LabRI Shiny Application

1. LabRI System

The LabRI (Laboratory Reference Interval) System was developed to facilitate the estimation and verification of laboratory test reference intervals, aiming to simplify its use for professionals unfamiliar with the RStudio environment or the R programming language. The system is based on the R programming language and integrates RMarkdown scripts (interactive documents combining R code and explanatory text) with a Shiny application (a web development tool in R that allows the creation of interactive interfaces). Additionally, the system includes automation scripts and an executable installer, as described below, to provide a streamlined user experience.

1.1. Processing Core (RMarkdown File)

The core of the LabRI tool is an RMarkdown script, specifically the LabRI_script.Rmd file, which reads laboratory datasets, allows the selection of a specific column corresponding to test results, and applies the "LabRI method" for estimating and verifying reference intervals. At the end of the process, the script generates an HTML report containing the detailed analysis results.

1.2. User Interface (Shiny Application)

To facilitate user interaction, a graphical interface was developed using Shiny, which serves as a base to encapsulate the RMarkdown script and eliminate the need for direct interaction with RStudio. The app.R file acts as the entry point for the Shiny application, responsible for loading the RMarkdown script and initializing the simplified graphical interface. This interface allows users to input data and view results in an intuitive and interactive manner.

1.3. Integration and Automated Execution: Initialization Files (.bat and .R)

The LabRI System's execution is automated through batch files (.bat), which perform specific functions to prepare and start the environment. To ensure that all necessary R packages for the LabRI method are properly installed, the install_packages.bat file is used. This file executes the install_packages.R script, which checks for required packages, installs missing ones, and updates outdated packages, ensuring an efficiently configured working environment.

The app.bat file, on the other hand, triggers the launch_app.R script, which uses the shiny.exe library to configure and launch the Shiny application in the Windows environment. The application runs locally on the localhost (127.0.0.1) address, automatically opening the user's default browser. This approach eliminates the need for direct interaction with RStudio, simplifying and streamlining the system startup process.

1.4. Distribution and Access (Available Versions)

The LabRI System is primarily distributed via an executable installer, developed using the Inno Setup Script (LabRI_installer_Script.iss), accessible in the repository GitHub LabRI Repository. This installer automatically organizes files, configures necessary directories, and creates two desktop shortcuts:

- **LabRI Package Installer Shortcut:** Executes the script associated with the install_packages.bat file, which checks for necessary R packages, installs missing ones, and updates outdated ones. Running this shortcut is essential for properly configuring the system's working environment, ensuring all dependencies are installed before using the Shiny application.
- **LabRI Shortcut:** Executes the script associated with the app.bat file, which initializes the Shiny application using the previously installed R packages. This shortcut will only function after the successful execution of the LabRI Package Installer Shortcut, as the Shiny application directly depends on the installed R packages. Therefore, if the packages are correctly installed by running the "LabRI Package Installer" Shortcut, executing the "LabRI" Shortcut will open the web interface of the "LabRI Shiny Application". In this interface, the user will need to select the dataset, choose the column containing the desired data, fill in the required information for study traceability, click on "Generate Report", and wait for the data to be analyzed and processed. The output will be an HTML report detailing the estimation and verification of the reference interval using the LabRI Method.

1.5. LabRI Method

The LabRI method is an adaptive approach composed of algorithms and sub-algorithms designed to estimate and verify reference intervals (RIs) in laboratory data. It is divided into two modules: the Estimation Module and the Verification Module.

1.5.1. Estimation Module

The Estimation Module employs an adaptive, multi-criteria approach to address challenges such as non-standard distributions and outliers. This module consists of four main stages:

1.5.1.1. Data Preprocessing

This stage involves cleaning data, removing noise, and handling outliers to improve dataset quality. Sub-algorithms like iboxplot and SIQR boxplot are sequentially applied to identify and remove extreme values. Additionally, the Deterministic Perturbation Technique introduces slight variations to repeated values, avoiding ties that could compromise the analysis.

1.5.1.2. Transformation and Box-Cox Application Decision

After preprocessing, the need to apply the Box-Cox transformation is evaluated. This transformation stabilizes variance and reduces skewness. The Sk PruneTails Sub-algorithm removes 2.5% of the upper and lower tails of the data to minimize the impact of outliers before calculating the Bowley skewness coefficient. If the transformation is indicated, parameters are optimized based on the maximum likelihood criterion (logLik).

1.5.1.3. Clustering, Truncation, and Limit Estimation

This stage identifies clusters in the distribution using algorithms like Expectation-Maximization (EM) and criteria like BIC. Depending on the number of clusters:

- **Single cluster:** The EM algorithm is directly used to estimate reference limits.
- **Multiple clusters:** When multiple clusters are identified, the distribution is truncated, and the reference limits are estimated using three algorithms: refineR, reflimR, and the EM algorithm (when applicable). Following this, a two-step Winsorization process is applied, and the centroids of both the lower and upper

reference limits are calculated using the K-means clustering approach, ensuring precise and robust determination of the reference intervals.

1.5.1.4. Confidence Interval Estimation

The Half Gaussian Confidence Interval Sub-algorithm calculates statistically robust confidence intervals for the reference limits, generating LRL_LabRI and URL_LabRI (Lower and Upper Reference Limits).

1.5.2. Verification Module

The Verification Module ensures the validity of estimated intervals through a rigorous three-level analysis:

- **First Level ~ Statistical Uncertainty:** Evaluates the degree of uncertainty associated with the estimated reference limits, ensuring the reliability of the results.
- **Second Level ~ Equivalence Test:** Compares the estimated reference limits with comparative reference intervals, assessing the practical significance of any observed differences.
- **Third Level ~ Concordance:** Uses Fleiss' Kappa test, Lin's Concordance Correlation Coefficient, and Flagging Rates.