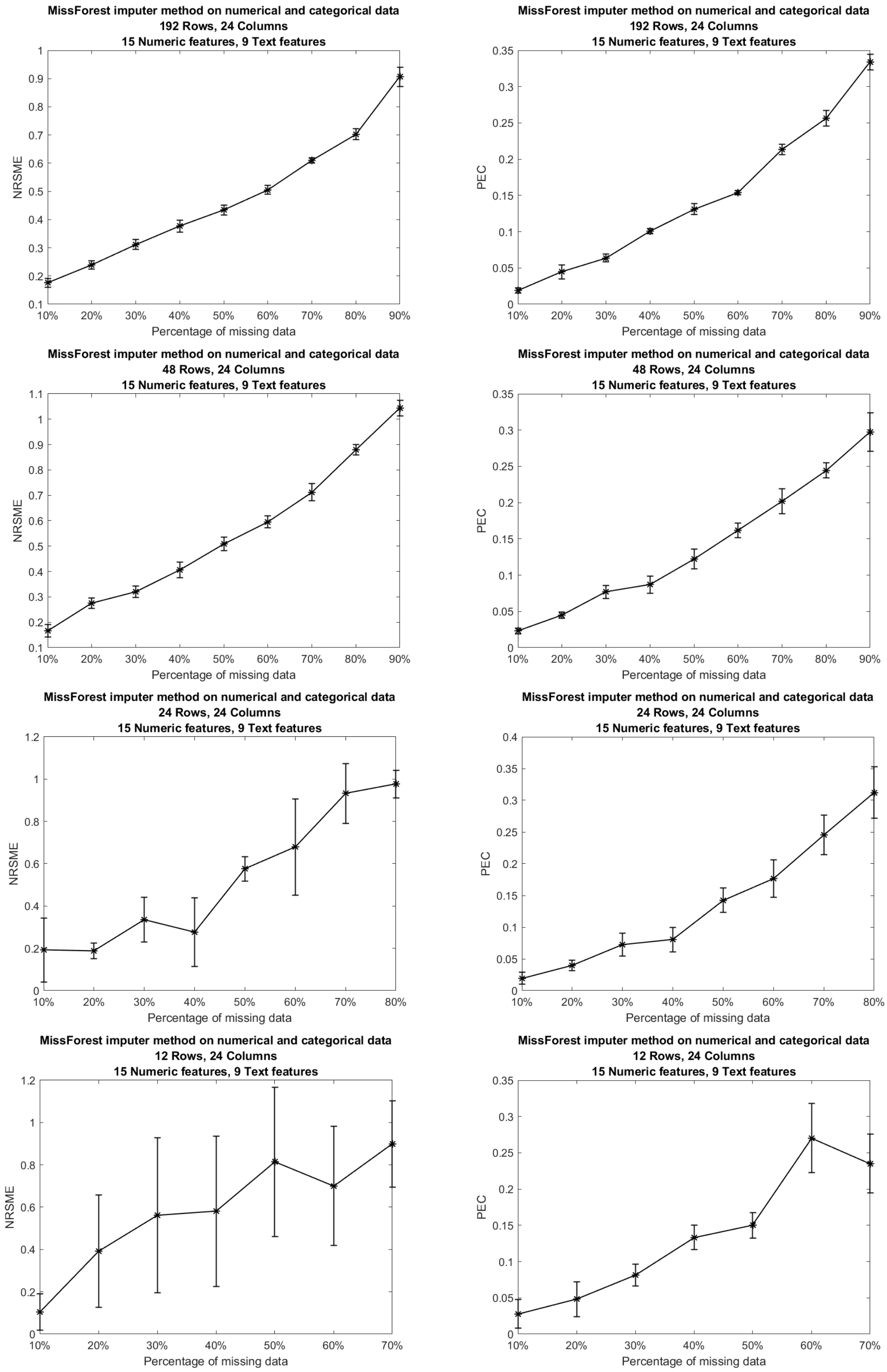


Figure 4.3: Left figures show the plots of average NRSME over different percentages of missing data while right figures show the plots of PEC over different percentages of missing data for datasets IX-XII. Asterisks represent the average NRSME or PEC and vertical lines represent the standard deviation of average NRSME or PEC calculated for each percentage of missing data after 5 runs.

4.2 Performance Analysis of the Outlier Detection Methods

We analyse the performance of different outlier detection methods such as leverage, local outlier factor and DBSCAN. Unfortunately, the results of this analysis are not incorporated in the final tool because of the limited time. The evaluation is performed on various outlier detection datasets obtained from [37]. These outlier detection datasets are of different dimensions. These datasets are labeled data for training and validation of outlier detection methods. Each datapoint of these datasets is labeled as true outlier or inlier by a specific outlier detection method.
For each outlier detection method studied here, we calculate outlier accuracy, inlier accuracy and total accuracy for these datasets. Outlier accuracy is defined as the percentage of accuracy between true outliers and outliers labeled by an outlier detection method in an outlier detection dataset. Inlier accuracy is defined as the percentage of accuracy between true inliers and inliers labeled by an outlier detection method in an outlier detection dataset. Total accuracy is defined as the percentage of accuracy between true labels and labels marked by an outlier detection method in an outlier detection dataset.

4.2.1 Leverage

The accuracy percentages of leverage method for different datasets are presented in table 4.1.

Table 4.1: The table represents the comparison of accuracy percentages of leverage with different datasets.

Accuracy percentage	**Parameter**	**Leverage**		
Outlier Detection Datasets	**Threshold**	**Outlier Accuracy**	**Inlier Accuracy**	**Total Accuracy**
Speech (3686,400) (1.65%)	0.2170 (2p/n)	0	100	98.35
Thyroid (3772,6) (2.5%)	0.0032 (2p/n)	54.84	94.54	93.56
Cardio (1831,21) (9.6%)	0.0229 (2p/n)	38.07	95.71	90.17
Arrhythmia (452,274) (15%)	0.9093 (1.5p/n)	0	100	85.40
Satellite (6435,36) (32%)	0.0112 (2p/n)	18.66	97.23	72.37
Ionosphere(351,33) (36%)	0.1880 (2p/n)	53.17	100	83.19

4.2.2 Local Outlier Factor

The accuracy percentages of local outlier factor method for different datasets are presented in table 4.2.

Table 4.2: The table represents the comparison of accuracy percentages of local outlier factor with different datasets.

Accuracy percentage	**Parameter**	**Local Outlier Factor**		
Outlier Detection Datasets	**Threshold**	**Outlier Accuracy**	**Inlier Accuracy**	**Total Accuracy**
Speech (3686,400) (1.65%)	0.9835 quantile	1.64	98.35	96.74
Thyroid (3772,6) (2.5%)	0.975 quantile	24.73	98.07	96.26
Cardio (1831,21) (9.6%)	0.904 quantile	17.05	91.18	84.05
Arrhythmia (452,274) (15%)	0.85 quantile	50	90.93	84.96
Satellite (6435,36) (32%)	0.68 quantile	42.39	72.81	63.19
Ionosphere(351,33) (36%)	0.64 quantile	73.81	85.33	81.20

Accuracy percentage	Parameter	Local Outlier Factor		
Outlier Detection Datasets	**Threshold**	**Outlier Accuracy**	**Inlier Accuracy**	**Total Accuracy**
Speech (3686,400) (5%)	0.95 quantile	8.20	95.06	93.63
Thyroid (3772,6) (5%)	0.95 quantile	43.01	95.95	94.65
Cardio (1831,21) (5%)	0.95 quantile	15.34	96.07	88.31
Arrhythmia (452,274) (5%)	0.95 quantile	24.24	98.19	87.39
Satellite (6435,36) (5%)	0.95 quantile	7.61	96.20	68.17
Ionosphere(351,33) (5%)	0.95 quantile	14.29	100	69.23

4.2.3 DBSCAN

The accuracy percentages of DBSCAN method for different datasets are presented in table 4.3.

Table 4.3: The table represents the comparison of accuracy percentages of DBSCAN with different datasets.

Accuracy percentage	Parameters		DBSCAN		
Outlier Detection Datasets	**Eps**	**MinPts**	**Outlier Accuracy**	**Inlier Accuracy**	**Total Accuracy**
Speech (3686,400) (1.65%)	26	500	0	100	98
Thyroid (3772,6) (2.5%)	0.09	10	77	94	94
Cardio (1831,21) (9.6%)	4.5	25	19	99.7	91
Arrhythmia (452,274) (15%)	275	300	23	99	88
Satellite (6435,36) (32%)	40	50	27	96	74
Ionosphere(351,33) (36%)	4	40	22	100	72

The performance of different outlier detection methods is studied as follows.

- Outlier accuracy is of primary concern while evaluating the performance of an outlier detection method since it is an accuracy measure of outliers in a dataset. In the context of the outlier accuracy, leverage and DBSCAN methods perform comparatively better than local outlier factor.
- There are different parameters to be set in outlier detection methods. The parameters play a significant role in finding outliers. Thus, special priority should be given in setting the parameters.

4.3 Demo

DataCleaningTool is a user friendly, free and open source data cleaning standalone application developed using Matlab App Designer 2018b version. DataCleaningTool app installation file can be found in the github repository [38]. The Matlab code can be accessed from github repository [39]. DataCleaningTool is a data cleaning application which consists of multiple widgets and buttons. The properties of DataCleaningTool are

- DataCleaningTool always opens in a full screen mode. The application can be resized to a reduced size.
- Each widget provides specific statistical information about the data.
- Each button aims to clean data by resolving inconsistencies, smoothing noisy data, identifying outliers, removing outliers or filling in missing observations.
- Each widget gets updated accordingly after each activity.
- All buttons are black in color. Pressing a button each time changes the button color from black to grey color and then again to black. The button remains grey in color until it completes its specific task and all widgets gets updated accordingly.
- Pressing any button turns the Undo button to blue color. The Undo button remains blue in color until last activity can be undone.
- Sliders and their corresponding edit boxes are interdependable.
- User can find help in using DataCleaningTool.

We demonstrate the DataCleaningTool using an example dataset 'demodata.csv'. The example dataset is obtained by tweaking the coronavirus dataset [40]. The example dataset is of dimension 127×12. The example dataset consists of the following features.

1. Serial_Number: Unique identifier to a country.
2. Country_Region: Name of the country.
3. Population_Size: Size of the population of the country.
4. tourism: Number of international arrivals in the country.
5. Date_FirstFatality: Date of the first fatality in the country.
6. Date_FirstConfirmedCase: Date of the first confirmed case in the country.
7. Latitude: Geographic coordinate of the country.
8. Longtitude: Geographic coordinate of the country.
9. mean_Age: Mean age of the population of the country.
10. Lockdown_Date: Date of the lockdown in the country.
11. Lockdown_Type: Level of the lockdown (full or partial) in the country.
12. Country_Code: Geographical code representing the country.

Using the example dataset, we will show how to clean a statistical dataset using DataCleaningTool developed in this thesis. The complete demo can be found in appendix B. First we wish to understand our data by doing a descriptive statistics analysis of our dataset. In Descriptive Statistics, we are describing and summarizing our data, either through numerical calculations or graphs. Secondly we distinguish id feature 'Serial_Number' from other numerical features. Next we detect inconsistent capitalization of feature names such as 'Serial_Number', 'Country_Region', 'Population_Size', 'tourism', 'Date_FirstFatality', 'Date_FirstConfirmedCase', 'Latitude', 'Longtitude', 'mean_Age', 'Lockdown_Date', 'Lockdown_Type', 'Country_Code' and unify inconsistent capitalization of feature names. Then we wish to extract data for the countries whose 'Population_Size' is greater than 'Tourism'. So we set cross-field validation constraint to remove irrelevant observations. Then we wish to extract data for the countries whose maximum 'Mean_Age' is 45. So we set the range constraint to remove irrelevant observations. We delete feature 'Longitude' since it contains a large percentage of missing observations. We illustrate missing observations by missingness plot and impute missing observations using missForest method. Lastly, we log transform the numerical feature 'Population_Size' which makes the feature less skewed.

4.3.1 Load data

The first step is to load the example data ‘demodata.csv’. We use Import Data with Features in Columns button to load the example data. We browse for the input file. The full path of the selected file is displayed and the file is loaded. Figures 4.4-4.6 illustrate how to load data in DataCleaningTool.

Figure 4.4: Step 1. Click Import Data with Features in Columns button.

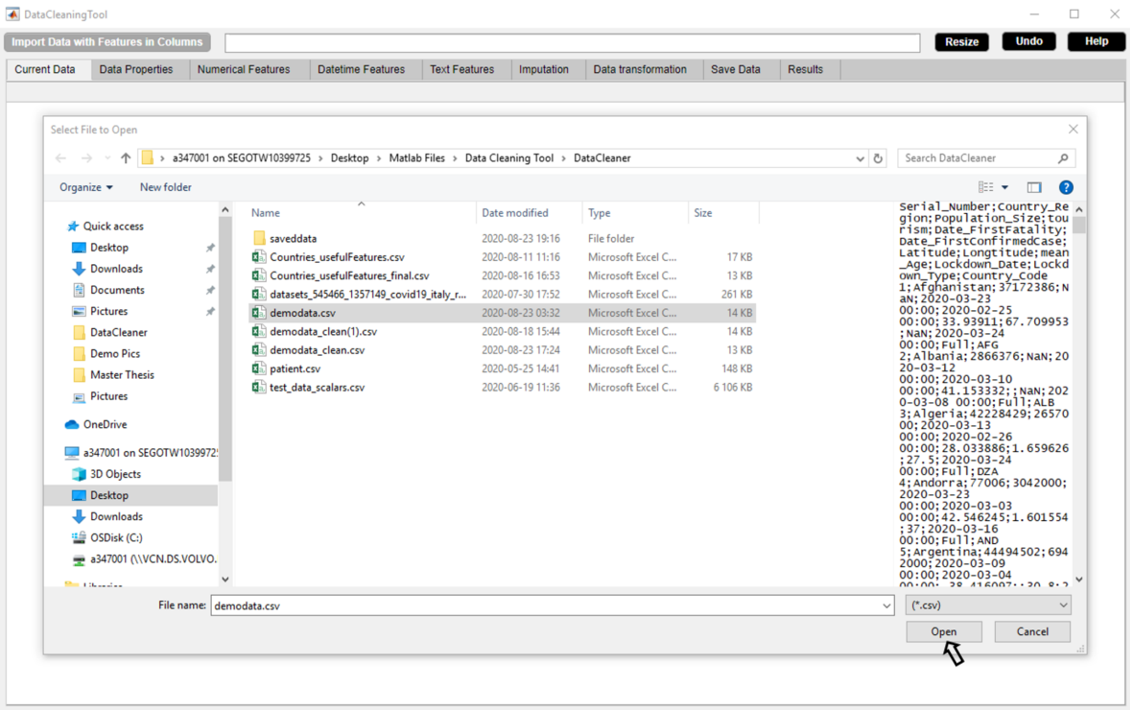


Figure 4.5: Step 2. Import Data with Features in Columns button in use turns grey in color and an open dialog box appears. Browse for an input file.

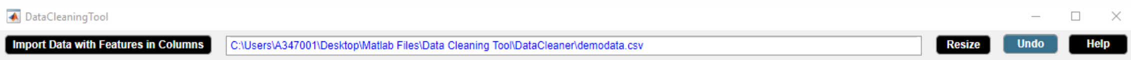


Figure 4.6: Step 3. Import Data with Features in Columns button returns back to its original color once it completes its task. The full path of the selected file is displayed and the file is loaded.

4.3.2 Show statistical information

Figure 4.7 shows the statistical information of the example data. Figures 4.8-4.10 shows the descriptive statistics of the numerical, the datetime and the text features respectively.

Figure 4.7: Statistical information of the example data is displayed in the Data Properties widget.

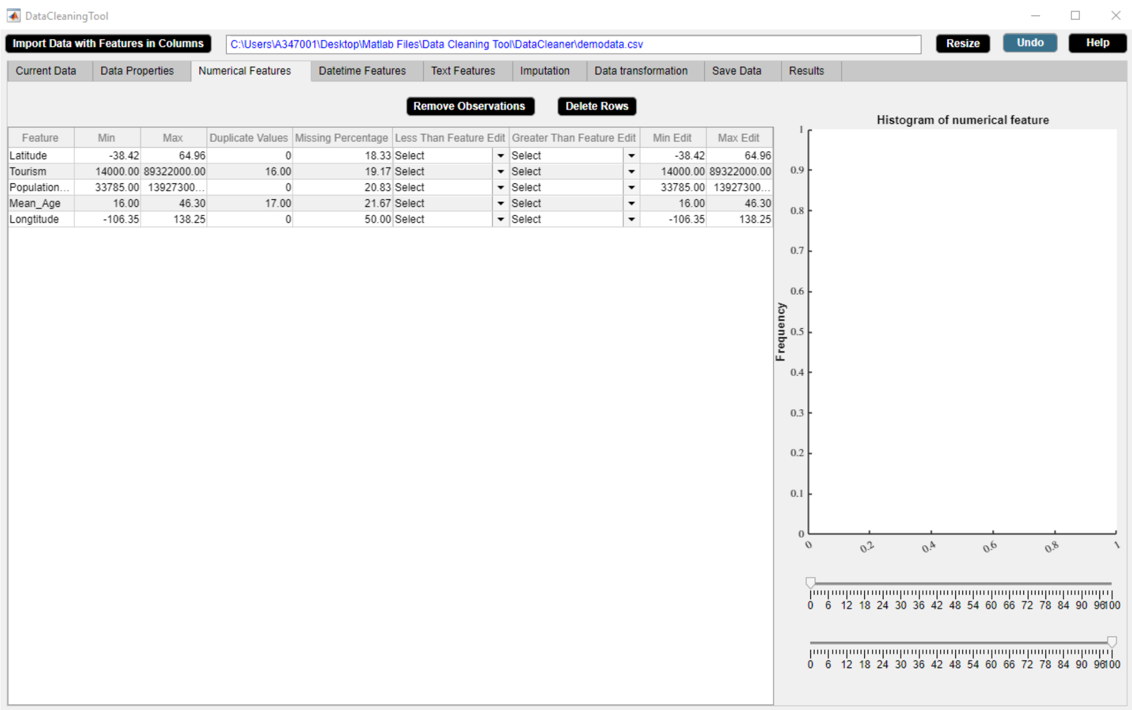


Figure 4.8: Descriptive statistics of numerical features is displayed in the Numerical Features widget.

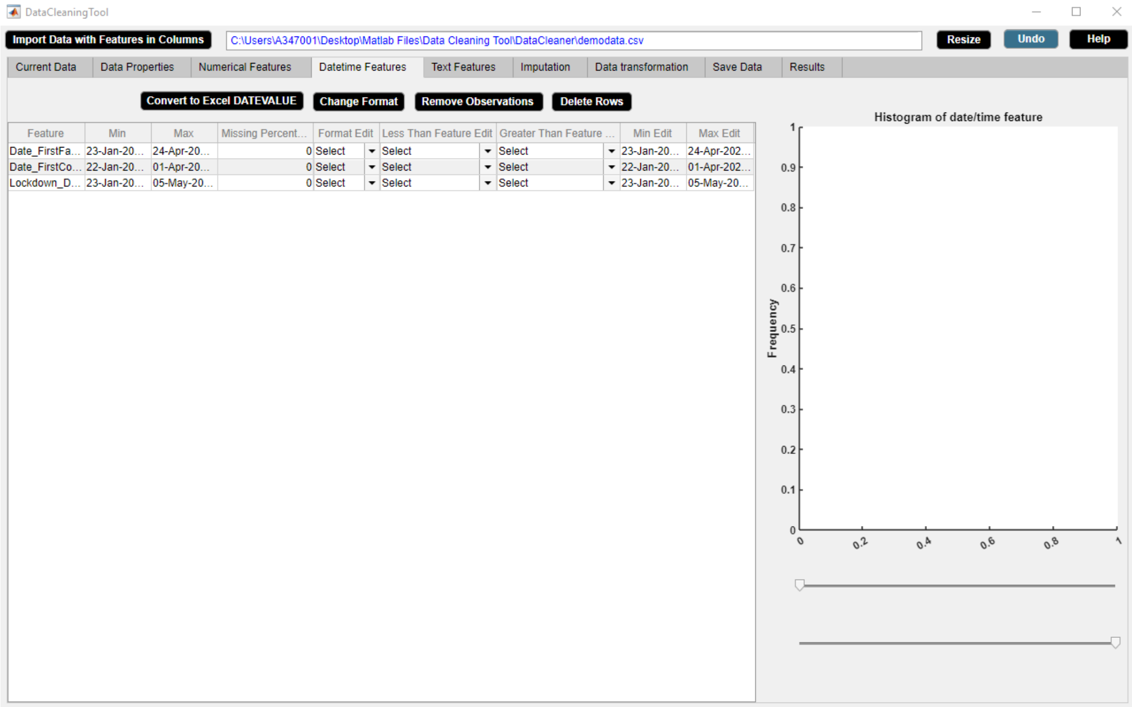


Figure 4.9: Descriptive statistics of datetime features is displayed in the Datetime Features widget.

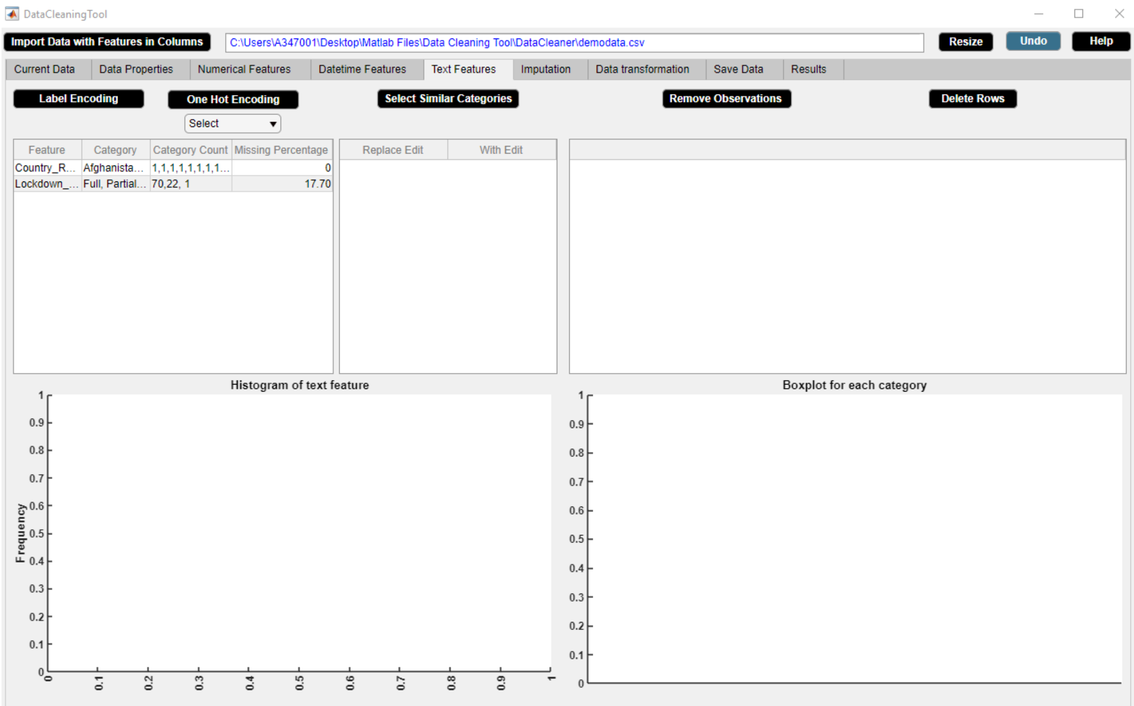


Figure 4.10: Descriptive statistics of text features is displayed in the Text Features widget.

4.3.3 Detect and rectify incorrect id data type

In the example data, ‘Serial_Number’ represents a unique identifier to a country. We select the feature ‘Serial_Number’ and use Id button to seperate id feature ‘Serial_Number’ from numerical features. Figures 4.11-4.12 illustrate how to detect incorrect id data type in DataCleaningTool.

Figure 4.11: Step 1. Select a feature from numerical or datetime or text list box. Click Id button.

Figure 4.12: Step 2. The selected numerical or datetime or text feature becomes id feature.

4.3.4 Detect and unify inconsistent capitalization of feature names

In the example data, the feature names ‘tourism’, ‘mean_Age’ have inconsistent capitalization. We use Feature Names button to capitalize each feature name so as to unify inconsistent capitalization. Figures 4.13-4.14 illustrate how to detect inconsistent feature names in DataCleaningTool.

Figure 4.13: Step 1. Select case from dropdown menu. Click Feature Names button.

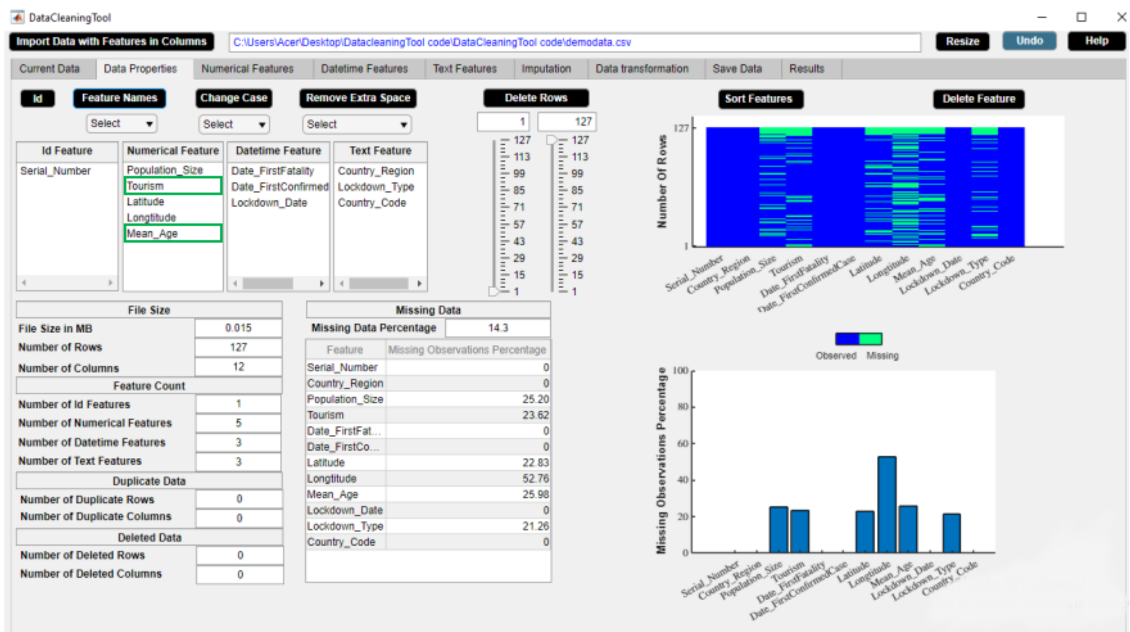


Figure 4.14: Step 2. Check that the feature names have consistent capitalization.

4.3.5 Set cross-field validation constraint and remove irrelevant observations

We use Remove Observations button to extract data for the countries whose 'Population_Size' is greater than 'Tourism'. Figures 4.15-4.16 illustrate how to set constraint in DataCleaningTool.

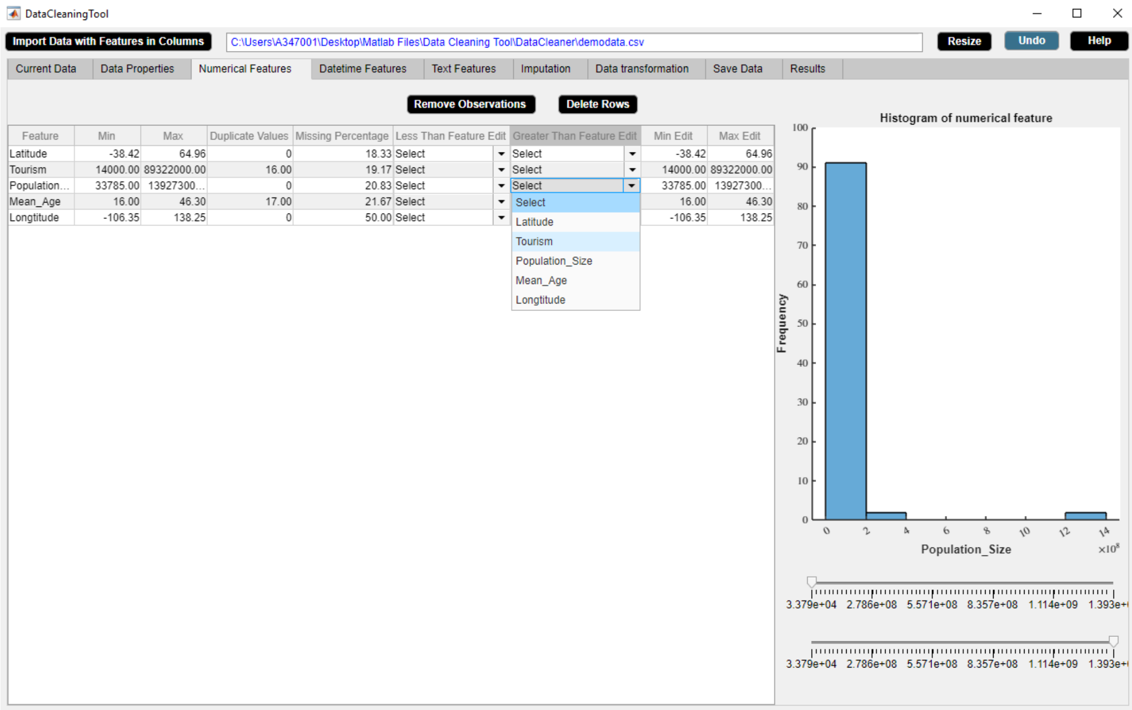


Figure 4.15: Step 1. Set constraint from Less or Greater Than Feature Edit dropdown menu.

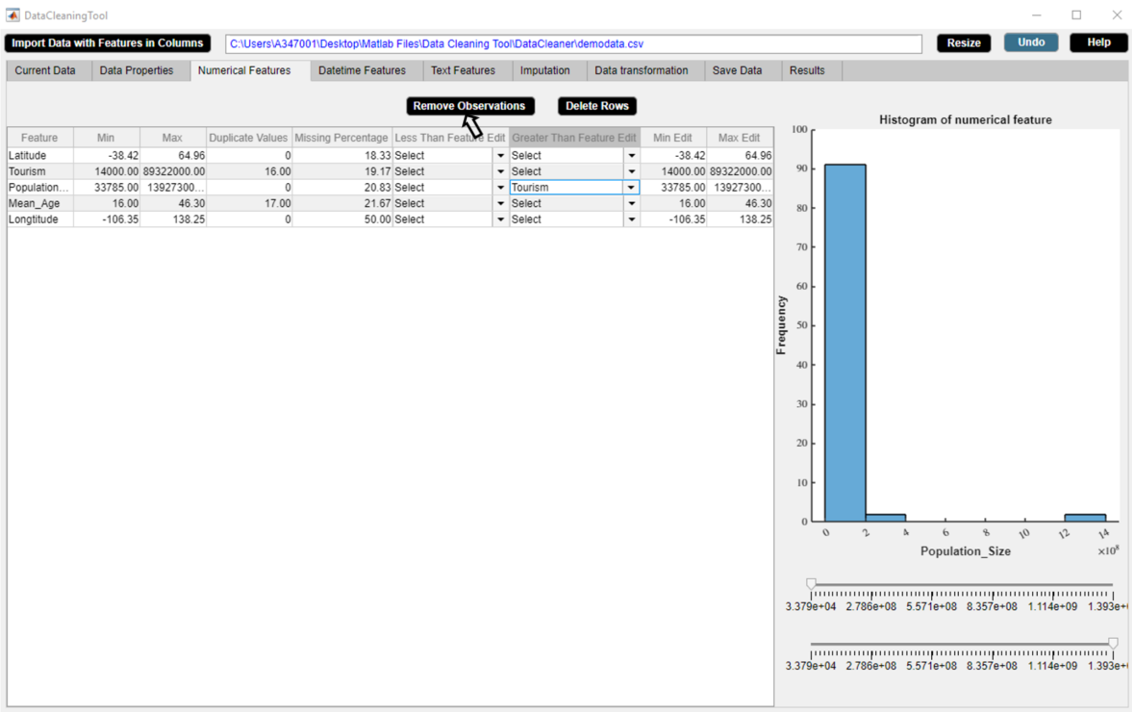


Figure 4.16: Step 2. Click Remove Observations button to replace irrelevant by missing.

4.3.6 Set range constraint and remove irrelevant observations

We use Delete Rows button to extract data for the countries whose maximum 'Mean_Age' of population is 45. Figures 4.31-4.32 illustrate how to set range constraint in DataCleaningTool.

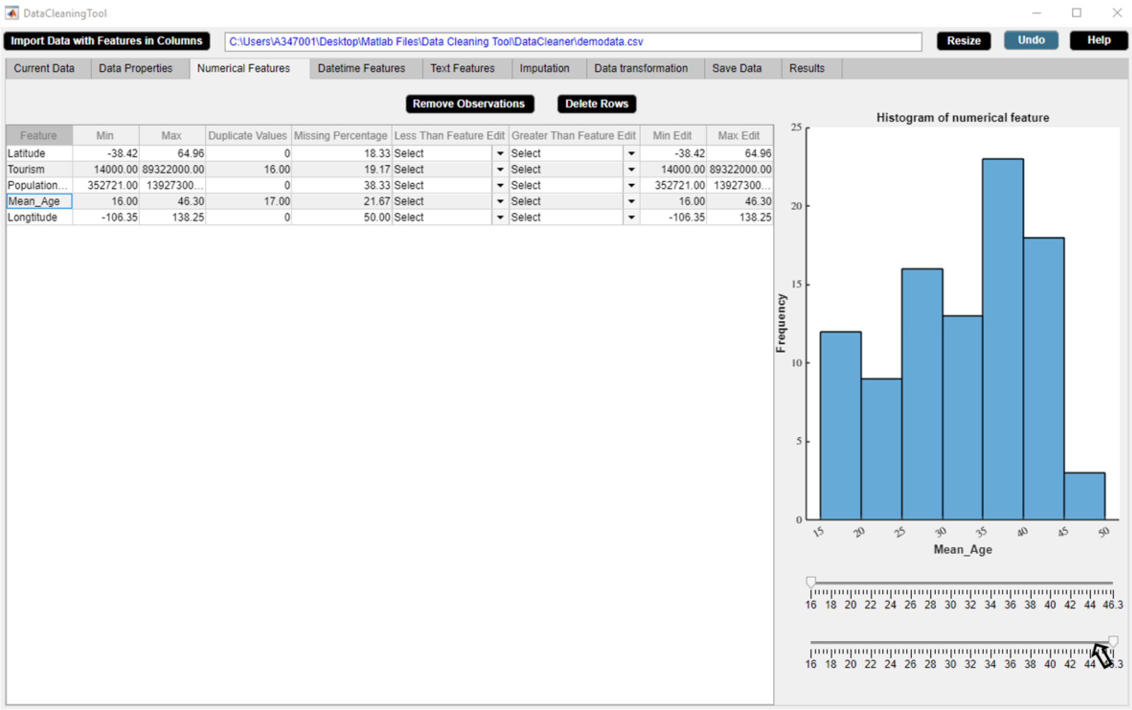


Figure 4.17: Step 1. Set maximum 'Mean_Age' as 45 from maximum slider or Max Edit box.

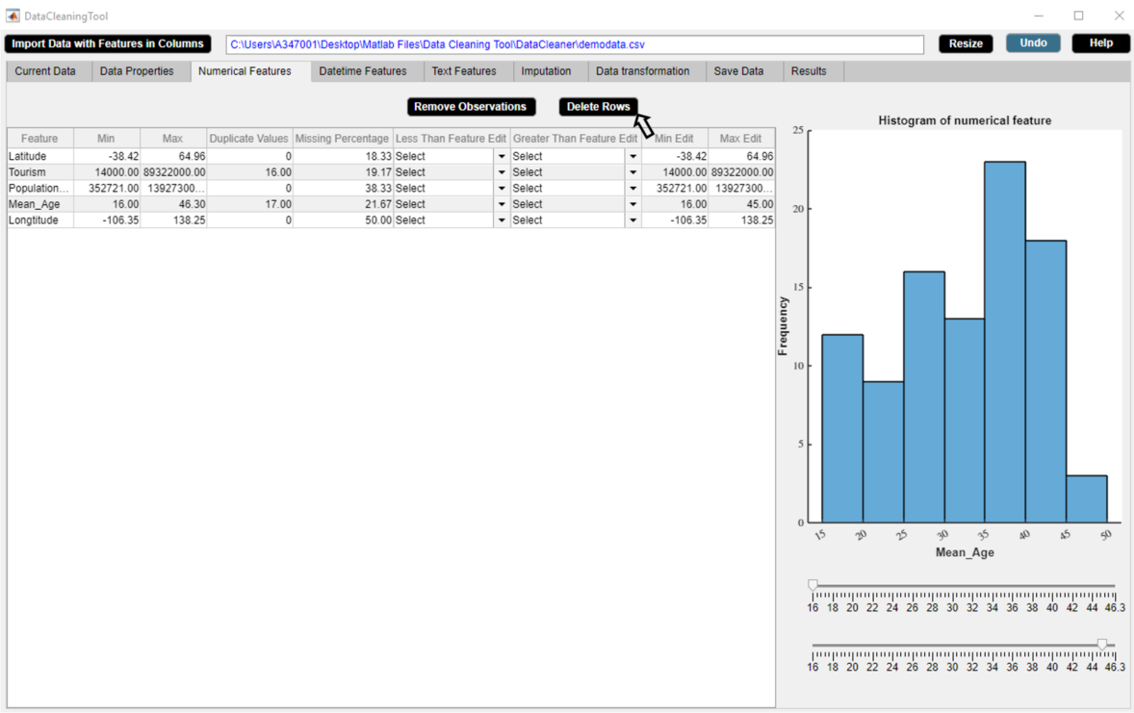


Figure 4.18: Step 2. Click Delete Rows button to delete rows containing irrelevant observations. The updated histogram of the selected feature appears on the left side of widget.

4.3.7 Label encoding

We use Label Encoding button to label encode the categorical feature 'Lockdown_Type'. Figures 4.19-4.20 illustrate how to label encode a categorical feature in DataCleaningTool.

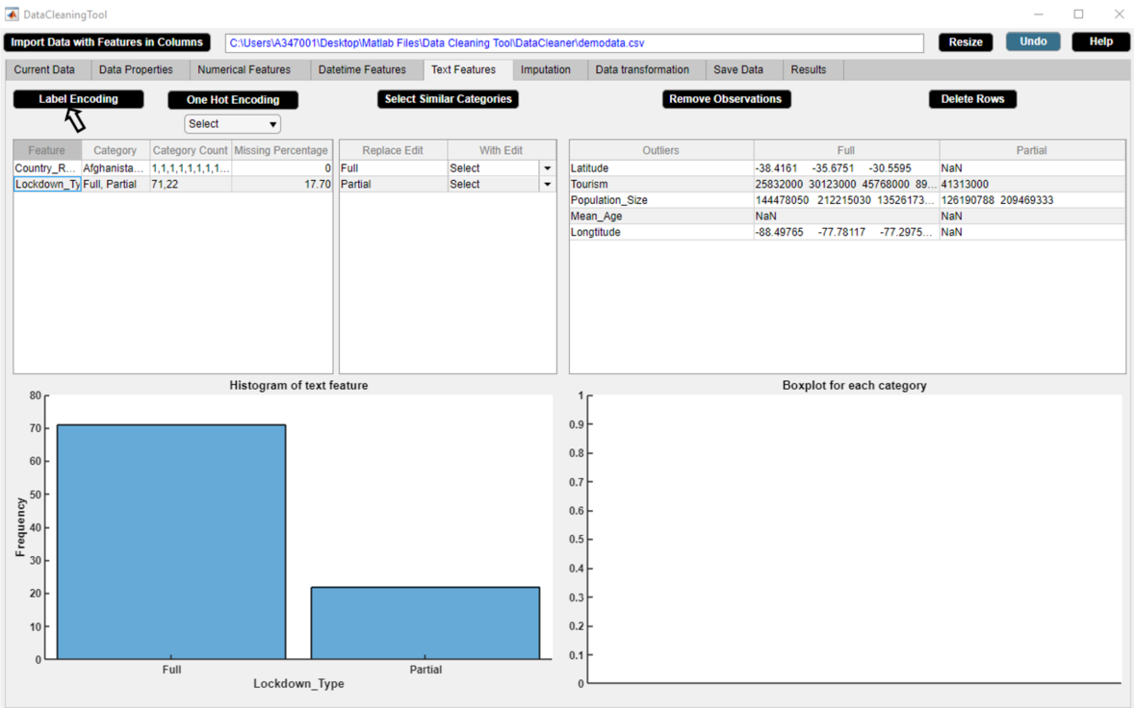


Figure 4.19: Step 1. Select categorical feature from Feature column of the text features descriptive statistics table. Click Label Encoding button.

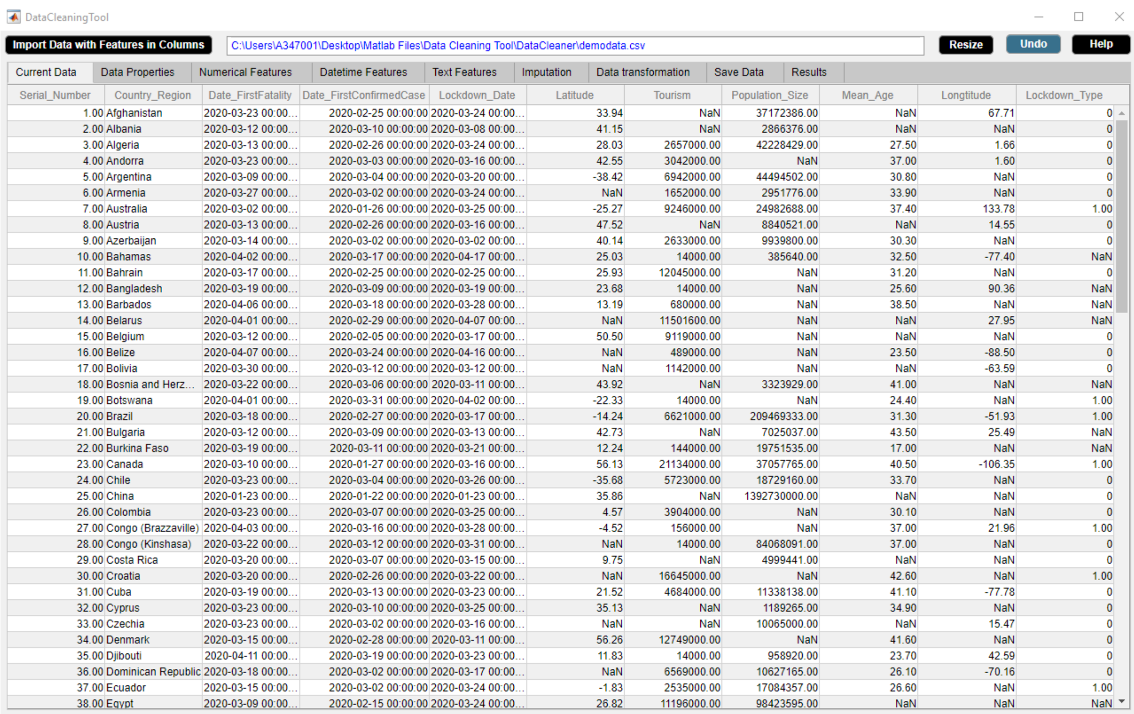


Figure 4.20: Step 2. Check that the text feature is label encoded in Current Data widget.

