STAT 5243 Project 2

Team 11: Ran Yan, Zijun Fu, Shayan Chowdhury, Tiantian Li

March 14, 2025

App deployed on **shinyapps.io** here:【https://5243-project2-team11.shinyapps.io/Desktop/】

GitHub Repo:【https://github.com/shayantist/STAT5243-Project2/】

1. Application Overview

For the project, we developed a comprehensive **R Shiny web application** for data analysis, meeting all the requirements specified in the class instructions. Our goal was to achieve an **"Advanced"** rating in all evaluation criteria. The application provides a **user-friendly interface** for **uploading datasets, cleaning and preprocessing data, feature engineering, and exploratory data analysis (EDA)**.

2. Application Features and User Guide

2.1 Data Loading

Feature Description: Our application provides a flexible data-loading feature. Users can upload files in **multiple formats**, including CSV, Excel (.xlsx/.xls), JSON, and RDS, allowing seamless integration of **different data sources**. Additionally, for users who want to explore the application without uploading their own data, built-in datasets such as the **Iris flower data set**, **US Economic Time Series**, and **US Crime Statistics** are readily available for quick access.

Usage Steps: To load a dataset, navigate to the ***Data Loading*** tab, where users can either **upload** a file or select one of the built-in datasets. After making a selection, clicking the ***Load Data*** button will initiate the loading process, and the dataset will be displayed in the ***Data Preview*** table for verification before further processing.

2.2 Data Cleaning and Preprocessing

Feature Description: This module integrates core data cleaning and preprocessing functions, covering four key areas: **missing value handling**, **duplicate removal**, **variable transformation**, and **text preprocessing**. ① Missing values can be addressed through **row deletion**, **constant value filling**, **statistical imputation** (mean/median/mode), or **interpolation** methods (linear, spline, nearest neighbor) for numerical columns, ensuring flexible solutions for various data quality issues. ② The duplicate removal function allows users to identify and **eliminate redundant records** based on selected columns, with options to retain the **first or last occurrence** or remove all duplicates. ③ Variable transformation supports **standardization**, **normalization**, **log transformation**, and **outlier removal** for numerical data, while categorical variables can be processed using one-hot encoding to enhance data quality. ④ For **text data**, the module offers **tokenization**, **stopword removal** (e.g. "the", "and", "is"), **lowercase conversion**, **punctuation removal**, and **spell correction** (only English for now), catering to natural language processing **(NLP)** requirements.

Usage Steps: Upon entering the ***Data Cleaning & Preprocessing*** tab, users can systematically perform cleaning tasks. First, they can identify problematic columns using the **missing value summary table**, select an appropriate handling method, and apply changes. Next, duplicate records can be removed by selecting key columns (e.g., order ID), defining retention rules, and executing the cleanup. For variable transformation, users can specify a target column and apply methods such as standardization, with immediate effects visible in the preview area. Text preprocessing requires selecting a text column and enabling options like stopword removal, which is executed with a single click. All modifications are **dynamically reflected in the data preview table**, allowing users to download the cleaned dataset or reload the original data for rollback, ensuring a flexible and controlled analysis workflow.

2.3 Feature Engineering

Feature Description: The feature engineering module enables users to enhance their datasets by **creating new variables** through **various transformations**. **Mathematical operations** allow the generation of new features by applying **addition, subtraction, multiplication,** or **division** to two numerical columns. **Binning methods** support both equal-width and quantile-based equal-frequency binning, helping categorize continuous data into meaningful groups. **Date processing** extracts specific components such as year, month, or day from datetime columns, while **aggregate statistics** compute values like **mean, median, mode, sum**, and **standard deviation** based on a grouping column. To facilitate analysis, real-time **visualization tools**, including **histograms**, **density plots**, and **boxplots**, dynamically display the distributions of newly created features.

Usage Steps: Upon entering the ***Feature Engineering*** tab, users first select a feature creation method, such as binning or aggregate statistics. They then specify the target column and configure parameters, such as the number of bins or grouping fields. Clicking ***Create Feature*** generates the new feature, automatically updating the data preview table and triggering the visualization module. Users can switch between histogram, density plot, or boxplot to analyze the feature distribution. The newly derived features can be directly utilized for further analysis or exported, enabling an **end-to-end data enhancement workflow**.

2.4 Exploratory Data Analysis (EDA)

Feature Description: The Exploratory Data Analysis (EDA) module helps users visualize and explore their data through interactive charts and graphs. For **univariate analysis**, users can generate **histograms, boxplots, density plots,** and **bar charts** to examine the distribution of individual variables. **Bivariate analysis** allows users to explore relationships between two variables using **scatter plots**, **line charts**, **grouped boxplots**, and **heatmaps**. The application also includes **correlation analysis**, where users can compute and visualize **Pearson, Spearman, or Kendall correlation matrices** as heatmaps. Additionally, dynamic filtering options enable users to refine their dataset using **range sliders** for numerical values or **multi-selection filters** for categorical variables, making it easier to focus on relevant data.

Usage Steps: To perform an exploratory data analysis, users should first navigate to the ***EDA*** tab, where they will find a variety of visualization options. They can begin by **selecting the variables** they wish to analyze and choosing the most appropriate **visualization type**, such as a scatter plot to examine relationships, a histogram to inspect data distribution, or a heatmap to explore correlations.Once the desired variables and visualization settings have been selected, users can click the ***Generate Plot*** button, which will instantly render an **interactive visualization**, allowing them to interact with the data dynamically. If further refinement is needed, users can utilize the ***Filtered Data View section***, where they can apply **custom filters** based on specific **numerical or categorical criteria**, enabling a more focused and insightful data exploration process.

3. Team Contributions

Team Member	**Contribution**
Ran Yan (ry2487) & Tiantian Li (tl3404)	**Report writing**
Zijun Fu (zf2342)	**Optimization and debugging of shiny app**
Shayan Chowdhury (sc4040)	**First draft of shiny app features, GitHub Repo, report editing**