4.3.8 One-hot encoding

We use One Hot Encoding button to one hot encode the categorical feature 'Country_Region'. Figures 4.21-4.22 illustrate how to one hot encode a categorical feature in DataCleaningTool.

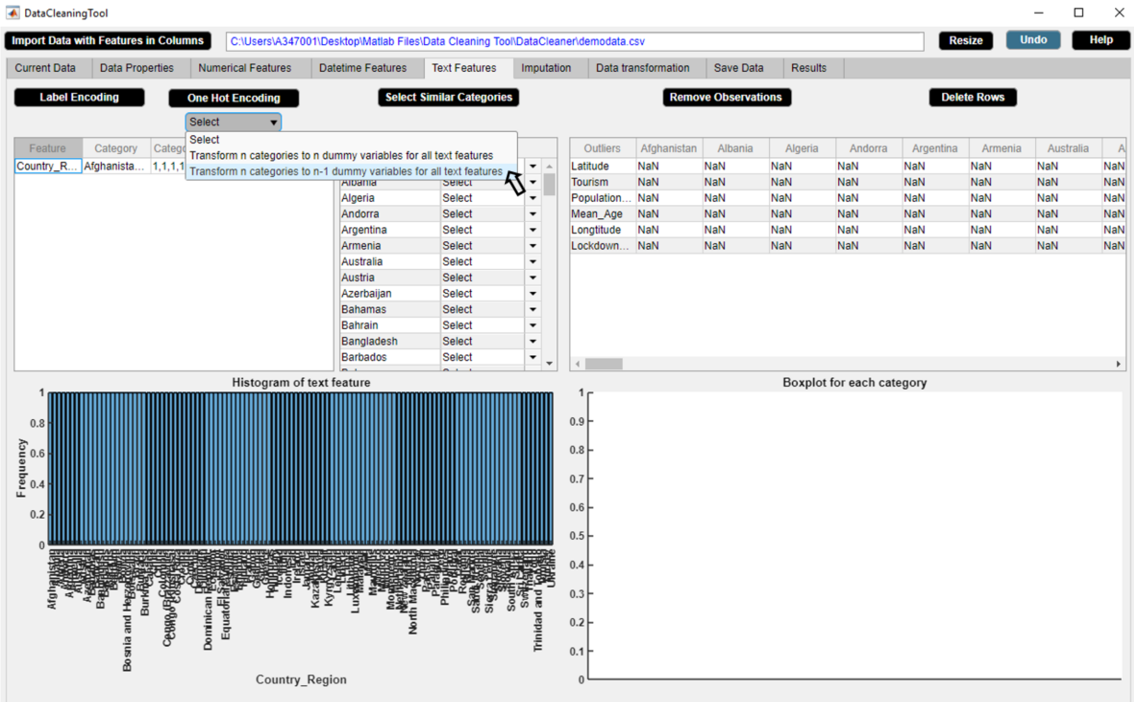


Figure 4.21: Step 1. Select categorical feature from Feature column of the text features descriptive statistics table. Select an option from dropdown menu. Click One Hot Encoding button.

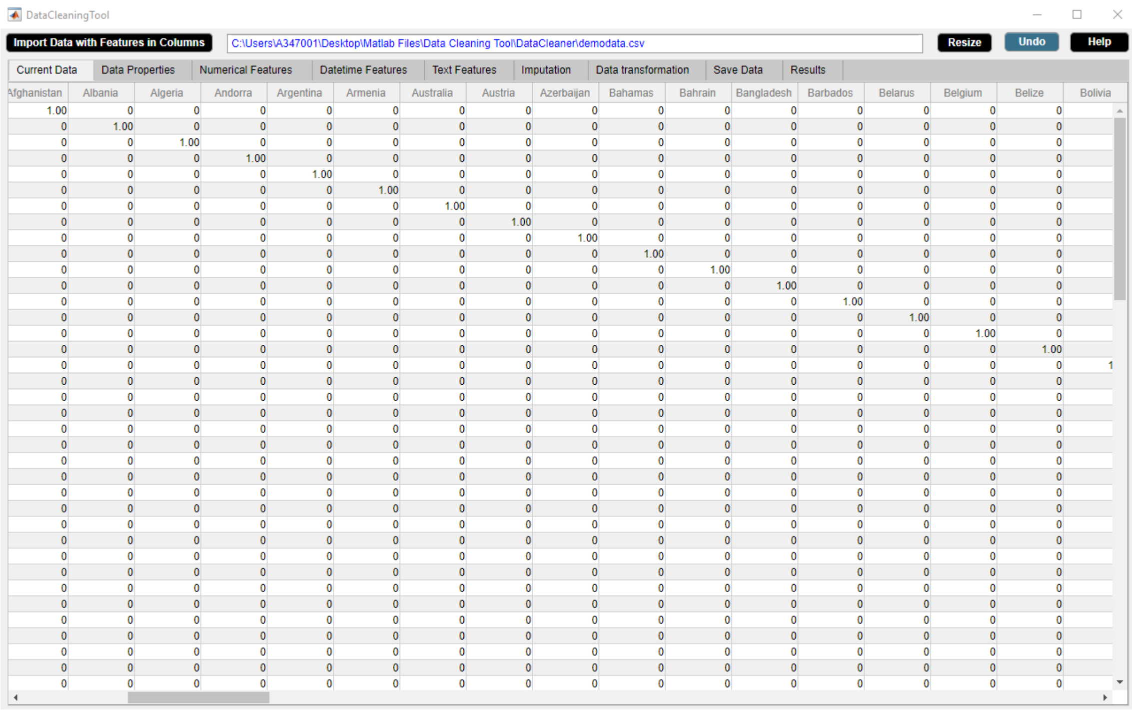


Figure 4.22: Step 2. Check that the text feature is one hot encoded in Current Data widget.

4.3.9 Drop feature with large number of missing observations

We use Delete Feature button to drop 'Longitude' feature which has a large number of missing values. Figures 4.23-4.24 illustrate how to drop a feature in DataCleaningTool.

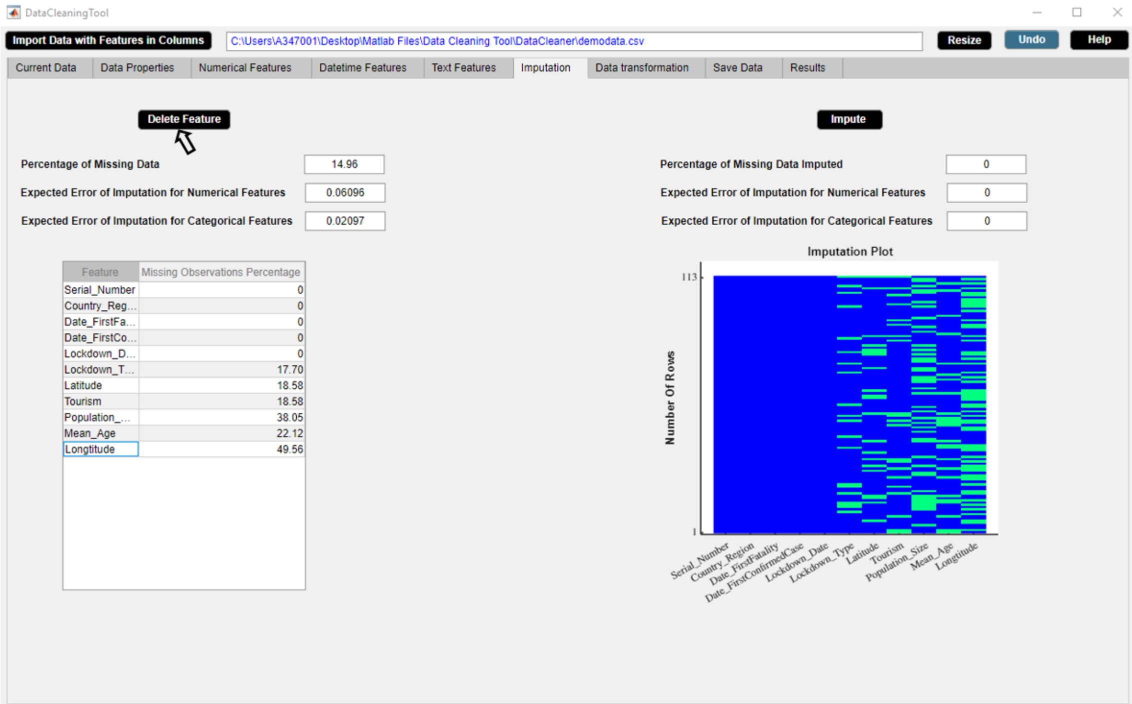


Figure 4.23: Step 1. Select a feature from Feature column of missing observations percentage table. Click Delete Feature button.

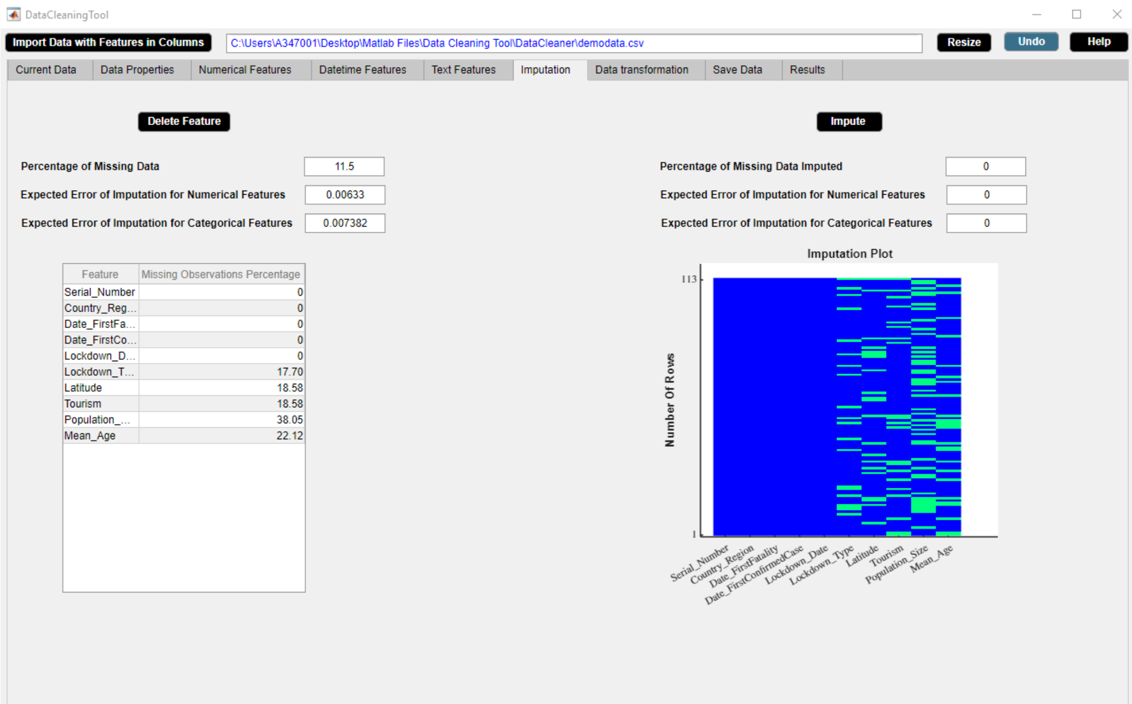


Figure 4.24: Step 2. Check that the selected feature is deleted.

4.3.10 Illustrate and impute missing observations

We use Impute button to impute missing values in the example data using missForest method. Figures 4.25-4.26 illustrate how to impute missing observations in DataCleaningTool.

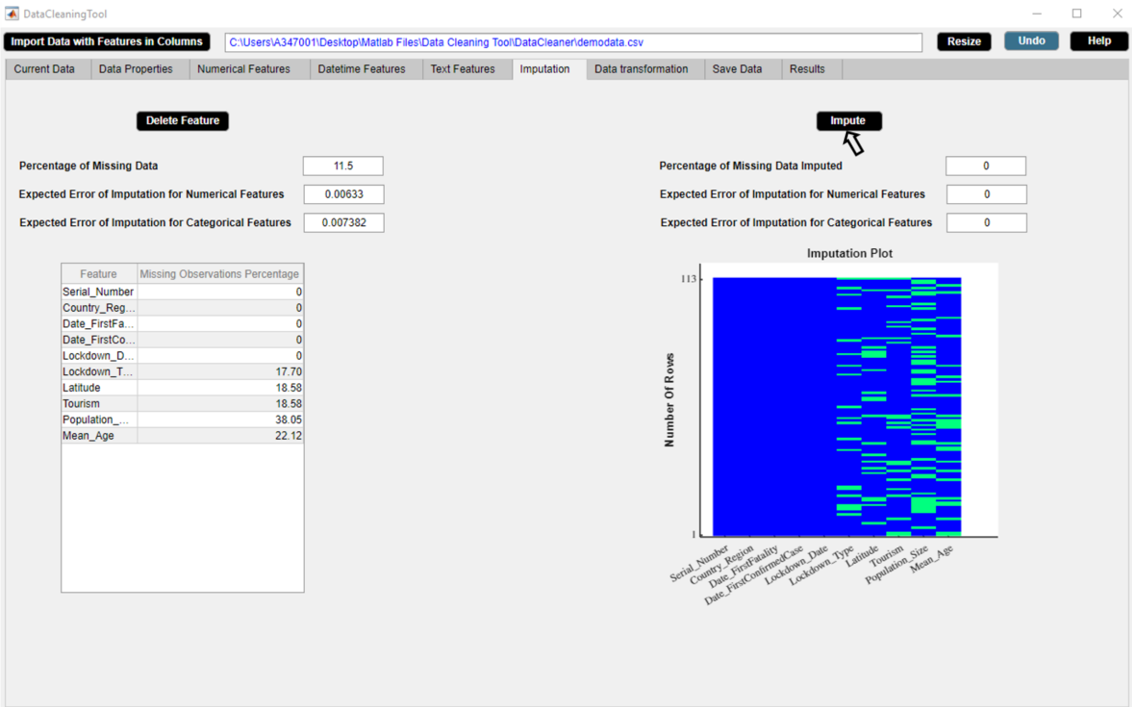


Figure 4.25: Step 1. Click Impute button.

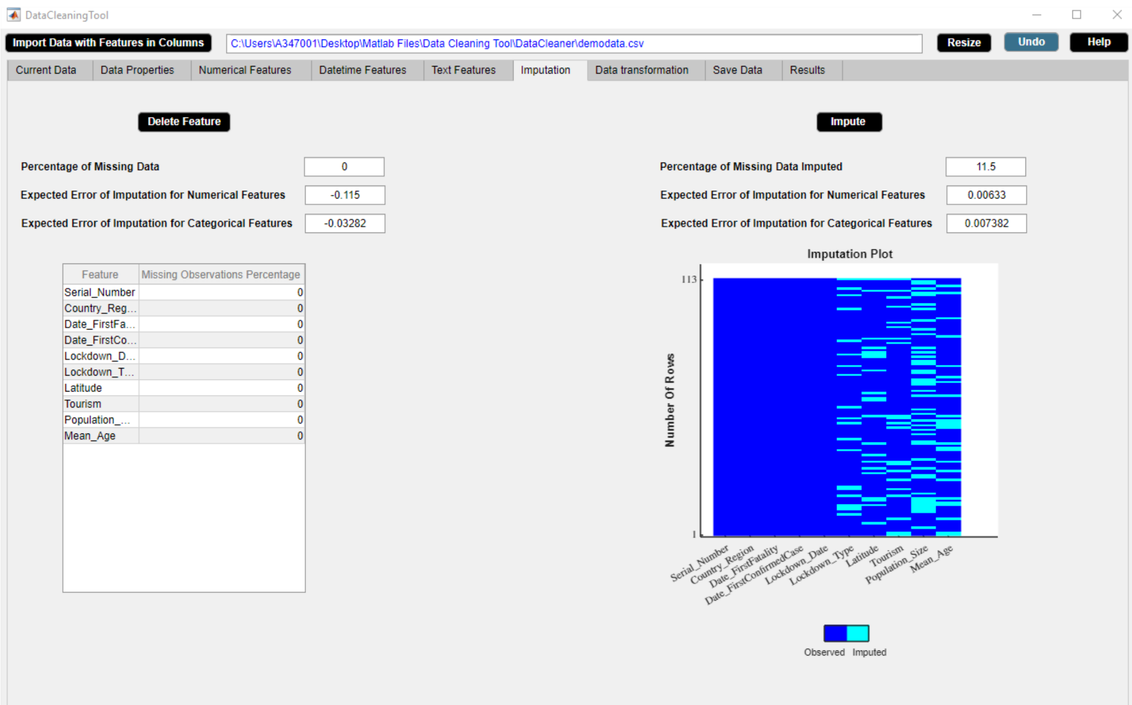


Figure 4.26: Step 2. Check that the missing observations are imputed.

4.3.11 Transform numerical features

We use Transform button to logarithmize 'Population_Size' in the example data. Figures 4.27-4.28 illustrate how to transform numerical features in DataCleaningTool.

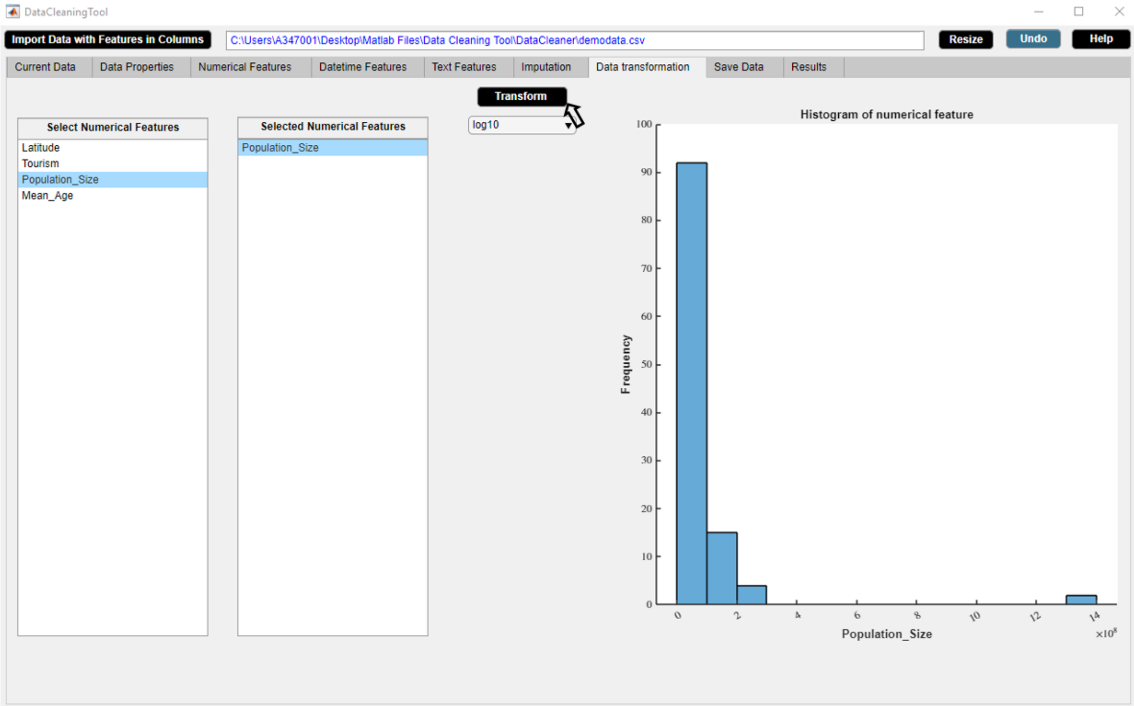


Figure 4.27: Step 1. Select numerical features from Select Numerical Features list box. Click Transform button.

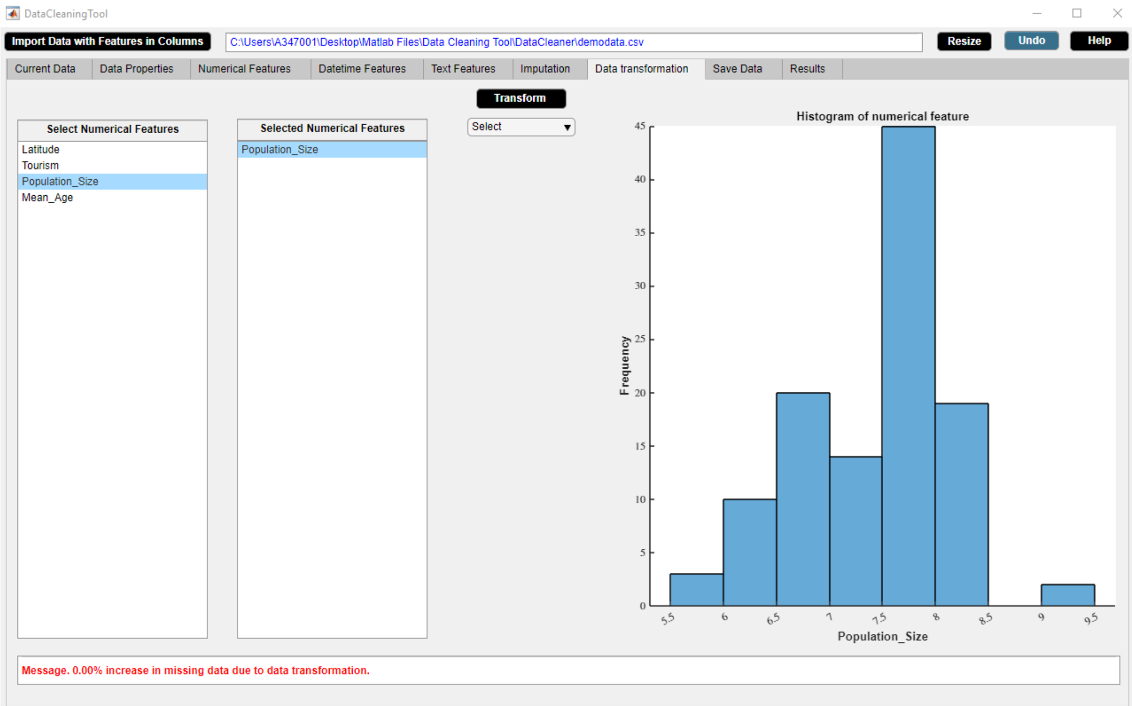


Figure 4.28: Step 2. Check that the numerical feature is transformed by histogram display.

4.3.12 Interactive data visualizations

We wish to sort features in plots according to increasing percentage of missing observations. Figures 4.29-4.30 illustrate how to operate on plots in DataCleaningTool by clicking a button.

Figure 4.29: Step 1. Click Sort Features button.

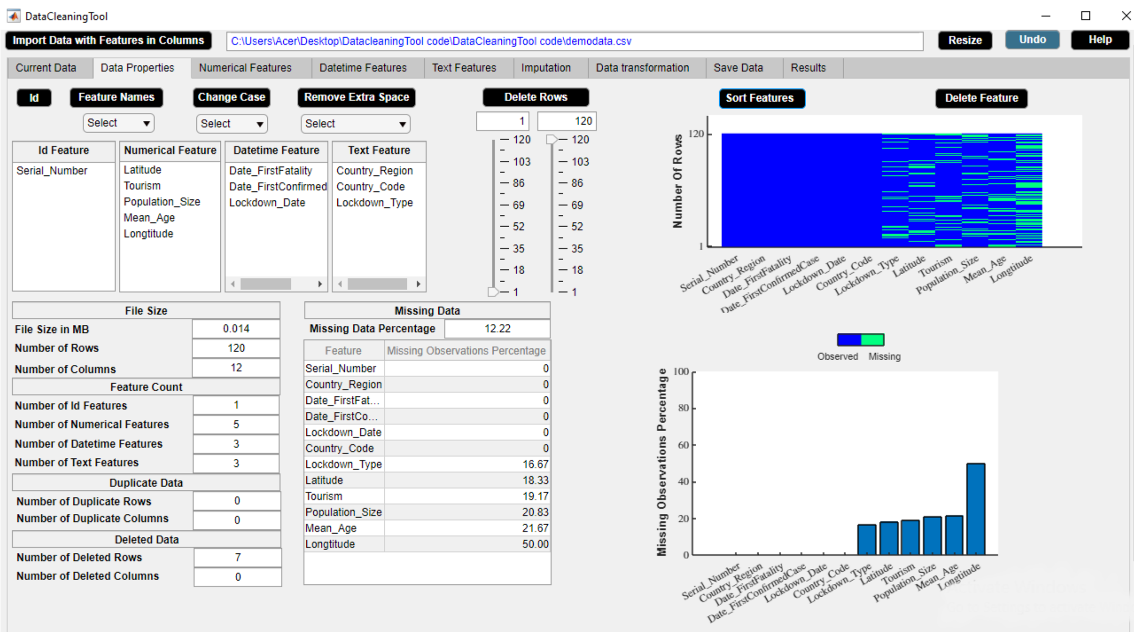


Figure 4.30: Step 2. Check that the plots are sorted by increasing percentage of missing observations.

We wish to delete rows containing irrevalent observations from histogram. Figures 4.31-4.32 illustrate how to manipulate plot in DataCleaningTool by moving a slider.

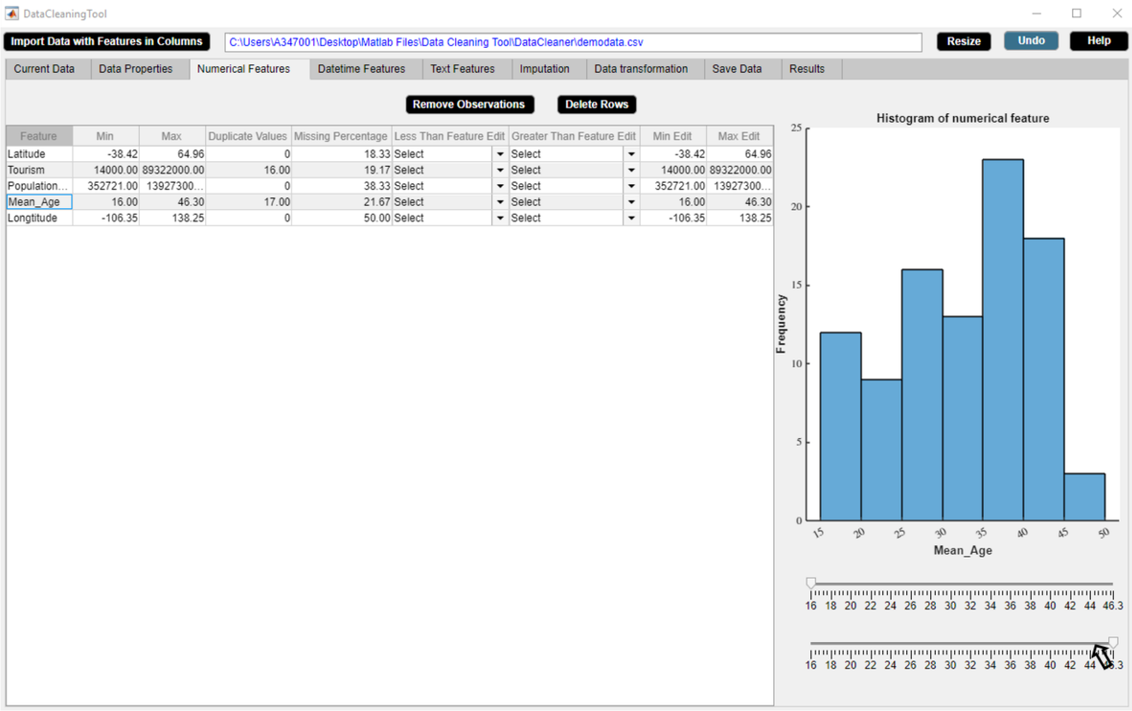


Figure 4.31: Step 1. Select maximum of the selected feature from maximum slider.

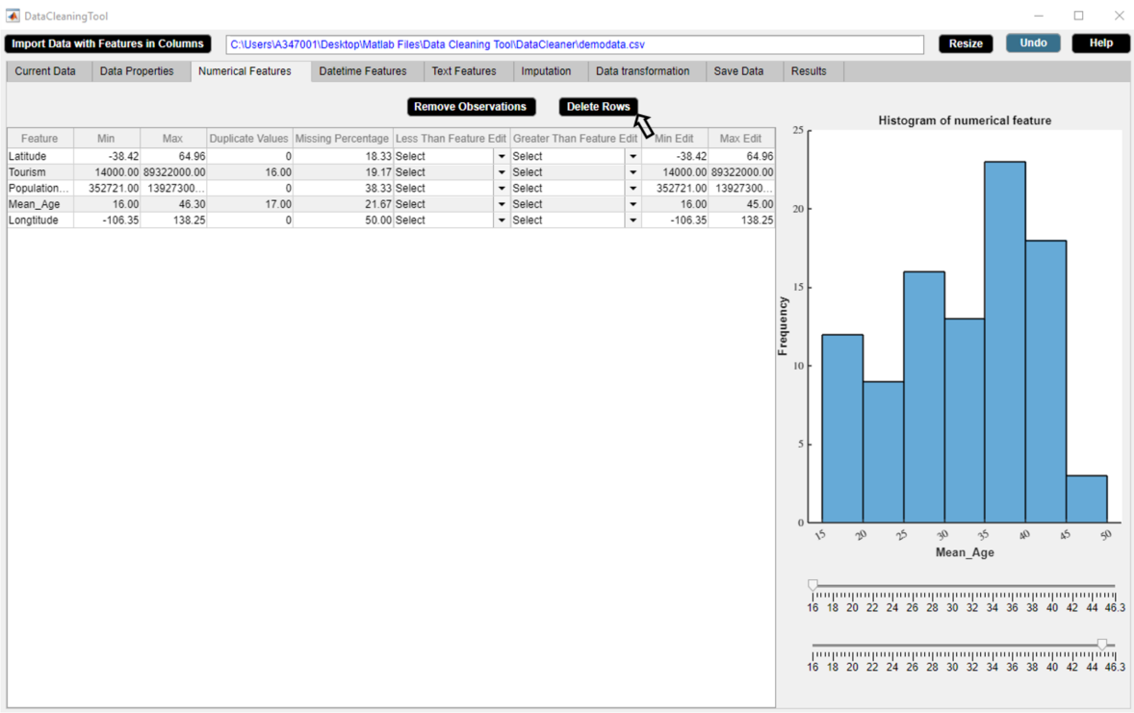


Figure 4.32: Step 2. Check that the maximum of the selected feature is edited in Max Edit box. Click Delete Rows button.

5
Conclusion

Data cleaning is a necessary step in data-driven analytics. Different data cleaning tasks target different data problems. In this thesis, we support the process of data cleaning. To support the study, the main outcome of the thesis work is the development of a user cooperative data cleaning tool. The chapter discusses two aspects of the thesis work. In Section 5.1, the contributions are summarized and in Section 5.2, the future directions of the work are discussed.

5.1 Contributions

DataCleaningTool is a user friendly standalone application that offers multiple data cleaning approaches in one platform. As compared to existing data cleaning tools, DataCleaningTool is designed with the following core competencies.

- The tool is not a black box.
- It is simple to use.
- It assists users in each step of cleaning data.
- It solves data inconsistency.
- It tackles noisy data.
- It performs missing data imputation for both continuous and categorical data at the same time using missForest algorithm.
- It deals with outliers.
- It provides interactive data visualization techniques.
- It is a free and open source software.

5.2 Future Work

Data cleaning involves a wide variety of cleaning tasks to detect and solve data problems and so there are many aspects one can focus on. Although DataCleaningTool tries to fix as many data problems as possible, there remains much room for improvement. Some of the aspects need to be focused are as follows.

- Automated Display of Data and Statistical Information of Data
  - In case of large volume of data, DataCleaningTool runs slow and it takes time to display the whole data. Thus, dealing with high volume data can be a future work. Since DataCleaningTool is a Matlab based application, one can generate a Matlab script to automatically connect to a SQL database, run an SQL query, and perform data cleaning on the imported data.
- Automated Data Type Discovery
  - We can automatically discover three basic data types such as numerical, text and date-time in DataCleaningTool. In future, one can discover further classification of data types such as ordinal and interval in DataCleaningTool.
- Removal of Unwanted Data
  - In DataCleaningTool, we can identify and remove unwanted data such as irrelevant observations which do not fit the specific problem to be solved by the user. Although we calculate the number of duplicate rows and columns in the data, we can not identify

and remove them in DataCleaningTool. In future, the task of identifying and removing duplicates can be implemented in DataCleaningTool.

- Outlier Detection
  - We only consider univariate outlier detection method in DataCleaningTool. Although we examined the performance of different multivariate outlier detection methods such as leverage, local outlier factor and DBSCAN, the methods are not implemented in DataCleaningTool owing to time constraints. A further project can be performed to explore the different multivariate outlier detection methods in DataCleaningTool.
- Missing Data Handling
  - We implement missForest method to impute missing values for mixed type data in DataCleaningTool. We also predict the performance of the missForest imputation method using the normalized root mean squared error for continuous data and the percentage of erroneous categorical entries for categorical data. In our tool, we do not impute datetime values. A further work can be done to implement the task of imputing datetime features in DataCleaningTool.
- Data Transformation
  - Common data transformations such as standardization, normalization, logarithm, exponential, square root and inverse are implemented in DataCleaningTool. There are multiple other mathematical functions that the values of a specific numerical feature can be transformed such that they are most suitable for the algorithm being used. For future work, it can be implemented in DataCleaningTool that the user can choose any mathematical function to transform a numerical feature accordingly.
- Data Visualization
  - We provide various interactive data visualization techniques so that the user can directly operate on the visualization to explore what they want. However, the data visualization techniques used in DataCleaningTool are univariate which helps to understand each feature of the data separately. Therefore, in future multivariate data visualization methods such scatter plot, heatmap and parallel coordinates plot can be implemented in DataCleaningTool for visualizing and analyzing high dimensional data..
- Further development
  - Another issue that is left to explore is the issue of multicollinearity. Multicollinearity is a serious issue in statistical learning models such as regression because it undermines the statistical significance of an independent variable.
  - The primary task of DataCleaningTool is data cleaning. In future, the data cleaning task can be extended to data analysis.

Bibliography

[1] Gali Halevi and Henk Moed. The evolution of big data as a research and scientific topic: Overview of the literature. *Research Trends*, 30:3–6, 01 2012.

[2] Mircea Trifu and Mihaela Laura Ivan. Big data : present and future big data : present and future. 2014.

[3] Openrefine [internet]. openrefine.org. 2020 [cited 7 september 2020]. available from: `https://openrefine.org/`.

[4] Data wrangler [internet]. vis.stanford.edu. 2020 [cited 7 september 2020]. available from: `http://vis.stanford.edu/wrangler/`.

[5] Winpure [internet]. winpure.com. 2020 [cited 7 september 2020]. available from: `https://winpure.com/`.

[6] rhiever/datacleaner [internet]. github. 2020 [cited 7 september 2020]. available from: `https://github.com/rhiever/datacleaner`.

[7] ekstroem/datamaid [internet]. github. 2020 [cited 7 september 2020]. available from: `https://github.com/ekstroem/dataMaid`.

[8] Sas [internet]. documentation.sas.com. 2020 [cited 7 september 2020]. available from: `https://documentation.sas.com/`.

[9] Time series data anomaly detection: A closer look [internet]. anodot. 2020 [cited 7 september 2020]. available from: `https://www.anodot.com/blog/closer-look-time-series-anomaly-detection/`.

[10] Anomaly detection - happiest minds [internet]. solutions. 2020 [cited 7 september 2020]. available from: `https://www.happiestminds.com/solutions/anomaly-detection/`.

[11] Won Kim, Byoung-Ju Choi, Eui Hong, Soo-Kyung Kim, and Doheon Lee. A taxonomy of dirty data. *Data Min. Knowl. Discov.*, 7:81–99, 01 2003.

[12] How to use spell checker with matlab? - matlab answers - matlab central [internet]. in.mathworks.com. 2020 [cited 7 september 2020]. available from: `https://in.mathworks.com/matlabcentral/answers/231219-how-to-use-spell-checker-with-matlab`.

[13] Set command window output display format - matlab format [internet]. mathworks.com. 2020 [cited 7 september 2020]. available from: `https://www.mathworks.com/help/matlab/ref/format.html`.

[14] Daniel J. Stekhoven and Peter Bühlmann. Missforest - non-parametric missing value imputation for mixed-type data. *Bioinformatics*, 28 1:112–8, 2012.

[15] Keigo Kimura and Tetsuya Yoshida. Non-negative matrix factorization with sparse features. pages 324–329, 11 2011.

[16] Joris M. Mooij, Jonas Peters, Dominik Janzing, Jakob Zscheischler, and Bernhard Schölkopf. Distinguishing cause from effect using observational data: methods and benchmarks. *CoRR*, abs/1412.3773, 2014.

[17] Aryana Jackson and Seán Lacey. The discrete fourier transformation for seasonality and anomaly detection on an application to rare data. ahead-of-print, 05 2020.

[18] Z.M. Nopiah, A. Lennie, S. Abdullah, M.Z. Nuawi, A.Z. Nuryazmin, and M.N. Baharin. The use of autocorrelation function in the seasonality analysis for fatigue strain data. *Journal of Asian Scientific Research*, 2(11):782–788, 2012.

[19] David C. Hoaglin and Roy E. Welsch. The hat matrix in regression and anova. 1978.

[20] Vijayakumar Veeramani, Nallam Divya, P. Sarojini, and K. Sonika. Isolation forest and local outlier factor for credit card fraud detection system. 04 2020.

[21] Joseph Dettori and Daniel Norvell. The anatomy of data. *Global Spine Journal*, 8:219256821774699, 01 2018.
[22] Nicholas Matthews. *Measurement, Levels of.* 01 2017.
[23] G. Darlington. *Dummy Variables.* 07 2005.
[24] Norazian Mohamed Noor. Roles of imputation methods for filling the missing values: A review. *Advances in Environmental Biology*, 7:3861–3869, 01 2013.
[25] Shinichi Nakagawa. Chapter 4 missing data : mechanisms , methods , andmessages. 2015.
[26] Donald B. Rubin. Inference and missing data. *Biometrika*, 63(3):581–592, 1976.
[27] Sutthipong Meeyai. Logistic regression with missing data: A comparison of handling methods, and effects of percent missing values. *Journal of Traffic and Logistics Engineering*, 2016.
[28] Leo Breiman. Random Forests. *Machine Learning*, 45(1):5–32, 2001.
[29] Celine Vens. *Random Forest*, pages 1812–1813. Springer New York, New York, NY, 2013.
[30] Adele Cutler, David Cutler, and John Stevens. *Random Forests*, volume 45, pages 157–176. 01 2011.
[31] Chapter 9 - noninvasive fracture characterization based on the classification of sonic wave travel times. In Siddharth Misra, Hao Li, and Jiabo He, editors, *Machine Learning for Subsurface Characterization*, pages 243 – 287. Gulf Professional Publishing, 2020.
[32] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM Comput. Surv.*, 41(3), July 2009.
[33] Markus Goldstein and Seiichi Uchida. A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data. *PLoS ONE*, 11, 2016.
[34] Charu C. Aggarwal. *Outlier Analysis.* Springer Publishing Company, Incorporated, 2nd edition, 2016.
[35] Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng, and Jörg Sander. Lof: Identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, SIGMOD '00, page 93–104, New York, NY, USA, 2000. Association for Computing Machinery.
[36] Auto data car price prediction regression [internet]. kaggle.com. 2020 [cited 17 september 2020]. available from: `https://www.kaggle.com/thorgodofthunder/auto-data-car-price-prediction-regression`.
[37] Shebuti Rayana. ODDS Library. `http://odds.cs.stonybrook.edu`. Stony Brook, NY: Stony Brook University, Department of Computer Science, 2016.
[38] Devosmita Chatterjee. DataCleaningTool. `https://github.com/devosmitachatterjee2018/DataCleaningTool/tree/main/Standalone%20Desktop%20App`, 2021.
[39] Devosmita Chatterjee. DataCleaningTool. `https://github.com/devosmitachatterjee2018/DataCleaningTool`, 2021.
[40] Covid-19 useful features by country [internet]. kaggle.com. 2020 [cited 9 september 2020]. available from: `https://www.kaggle.com/ishivinal/covid19-useful-features-by-country`.
[41] Standard score [internet]. en.wikipedia.org. 2020 [cited 14 september 2020]. available from: `https://en.wikipedia.org/wiki/Standard_score`.

A

Appendix A: Performance Analysis of MissForest Method

Table A.1: The table represents the comparison of NRSME values for datasets of different sizes with different percentages of missing values. The empty cells represent that computation is not feasible due to high missing data percentage.

NRSME	**Percentage of missing data**								
Nature of data	10%	20%	30%	40%	50%	60%	70%	80%	90%
Overdetermined $n = 120$, $p = 15$ $n = 8p$	0.1297	0.1957	0.2629	0.3307	0.4006	0.4886	0.5919	0.7462	0.9394
Overdetermined $n = 48$, $p = 15$ $n = 2p$	0.1588	0.2558	0.3692	0.4427	0.5371	0.6637	0.8100	0.9751	-
Equal $n = 24$, $p = 15$ $n = p$	0.2076	0.3434	0.5057	0.6191	0.7450	0.8105	0.9387	-	-
Underdetermined $n = 12$, $p = 15$ $n = 0.5p$	0.2319	0.4173	0.5012	0.6034	0.7781	-	-	-	-

Table A.2: The table represents the comparison of PEC values for datasets of different sizes with different percentages of missing values. The empty cells represent that computation is not feasible due to high missing data percentage.

PEC	**Percentage of missing data**								
Nature of data	10%	20%	30%	40%	50%	60%	70%	80%	90%
Overdetermined $n = 72$, $p = 9$ $n = 8p$	0.0278	0.0552	0.0830	0.1086	0.1315	0.1802	0.2228	0.2549	0.3225
Overdetermined $n = 18$, $p = 9$ $n = 2p$	0.0198	0.0444	0.0716	0.0975	0.1160	0.1593	0.1556	0.2667	-
Equal $n = 9$, $p = 9$ $n = p$	0.0346	0.0642	0.0667	0.1210	0.1605	0.1951	0.2272	-	-
Underdetermined $n = 4$, $p = 9$ $n = 0.5p$	0.0389	0.0722	0.0833	0.1278	-	-	-	-	-

Table A.3: The table represents the comparison of NRSME values for continuous datasets of different sizes with different percentages of missing values. The empty cells represent that computation is not feasible due to high missing data percentage.

NRSME	**Percentage of missing data**								
Nature of data	10%	20%	30%	40%	50%	60%	70%	80%	90%
Overdetermined $n = 192$, $p = 24$ $n = 8p$	0.1760	0.2400	0.3119	0.3773	0.4339	0.5055	0.6097	0.7026	0.9060
Overdetermined $n = 48$, $p = 24$ $n = 2p$	0.1662	0.2759	0.3194	0.4057	0.5084	0.5952	0.7122	0.8802	1.0429
Equal $n = 24$, $p = 24$ $n = p$	0.1920	0.1879	0.3350	0.2766	0.5758	0.6796	0.9322	0.9764	-
Underdetermined $n = 12$, $p = 24$ $n = 0.5p$	0.1034	0.3924	0.5625	0.5808	0.8146	0.7004	0.8986	-	-

Table A.4: The table represents the comparison of PEC values for datasets of different sizes with different percentages of missing values. The empty cells represent that computation is not feasible due to high missing data percentage.

PEC	**Percentage of missing data**								
Nature of data	10%	20%	30%	40%	50%	60%	70%	80%	90%
Overdetermined $n = 192$, $p = 24$ $n = 8p$	0.0192	0.0448	0.0638	0.1008	0.1311	0.1539	0.2134	0.2567	0.3338
Overdetermined $n = 48$, $p = 24$ $n = 2p$	0.0231	0.0449	0.0769	0.0870	0.1222	0.1620	0.2019	0.2444	0.2972
Equal $n = 24$, $p = 24$ $n = p$	0.0194	0.0398	0.0722	0.0806	0.1426	0.1769	0.2454	0.3120	-
Underdetermined $n = 12$, $p = 24$ $n = 0.5p$	0.0278	0.0481	0.0815	0.1333	0.1500	0.2704	0.2352	-	-

B

Appendix B: Complete Demo

Overview

Presently, large amount of data generated by organizations drives its business decisions. The data is usually inconsistent, inaccurate and incomplete. Poor data quality may lead to incorrect decisions for the organizations and hence, negatively affect organizations. Thus, high quality data is of utmost priority to use the data effectively. Data cleaning is the ultimate way to solve the data quality issues. But, data cleaning is really a time consuming task. Thus, tools which can help with the task are needed. This demands data cleaning tools for systematically examining data for errors and automatically cleaning them using algorithms. These data cleaning tools help organizations save time and increase their efficiency.
DataCleaningTool is a user friendly, free and open source data cleaning standalone application developed to achieve the task of data cleaning in a cooperative way. This application is able to identify the potential data problems and report results and recommendations such that users can clean data effectively with its assistance. The major data problems encountered by DataCleaningTool and the possible approaches to fix them are as follows.

Incorrect data type

- Example: Numerical instead of string entries.
- Possible Approach: Set data type constraint.

Inconsistent feature names or columns

- Example: Feature names or columns have inconsistent capitalizations.
- Possible Approach: Use uppercase or lowercase characters.

Typographical errors

- Example: Extra white spaces.
- Possible Approach: Remove extra white spaces.

Nonsensical data

- Example: Age = -1.
- Possible Approach: Set range constraint to variable - Age ≥ 0.

Extrapolation errors

- Example: A model of glacial retreat: $V = 100 - 2t$ where V = volume of ice, t = time variable, and $t = 0$ AD. If we extrapolate to earlier than $t = 0$, then ice volume becomes bigger. Mathematically, we can extrapolate back in time but then the ice volume of the glacier would exceed the total volume of the earth which is absurd.
- Possible Approach: Set range constraint to variable - $t \geq 0$.

Truncation error (Volvo)

- Example: Difference between the actual value (2.99792458×10^8) and the truncated value up to two decimals (2.99×10^8).
- Possible Approach: Use long format [13].

Time stamp errors (Volvo)

- Example: The first failure time can show time prior to when the electric vehicles were produced if the vehicle clock has not been correctly set.
- Possible Approach: Set cross field validation constraint to variable - first failure time of a vehicle > time when the vehicle was produced.

Fault code count (Volvo)

- Example: Fault codes stored by the on-board computer diagnostic system notify about a problem found in the car. Sometimes although an issue is notified, failure count = 0.
- Possible Approach: Set range constraint to variable - Failure count > 0.

Missing data

- Example: NaN or ‘ ’.
- Possible Approach: Imputation using MissForest method. [14].

Outliers

- Example: Fraudulent credit card transactions.
- Possible Approach: Z-score [41].

App Installation

DataCleaningTool is a standalone application that can run on Windows platform. DataCleaningTool is a standalone application created from Matlab functions so that it can be used to run Matlab compiled program on computers that do not have Matlab installed. The Matlab Compiler Runtime enables to run standalone application compiled within Matlab. The DataCleaningTool app installation package is already provided with Matlab Compiler Runtime. The following steps show how to install DataCleaningTool application.

- Open app installation folder 'Standalone Desktop App'.
- There are three folders 'for_redistribution', 'for_redistribution_files_only', 'for_testing' present in the folder 'Standalone Desktop App'. Open 'for_redistribution' folder.
- Install 'DataCleaningTool.exe' file from 'for_redistribution' folder.
- Click Finish.

Getting Started

DataCleaningTool is a data cleaning application which consists of multiple widgets and buttons. DataCleaningTool is shown in figure B.1. The properties of DataCleaningTool are

- DataCleaningTool always opens in a full screen mode. The application can be resized to a reduced size.
- Each widget provides specific statistical information about the data.
- Each button aims to clean data by resolving inconsistencies, smoothing noisy data, identifying outliers, removing outliers or filling in missing observations.
- All buttons are black in color. Pressing a button each time changes the button color from black to grey color and then again to black. The button remains grey in color until it completes its specific task and all widgets gets updated accordingly.
- Pressing any button turns the Undo button to blue color. The Undo button remains blue in color until last activity can be undone.
- Sliders and their corresponding edit boxes are interdependable.
- User can find help in using DataCleaningTool.

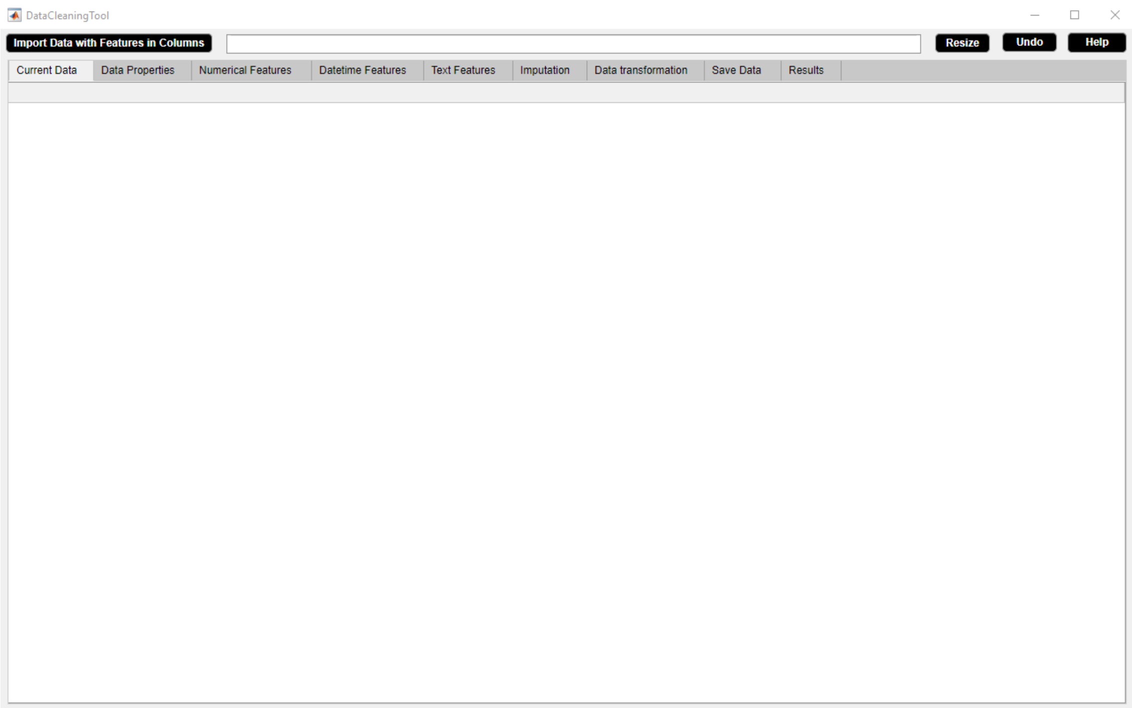


Figure B.1: DataCleaningTool.

We demonstrate the DataCleaningTool using an example dataset ‘demodata.csv’. The example dataset is obtained by tweaking the coronavirus dataset [40]. The example dataset is of dimension 127×12. The example dataset consists of the following features.

1. Serial_Number: Unique identifier to a country.
2. Country_Region: Name of the country.
3. Population_Size: Size of the population of the country.
4. tourism: Number of international arrivals in the country.
5. Date_FirstFatality: Date of the first fatality in the country.
6. Date_FirstConfirmedCase: Date of the first confirmed case in the country.
7. Latitude: Geographic coordinate of the country.
8. Longtitude: Geographic coordinate of the country.
9. mean_Age: Mean age of the population of the country.
10. Lockdown_Date: Date of the lockdown in the country.
11. Lockdown_Type: Level of the lockdown (full or partial) in the country.

12. Country_Code: Geographical code representing the country.

Using the example dataset, we will show the steps how to clean data using the DataCleaningTool.

B.1 Import Data with Features in Columns Button

Loads data from comma-separated (.csv), Excel (.xlsx), tab-delimited (.txt), data (.dat) files and then reads the data into table.

Application

- Reduce truncation errors upto 15 decimal places using long decimal format.

Example

Step 1: Click **Import Data with Features in Columns** button.

Step 2: **Import Data with Features in Columns** button in use turns grey in color and an open dialog box appears. Browse for an input file.

Step 3: **Import Data with Features in Columns** button returns back to its original color once it completes its task. The full path of the selected file is displayed and the file is loaded.

We use **Import Data with Features in Columns** button to load the example data 'demodata.csv'. Figures B.2-B.4 illustrate how to use **Import Data with Features in Columns** button.

Figure B.2: Step 1. Import Data with Features in Columns Button

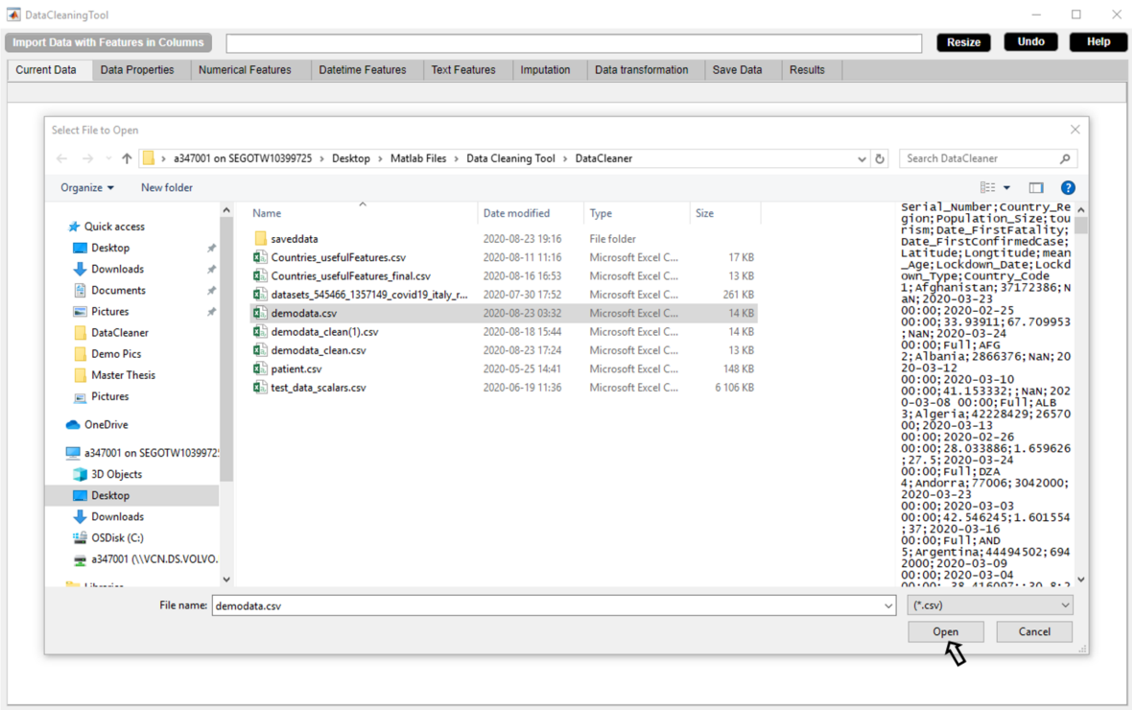


Figure B.3: Step 2. Import Data with Features in Columns Button

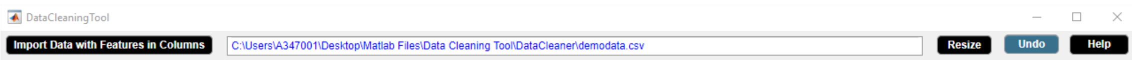


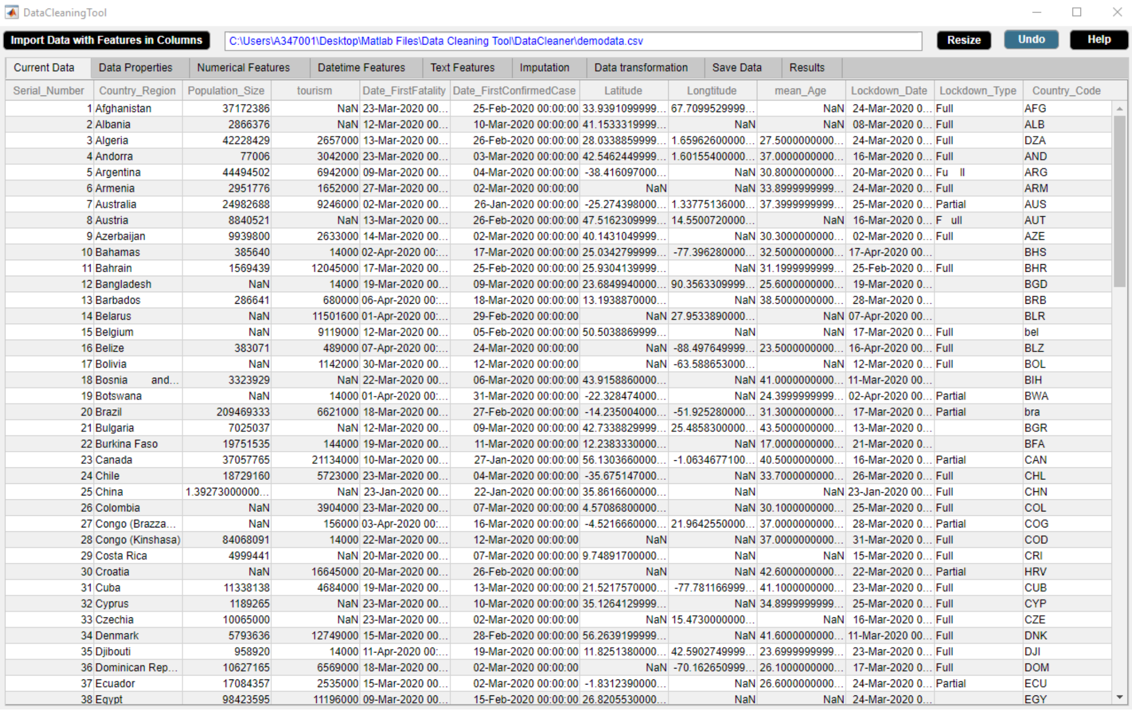
Figure B.4: Step 3. Import Data with Features in Columns Button

Data Cleaning Widgets

B.2 Current Data Widget

The **Current Data** widget displays the input data in table format. The **Current Data** widget is shown in figure B.5. The properties of the Current Data widget are as follows.

- The widget shows the presence of round off errors in numerical features.
- The widget shows the presence of inconsistent capitalization of feature names and features.
- The widget shows the existence of extra whitespaces in text features.
- Default datetime format is ‘dd-MMM-yyyy HH:mm:ss’ for datetime features.
- The widget shows the presence of missing numerical observations represented by NaNs.
- The widget shows the presence of missing datetime observations represented by NaTs.
- The widget shows the presence of missing text observations represented by empty strings.
- The updated table can be visualized after each activity since the widget gets updated accordingly.



Serial_Number	Country_Region	Population_Size	tourism	Date_FirstFatality	Date_FirstConfirmedCase	Latitude	Longtitude	mean_Age	Lockdown_Date	Lockdown_Type	Country_Code
1	Afghanistan	37172386	NaN	23-Mar-2020 00...	25-Feb-2020 00:00:00	33.9391099999...	67.7099529999...	NaN	24-Mar-2020 0...	Full	AFG
2	Albania	2866376	NaN	12-Mar-2020 00...	10-Mar-2020 00:00:00	41.1533319999...	NaN	NaN	08-Mar-2020 0...	Full	ALB
3	Algeria	42228429	2657000	13-Mar-2020 00...	26-Feb-2020 00:00:00	28.0338859999...	1.65962600000...	27.5000000000...	24-Mar-2020 0...	Full	DZA
4	Andorra	77006	3042000	23-Mar-2020 00...	03-Mar-2020 00:00:00	42.5462449999...	1.60155400000...	37.0000000000...	16-Mar-2020 0...	Full	AND
5	Argentina	44494502	6942000	09-Mar-2020 00...	04-Mar-2020 00:00:00	-38.416097000...	NaN	30.8000000000...	20-Mar-2020 0...	Fu ll	ARG
6	Armenia	2951776	1652000	27-Mar-2020 00...	02-Mar-2020 00:00:00	NaN	NaN	33.8999999999...	24-Mar-2020 0...	Full	ARM
7	Australia	24982688	9246000	02-Mar-2020 00...	26-Jan-2020 00:00:00	-25.274398000...	1.33775136000...	37.3999999999...	25-Mar-2020 0...	Partial	AUS
8	Austria	8840521	NaN	13-Mar-2020 00...	26-Feb-2020 00:00:00	47.5162309999...	14.5500720000...	NaN	16-Mar-2020 0...	F ull	AUT
9	Azerbaijan	9939800	2633000	14-Mar-2020 00...	02-Mar-2020 00:00:00	40.1431049999...	NaN	30.3000000000...	02-Mar-2020 0...	Full	AZE
10	Bahamas	385640	14000	02-Apr-2020 00:...	17-Mar-2020 00:00:00	25.0342799999...	-77.396280000...	32.5000000000...	17-Apr-2020 00...		BHS
11	Bahrain	1569439	12045000	17-Mar-2020 00...	25-Feb-2020 00:00:00	25.9304139999...	NaN	31.1999999999...	25-Feb-2020 0...	Full	BHR
12	Bangladesh	NaN	14000	19-Mar-2020 00...	09-Mar-2020 00:00:00	23.6849940000...	90.3563309999...	25.6000000000...	19-Mar-2020 0...		BGD
13	Barbados	286641	680000	06-Apr-2020 00:...	18-Mar-2020 00:00:00	13.1938870000...	NaN	38.5000000000...	28-Mar-2020 0...		BRB
14	Belarus	NaN	11501600	01-Apr-2020 00:...	29-Feb-2020 00:00:00	NaN	27.9533890000...	NaN	07-Apr-2020 00...		BLR
15	Belgium	NaN	9119000	12-Mar-2020 00...	05-Feb-2020 00:00:00	50.5038869999...	NaN	NaN	17-Mar-2020 0...	Full	bel
16	Belize	383071	489000	07-Apr-2020 00:...	24-Mar-2020 00:00:00	NaN	-88.497649999...	23.5000000000...	16-Apr-2020 00...	Full	BLZ
17	Bolivia	NaN	1142000	30-Mar-2020 00...	12-Mar-2020 00:00:00	NaN	-63.588653000...	NaN	12-Mar-2020 0...	Full	BOL
18	Bosnia and...	3323929	NaN	22-Mar-2020 00...	06-Mar-2020 00:00:00	43.9158860000...	NaN	41.0000000000...	11-Mar-2020 00...		BIH
19	Botswana	NaN	14000	01-Apr-2020 00:...	31-Mar-2020 00:00:00	-22.328474000...	NaN	24.3999999999...	02-Apr-2020 00...	Partial	BWA
20	Brazil	209469333	6621000	18-Mar-2020 00...	27-Feb-2020 00:00:00	-14.235004000...	-51.925280000...	31.3000000000...	17-Mar-2020 0...	Partial	bra
21	Bulgaria	7025037	NaN	12-Mar-2020 00...	09-Mar-2020 00:00:00	42.7338829999...	25.4858300000...	43.5000000000...	13-Mar-2020 0...		BGR
22	Burkina Faso	19751535	144000	19-Mar-2020 00...	11-Mar-2020 00:00:00	12.2383330000...	NaN	17.0000000000...	21-Mar-2020 0...		BFA
23	Canada	37057765	21134000	10-Mar-2020 00...	27-Jan-2020 00:00:00	56.1303660000...	-1.0634677100...	40.5000000000...	16-Mar-2020 0...	Partial	CAN
24	Chile	18729160	5723000	23-Mar-2020 00...	04-Mar-2020 00:00:00	-35.675147000...	NaN	33.7000000000...	26-Mar-2020 0...	Full	CHL
25	China	1.39273000000...	NaN	23-Jan-2020 00...	22-Jan-2020 00:00:00	35.8616600000...	NaN	NaN	23-Jan-2020 00...	Full	CHN
26	Colombia	NaN	3904000	23-Mar-2020 00...	07-Mar-2020 00:00:00	4.57086800000...	NaN	30.1000000000...	25-Mar-2020 0...	Full	COL
27	Congo (Brazza...	NaN	156000	03-Apr-2020 00:...	16-Mar-2020 00:00:00	-4.5216660000...	21.9642550000...	37.0000000000...	28-Mar-2020 0...	Partial	COG
28	Congo (Kinshasa)	84068091	14000	22-Mar-2020 00...	12-Mar-2020 00:00:00	NaN	NaN	37.0000000000...	31-Mar-2020 0...	Full	COD
29	Costa Rica	4999441	NaN	20-Mar-2020 00...	07-Mar-2020 00:00:00	9.74891700000...	NaN	NaN	15-Mar-2020 0...	Full	CRI
30	Croatia	NaN	16645000	20-Mar-2020 00...	26-Feb-2020 00:00:00	NaN	NaN	42.6000000000...	22-Mar-2020 0...	Partial	HRV
31	Cuba	11338138	4684000	19-Mar-2020 00...	13-Mar-2020 00:00:00	21.5217570000...	-77.781166999...	41.1000000000...	23-Mar-2020 0...	Full	CUB
32	Cyprus	1189265	NaN	23-Mar-2020 00...	10-Mar-2020 00:00:00	35.1264129999...	NaN	34.8999999999...	25-Mar-2020 0...	Full	CYP
33	Czechia	10065000	NaN	23-Mar-2020 00...	02-Mar-2020 00:00:00	NaN	15.4730000000...	NaN	16-Mar-2020 0...	Full	CZE
34	Denmark	5793636	12749000	15-Mar-2020 00...	28-Feb-2020 00:00:00	56.2639199999...	NaN	41.6000000000...	11-Mar-2020 00...	Full	DNK
35	Djibouti	958920	14000	11-Apr-2020 00:...	19-Mar-2020 00:00:00	11.8251380000...	42.5902749999...	23.6999999999...	23-Mar-2020 0...	Full	DJI
36	Dominican Rep...	10627165	6569000	18-Mar-2020 00...	02-Mar-2020 00:00:00	NaN	-70.162650999...	26.1000000000...	17-Mar-2020 0...	Full	DOM
37	Ecuador	17084357	2535000	15-Mar-2020 00...	02-Mar-2020 00:00:00	-1.8312390000...	NaN	26.6000000000...	24-Mar-2020 0...	Partial	ECU
38	Egypt	98423595	11196000	09-Mar-2020 00...	15-Feb-2020 00:00:00	26.8205530000...	NaN	NaN	24-Mar-2020 0...		EGY

Figure B.5: Current Data Widget.

B.3 Data Properties Widget

The Data Properties widget displays several statistical aspects of the data. The Data Properties widget is shown in figure B.6. The properties of the Data Properties widget are as follows.

- The widget automatically discovers the datatypes of features of the input data set and shows the numerical features, the datetime features and the text features separately.
- The widget summarizes the characteristics of a data set such as file size in megabytes, number of rows and columns, number of id, numerical, datetime and text features, number of duplicate rows and columns, and number of deleted rows and columns.
- The widget shows the percentage of missing observations in the data set and the percentage of missing observations in each feature. The widget presents two visual methods for missing data - the missingness plot and the missing observations percentage plot. The missingness plot indicates the missing value occurence in the data. The missing observations percentage plot indicates the percentage of missing observations in each feature. This study of missing data helps to determine the missing data mechanism and hence choose strategies like listwise deletion, pairwise deletion, dropping features, imputation which can be applied to handle missing data so that they can be used for analysis and modelling.
- The information in the widget gets updated after each activity.

Figure B.6: Data Properties Widget.

B.3.1 Id Button

Separates id features from numerical or datetime or text features. Here id feature represents a unique identifier field in the data.

Application

- Avoid overfitting problem which occurs due to a unique identifier among features.

Example

Step 1: Select a feature from **Numerical Feature** or **Datetime Feature** or **Text Feature** list box in the **Data Properties** widget.
Step 2: Click **Id** button.
Step 3: **Id** button in use turns grey in color.
Step 4: **Id** button returns back to its original color once it completes its task.

In the example data, Serial_Number represents unique identifier to a country. We use **Id** button to seperate id feature 'Serial_Number' from numerical features. Figures B.7-B.10 illustrate how to use **Id** button.

Figure B.7: Step 1. Id Button

Figure B.8: Step 2. Id Button

Figure B.9: Step 3. Id Button

Figure B.10: Step 4. Id Button

B.3.2 Feature Names Button

Changes letter case of all feature names to one of the cases - lower case or upper case or capitalized case.

Application

- Fix structural errors such as unify inconsistent capitalization of feature names.

Example

Step 1: Check if there is any inconsistency in feature names capitalization.
Step 2: Select case from **Feature Names** dropdown menu.
Step 3: Click **Feature Names** button.
Step 4: **Feature Names** button in use turns grey in color.
Step 5: **Feature Names** button returns back to its original color once it completes its task.

In the example data, the feature names 'Serial_Number', 'Country_Region', 'Population_Size', 'tourism', 'Date_F- irstFatality', 'Date_FirstConfirmedCase', 'Latitude', 'Longtitude', 'mean_Age', 'Lockdown_Date', 'Lockdown_T- ype', and 'Country_Code' have inconsistent capitalization. We use **Feature Names** button to capitalize first letter of each feature name so as to unify inconsistent capitalization of feature names. Figures B.11-B.15 illustrate how to use **Feature Names** button.

Figure B.11: Step 1. Feature Names Button

Figure B.12: Step 2. Feature Names Button

Figure B.13: Step 3. Feature Names Button

Figure B.14: Step 4. Feature Names Button

Figure B.15: Step 5. Feature Names Button

B.3.3 Change Case Button

Change letter case of a feature to one of the cases- lower case or upper case or capitalized case.
Application

- Fix structural errors such as unify inconsistent capitalization of a feature column.

Example

Step 1: Check if there is any inconsistency in feature capitalization in the **Current Data** widget.
Step 2: Select case from **Change Case** dropdown menu.
Step 3: Select the inconsistent feature from **Numerical Feature** or **Datetime Feature** or **Text Feature** list box in the **Data Properties** widget.
Step 4: Click **Change Case** button.
Step 5: **Change Case** button in use turns grey in color.
Step 6: **Change Case** button returns back to its original color once it completes its task.
Step 7: Check the change in **Current Data** widget.

In the example data, the feature column ‘Country_Code’ has inconsistent capitalization. The whole feature column ‘Country_Code’ is in upper case except fifteenth observation ‘bel’ and twenth observation ‘bra’. We use **Change Case** button to change the whole column to upper case so as to unify inconsistent capitalization of the feature. Figures B.16-B.22 illustrate how to use **Change Case** button.

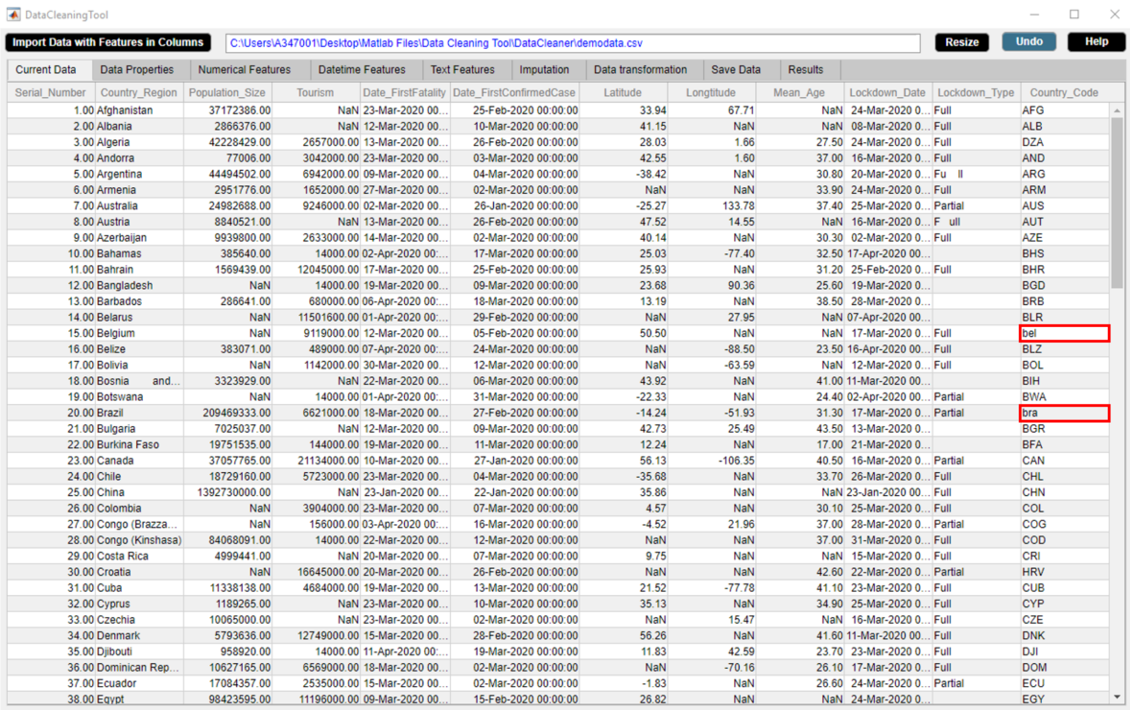


Figure B.16: Step 1. Change Case Button

Figure B.17: Step 2. Change Case Button

Figure B.18: Step 3. Change Case Button

Figure B.19: Step 4. Change Case Button

Figure B.20: Step 5. Change Case Button

Figure B.21: Step 6. Change Case Button

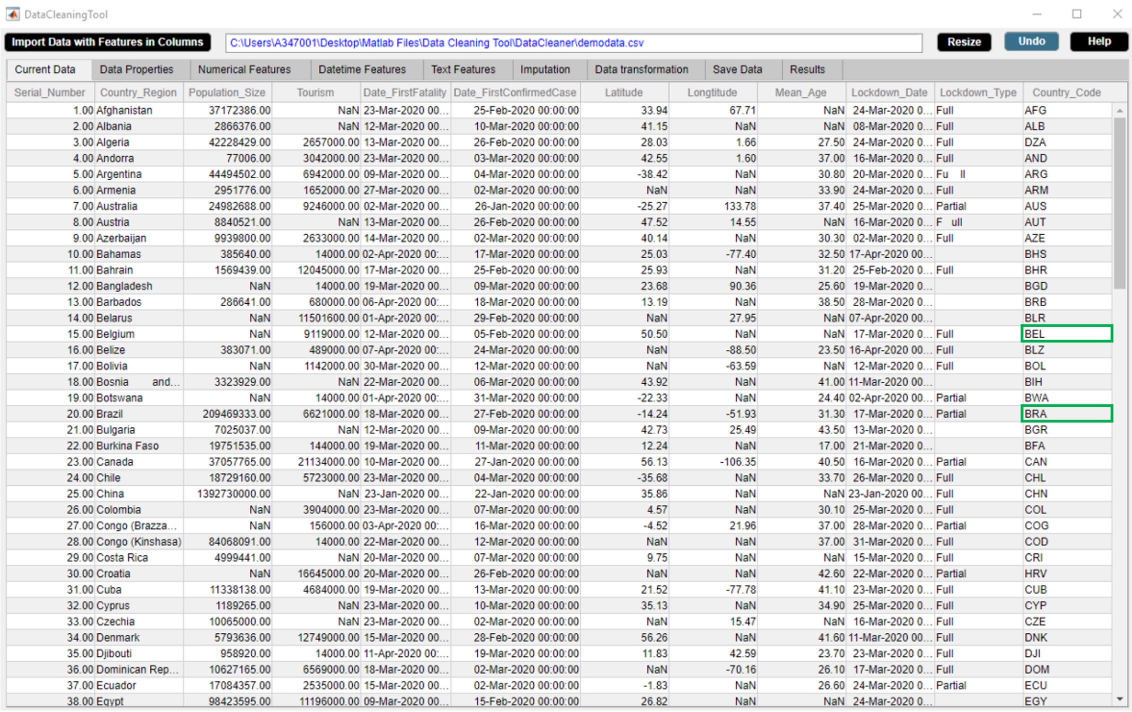


Figure B.22: Step 7. Change Case Button

B.3.4 Remove Extra Space Button

Removes either all spaces or to only one whitespace in a string of a feature.
Application

- Fix structural errors such as typographical errors.

Example

Step 1: Check if there is any extra space in a feature in the **Current Data** widget.
Step 2: Select any one option from **Remove Extra Space** dropdown menu.
Step 3: Select the feature from **Numerical Features** or **Datetime Features** or **Text Features** list box in the **Data Properties** widget.
Step 4: Click **Remove Extra Space** button.
Step 5: **Remove Extra Space** button in use turns grey in color.
Step 6: **Remove Extra Space** button returns back to its original color once it completes its task.
Step 7: Check the change in **Current Data** widget.

In the example data, the feature 'Lockdown_type' is either 'Full' or 'Partial'. The fifth and eighth observations in feature column 'Country_Code' are 'Fu ll' and 'F ull'. We use **Remove Extra Space** button to remove all spaces in the whole column. Figures B.23-B.29 illustrate how to use **Remove Extra Space** button.

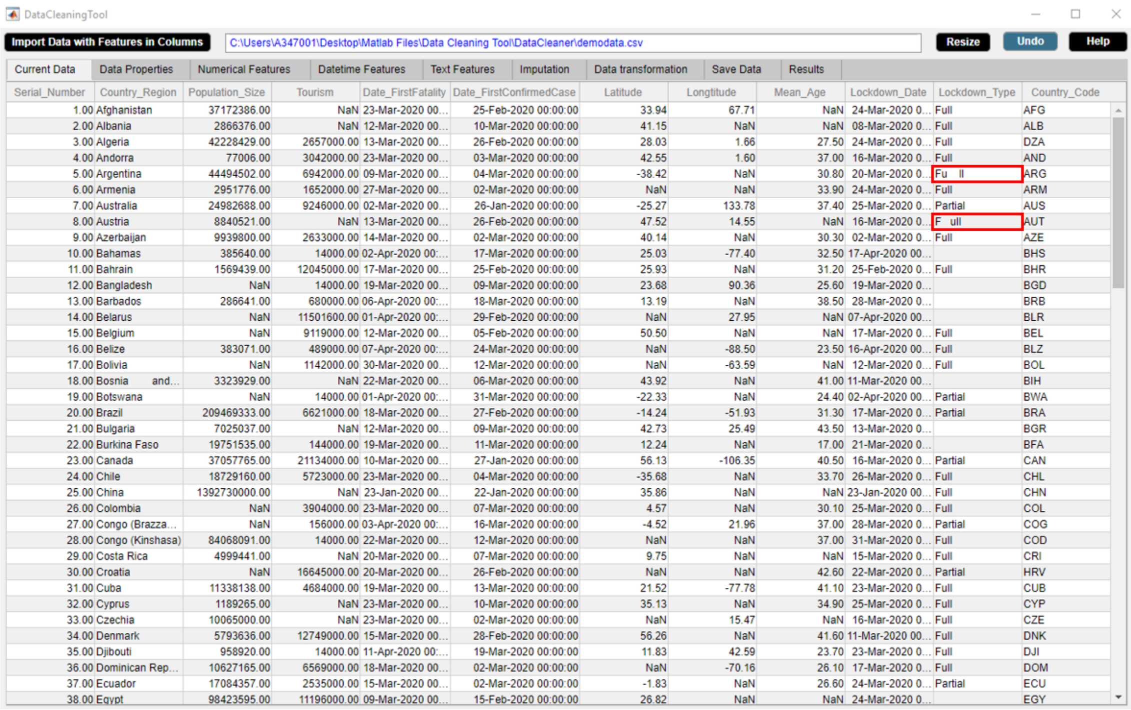


Figure B.23: Step 1. Remove Extra Space Button

Figure B.24: Step 2. Remove Extra Space Button

Figure B.25: Step 3. Remove Extra Space Button

Figure B.26: Step 4. Remove Extra Space Button

Figure B.27: Step 5. Remove Extra Space Button

Figure B.28: Step 6. Remove Extra Space Button

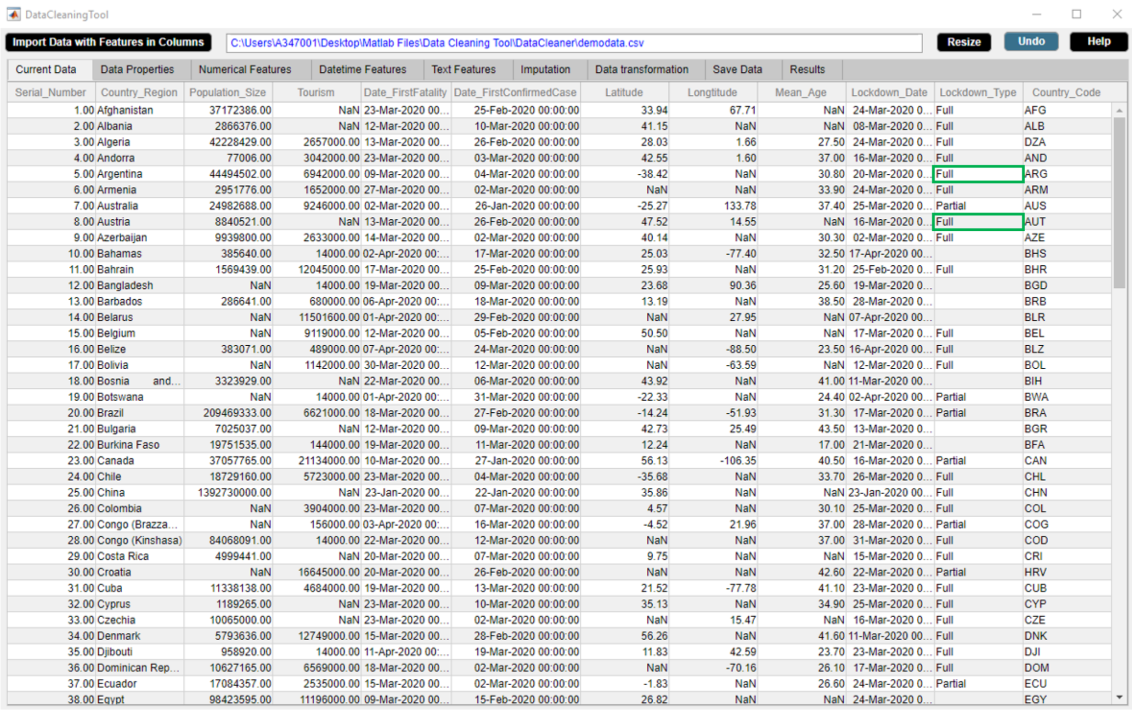


Figure B.29: Step 7. Remove Extra Space Button

Again, the eighteenth observation of the feature 'Country_region' is 'Bosnia   and Herzegovina'. We use **Remove Extra Space** button to remove to single white space in the whole column. Figures B.30-B.36 illustrate how to use **Remove Extra Space** button to remove to single white space.

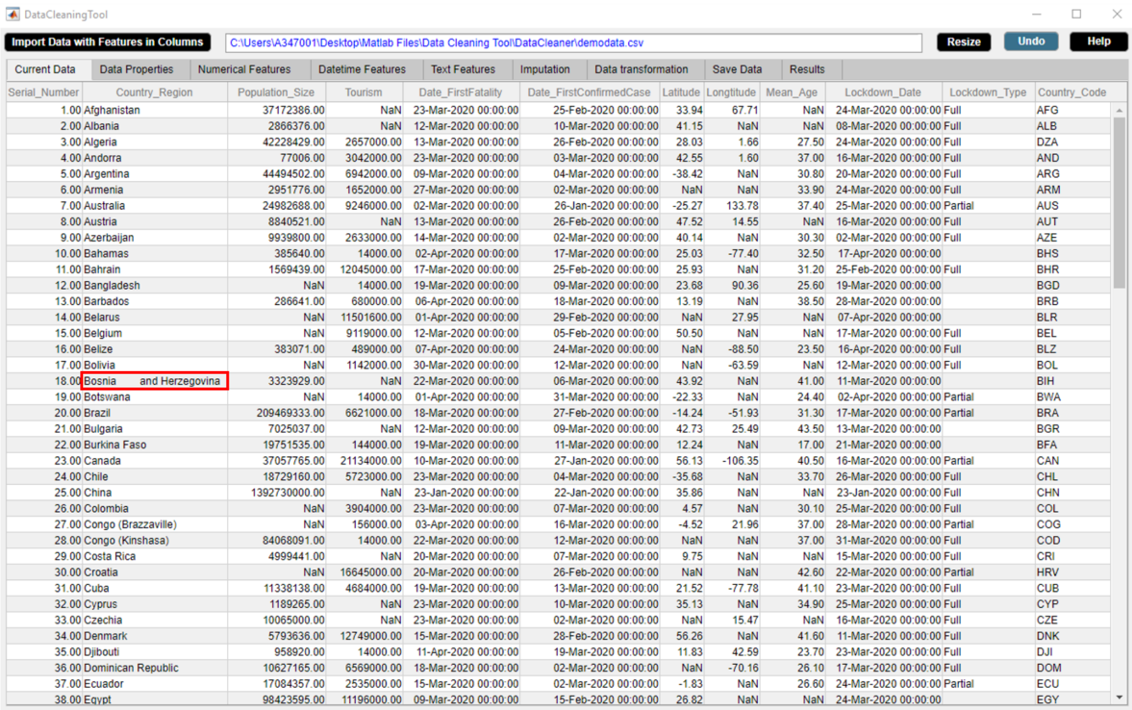


Figure B.30: Step 1. Remove Extra Space Button

Figure B.31: Step 2. Remove Extra Space Button

Figure B.32: Step 3. Remove Extra Space Button

Figure B.33: Step 4. Remove Extra Space Button

Figure B.34: Step 5. Remove Extra Space Button

Figure B.35: Step 6. Remove Extra Space Button

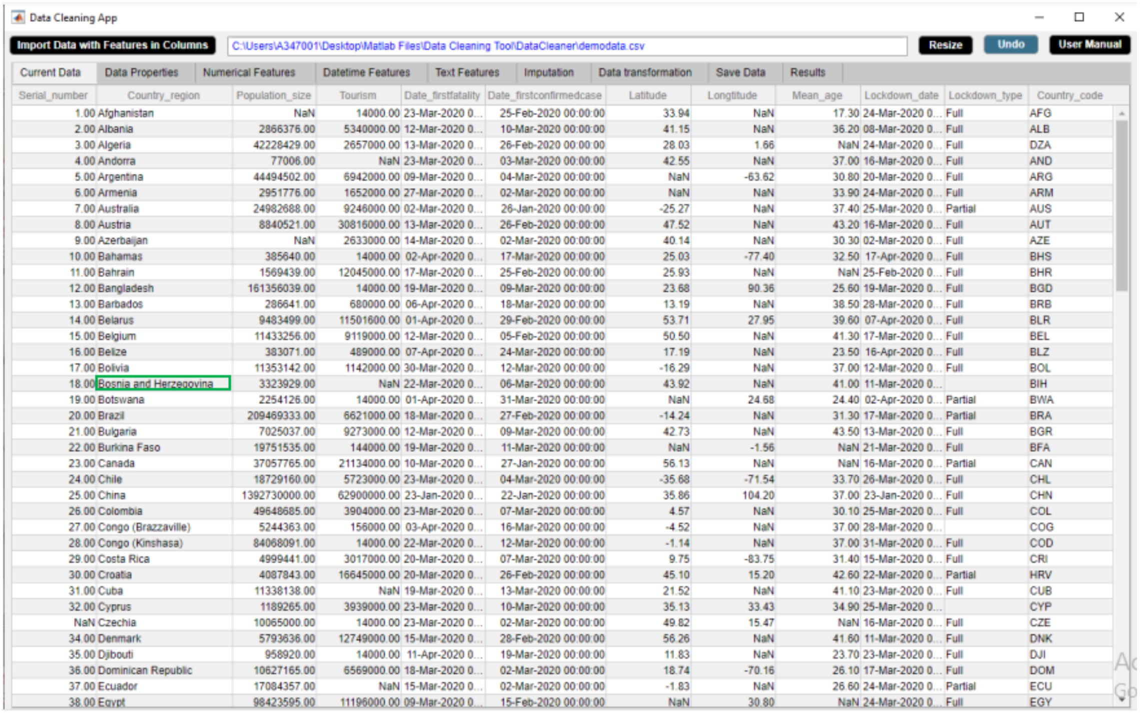


Figure B.36: Step 7. Remove Extra Space Button

B.3.5 Delete Rows Button

Deletes rows from data.
Application

- Delete rows containing a large number of missing observations.

Example

Step 1: Select minimum row number from minimum slider and maximum row number from maximum slider.
Step 2: Click **Delete Rows** button.
Step 3: **Delete Rows** button in use turns grey in color.
Step 4: **Delete Rows** button returns back to its original color once it completes its task.

The example data contains a large number of missing values in the last 7 rows. We use **Delete Rows** button to delete the last 7 rows of the data. Figures B.37-B.40 illustrate how to use **Delete Rows** button.

Figure B.37: Step 1. Delete Rows Button

Figure B.38: Step 2. Delete Rows Button

Figure B.39: Step 3. Delete Rows Button

Figure B.40: Step 4. Delete Rows Button

B.3.6 Sort Features Button

Sorts features in ascending order by missing observations percentage.
Example

Step 1: Click **Sort Features** button.
Step 2: **Sort Features** button in use turns grey in color.
Step 3: **Sort Features** button returns back to its original color once it completes its task.

We use **Sort Features** button to sort the features of the example data by increasing missing observations percentage. Figures B.41-B.43 illustrate how to use **Sort Features** button.

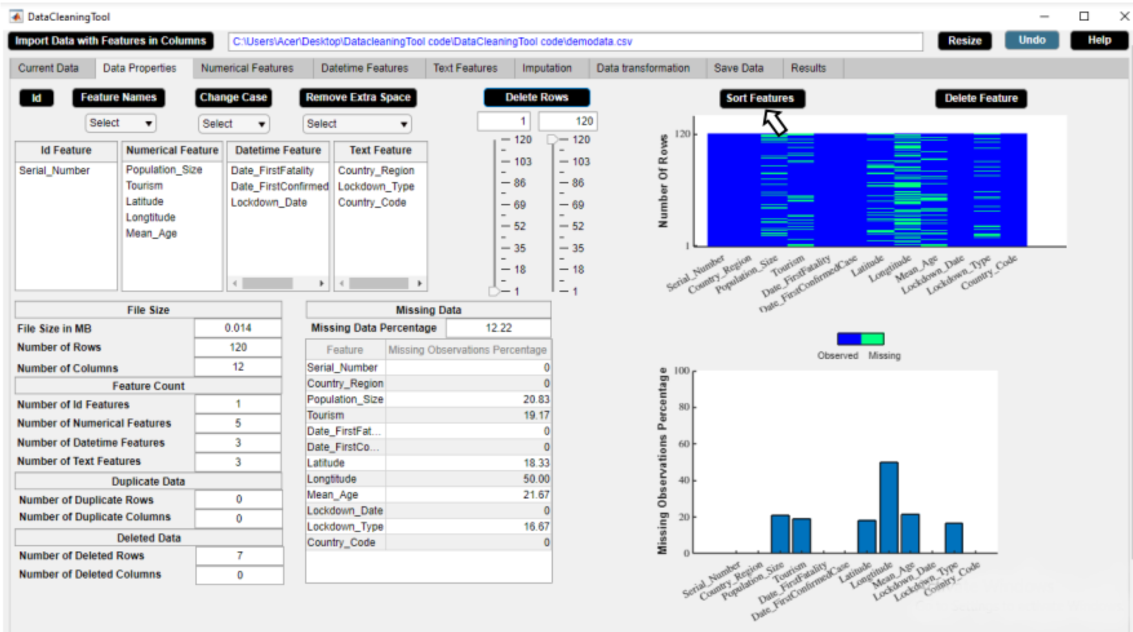


Figure B.41: Step 1. Sort Features Button

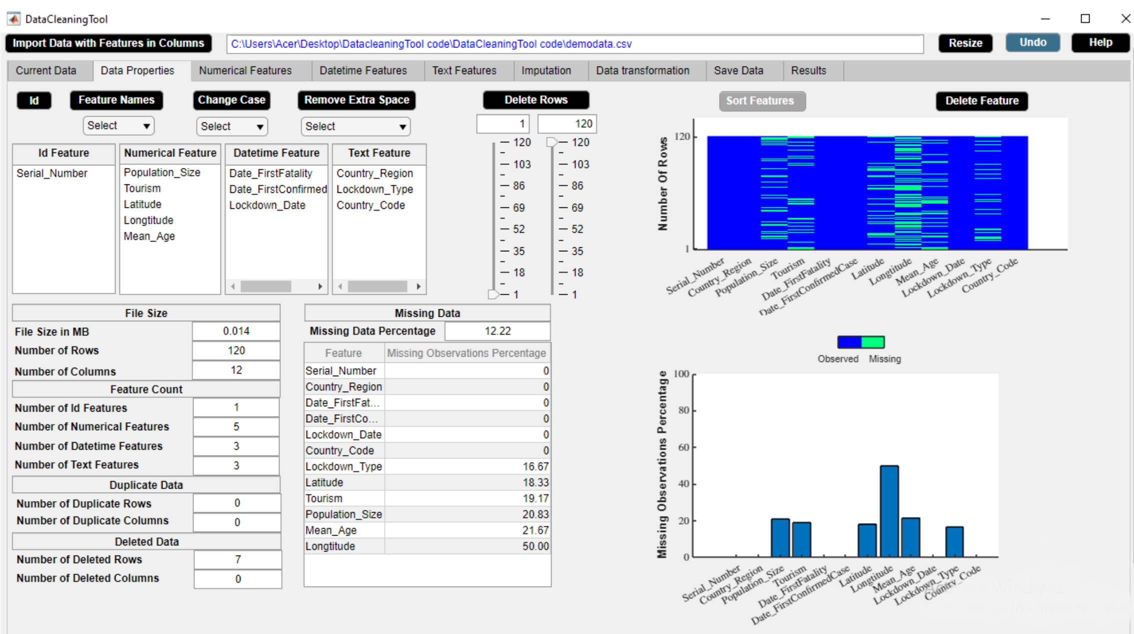


Figure B.42: Step 2. Sort Features Button

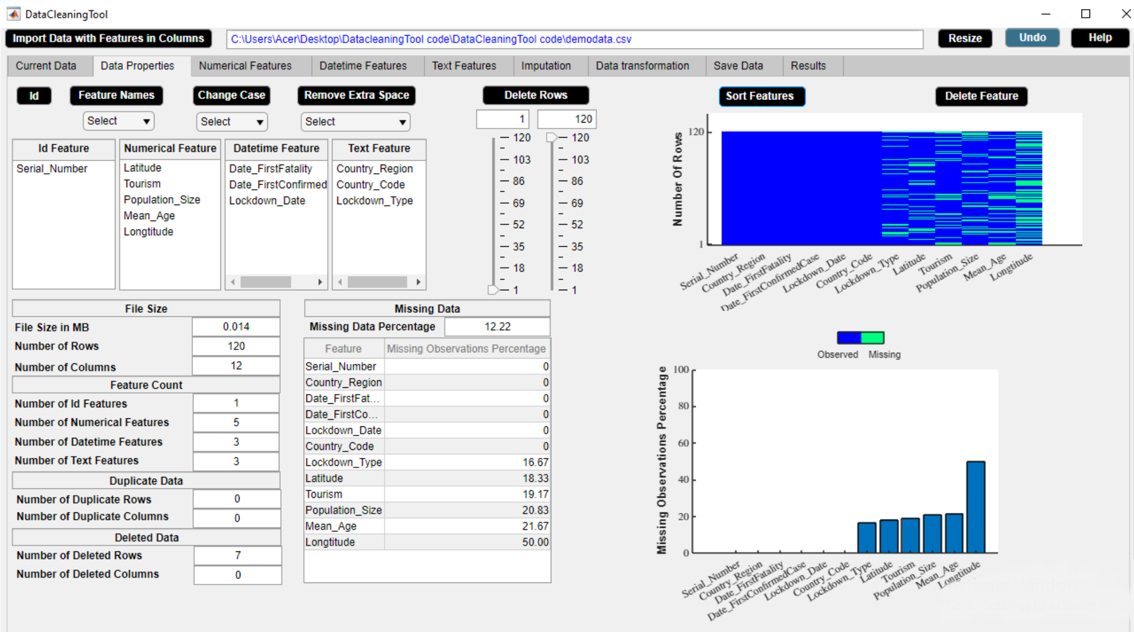


Figure B.43: Step 3. Sort Features Button

B.3.7 Delete Feature Button

Delete a feature from data.

Application

- Delete an unwanted or irrelevant feature.
- Delete a feature containing a large number of missing observations.

Example

Step 1: Select a feature from **Feature** column of missing observations percentage table.
Step 2: Click **Delete Feature** button.
Step 3: **Delete Feature** button in use turns grey in color.
Step 4: **Delete Feature** button returns back to its original color once it completes its task.

From a data analyst's point of view, 'Country_Code' is an irrelevant feature in the example data. We use **Delete Feature** button to delete 'Country_Code' feature. Figures B.44-B.47 illustrate how to use **Delete Feature** button.

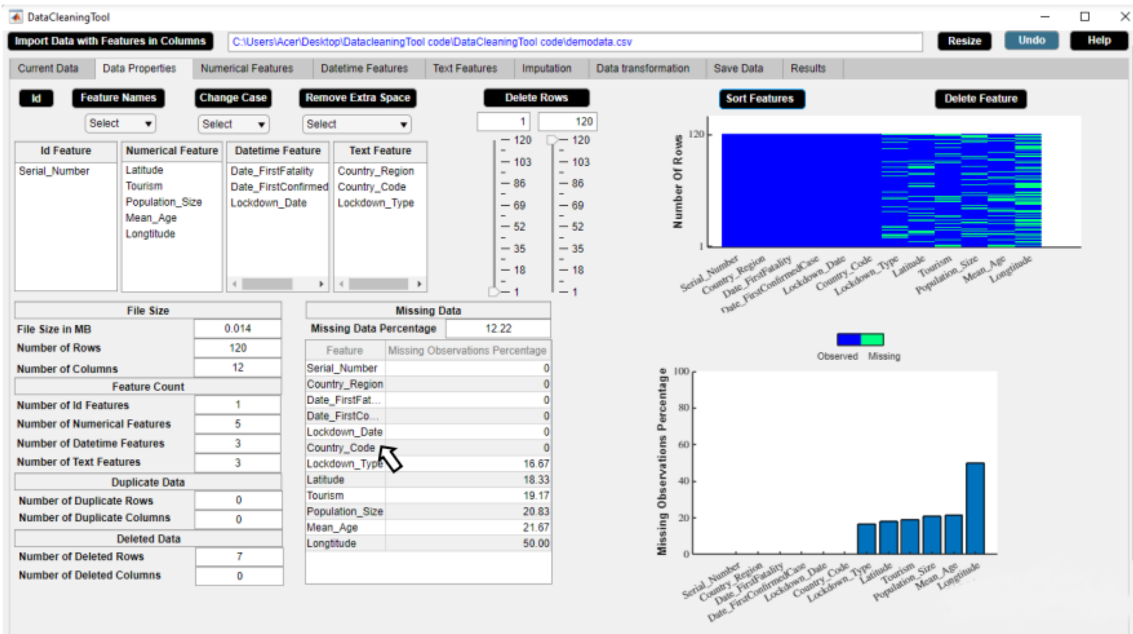


Figure B.44: Step 1. Delete Feature Button

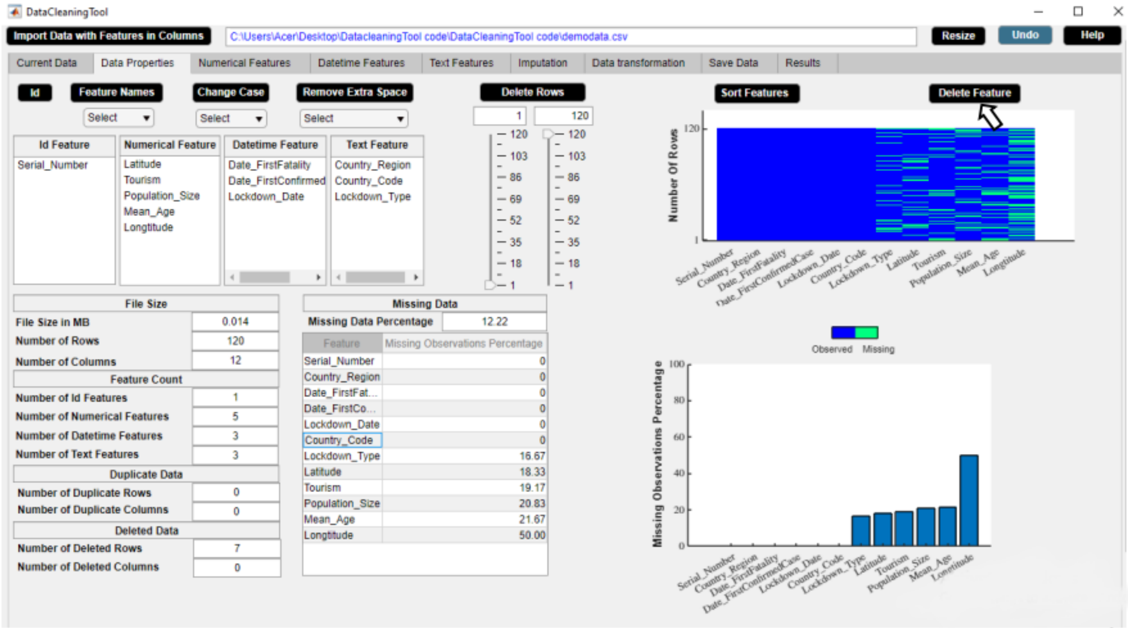


Figure B.45: Step 2. Delete Feature Button

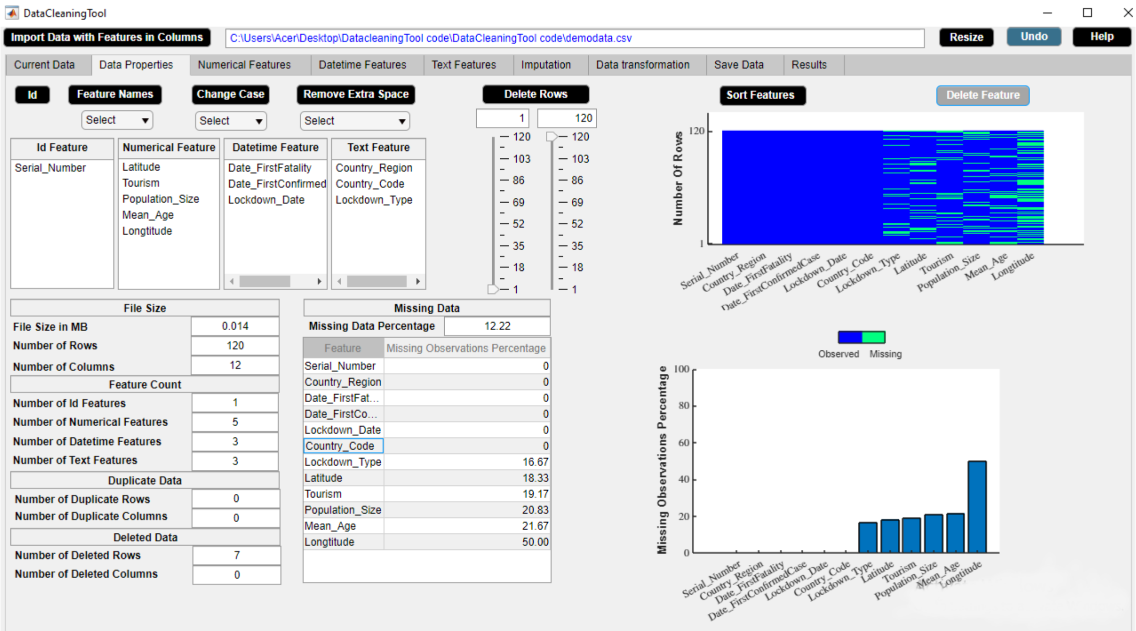


Figure B.46: Step 3. Delete Feature Button

Figure B.47: Step 4. Delete Feature Button

B.4 Numerical Features Widget

The Numerical Features widget displays statistical description of the numerical data. The Numerical Features widget is shown in figure B.48. The properties of the Numerical Features widget are as follows.

- The widget shows the descriptive statistics of each numerical feature of the data such as minimum observation and maximum observation of the feature. Descriptive statistics of a feature gives a quantitative description of a feature.
- The widget shows the duplicate observations present in each numerical feature and the missing observations percentage of each numerical feature. Duplicate observation can be an error in the data and could possibly influence later analyses of the data.
- Cross validation constraint and range constraint can be set in the widget. This will result in some unwanted numerical observations.
- The statistical information of the numerical data in the widget gets updated after each activity.

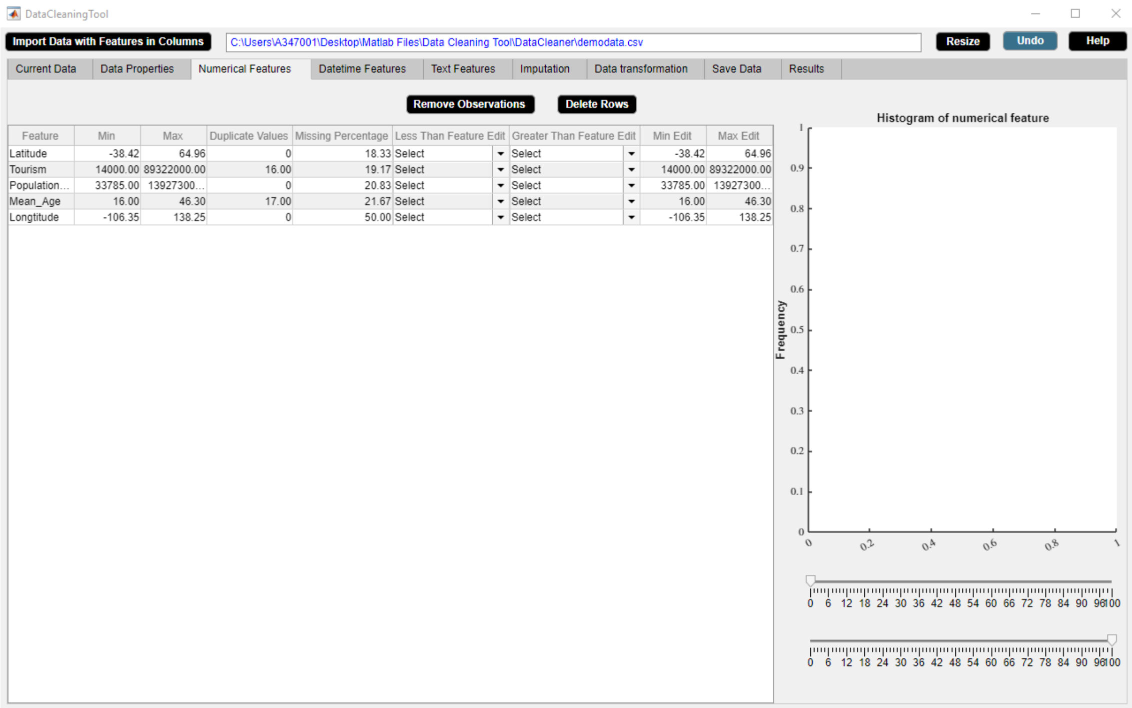


Figure B.48: Numerical Features Widget.

B.4.1 Numerical Feature Cell Selection Button

Displays histogram of a numerical feature.

Application

- Outlier visualization technique.

Example

Step 1: Select a numerical feature from **Feature** column of the numerical features descriptive statistics table.

Step 2: A histogram of the selected numerical feature appears in the right side of the **Numerical Features** widget and the sliders get updated accordingly.

We use **Numerical Feature Cell Selection** button to visualize the histogram of 'Population_Size' feature. Figures B.49-B.50 illustrate how to use **Numerical Feature Cell Selection** button.

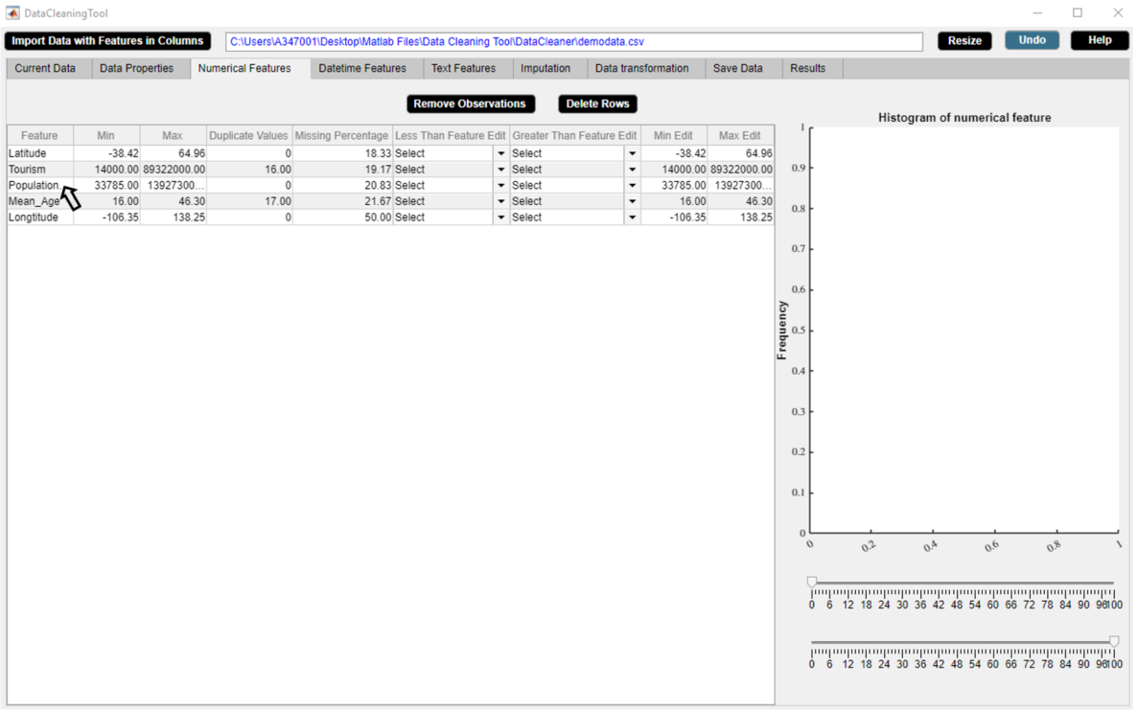


Figure B.49: Step 1. Numerical Feature Cell Selection Button

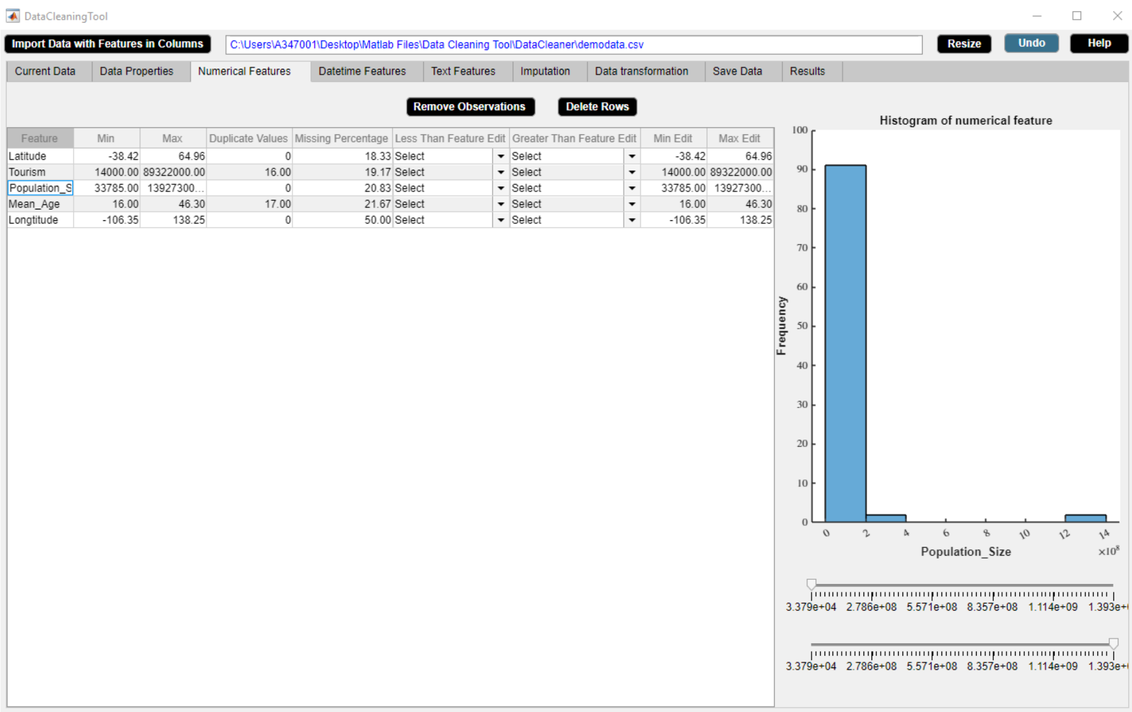


Figure B.50: Step 2. Numerical Feature Cell Selection Button

B.4.2 Remove Observations Button

Replaces unwanted numerical observations by missing values.

Application

- Removes unwanted or irrelevant observations.

Example

Step 1: Choose constraint from **Less Than Feature Edit** dropdown menu or **Greater Than Feature Edit** dropdown menu or **Min Edit** box or **Max Edit** box in the **Numerical Features** widget.
Step 2: Click **Remove Observations** button.
Step 3: **Remove Observations** button in use turns grey in color.
Step 4: **Remove Observations** button returns back to its original color once it completes its task.

We wish to prepare the data for analysis for the countries whose ‘Population_Size’ is greater than ‘tourism’. We use **Remove Observations** button to extract data for the countries whose ‘Population_Size’ is greater than ‘Tourism’. Figures B.51-B.54 illustrate how to use **Remove Observations** button.

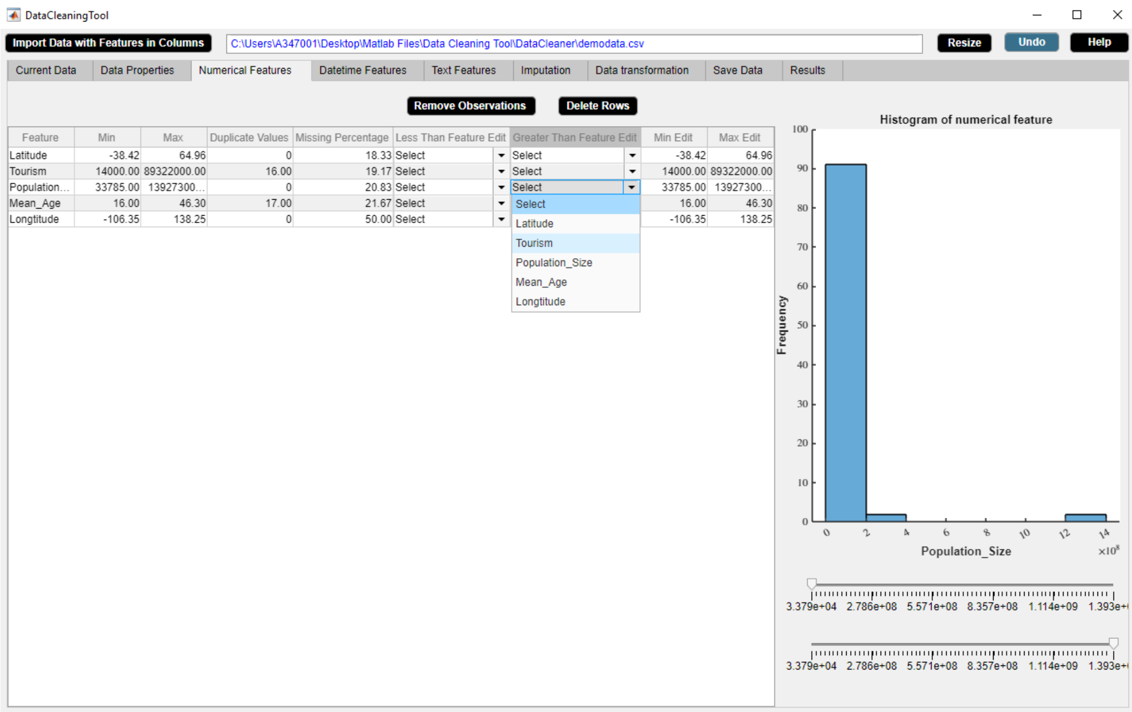


Figure B.51: Step 1. Remove Observations Button

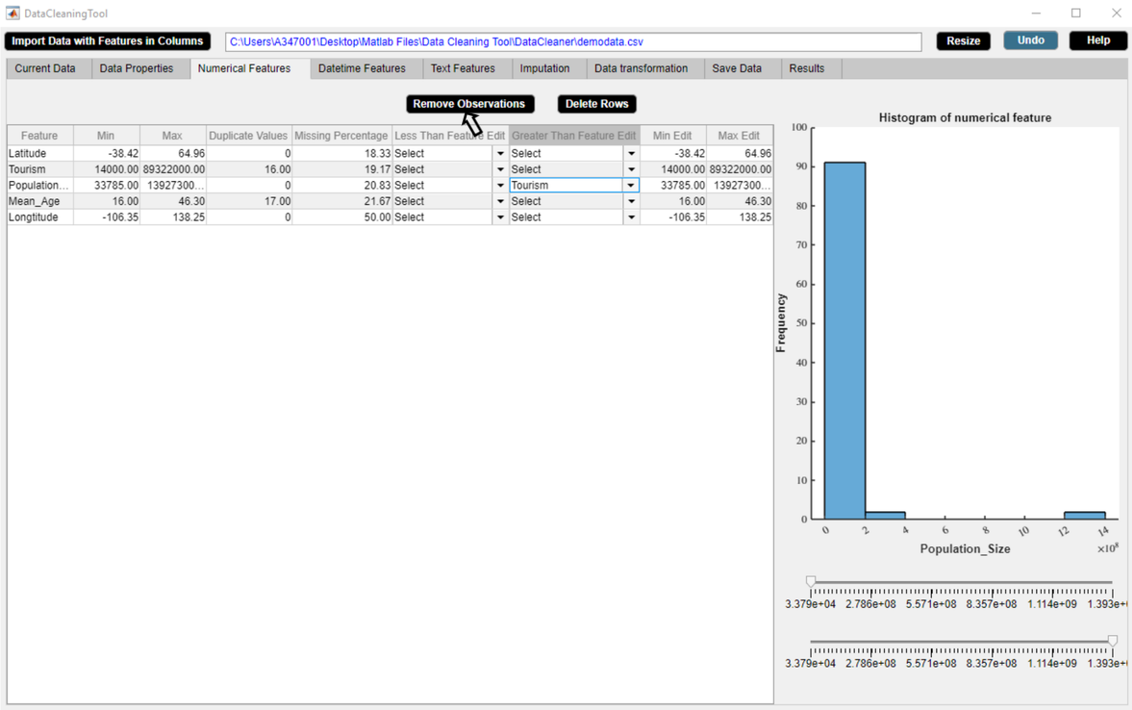


Figure B.52: Step 2. Remove Observations Button

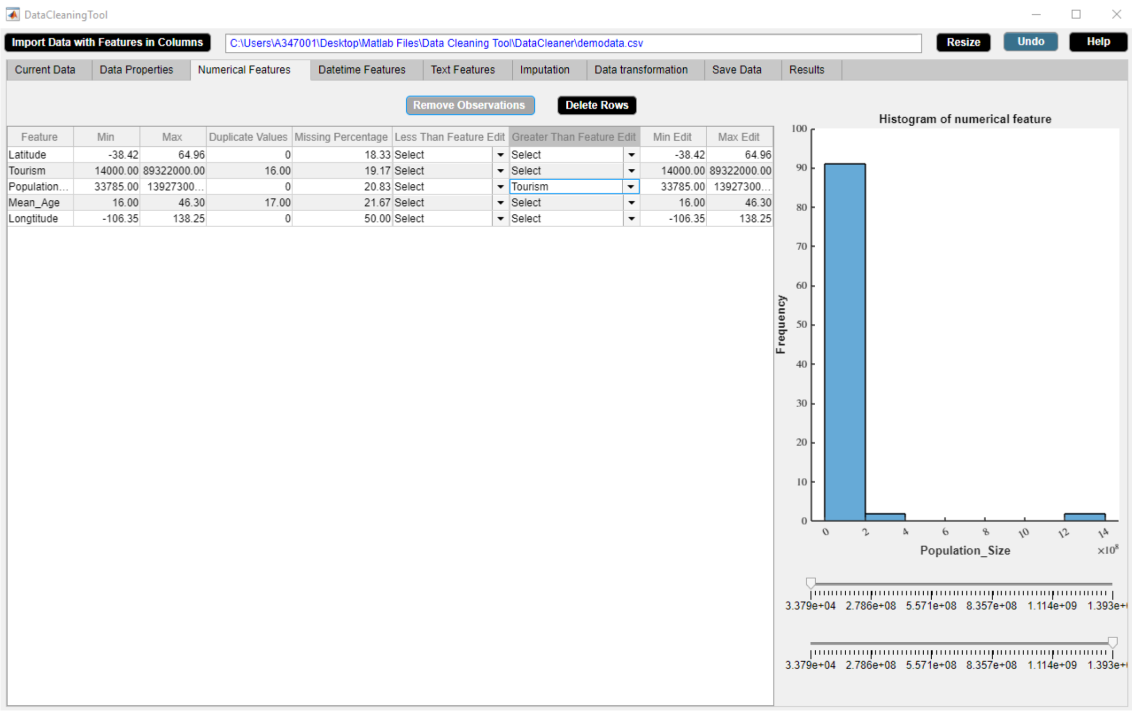


Figure B.53: Step 3. Remove Observations Button

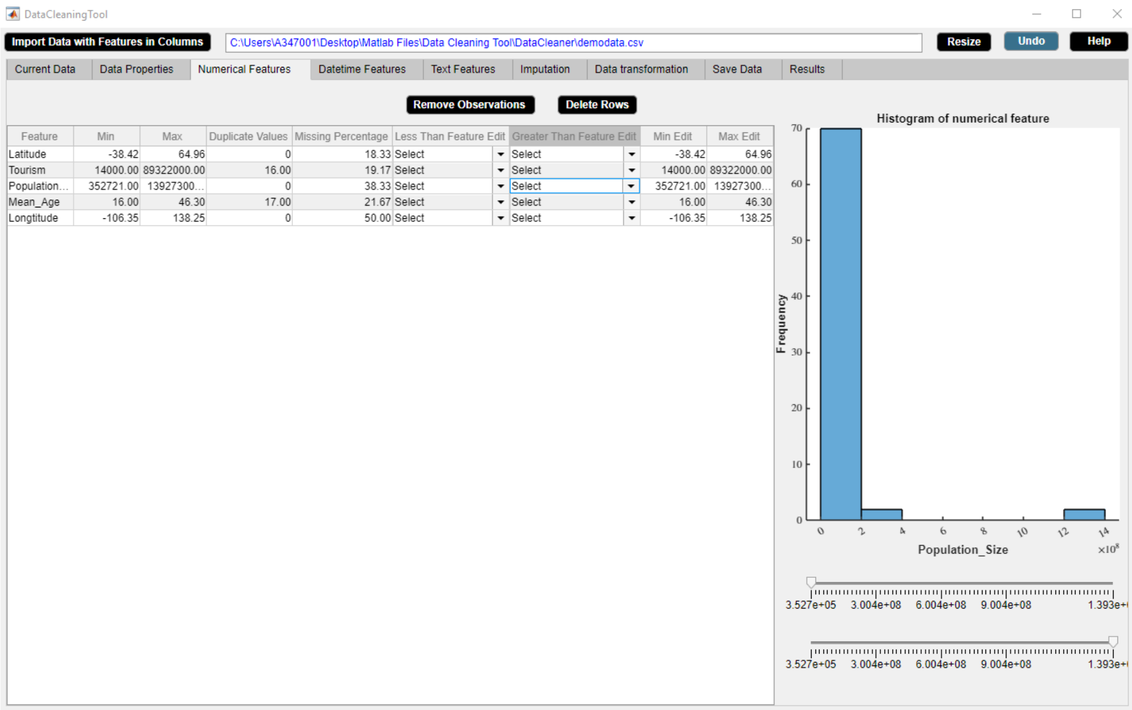


Figure B.54: Step 4. Remove Observations Button

B.4.3 Delete Rows Button

Deletes rows with unwanted numerical observations.

Application

- Delete unwanted or irrelevant rows.
- Delete rows containing a large number of missing observations.

Example

Step 1: Select a numerical feature from **Feature** column of the numerical features descriptive statistics table.

Step 2: A histogram of the selected numerical feature appears in the right side of the **Numerical Features** widget and the sliders get updated accordingly. Choose constraint from **Less Than Feature Edit** dropdown menu or **Greater Than Feature Edit** dropdown menu or **Min Edit** box or **Max Edit** box of the numerical features descriptive statistics table in the **Numerical Features** widget. Also, minimum value and maximum value can be selected from sliders.

Step 3: Click **Delete Rows** button.

Step 4: **Delete Rows** button in use turns grey in color.

Step 5: **Delete Rows** button returns back to its original color once it completes its task.

We wish to prepare the data for analysis for the countries whose maximum 'Mean_age' is 45. We use **Delete Rows** button to extract data for the countries whose maximum 'Mean_age' is 45. Figures B.55-B.59 illustrate how to use **Delete Rows** button.

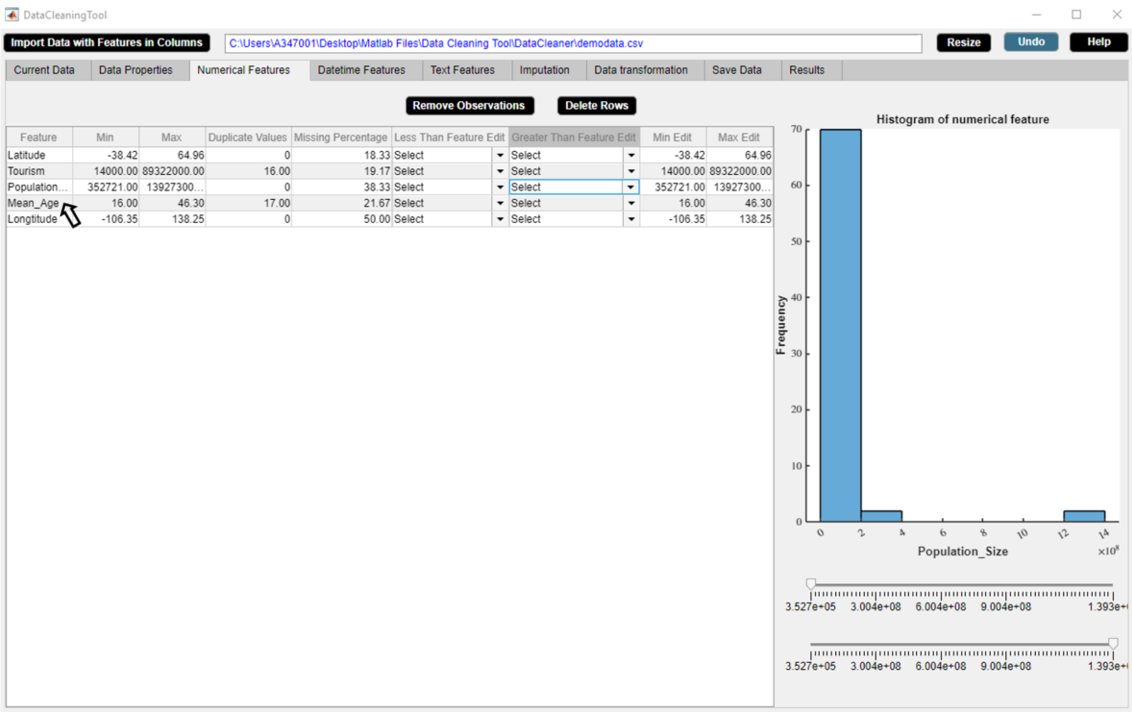


Figure B.55: Step 1. Delete Rows Button

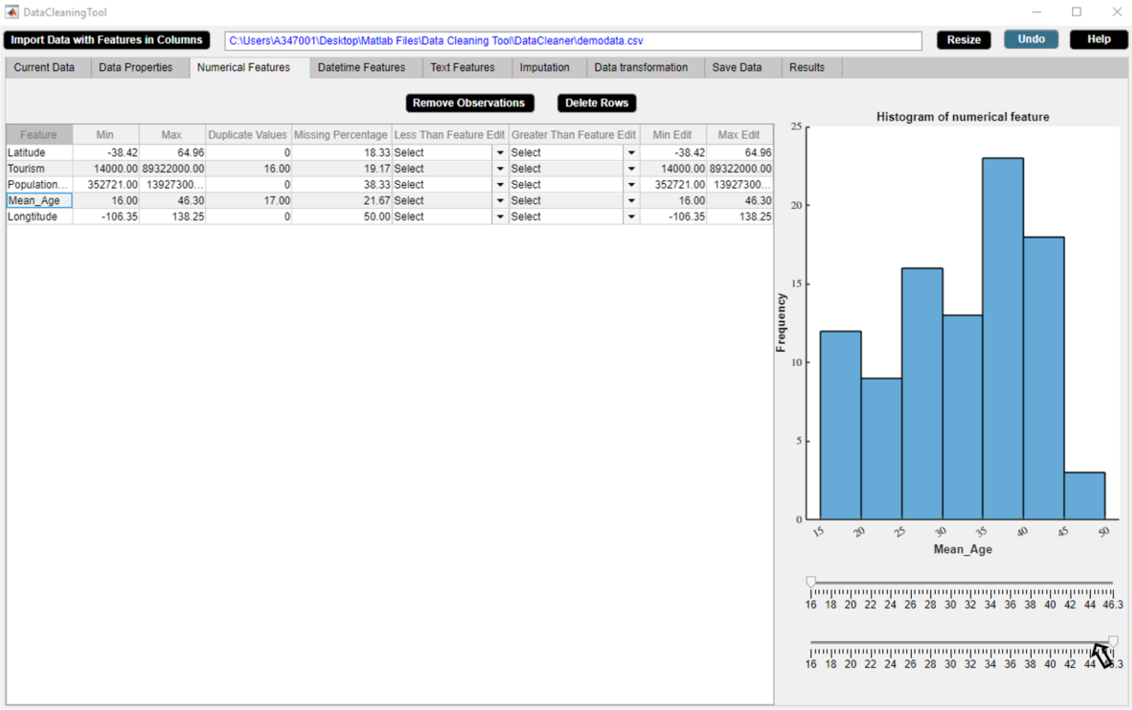


Figure B.56: Step 2. Delete Rows Button

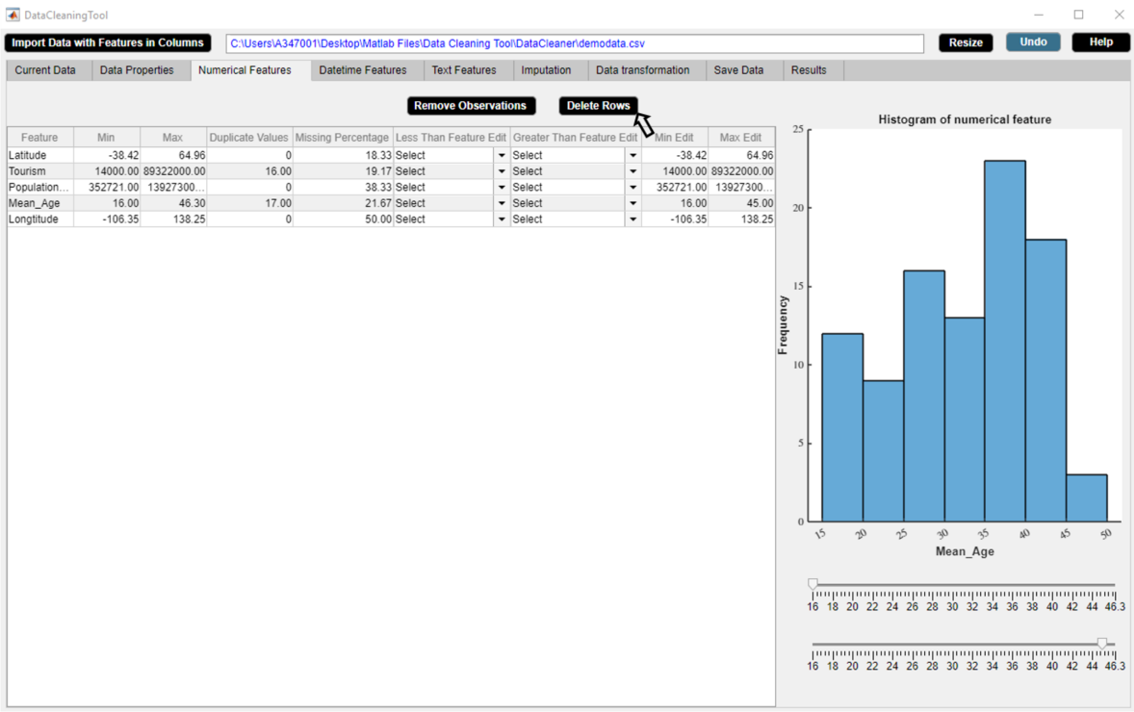


Figure B.57: Step 3. Delete Rows Button

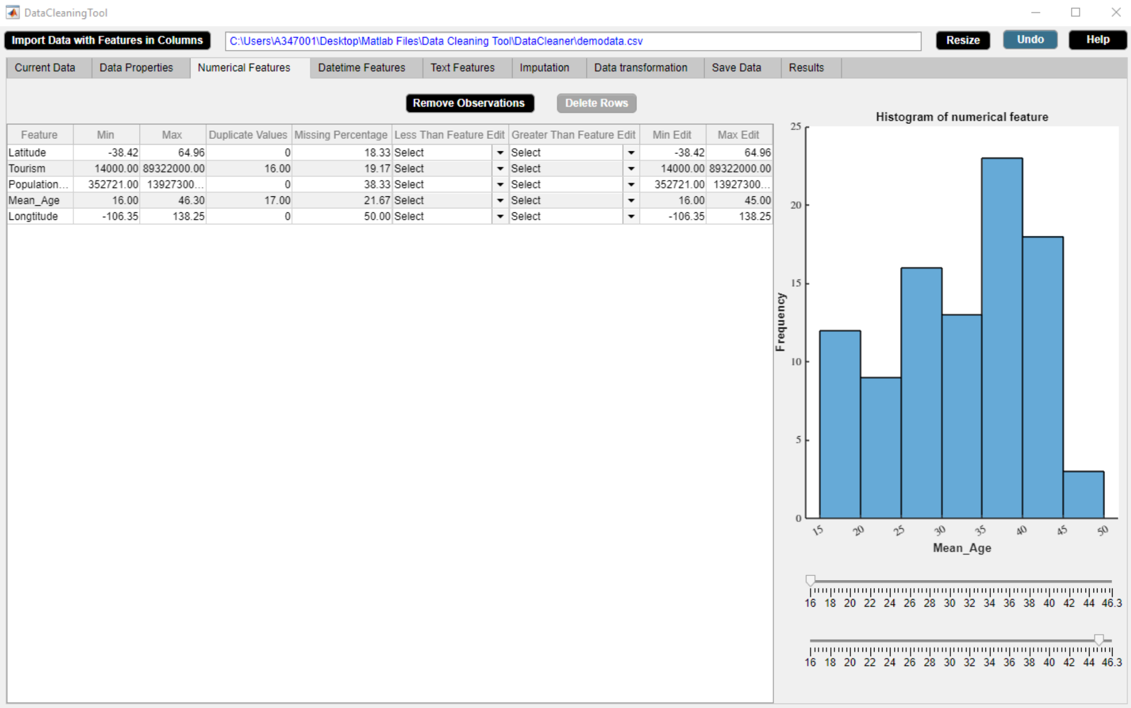


Figure B.58: Step 4. Delete Rows Button

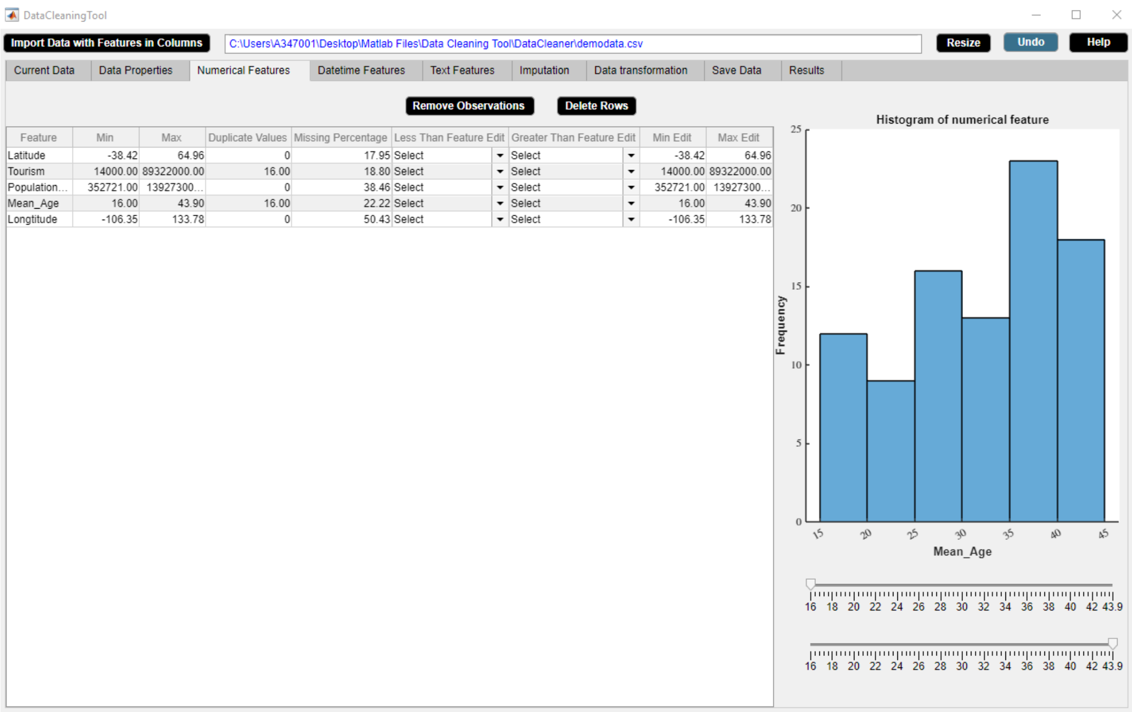


Figure B.59: Step 5. Delete Rows Button

B.5 Datetime Features Widget

The Datetime Features widget displays statistical description of the datetime data. The Datetime Features widget is shown in figure B.60. The properties of the Datetime Features widget are as follows.

- The widget shows the descriptive statistics of each datetime feature of the data such as minimum observation and maximum observation of the feature.
- The widget also shows the missing observations percentage of each datetime feature.
- Datetime format can be changed.
- Cross validation constraint and range constraint can be set in the widget for each datetime feature. This will result in some unwanted datetime observations.
- The statistical information of the datetime data in the widget gets updated after each activity.

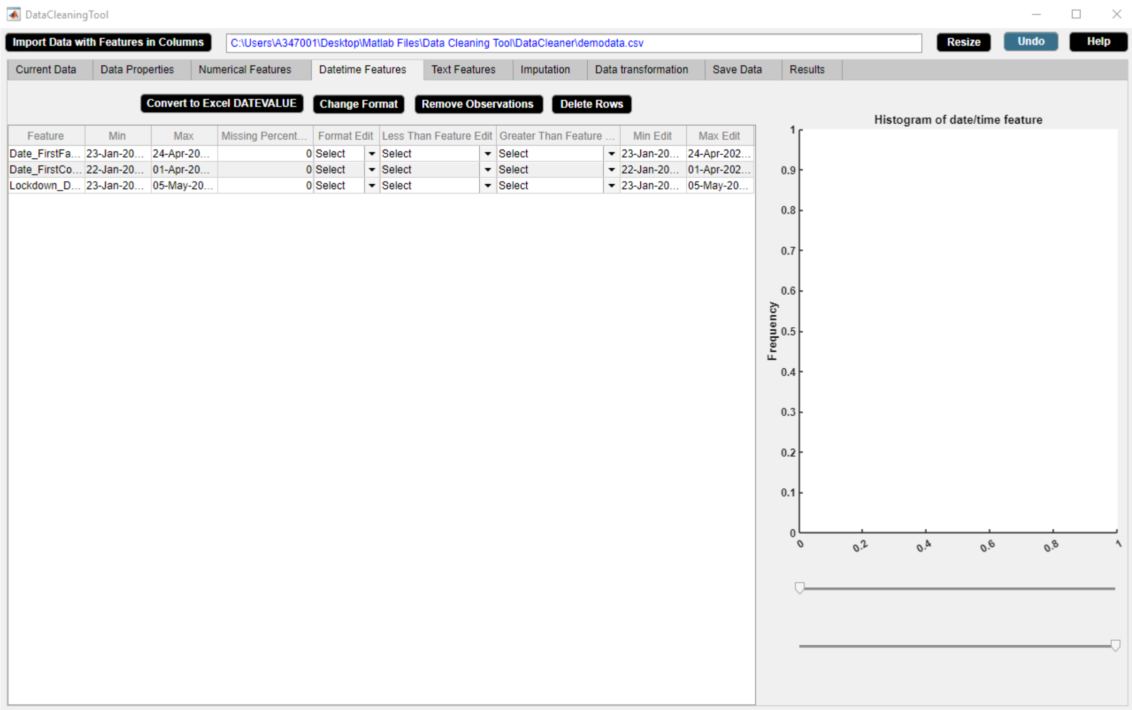


Figure B.60: Datetime Features Widget.

B.5.1 Datetime Feature Cell Selection Button

Displays histogram of a datetime feature.

Application

- Outlier visualization technique.

Example

Step 1: Select a datetime feature from **Feature** column of the datetime features descriptive statistics table.

Step 2: A histogram of the selected datetime feature appears in the right side of the **Datetime Features** widget and the sliders get updated accordingly.

We use **Datetime Feature Cell Selection** button to visualize the histogram of 'Date_FirstConfirmedCase' feature. Figures B.61-B.62 illustrate how to use **Datetime Feature Cell Selection** button.

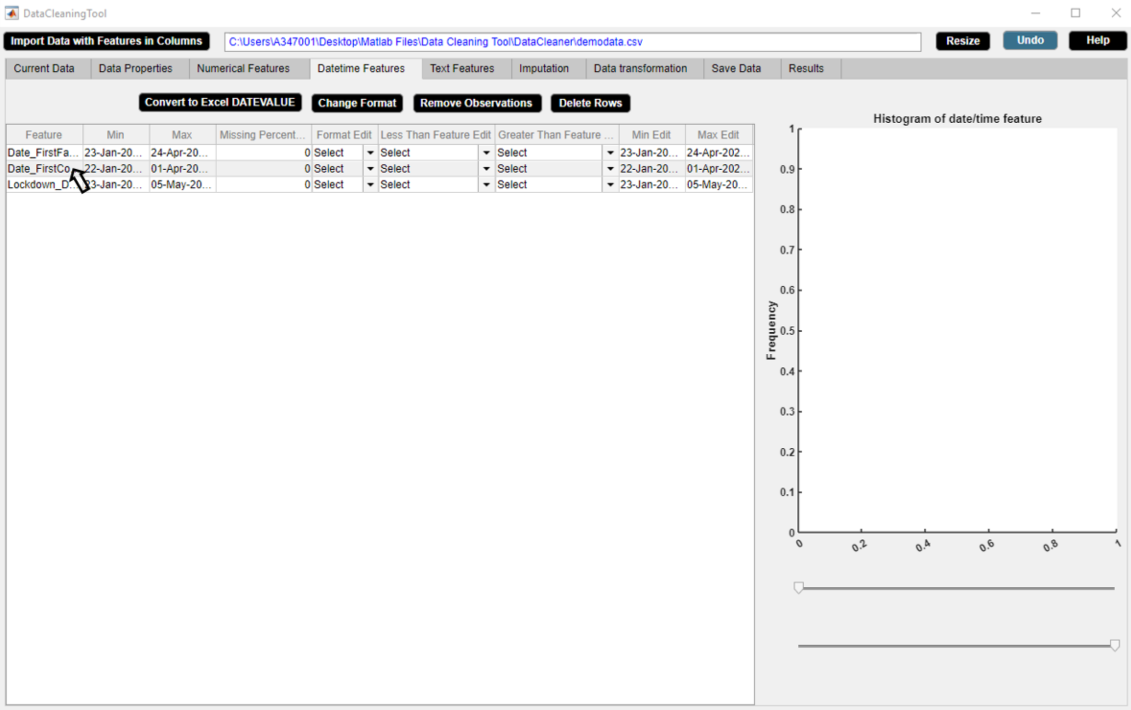


Figure B.61: Step 1. Datetime Feature Cell Selection Button

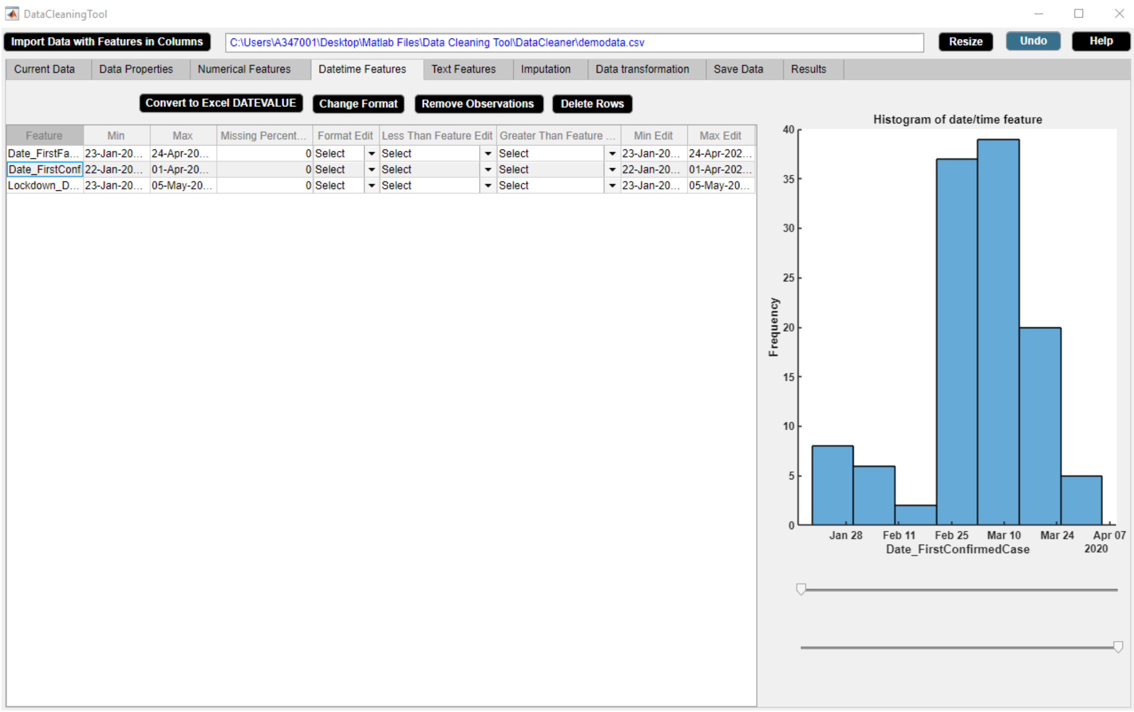


Figure B.62: Step 2. Datetime Feature Cell Selection Button

B.5.2 Convert To Excel DATEVALUE Button

Converts datetime to Excel DATEVALUE. First it transforms datetime to Matlab serial date number and then to Excel serial date number. MATLAB date numbers start from January 1, 0000 A.D., and hence there is a difference of 693960 relative to the Excel date system which uses January 1, 1900, as starting point.

B.5.3 Change Format Button

Changes datetime format.
Example

Step 1: Select a datetime format from **Format Edit** dropdown menu of the datetime features descriptive statistics table.
Step 2: Click **Change Format** button.
Step 3: **Change Format** button in use turns grey in color.
Step 4: **Change Format** button returns back to its original color once it completes its task.
Step 5: Check the datetime format in the **Current Data** widget.

We use **Change Format** button to change the datetime format of all the datetime features to 'yyyy-MM-dd HH:mm:ss'. Figures B.63-B.67 illustrate how to use **Change Format** button.

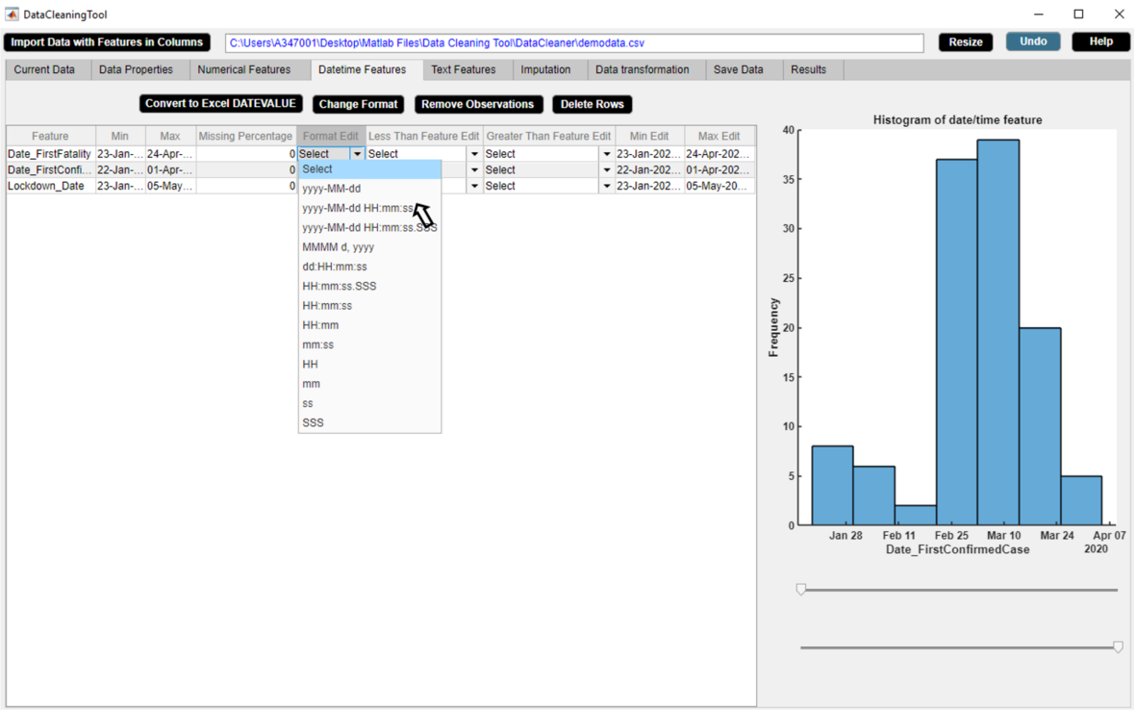


Figure B.63: Step 1. Change Format Button

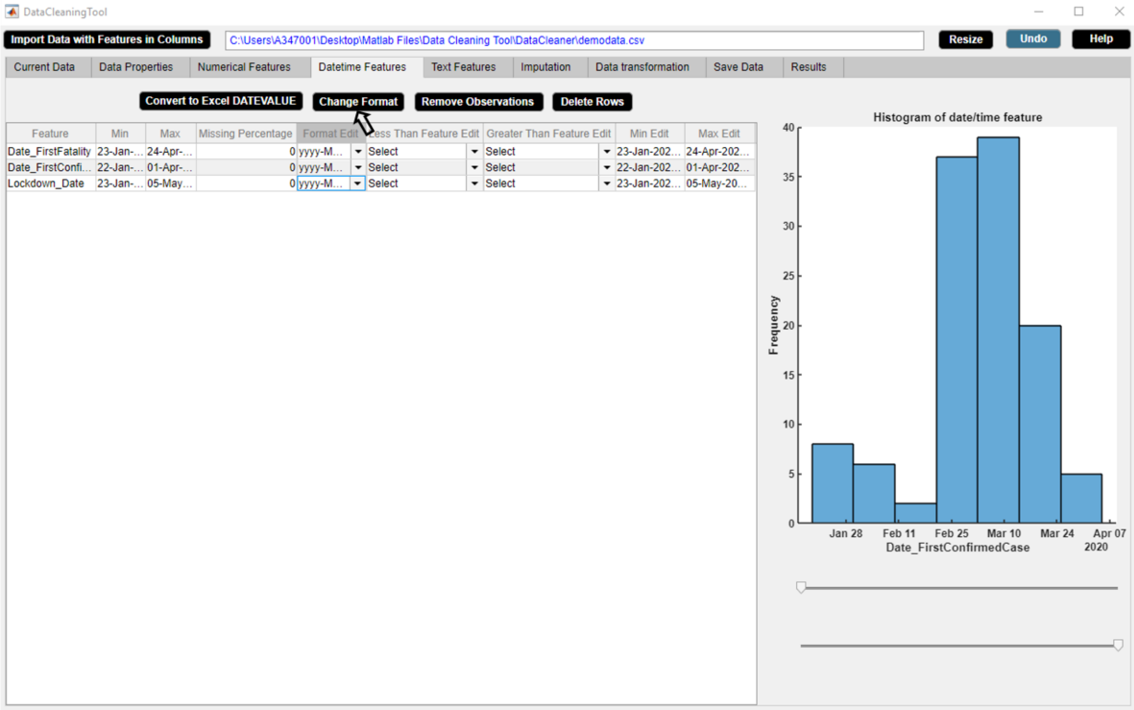


Figure B.64: Step 2. Change Format Button

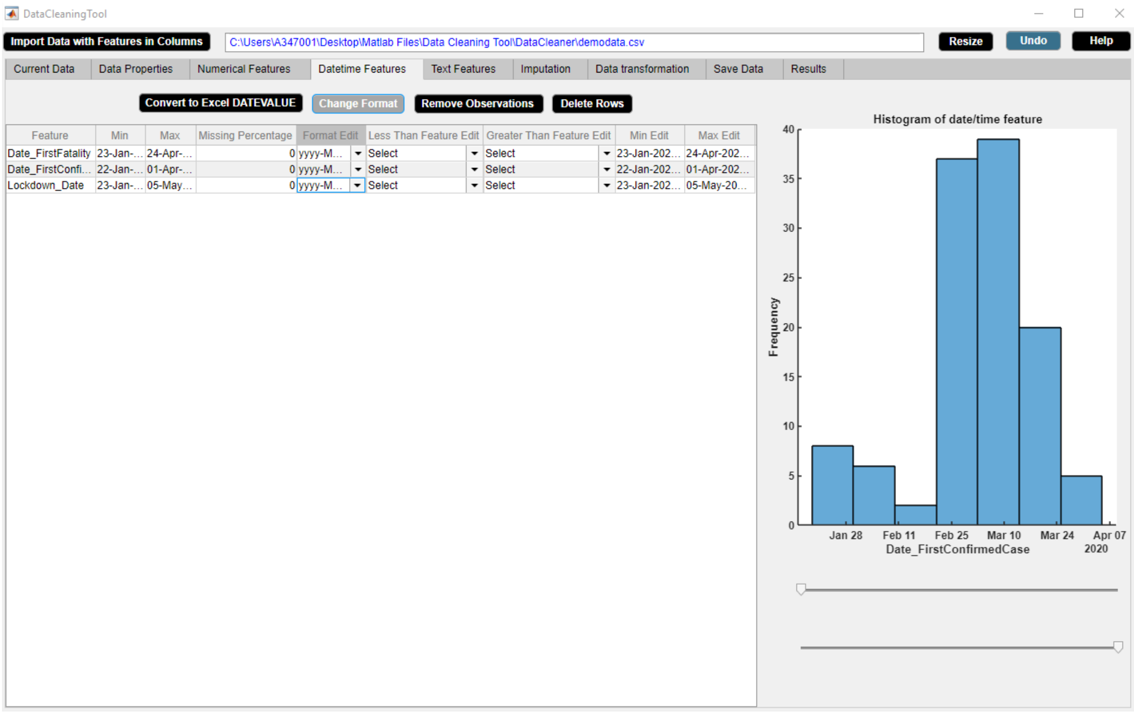


Figure B.65: Step 3. Change Format Button

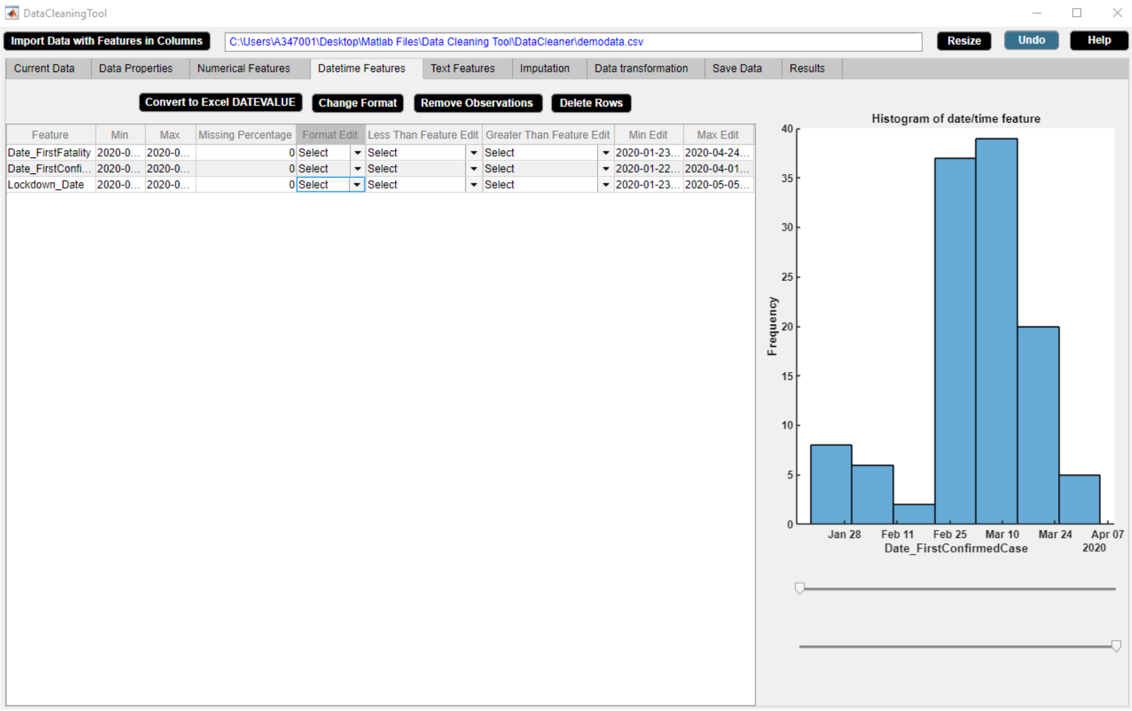


Figure B.66: Step 4. Change Format Button

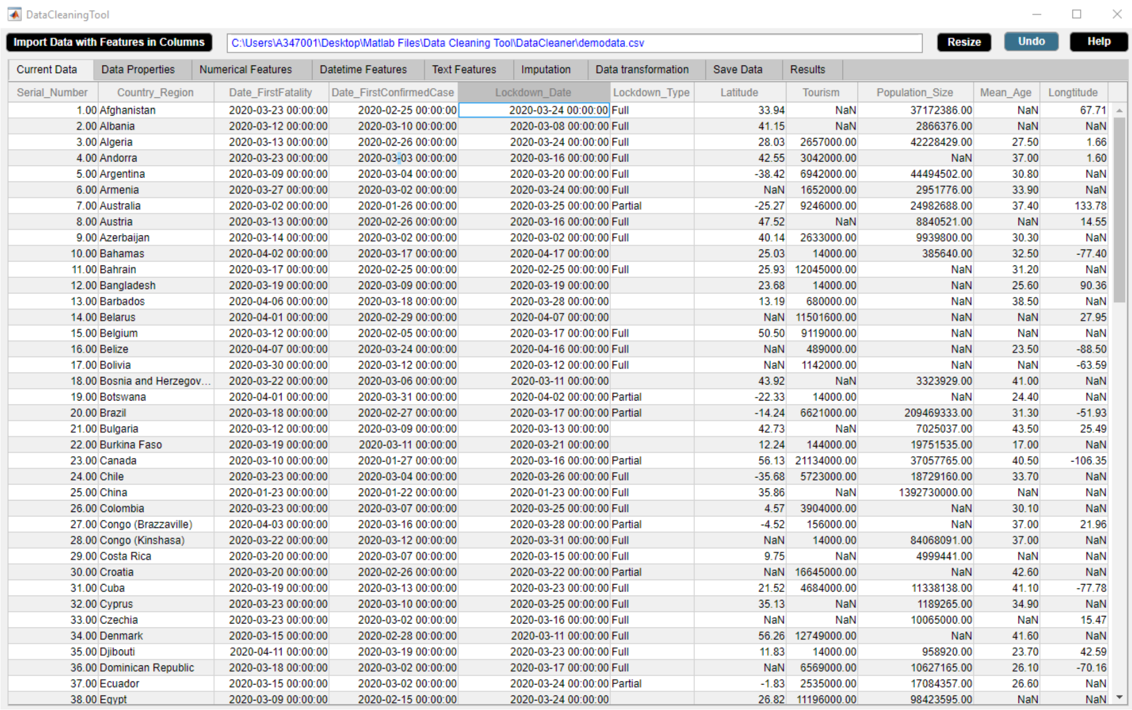


Figure B.67: Step 5. Change Format Button

B.5.4 Remove Observations Button

Replaces unwanted datetime observations by missing values.
Application
- Remove unwanted or irrelevant observations.

B.5.5 Delete Rows Button

Deletes rows with unwanted datetime observations.
Application

- Delete unwanted or irrelevant rows.
- Delete rows containing a large number of missing observations.

Example

Step 1: Choose constraint from **Less Than Feature Edit** dropdown menu or **Greater Than Feature Edit** dropdown menu or **Min Edit** box or **Max Edit** box of the datetime features descriptive statistics table in the **Datetime Features** widget.
Step 2: Click **Delete Rows** button.
Step 3: **Delete Rows** button in use turns grey in color.
Step 4: **Delete Rows** button returns back to its original color once it completes its task.

We wish to prepare the data for analysis for the countries whose 'Date_FirstConfirmedCase' is less than 'Date_FirstFatality'. We use **Delete Rows** button to extract data for the countries whose 'Date_FirstConfirmed- Case' is less than 'Date_FirstFatality'. Figures B.68-B.71 illustrate how to use **Delete Rows** button.

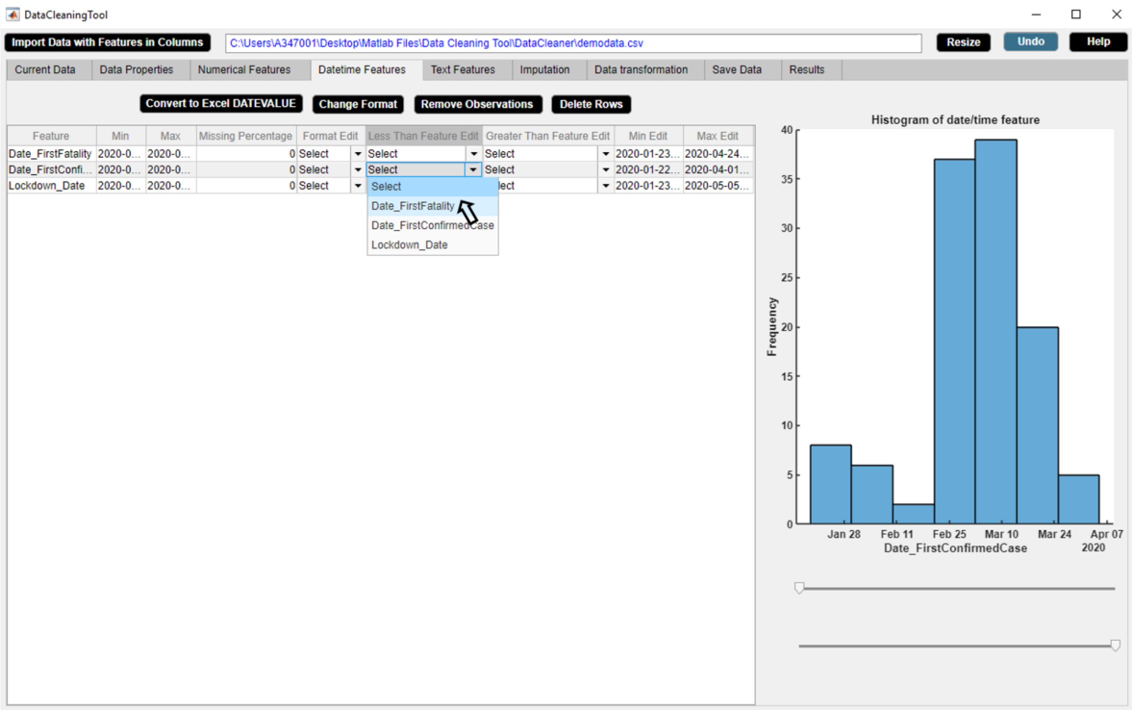


Figure B.68: Step 1. Delete Rows Button

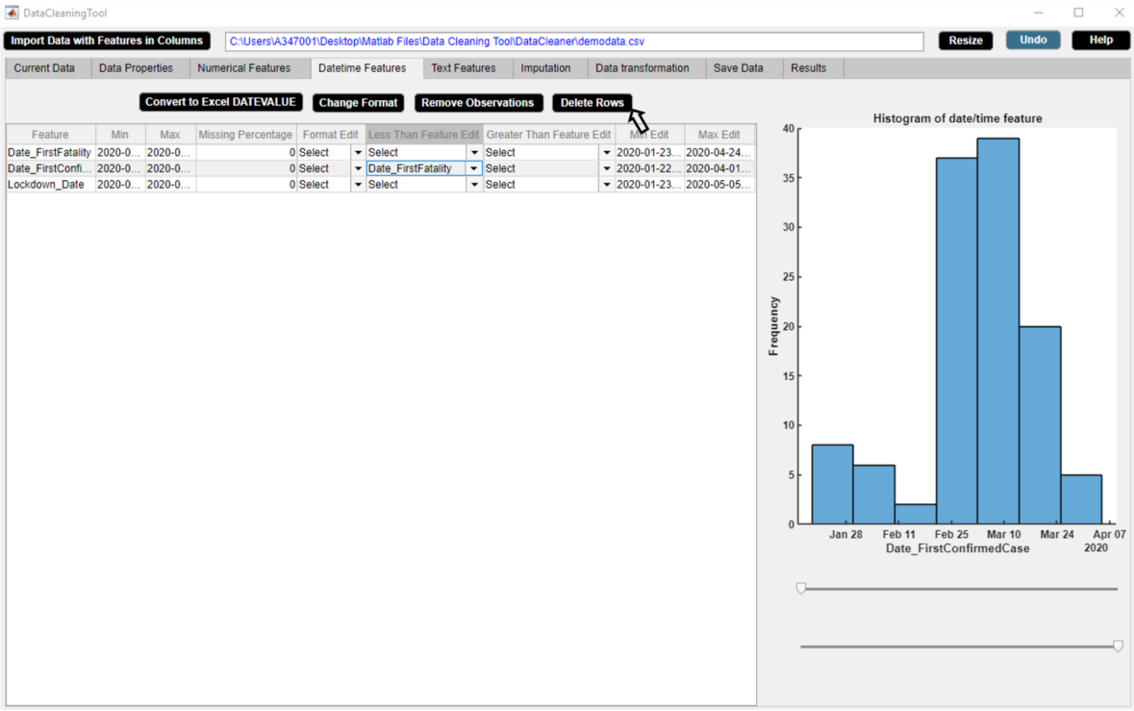


Figure B.69: Step 2. Delete Rows Button

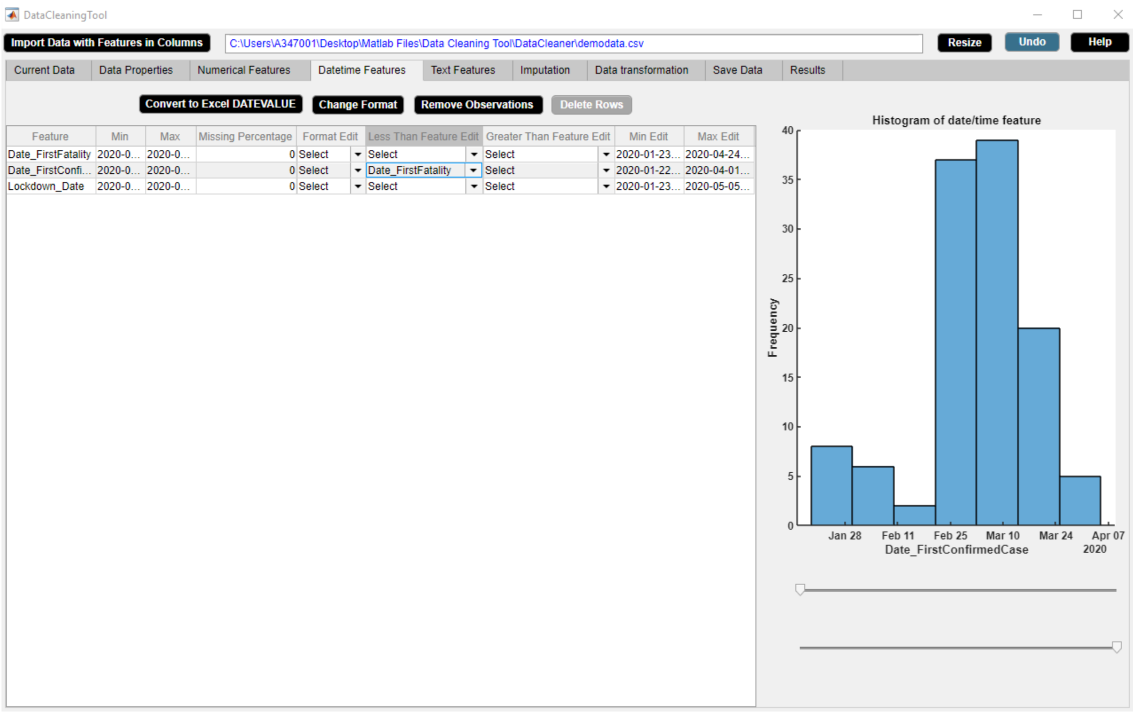


Figure B.70: Step 3. Delete Rows Button

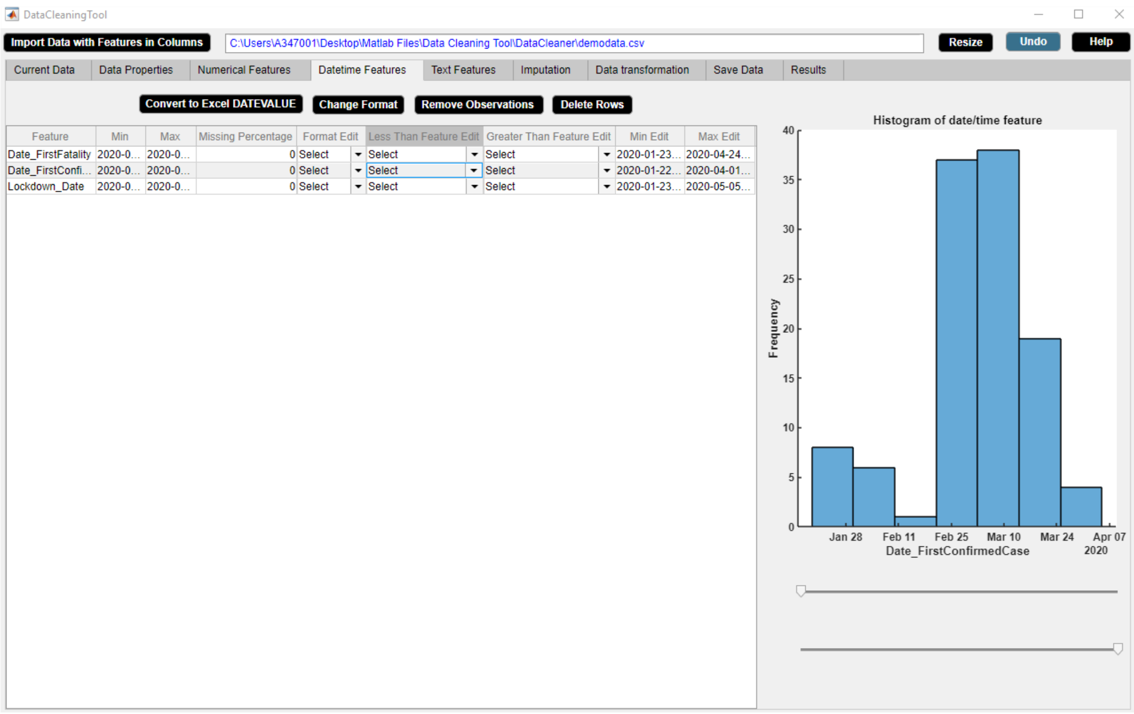


Figure B.71: Step 4. Delete Rows Button

B.6 Text Features Widget

The Text Features widget displays statistical description of the text data. The Text Features widget is shown in figure B.72. The properties of the Text Features widget are as follows.

- The widget shows the descriptive statistics of each text feature of the data such as categories and categories count of the feature.
- The widget also shows the missing observations percentage of each text feature.
- The statistical information of the text data in the widget gets updated after each activity.

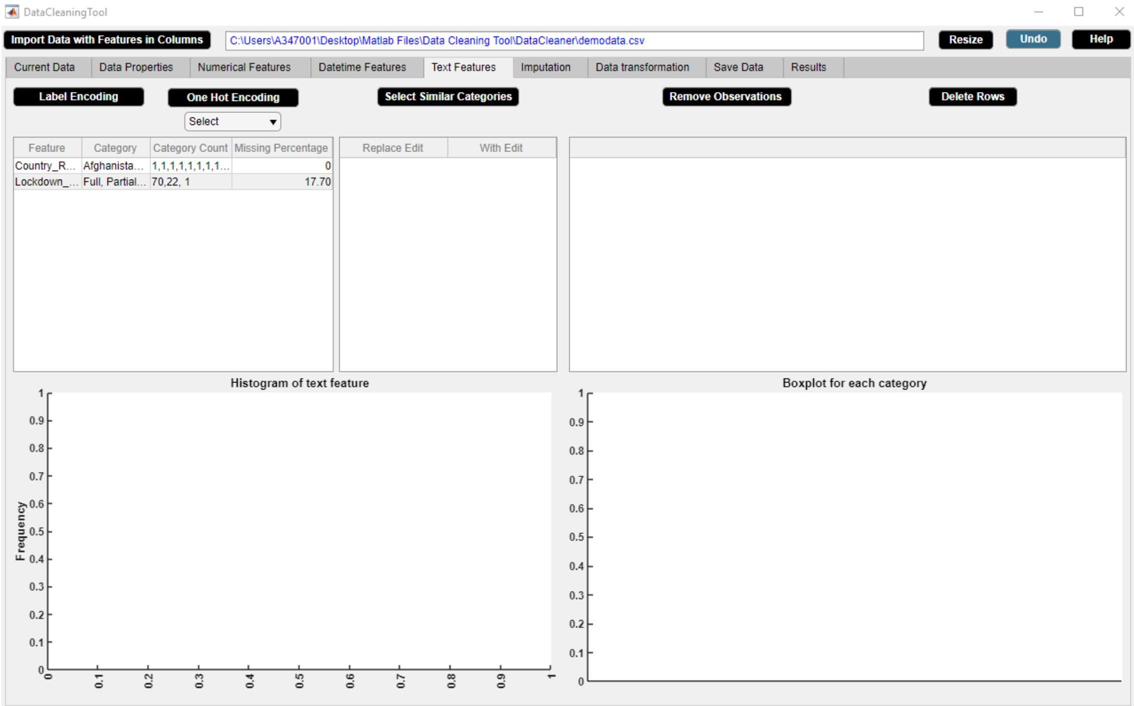


Figure B.72: Text Features Widget.

B.6.1 Select Similar Categories Button

Replaces categories with similar ones.
Example

Step 1: Select a text feature from feature column of the text features descriptive statistics table.
Step 2: Select similar category from **With Edit** dropdown menu.
Step 3: Click **Select Similar Categories** button.
Step 4: **Select Similar Categories** button in use turns grey in color.
Step 5: **Select Similar Categories** button returns back to its original color once it completes its task.

We use **Select Similar Categories** button to refer 'Total' as 'Full' in the example data. Figures B.73-B.77 illustrate how to use **Select Similar Categories** button.

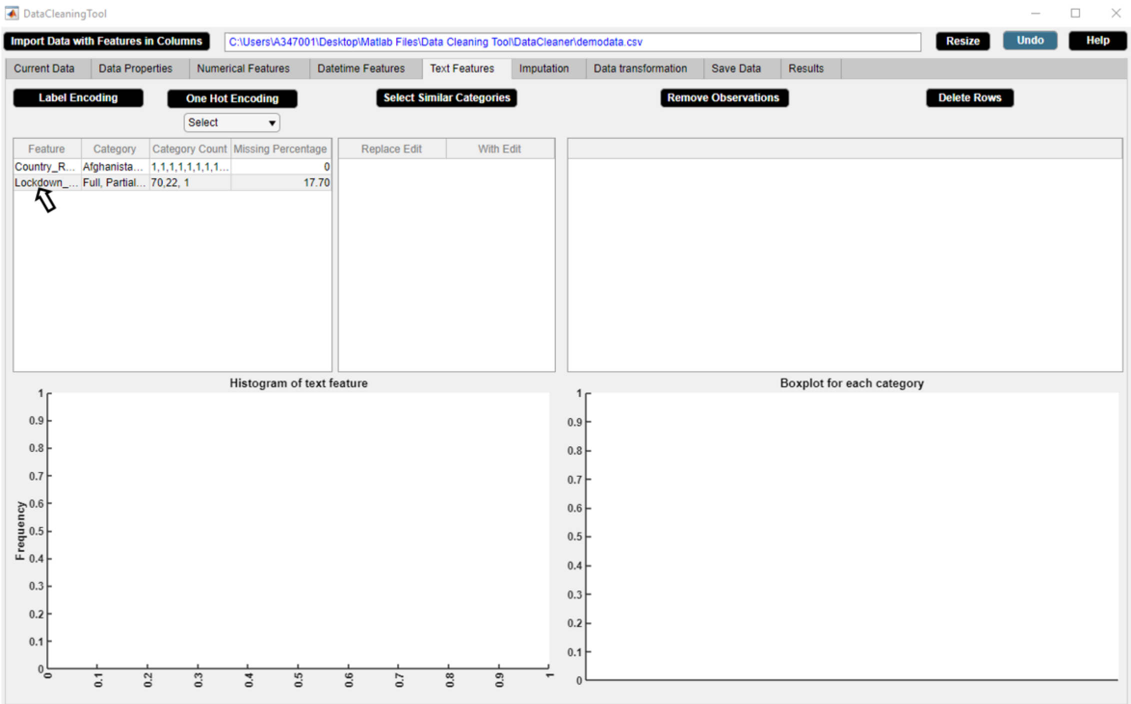


Figure B.73: Step 1. Select Similar Categories Button

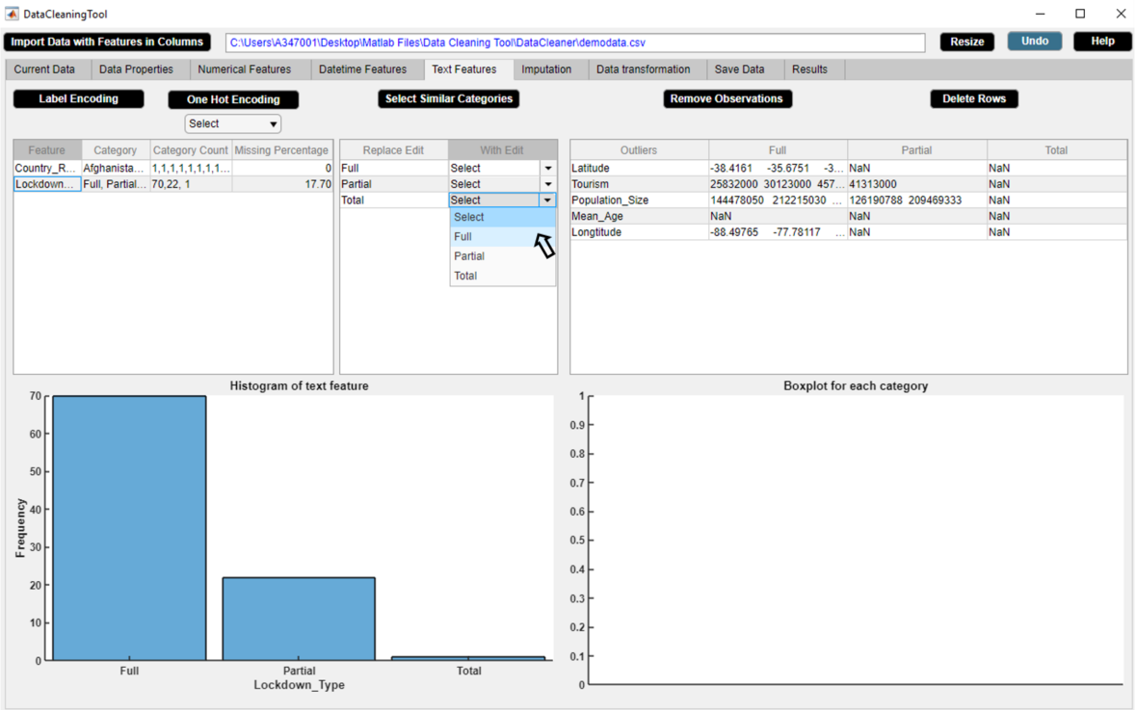


Figure B.74: Step 2. Select Similar Categories Button

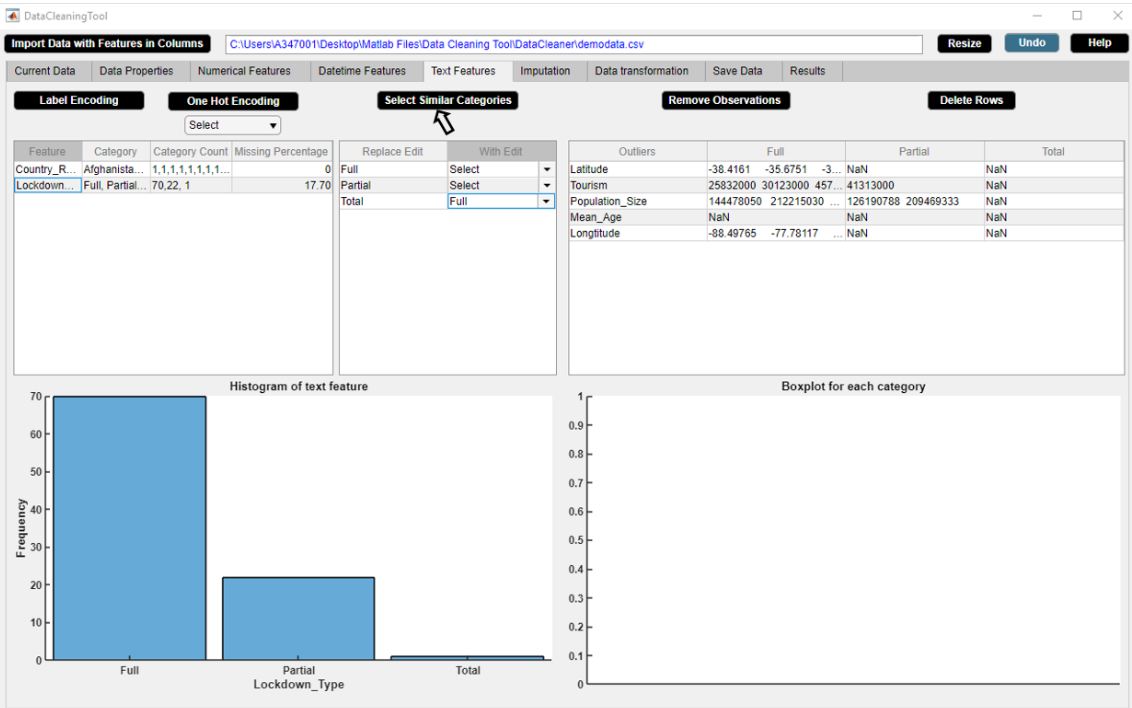


Figure B.75: Step 3. Select Similar Categories Button

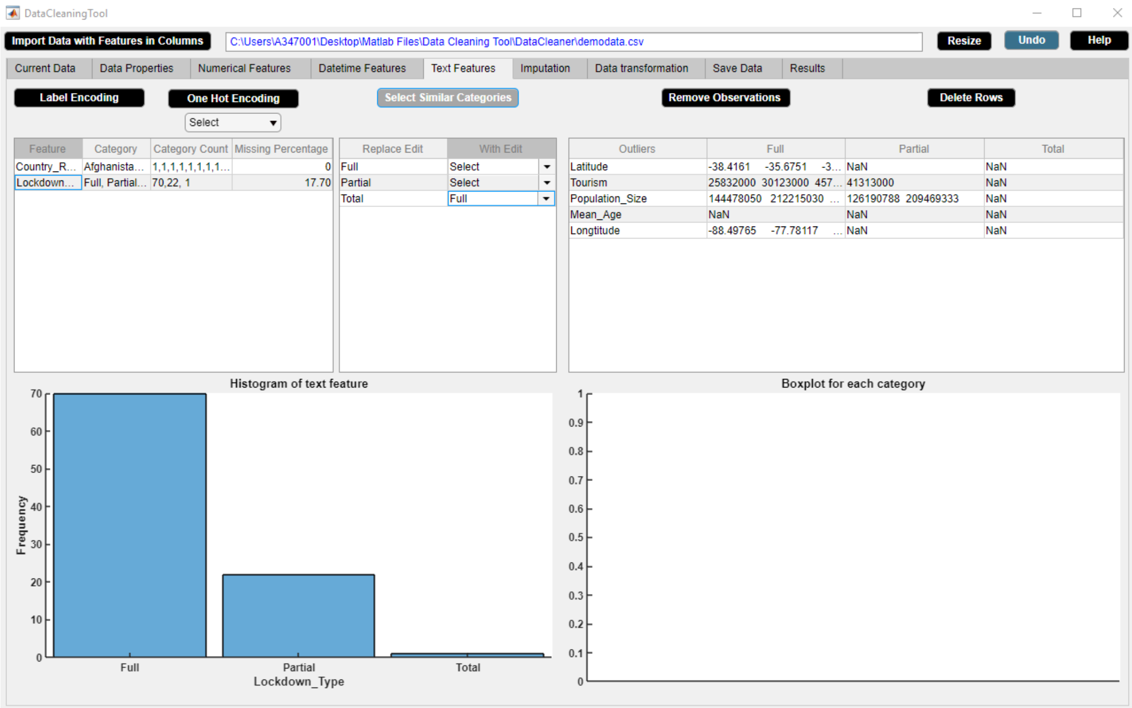


Figure B.76: Step 4. Select Similar Categories Button

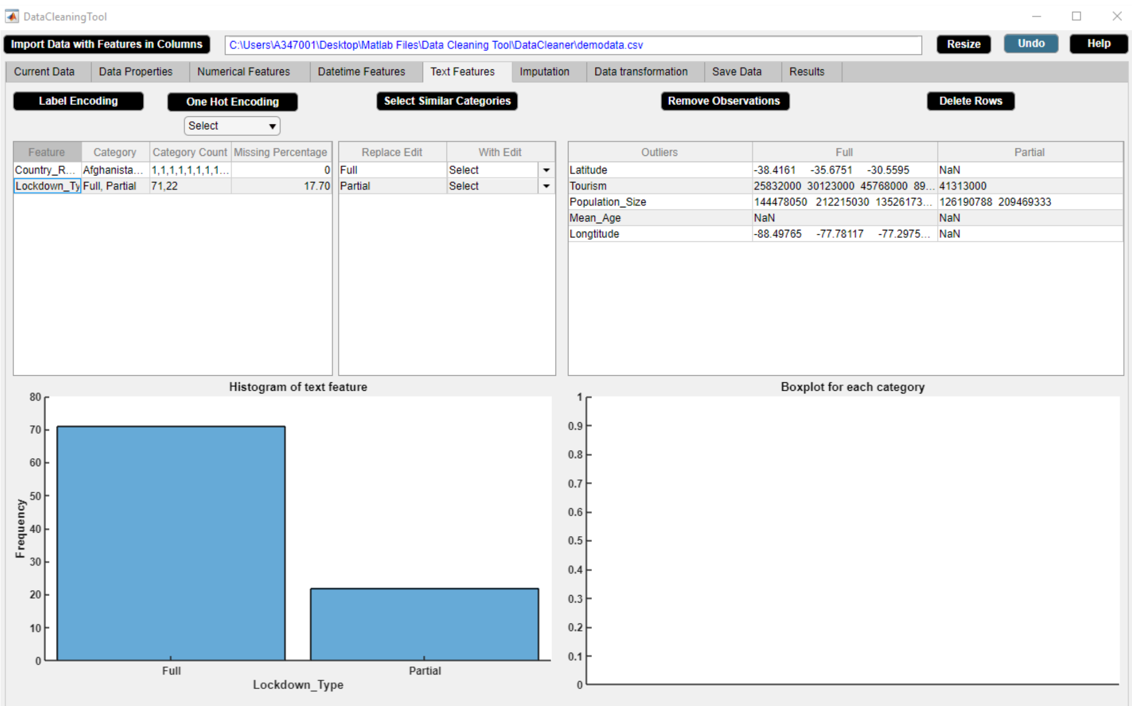


Figure B.77: Step 5. Select Similar Categories Button

B.6.2 Text Feature Cell Selection Button

Displays histogram of a text feature.
Application

- Outlier visualization technique.

Example

Step 1: Select a text feature from **Feature** column of the text features descriptive statistics table.
Step 2: A histogram of the selected text feature appears in the lower left side of the **Text Features** widget. Select a numerical feature from **Outliers** column of the right hand side table.
Step 3: A box plot of the selected numerical feature versus the selected text feature appears in the lower right side of the **Text Features** widget.

We use **Text Feature Cell Selection** button to visualize the histogram of 'Lockdown_Type' feature and the box plot of 'Mean_Age' versus 'Lockdown_Type'. It can be seen from the histogram of 'Lockdown_Type' that there are more countries with 'Full' lockdown rather than with 'Partial' lockdown. It can be seen from the box plot of 'Mean_Age' versus 'Lockdown_Type' that 'Mean_Age' of the population is larger for the countries with 'Full' lockdown rather than for the countries with 'Partial' lockdown. Figures B.78-B.80 illustrate how to use **Text Feature Cell Selection** button.

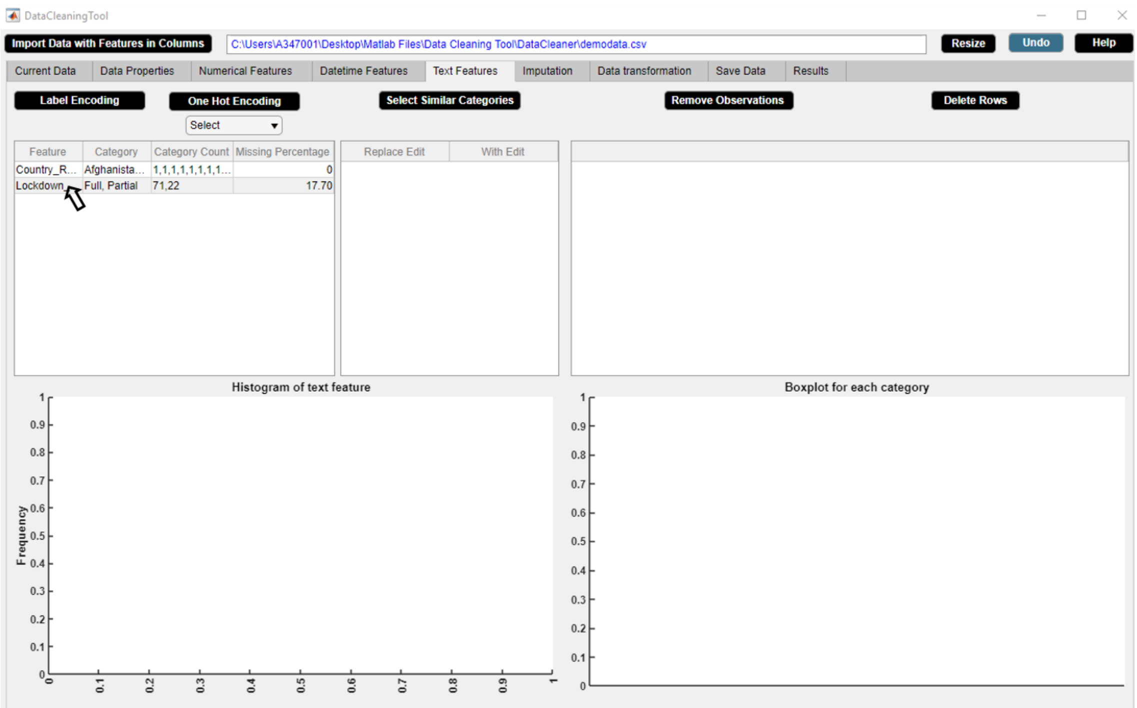


Figure B.78: Step 1. Text Feature Cell Selection Button

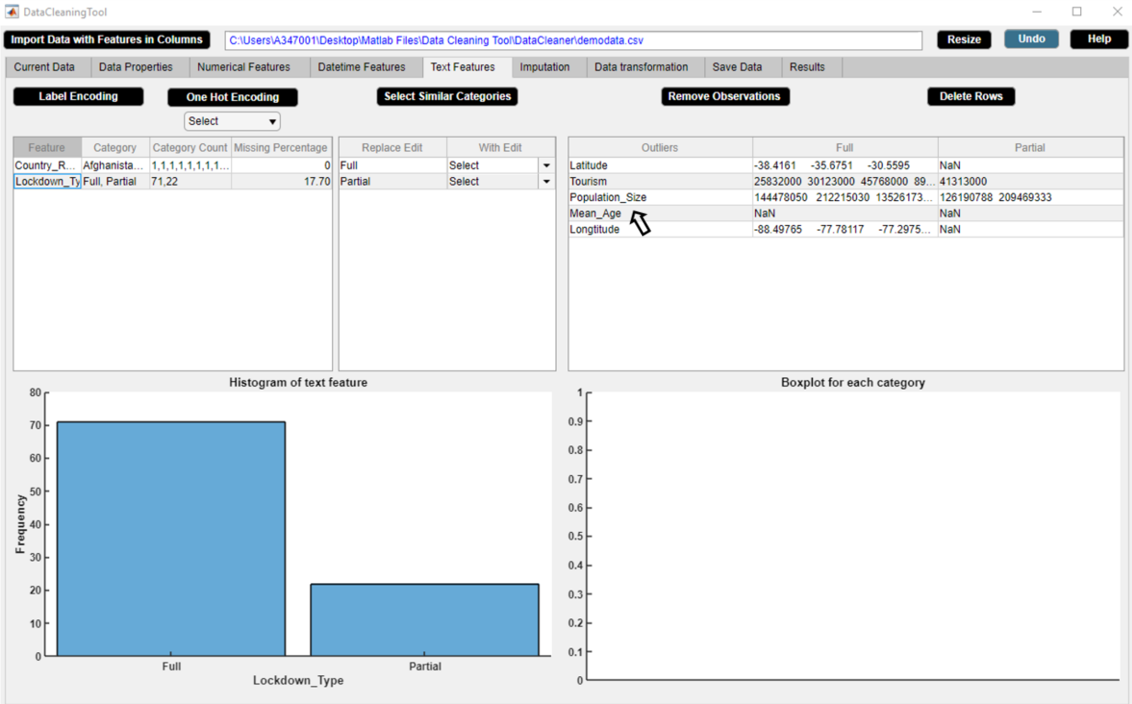


Figure B.79: Step 2. Text Feature Cell Selection Button

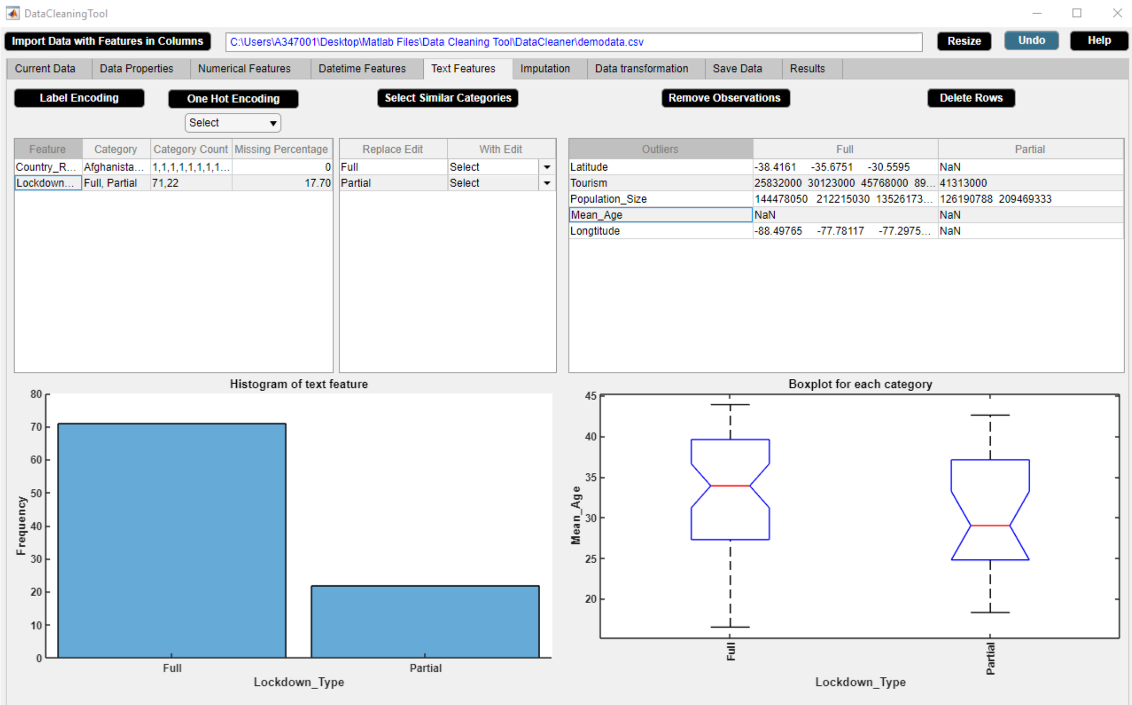


Figure B.80: Step 3. Text Feature Cell Selection Button

B.6.3 Label Encoding Button

Assigns each category of a categorical feature a value from 0 to n-1 where n is the number of categories. Note that label encoding is an encoding approach usually for handling ordinal categorical features.

Example

Step 1: Select a categorical feature from **Feature** column of the text features descriptive statistics table.

Step 2: Click **Label Encoding** button.

Step 3: **Label Encoding** button in use turns grey in color.

Step 4: **Label Encoding** button returns back to its original color once it completes its task.

Step 5: Check the change in **Current Data** widget.

We use **Label Encoding** button if we wish to label encode the categorical feature 'Lockdown_Type'. Figures B.81-B.85 illustrate how to use **Label Encoding** button.

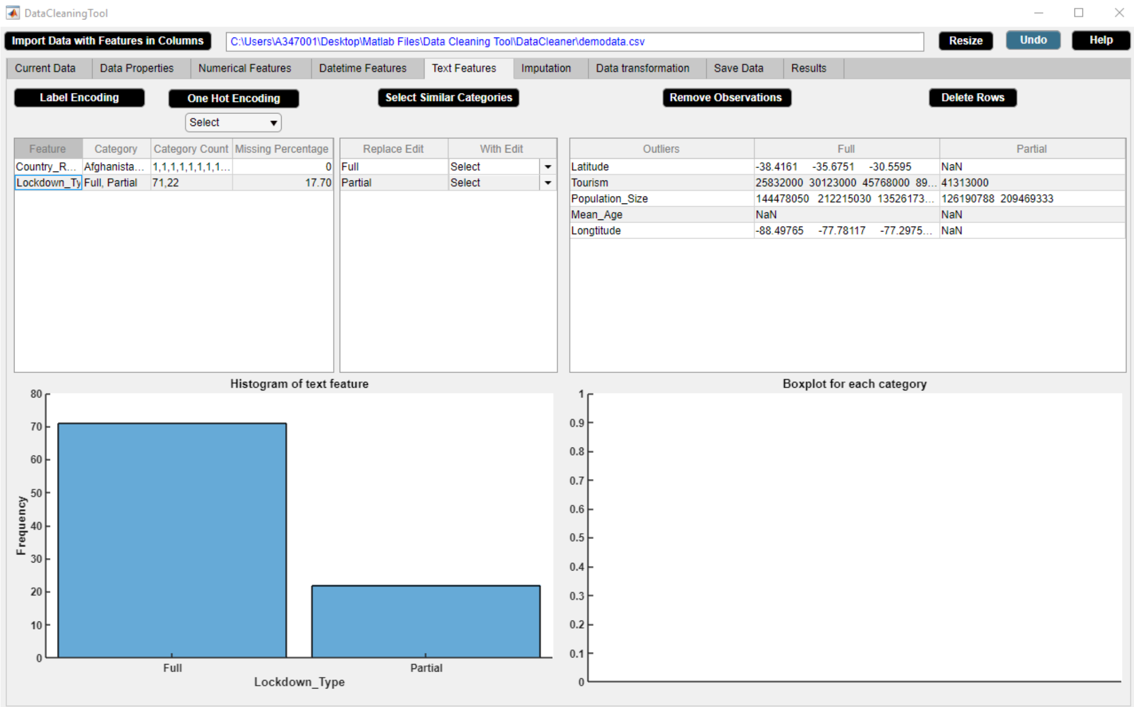


Figure B.81: Step 1. Label Encoding Button

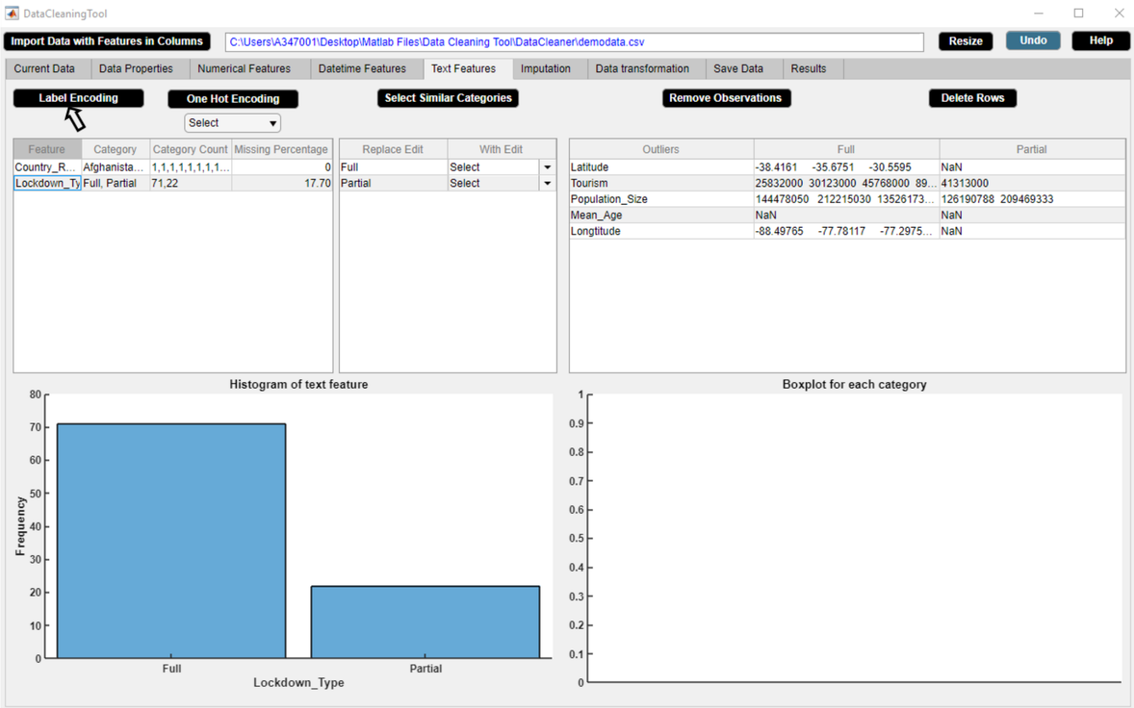


Figure B.82: Step 2. Label Encoding Button

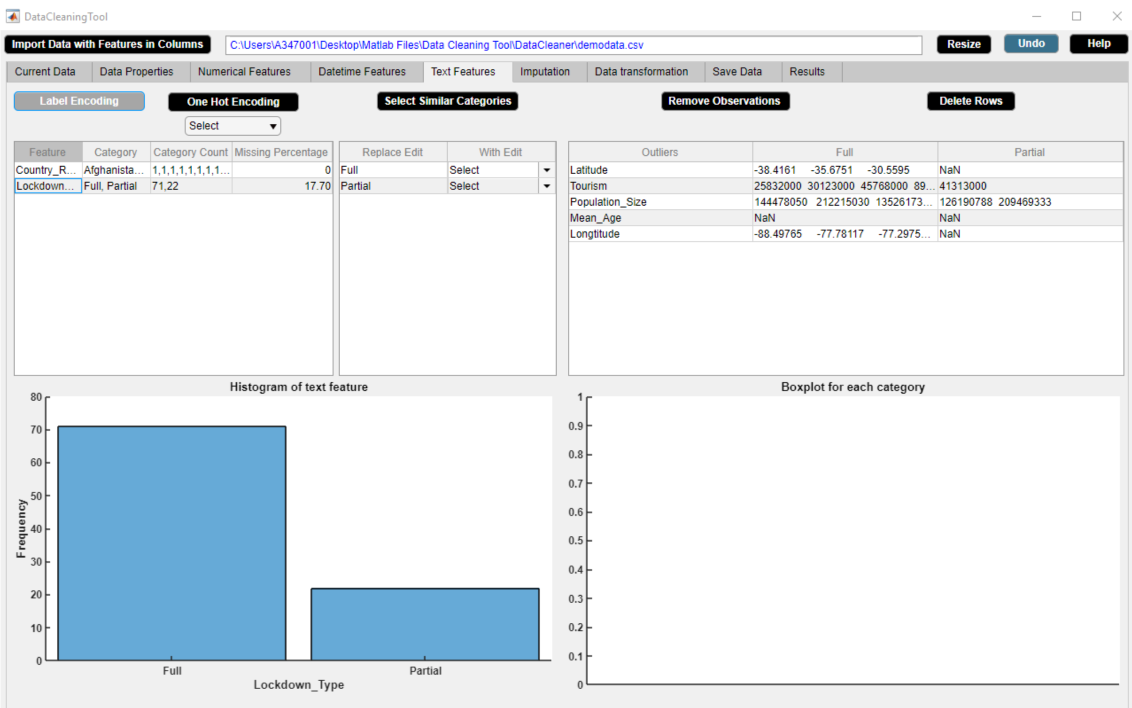


Figure B.83: Step 3. Label Encoding Button

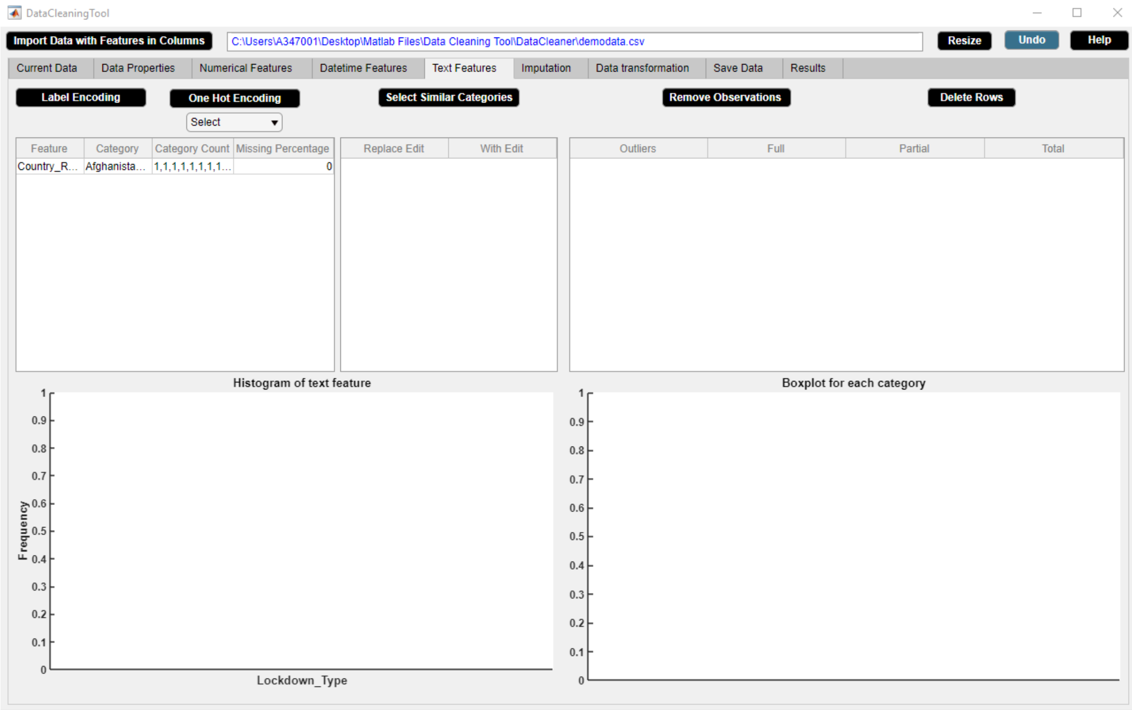


Figure B.84: Step 4. Label Encoding Button

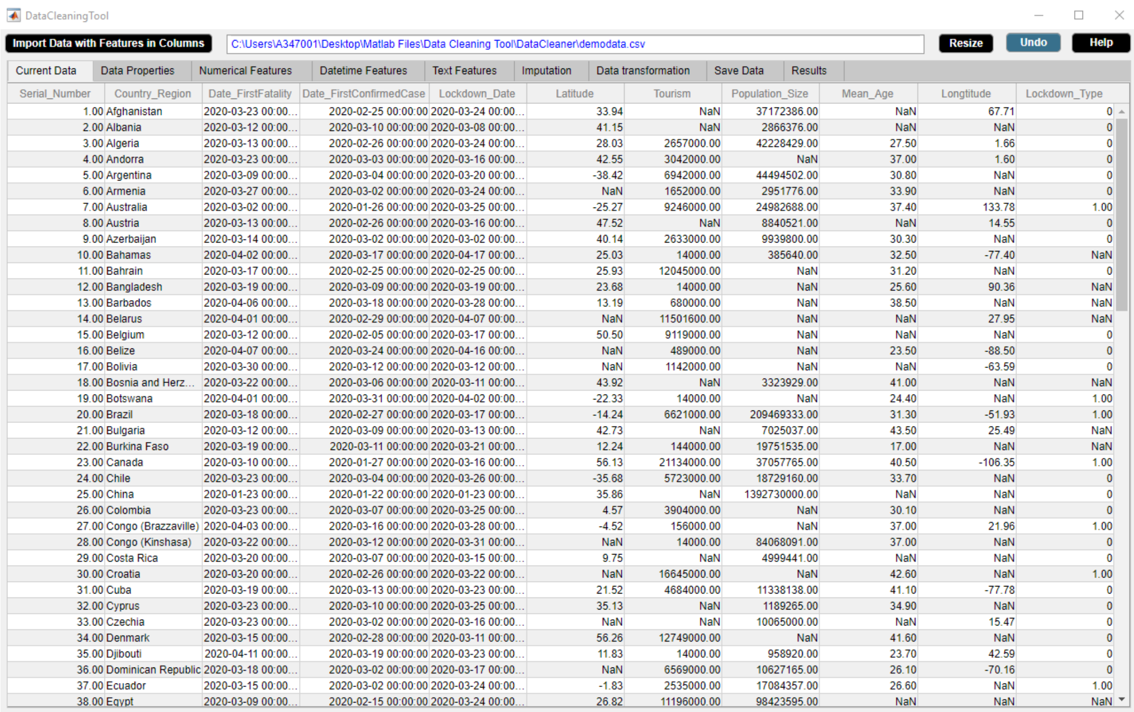


Figure B.85: Step 5. Label Encoding Button

B.6.4 One Hot Encoding Button

Transforms n categories to either n or n-1 dummy variables for a categorical feature. Note that one-hot encoding is an encoding approach usually for handling nominal categorical features.

Example

Step 1: Select a categorical feature from **Feature** column of the text features descriptive statistics table.

Step 2: Select any one option from **One Hot Encoding** dropdown menu. We transform n categories of a categorical feature to n dummy variables for methods such as singular value decomposition whereas n-1 dummy variables for methods such as regression.

Step 3: Click **One Hot Encoding** button.

Step 4: **One Hot Encoding** button in use turns grey in color.

Step 5: **One Hot Encoding** button returns back to its original color once it completes its task.

Step 6: Check the change in **Current Data** widget.

We use **One Hot Encoding** button if we wish to one hot encode the categorical feature 'Country_Region'. Figures B.86-B.91 illustrate how to use **One Hot Encoding** button.

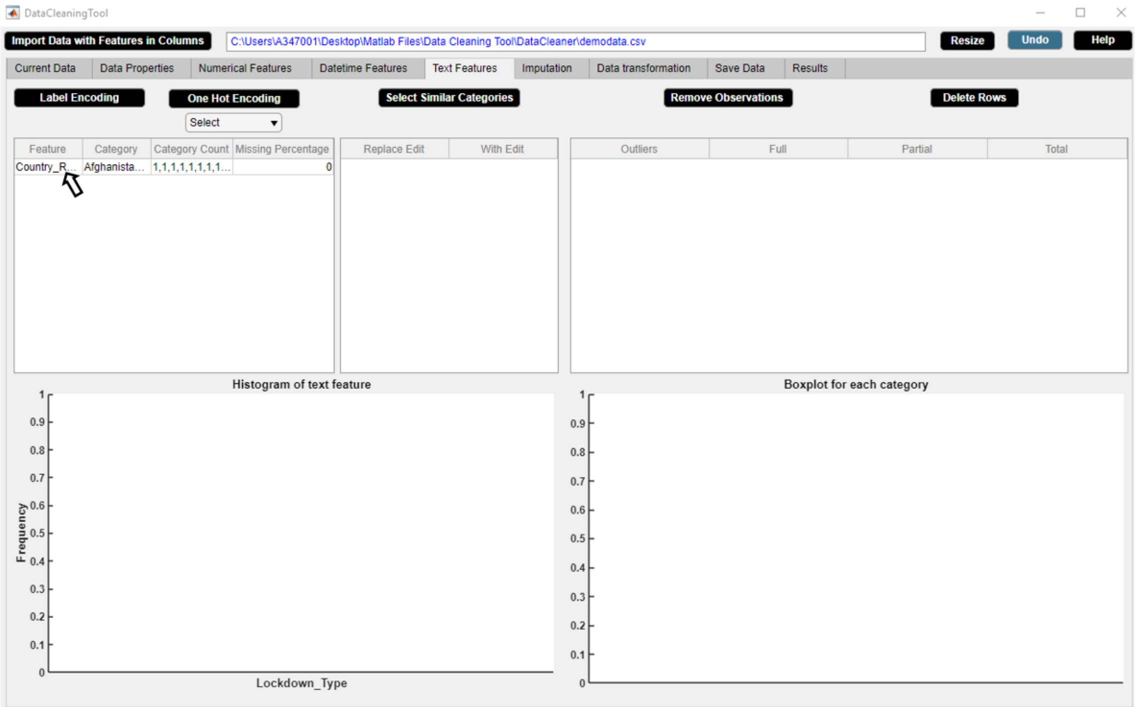


Figure B.86: Step 1. One Hot Encoding Button

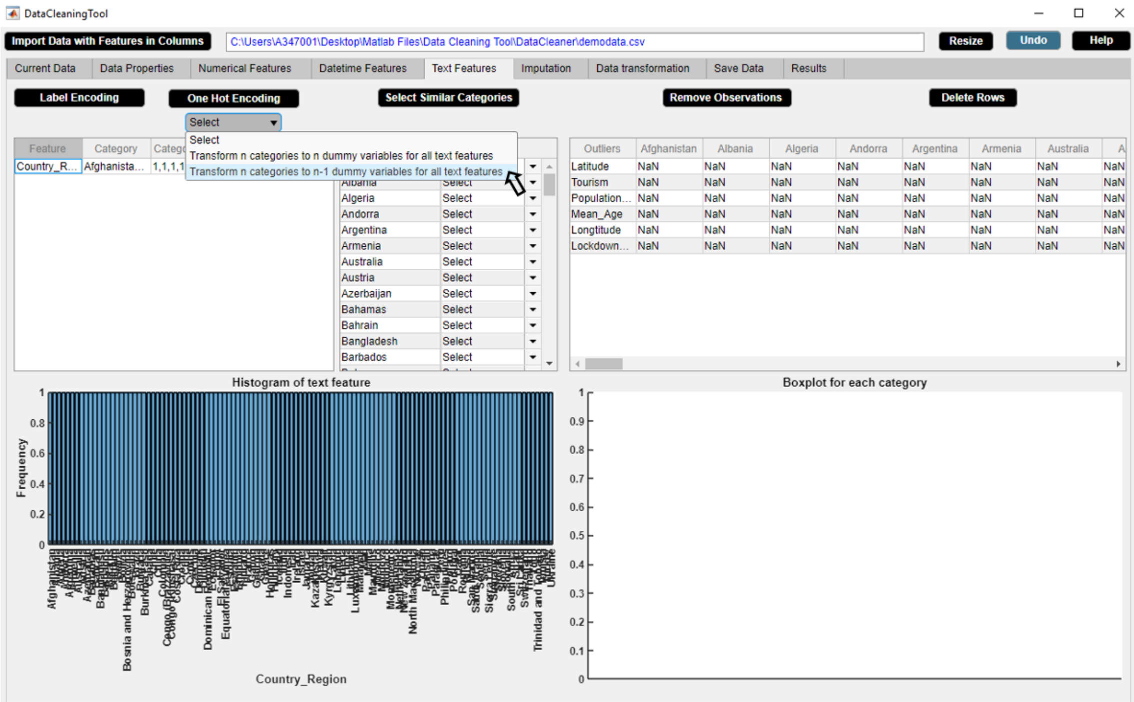


Figure B.87: Step 2. One Hot Encoding Button

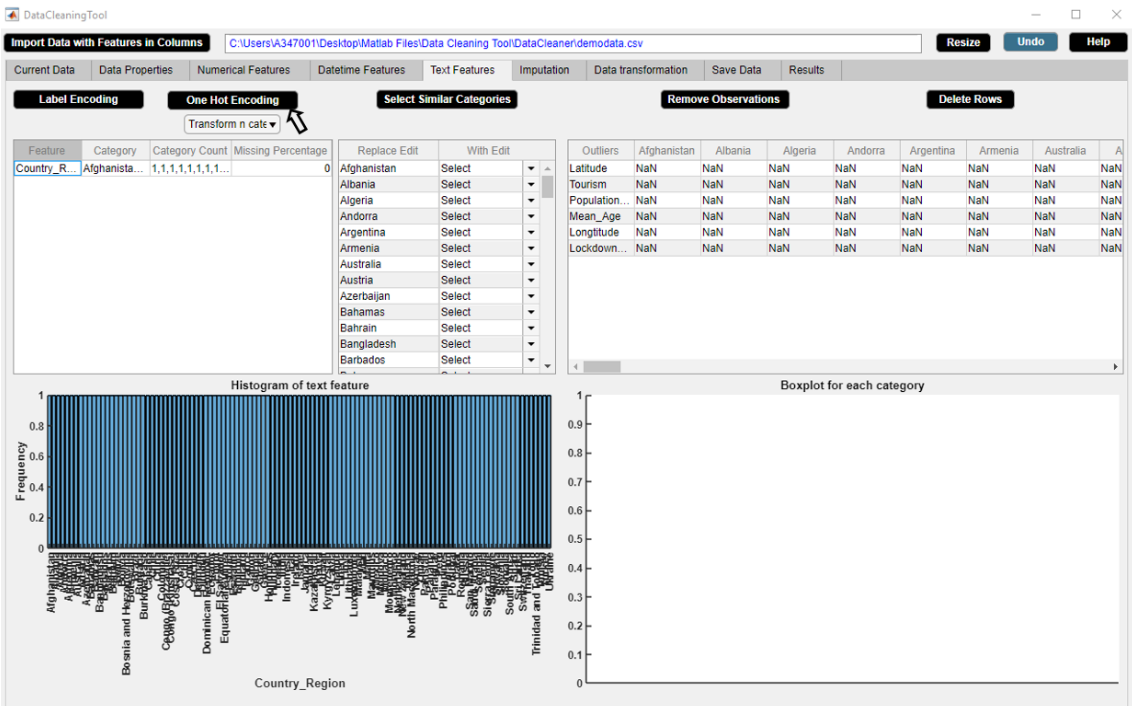


Figure B.88: Step 3. One Hot Encoding Button

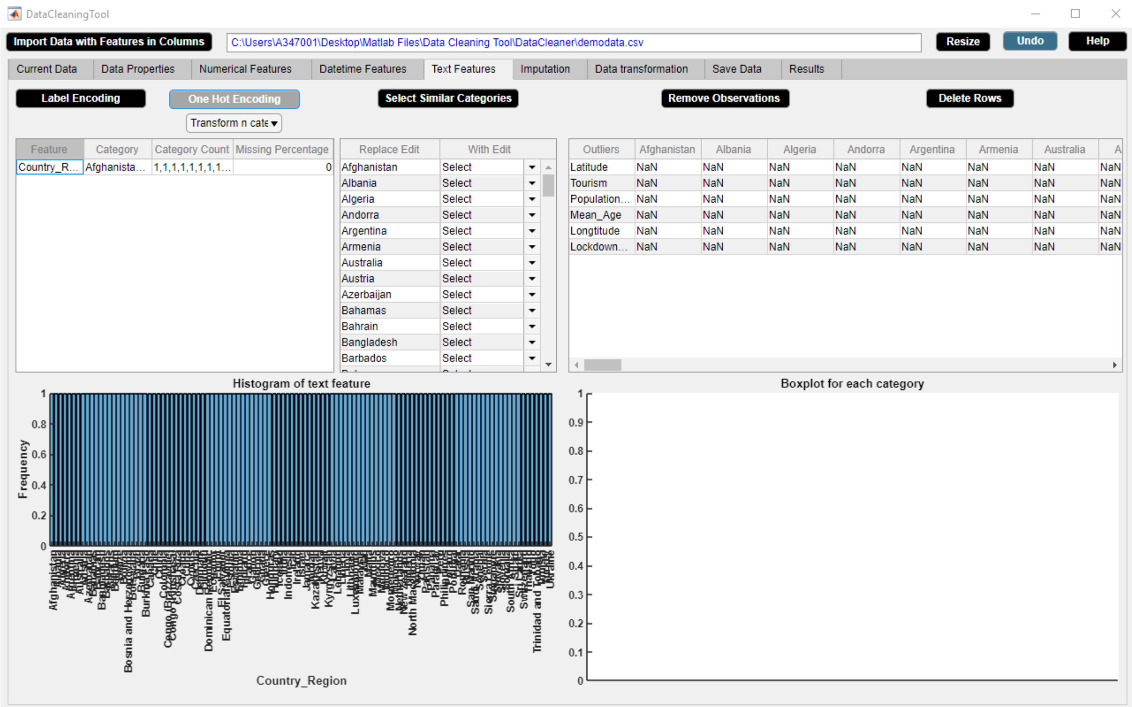


Figure B.89: Step 4. One Hot Encoding Button

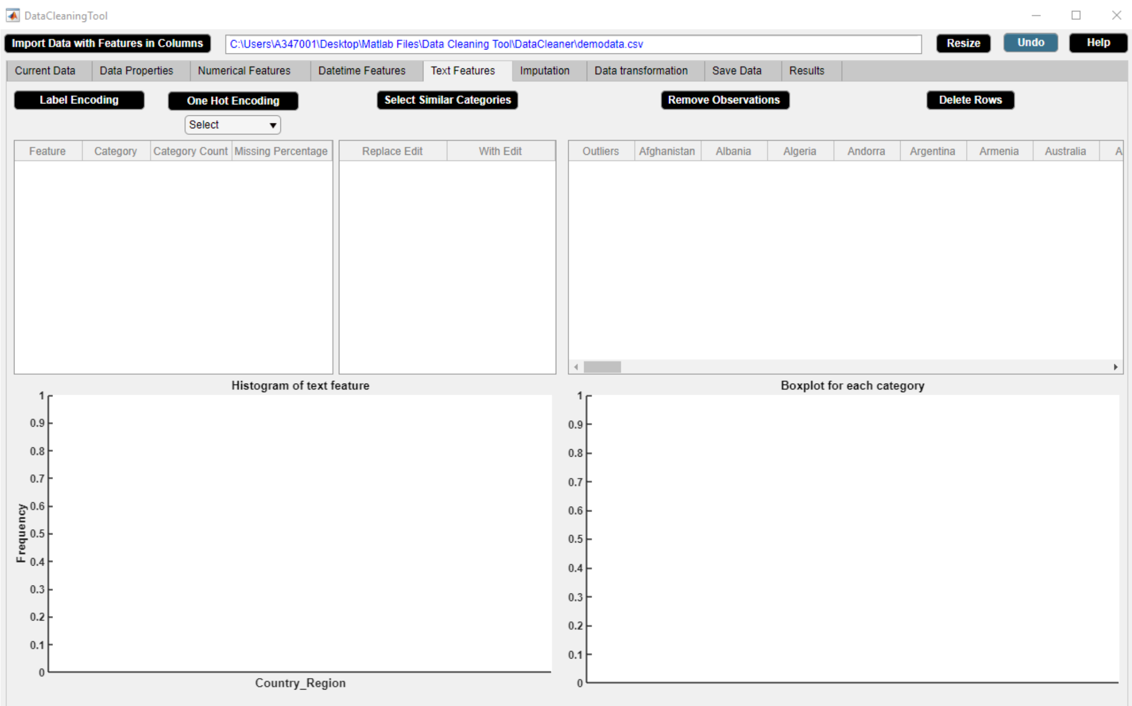


Figure B.90: Step 5. One Hot Encoding Button

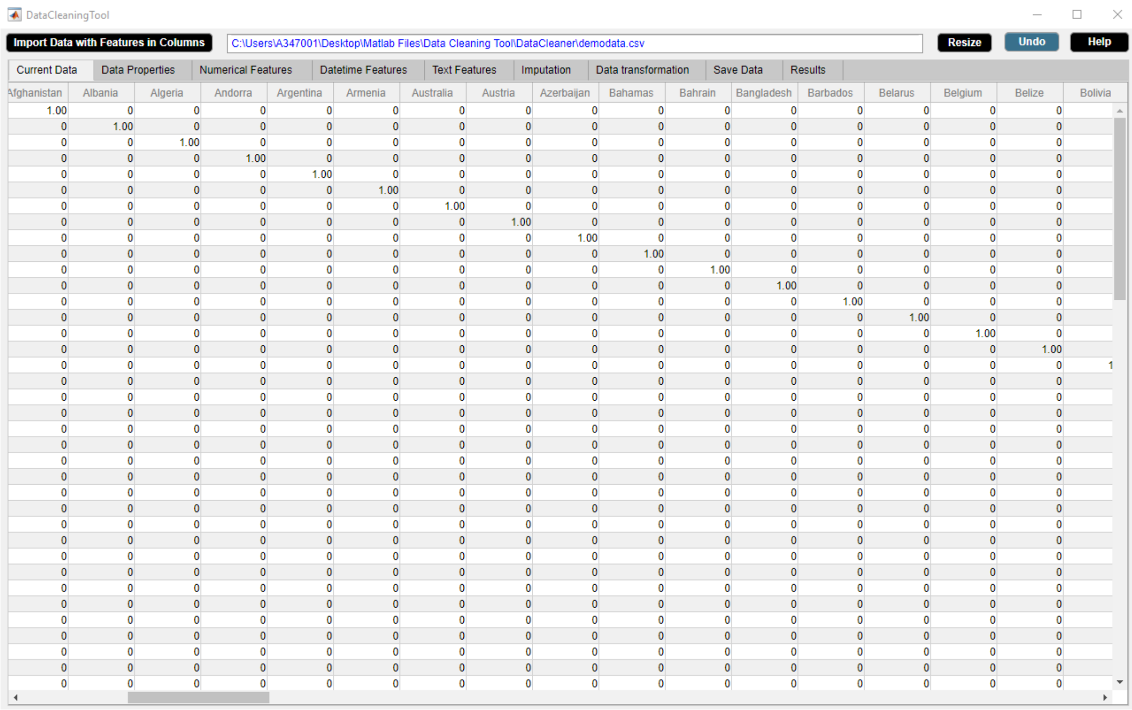


Figure B.91: Step 6. One Hot Encoding Button

B.6.5 Remove Observations Button

Replaces outliers by missing values.
Application
- Removes outliers.

B.6.6 Delete Rows Button

Deletes rows with outliers.
Application
- Deletes rows containing outliers.

B.7 Imputation Widget

The Imputation widget displays information about the missing data and the expected error of imputation for numerical and categorical features. The Imputation widget is shown in figure B.92. The properties of the Imputation widget are as follows.

- The widgets shows information about missing data such as percentage of missing data, expected error of imputation for numerical and categorical features. The performance analysis results of the missForest method discussed in chapter 4 is used to predict the expected error of imputation for numerical and categorical features for the specific ratio of data and percentage of missing data.
- The widget also presents the missing observations percentage table and the missingness plot.
- If datetime observations are missing, a message stating that datetime imputation is possible appears in red color in the lower side of the Imputation widget.
- The information of the missing data in the widget gets updated after each activity.

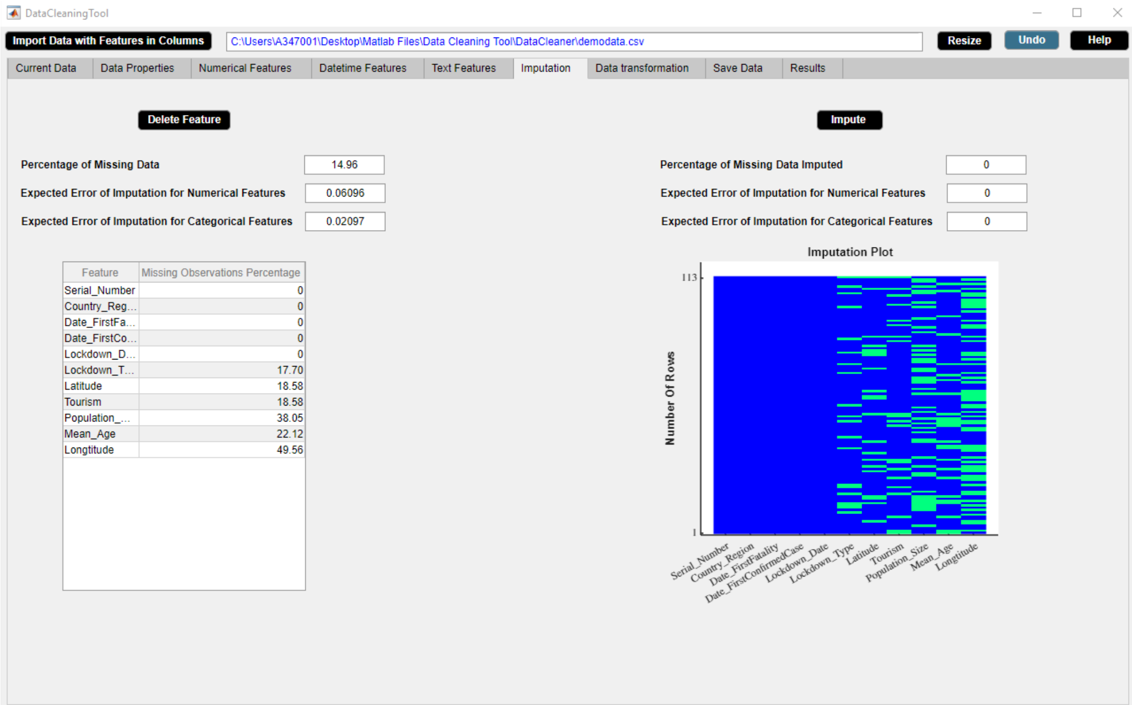


Figure B.92: Imputation Widget.

B.7.1 Delete Feature Button

Delete a feature from data.

Application

- Delete an unwanted or irrelevant feature.
- Delete a feature containing a large number of missing observations.

Example

Step 1: Select a feature from **Feature** column of missing observations percentage table.
Step 2: Click **Delete Feature** button.
Step 3: **Delete Feature** button in use turns grey in color.
Step 4: **Delete Feature** button returns back to its original color once it completes its task.

In the example data, 'Longitude' has a large number of missing values. We use **Delete Feature** button to delete 'Longitude' feature. Figures B.93-B.96 illustrate how to use **Delete Feature** button.

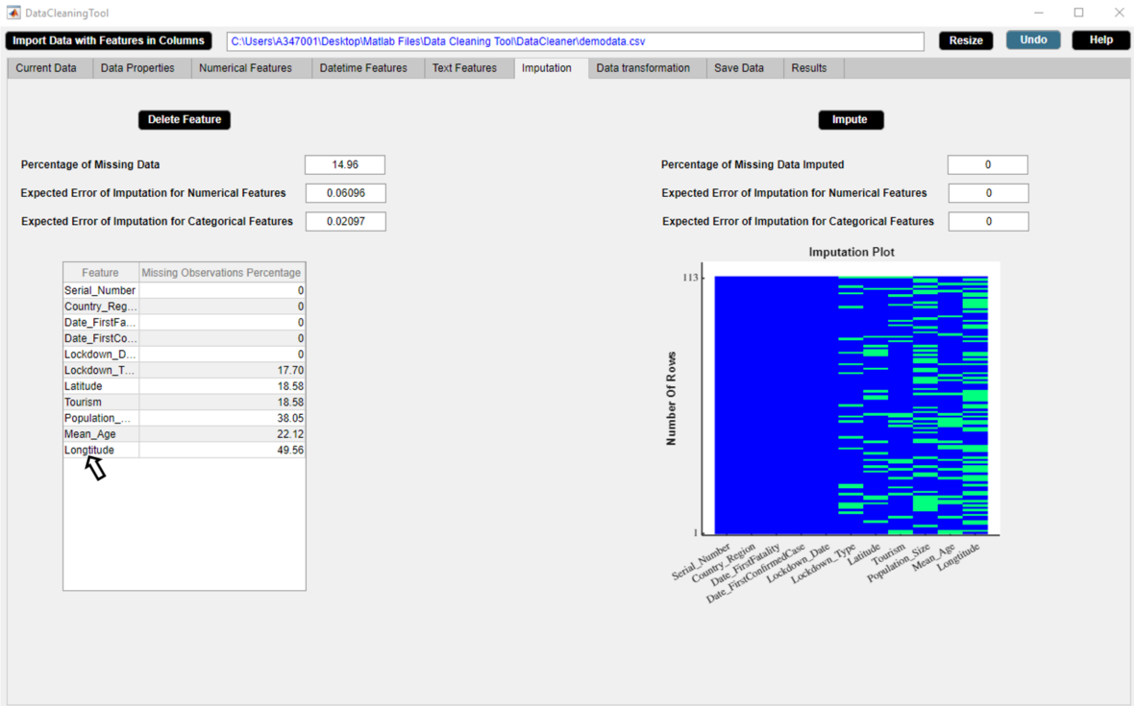


Figure B.93: Step 1. Delete Feature Button

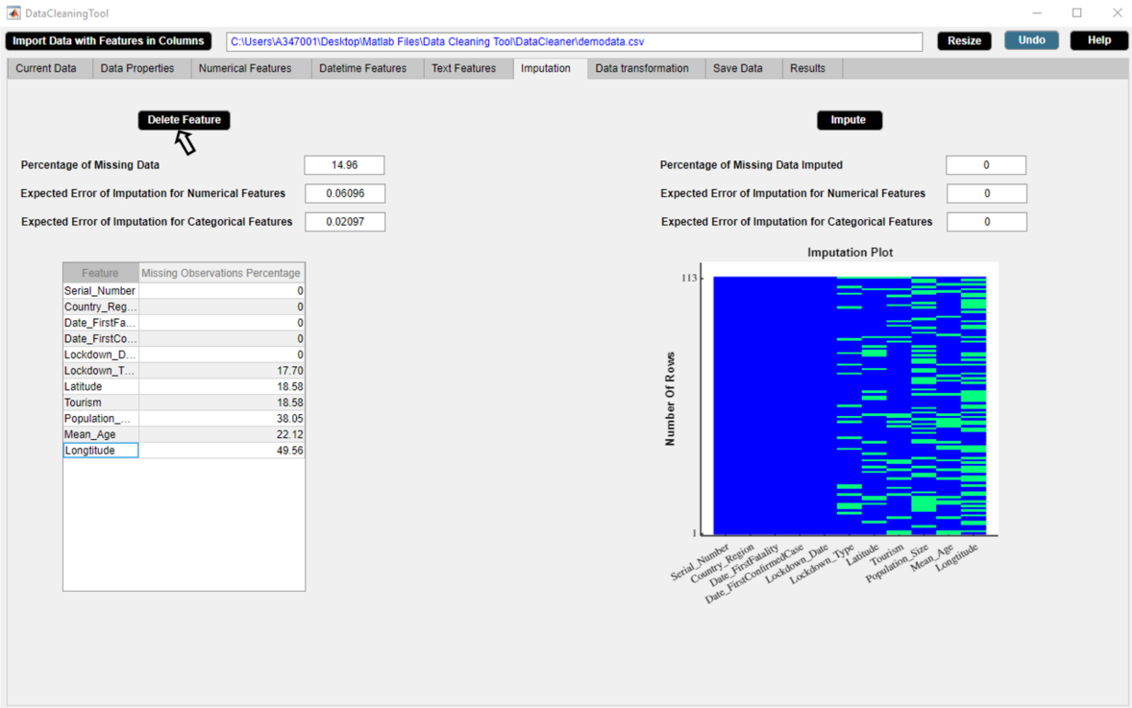


Figure B.94: Step 2. Delete Feature Button

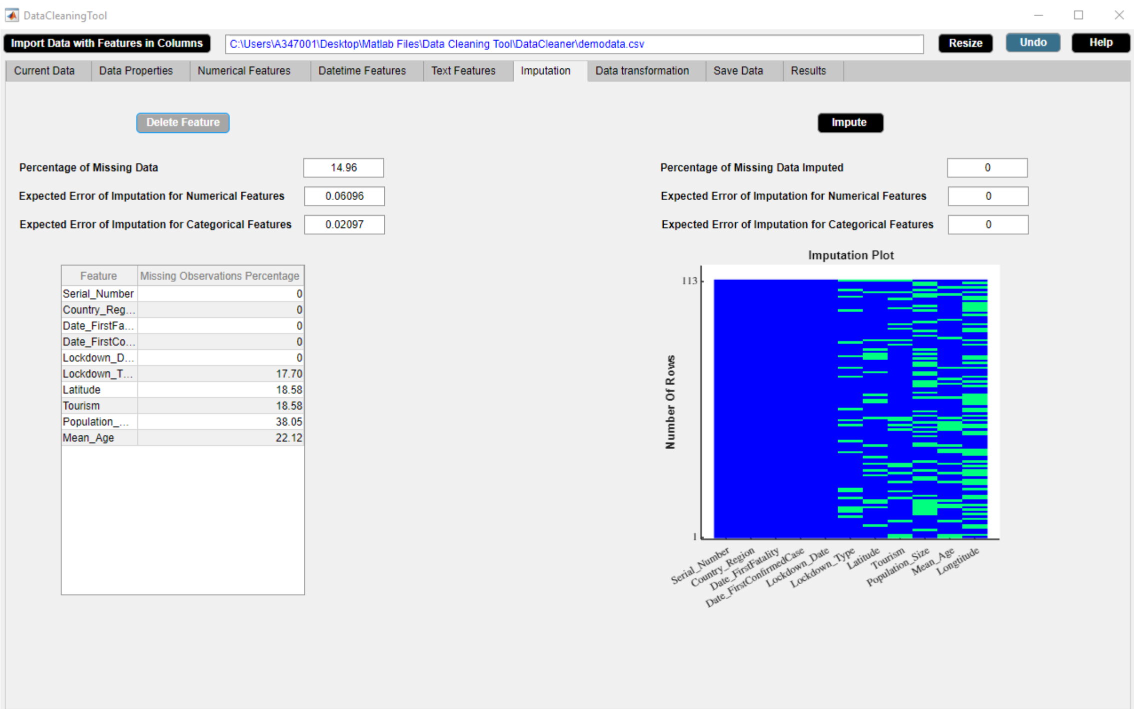


Figure B.95: Step 3. Delete Feature Button

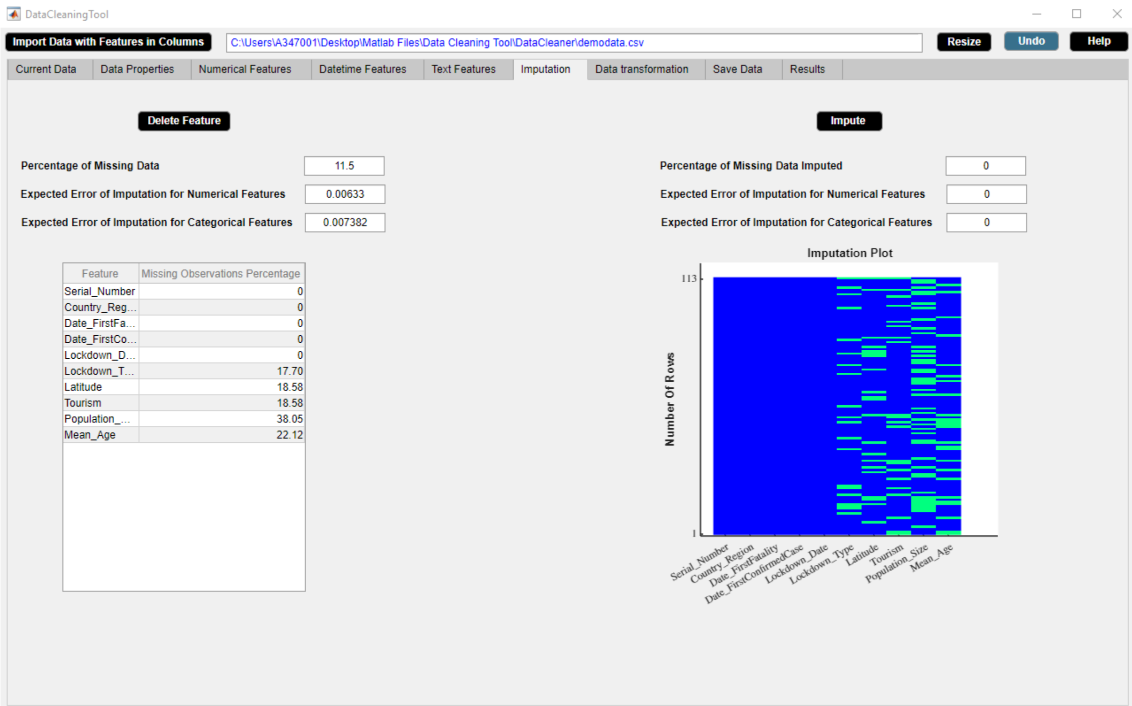


Figure B.96: Step 4. Delete Feature Button

B.7.2 Impute Button

Replaces missing values by estimated ones using missForest algorithm.
Application

- Impute missing observations.

Example

Step 1: Click **Impute** button.
Step 2: **Impute** button in use turns grey in color. If datetime observations are missing, a message stating that datetime imputation is not possible appears in red color in the lower side of the **Imputation** widget.
Step 3: **Impute** button returns back to its original color once it completes its task.

We use **Impute** button to impute missing values in the example data. Figures B.97-B.99 illustrate how to use **Impute** button.

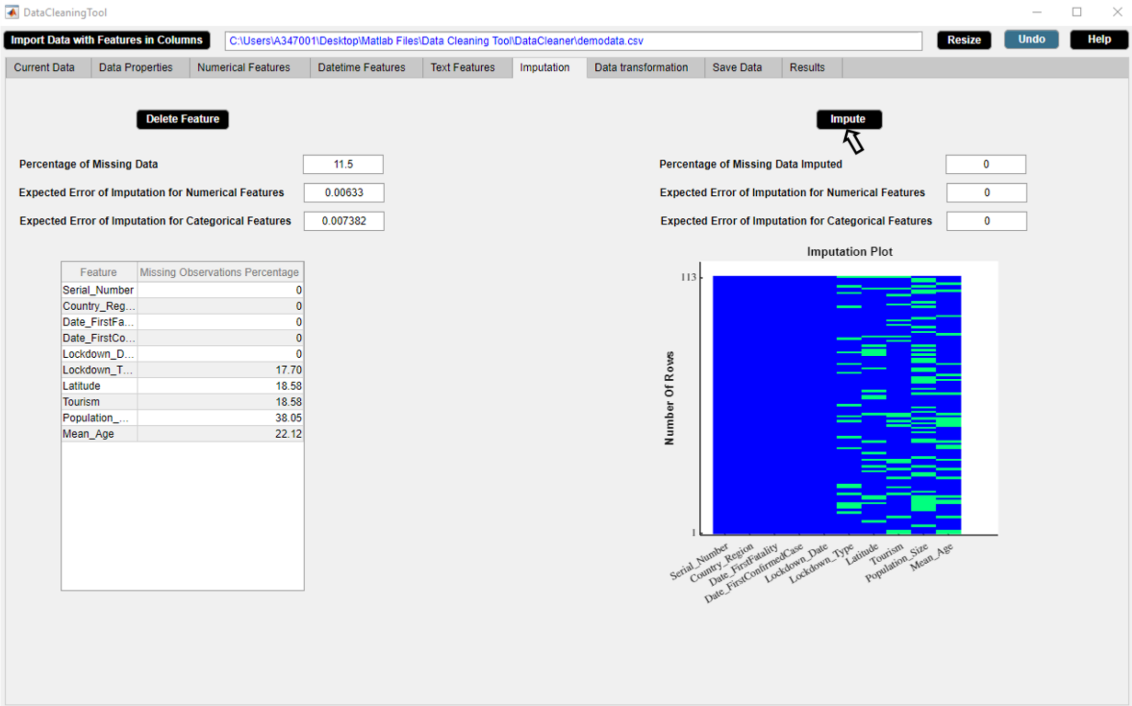


Figure B.97: Step 1. Impute Button

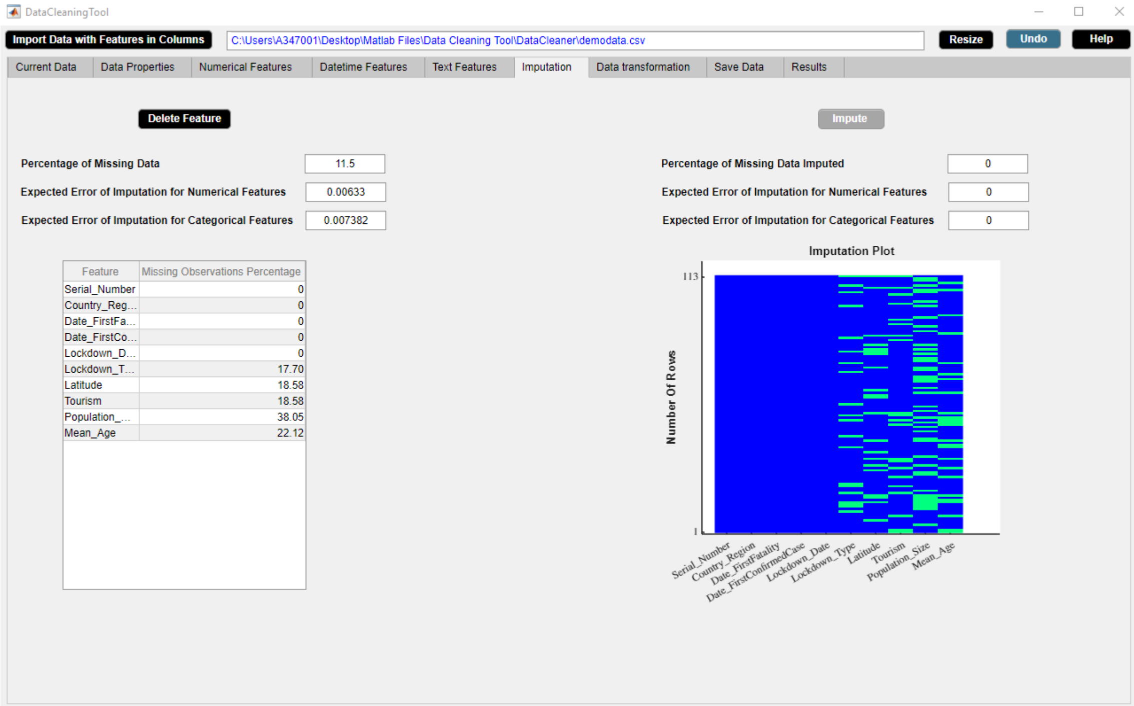


Figure B.98: Step 2. Impute Button

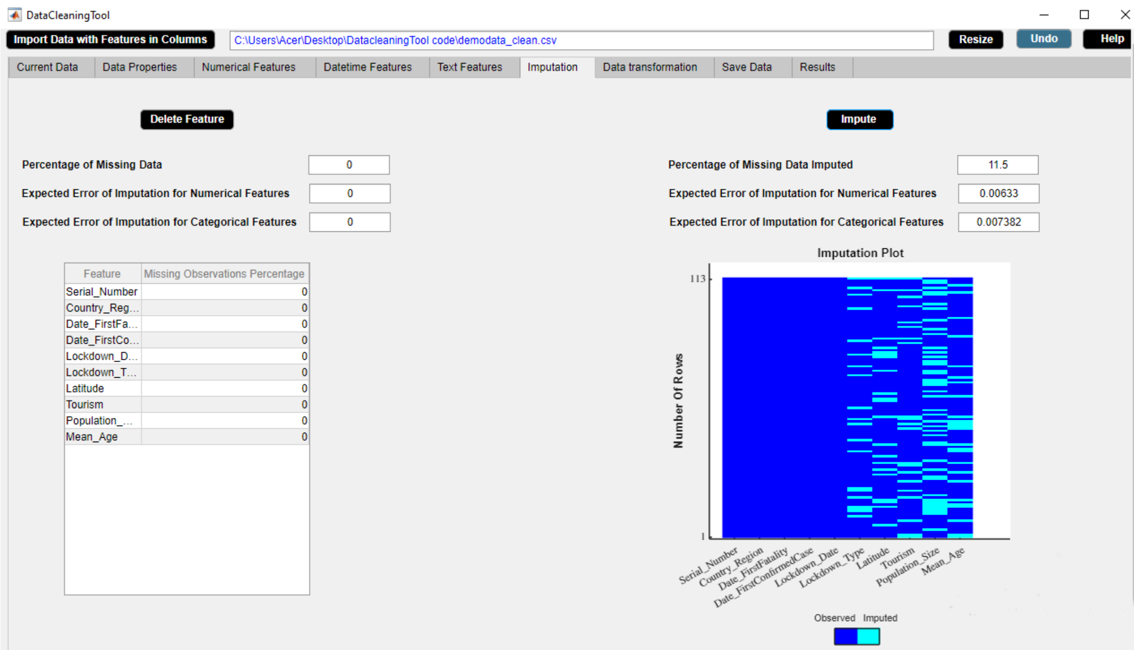


Figure B.99: Step 3. Impute Button

B.8 Data Transformation Widget

The Data Transformation widget displays the numerical features of the data on which data transformation can only be applied. The Data Transformation widget is shown in figure B.100. The properties of the Data Transformation widget are as follows.

- The widget presents the numerical features of the data.
- The numerical features of the data in the widget gets updated after each activity.

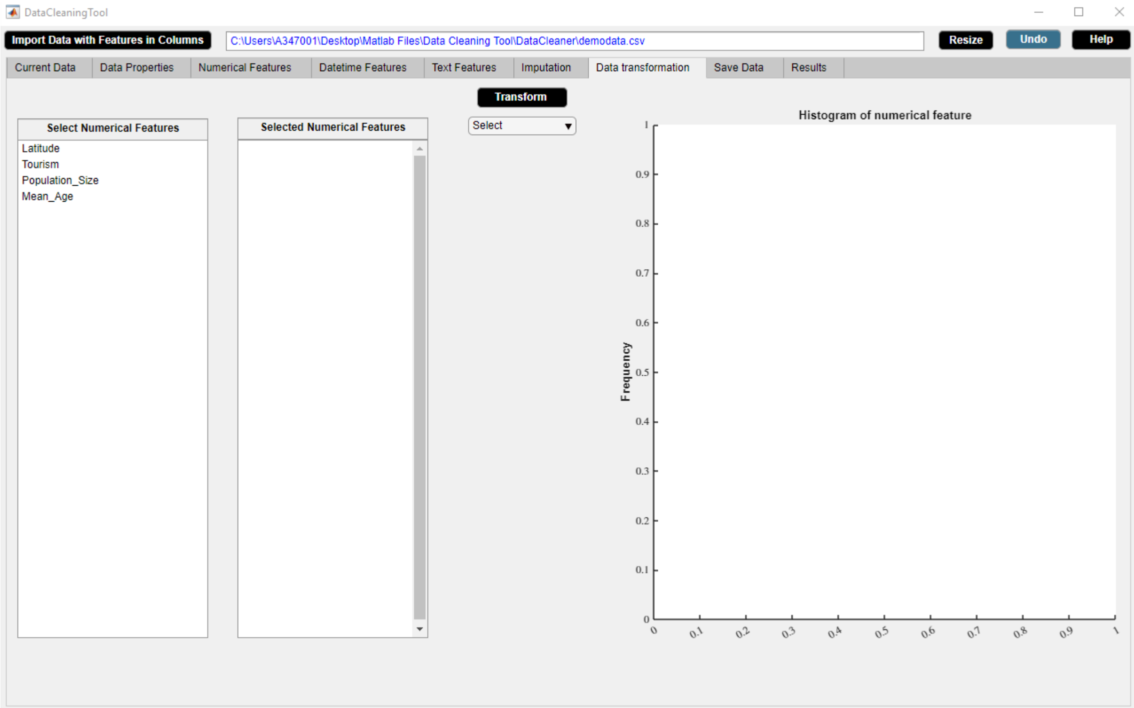


Figure B.100: Data Transformation Widget.

B.8.1 Transform Button

Standardize or normalize or logarithm or exponential or square root or inverse transform selected numerical features.

Application

- Outliers.

Example

Step 1: Select numerical feature/features from **Select Numerical Features** list box. Select an option from **Transform** dropdown menu. Here 'mean 0 and standard deviation' represents standardize, 'between 0 and 1' represents normalize, 'ln' represents natural logarithm transform, 'log10' represents logarithm base 10 transform, 'log2' represents logarithm base 2 transform, 'exp' represents natural exponential transform, 'sqrt' represents square root transform and 'reciprocal' represents inverse transform.

Step 2: Click **Transform** button.

Step 3: **Transform** button in use turns grey in color.

Step 4: **Transform** button returns back to its original color once it completes its task. A message regarding the percentage increase in missing data due to data transformation appears in red color in the lower side of the **Data Transformation** widget. Select the numerical feature from **Selected Numerical Features** list box.

Step 5: A histogram of the selected numerical feature appears in the right hand side of the **Data Transformation** widget.

We use **Transform** button to logarithmize 'Population_Size' in the example data. When we logarithmize 'Population_Size', the distribution becomes symmetric. Figures B.101-B.105 illustrate how to use **Transform** button.

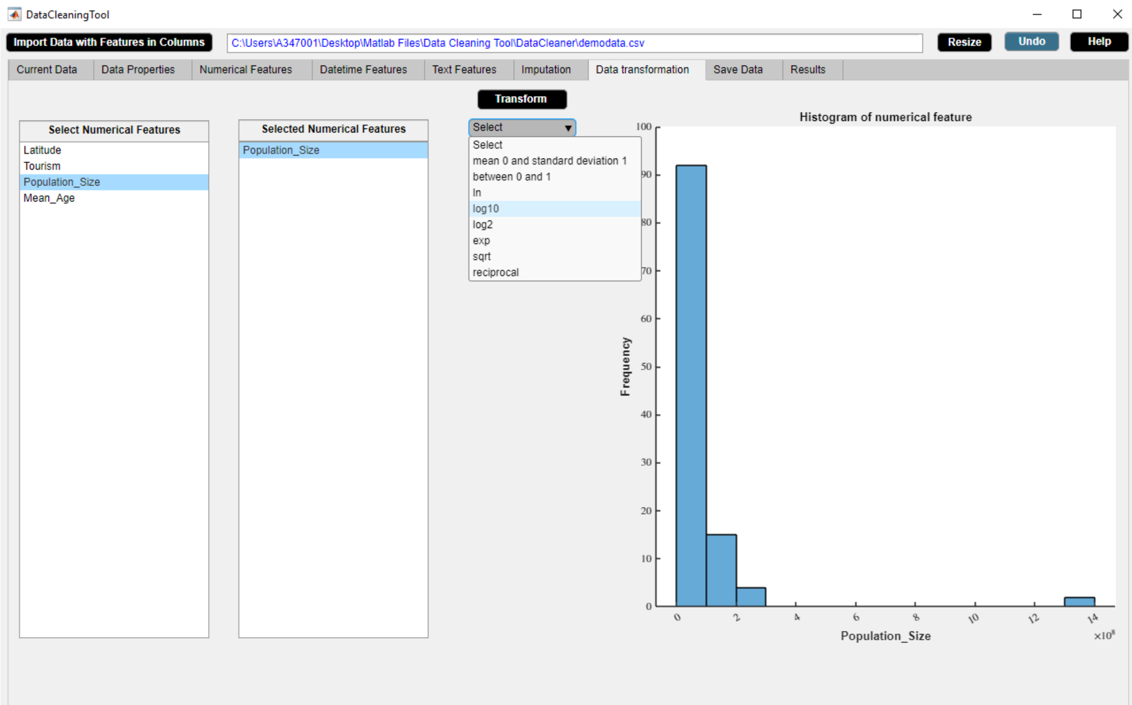


Figure B.101: Step 1. Transform Button

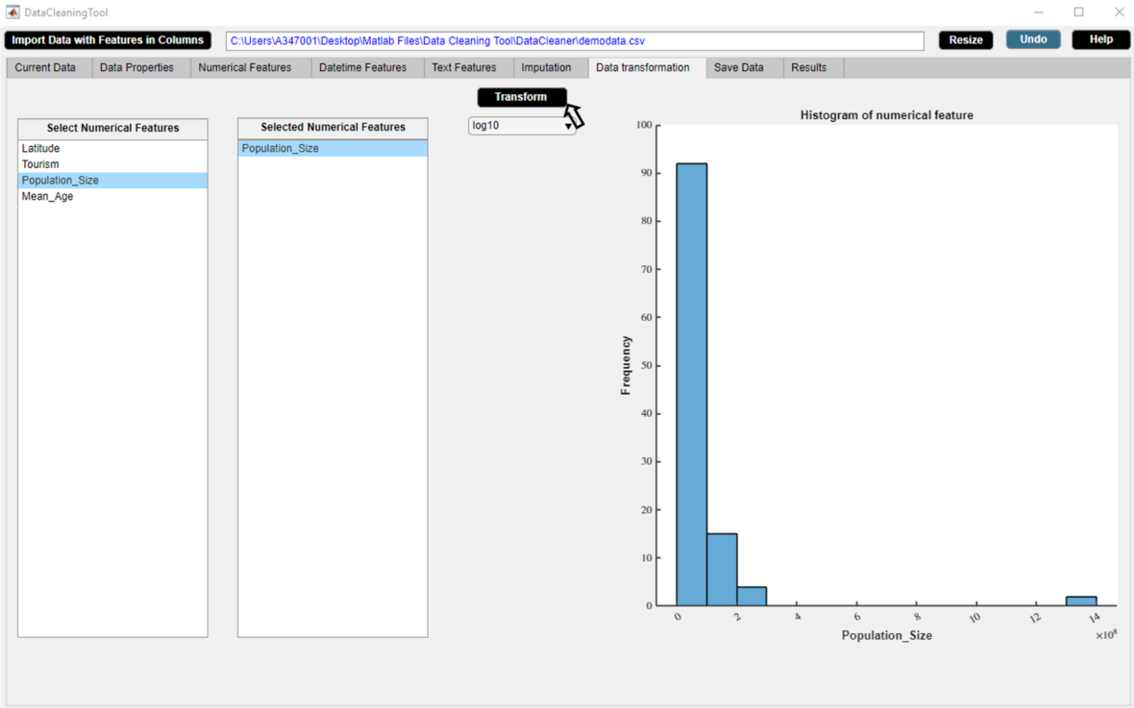


Figure B.102: Step 2. Transform Button

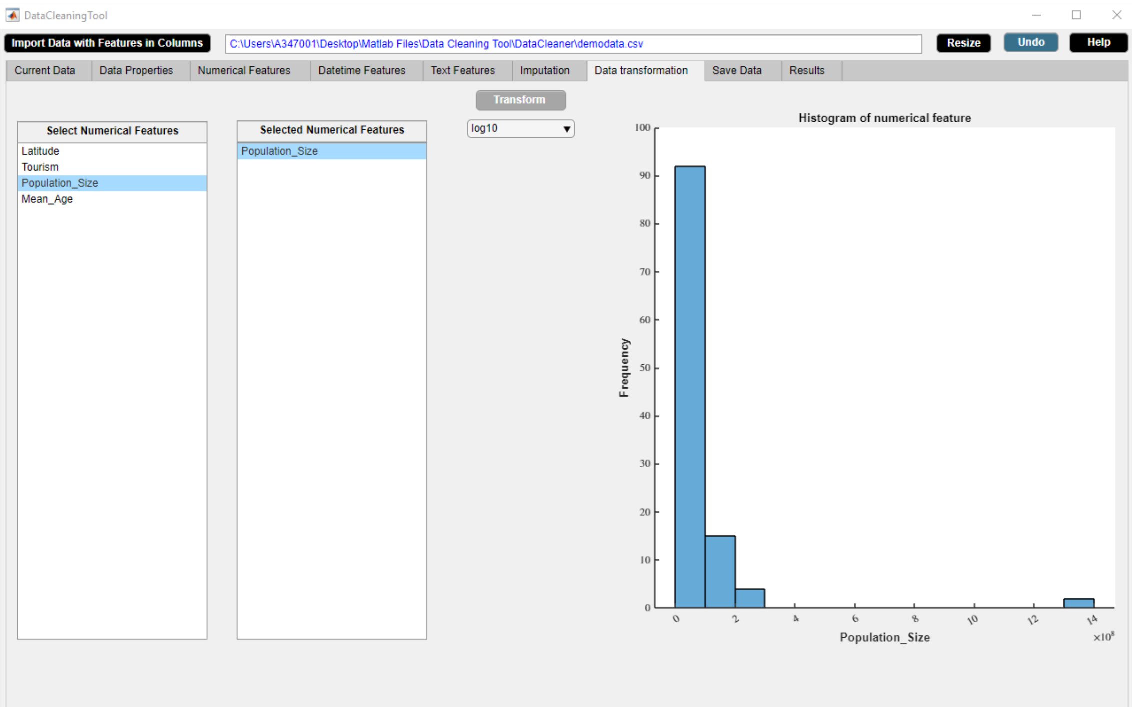


Figure B.103: Step 3. Transform Button

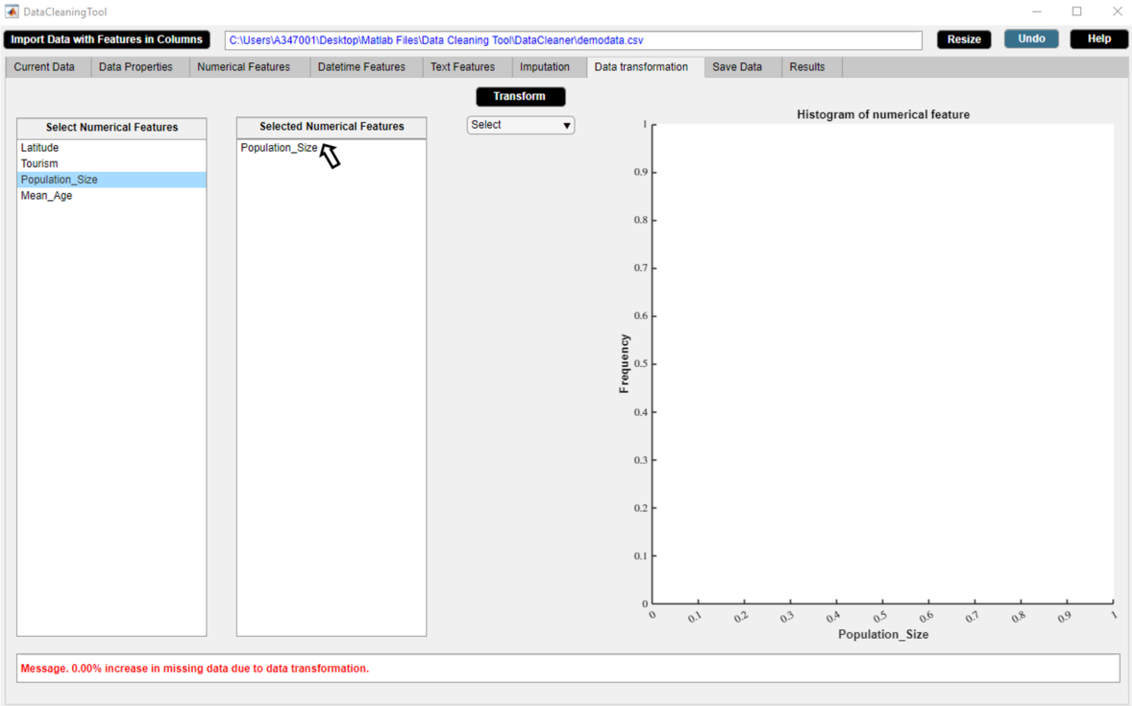


Figure B.104: Step 4. Transform Button

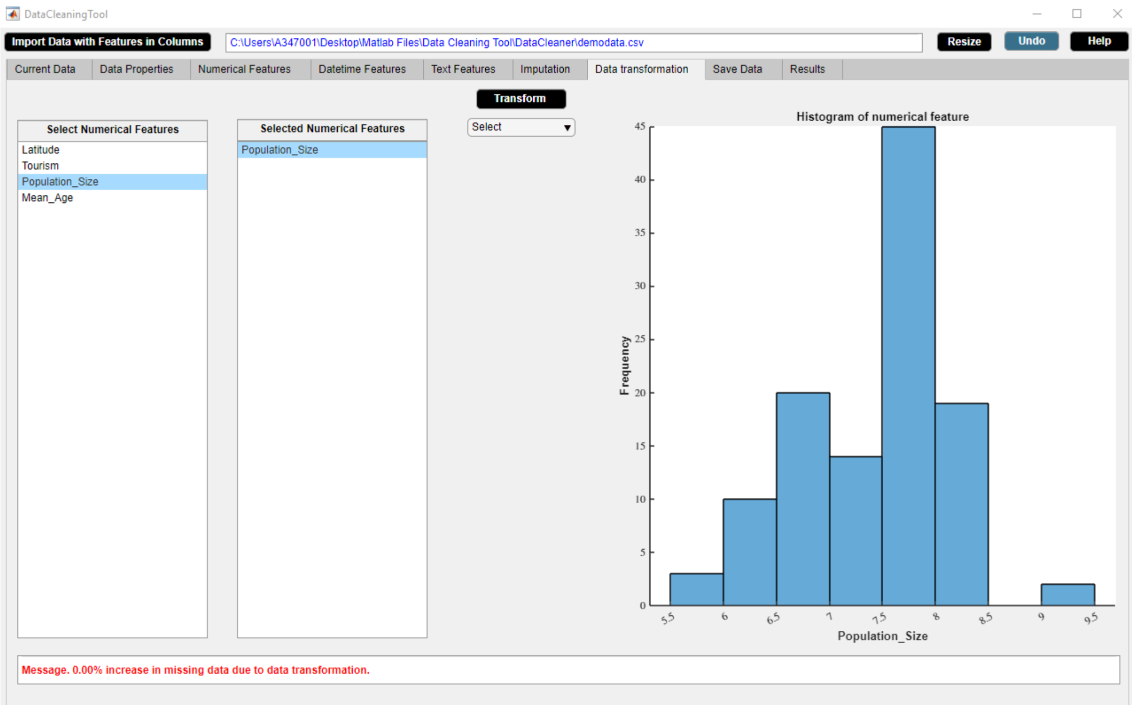


Figure B.105: Step 5. Transform Button

B.9 Save Data

The Save Data widget displays the full paths of the saved files. The Save Data widget is shown in figure B.106. The properties of the Save Data widget are as follows.

- The widget saves data in csv or xlsx format after data cleaning.
- Data can be saved for multiple times after each activity.
- The full paths of the saved files are displayed.

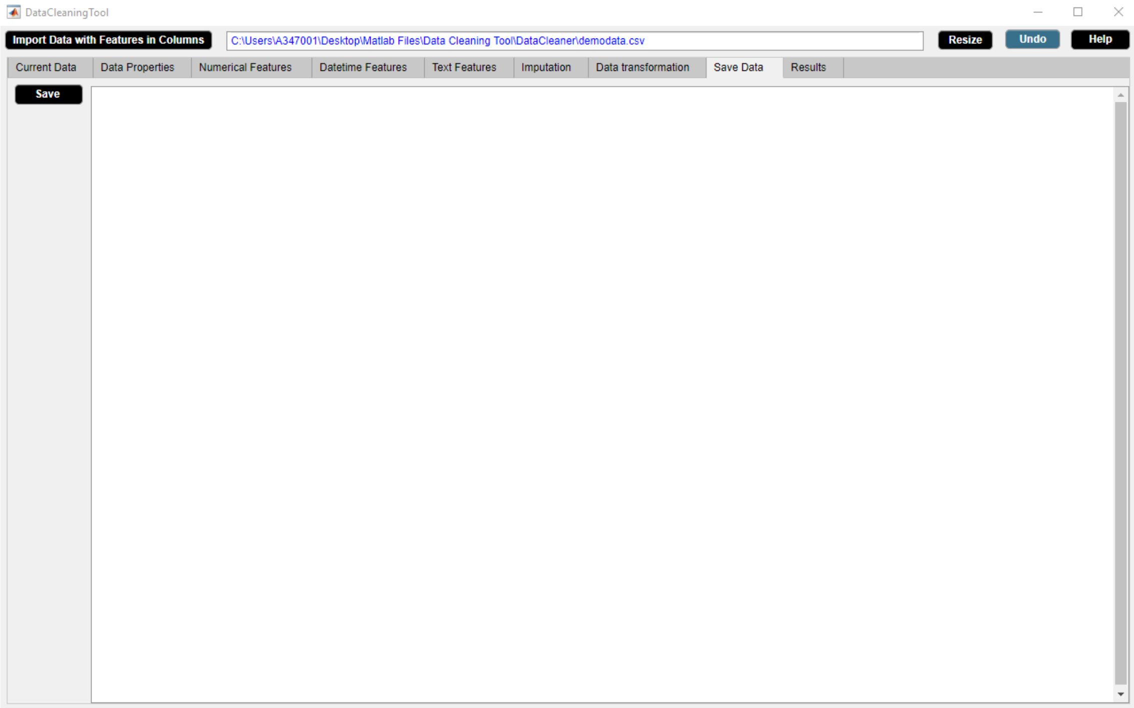


Figure B.106: Save Data Widget.

B.9.1 Save Button

Saves as comma-separated (.csv) or Excel (.xlsx) file.
Example

Step 1: Click **Save** button.
Step 2: **Save** button in use turns grey in color.
Step 3: **Save** button returns back to its original color once it completes its task.

We use **Save** button to save the example data in csv format. Figures B.107-B.110 illustrate how to use **Save** button.

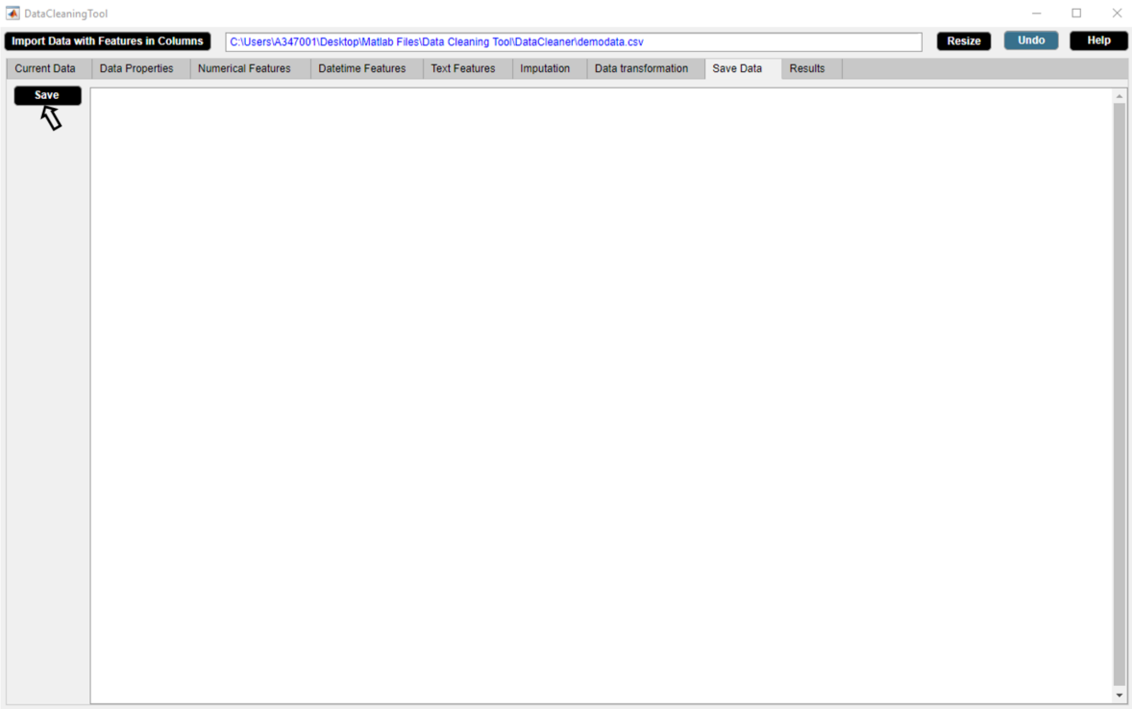


Figure B.107: Step 1. Save Button

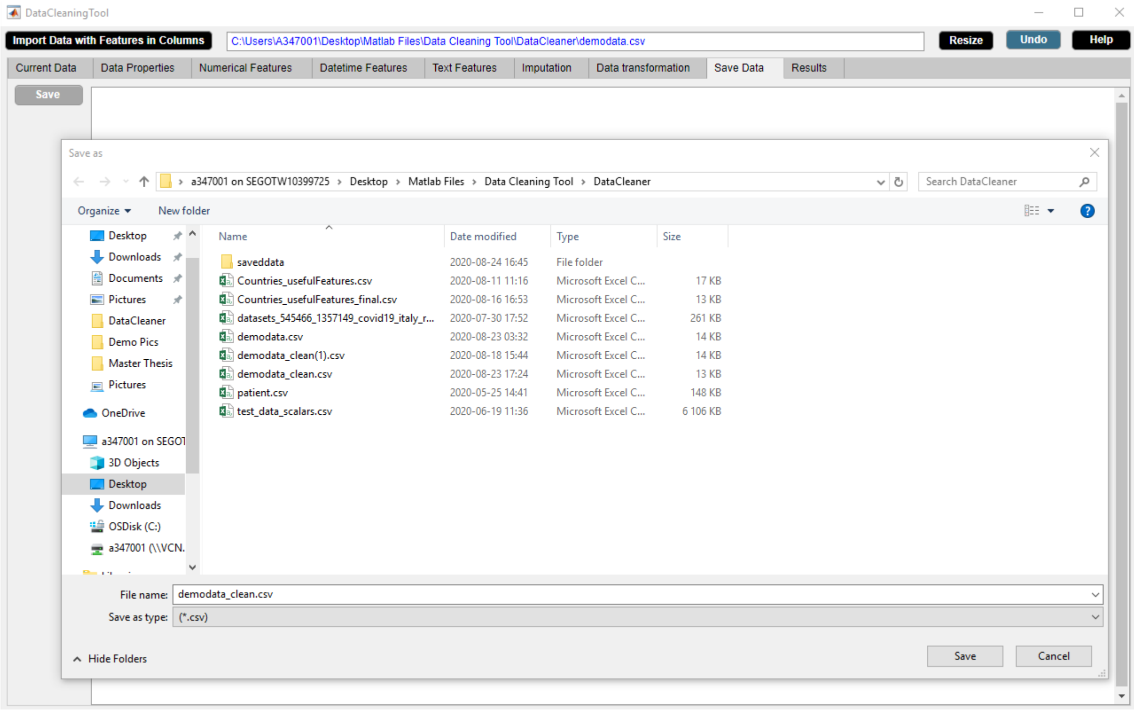


Figure B.108: Step 2. Save Button

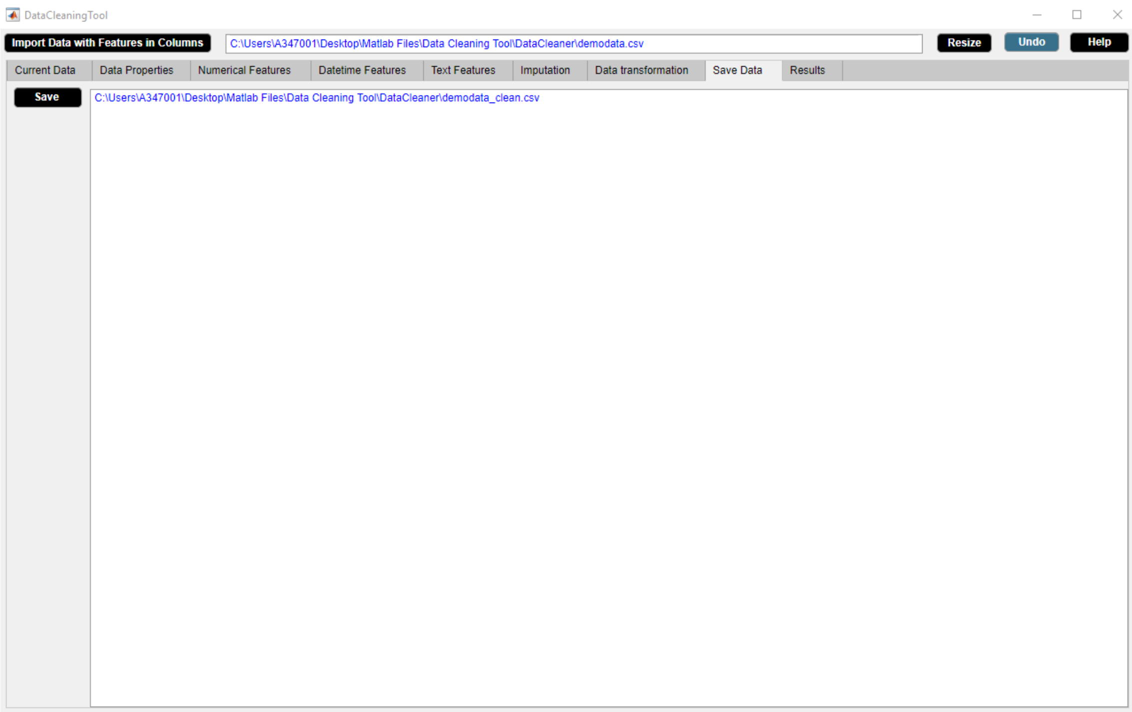


Figure B.109: Step 3. Save Button

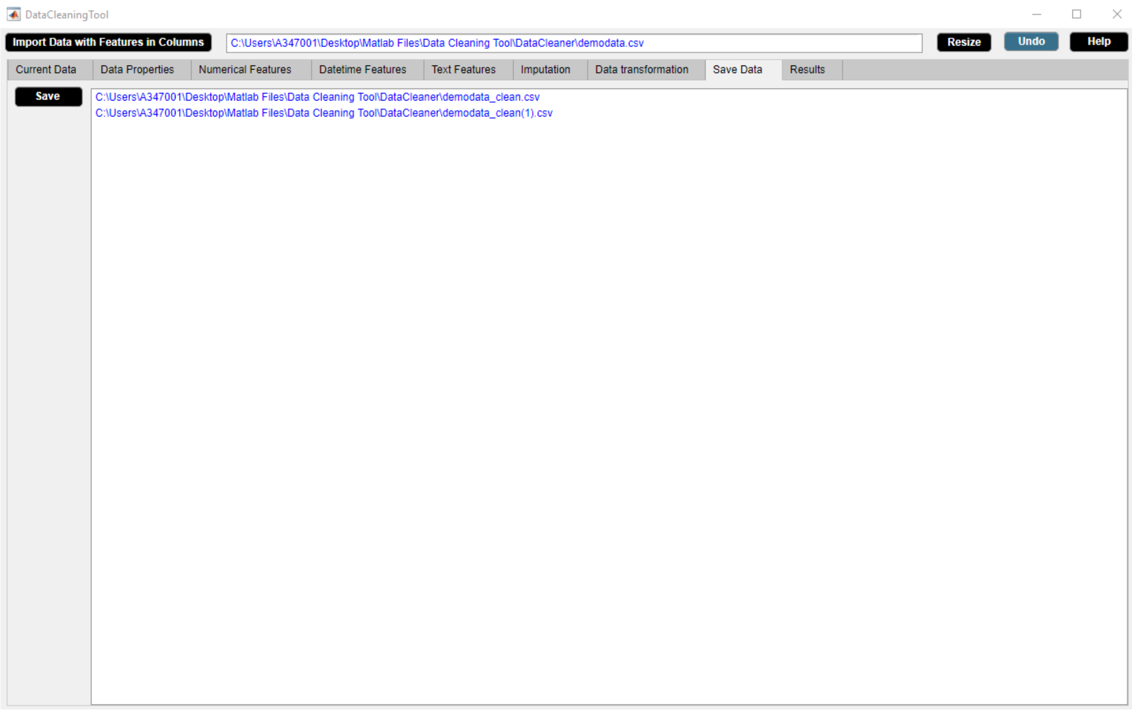


Figure B.110: Step 4. Save Button

B.10 Results

The Results widget displays information about the final report. The Results widget is shown in figure B.111. The properties of the Results widget are as follows.

- The widget generates results in pdf format after data cleaning. The results contains a detailed report of all the changes made in DataCleaningTool.
- Results can be generated containing a detailed report of specific changes made in DataCleaningTool.
- Results can be generated for multiple times after each activity.
- The full paths of the results are displayed.

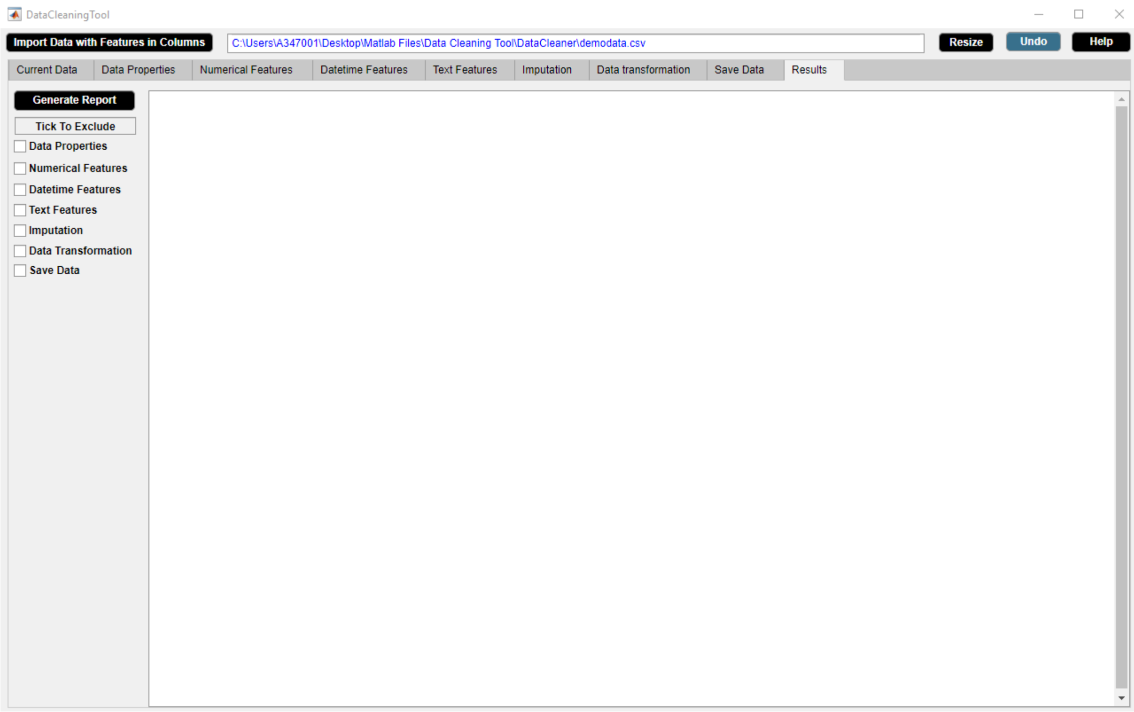


Figure B.111: Results Widget.

B.10.1 Generate Report Button

Generate pdf file containing results.

Example

Step 1: Click **Generate Report** button.

Step 2: **Generate Report** button in use turns grey in color.

Step 3: **Generate Report** button returns back to its original color once it completes its task.

We use **Generate Report** button to save the example data in csv format. Figures B.112-B.115 illustrate how to use **Generate Report** button.

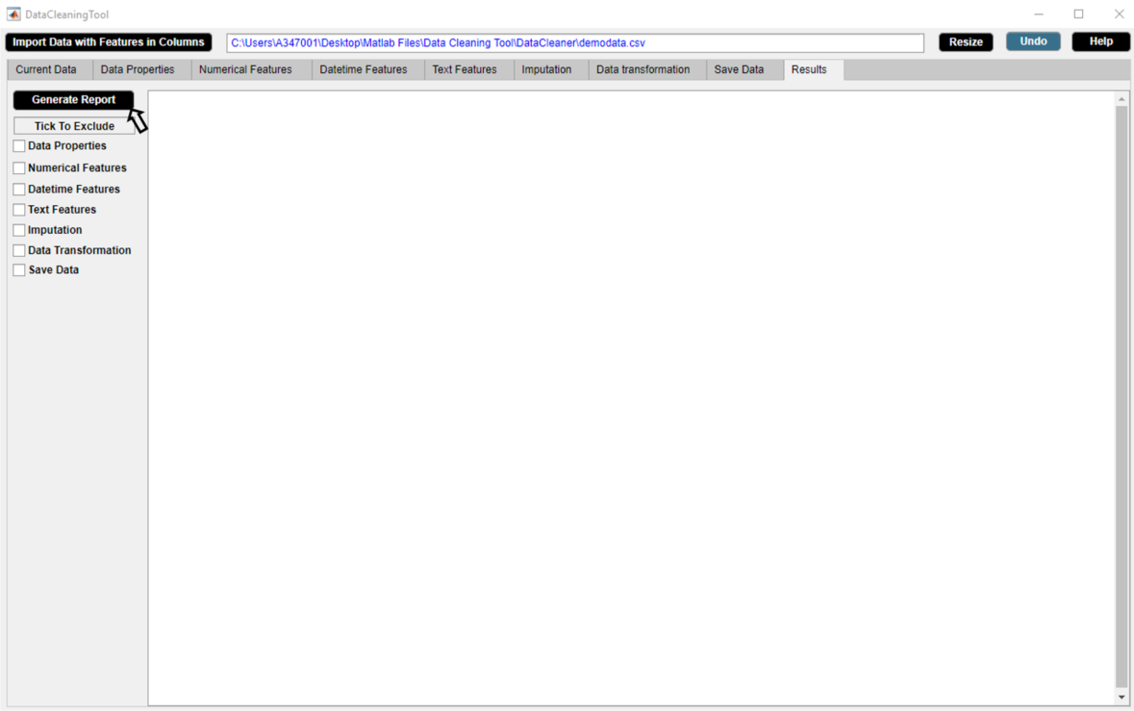


Figure B.112: Step 1. Generate Report Button

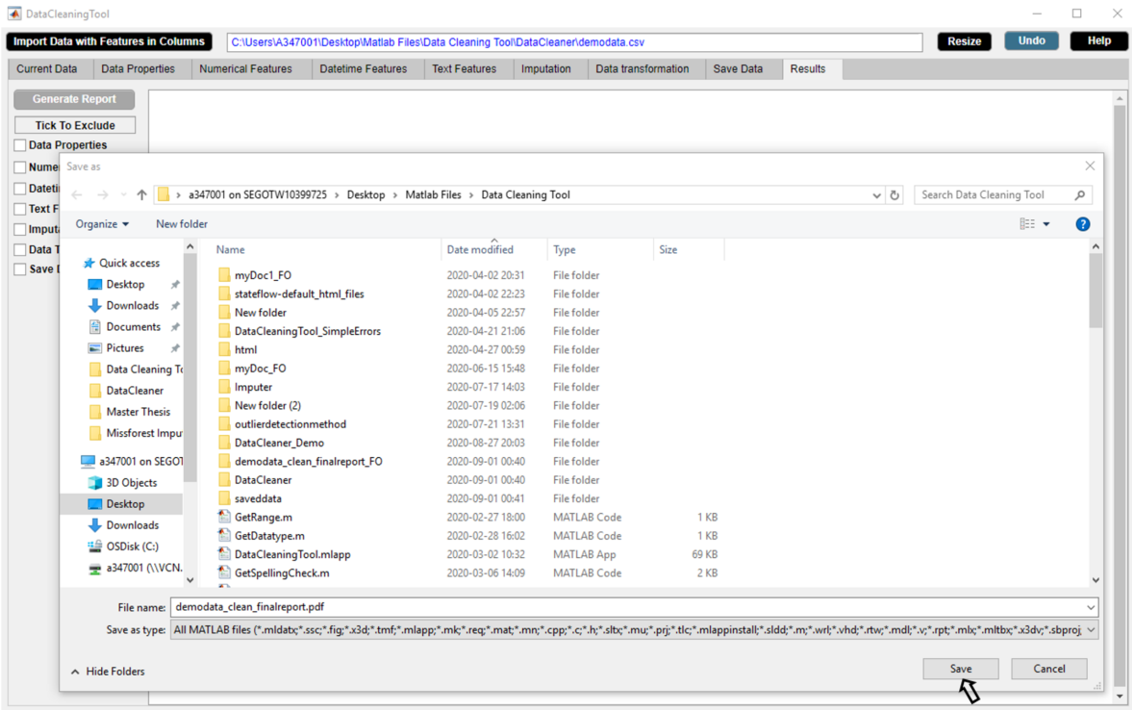


Figure B.113: Step 2. Generate Report Button

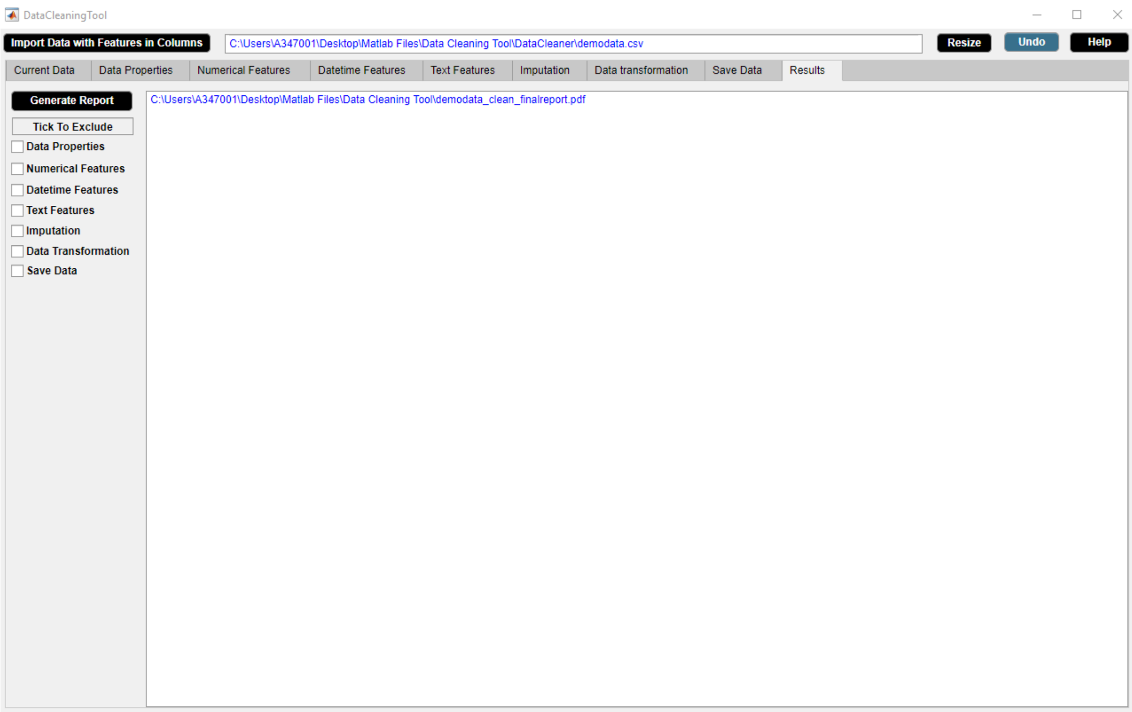


Figure B.114: Step 3. Generate Report Button

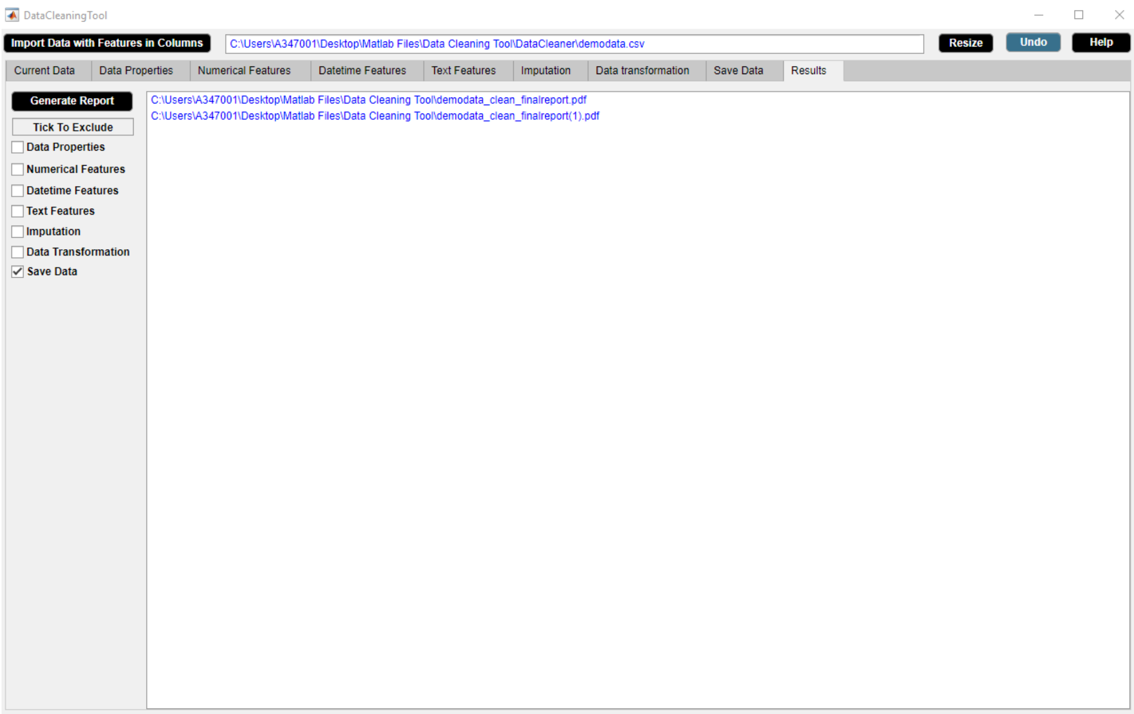


Figure B.115: Step 4. Generate Report Button

B.11 Other Attributes

Other attributes include the following three buttons which are present in the upper right side of the DataCleaningTool B.1.

B.11.1 Resize Button

Resizes the DataCleaningTool to a reduced size.

B.11.2 Undo Button

Performs the last activity and all the widgets get updated accordingly.

B.11.3 Help Button

Generates user manual of DataCleaningTool in pdf format.