BUILDING A SMALL APP FOR VISUALIZING MONTHLY EXPENSES

A PROJECT REPORT

Submitted by

SURYA S (2303811724321115)

in partial fulfillment of requirements for the award of the course

AGI1252-FUNDAMENTALS OF DATA SCIENCE USING R

in

ARTIFICIAL INTELLIGENCE AND DATA SCIENCE

K. RAMAKRISHNAN COLLEGE OF TECHNOLOGY

(An Autonomous Institution, affiliated to Anna University Chennai and Approved by AICTE, New Delhi)

SAMAYAPURAM – 621 112

JUNE-2025

K. RAMAKRISHNAN COLLEGE OF TECHNOLOGY (AUTONOMOUS)

SAMAYAPURAM – 621 112

BONAFIDE CERTIFICATE

Certified that this project report on **"BUILDING A SMALL APP FOR VISUALIZING MONTHLY EXPENSES"** is the bonafide work of **SURYA S (2303811724321115)** who carried out the project work during the academic year 2024 - 2025 under my supervision.

SIGNATURE

Dr.T.AVUDAIAPPAN, M.E.,Ph.D.,

HEAD OF THE DEPARTMENT

ASSOCIATE PROFESSOR

Department of Artificial Intelligence

K.Ramakrishnan College of Technology (Autonomous)

Samayapuram–621112.

SIGNATURE

Ms.S.Murugavalli., M.E.,(Ph.D).,

SUPERVISOR

ASSISTANT PROFESSOR

Department of Artificial Intelligence

K.Ramakrishnan College of Technology (Autonomous)

Samayapuram–621112.

Submitted for the viva-voce examination held on …02.06.2025……

INTERNAL EXAMINER

EXTERNAL EXAMINER

DECLARATION

I declare that the project report on **"BUILDING A SMALL APP FOR VISUALIZING MONTHLY EXPENSES"** is the result of original work done by us and best of our knowledge, similar work has not been submitted to **"ANNA UNIVERSITY CHENNAI"** for the requirement of Degree of **BACHELOR OF ENGINEERING**. This project report is submitted on the partial fulfilment of the requirement of the completion of the course **AGI1252 – FUNDAMENTALS OF DATA SCIENCE USING R**

.

Signature

SURYA S

Place: Samayapuram

Date : 02-06-2025

ACKNOWLEDGEMENT

It is with great pride that I express our gratitude and in-debt to our institution "**K.Ramakrishnan College of Technology (Autonomous)**", for providing us with the opportunity to do this project.

I glad to credit honourable chairman **Dr. K. RAMAKRISHNAN**, **B.E.,** for having provided for the facilities during the course of our study in college.

I would like to express our sincere thanks to our beloved Executive Director **Dr. S. KUPPUSAMY, MBA, Ph.D.,** for forwarding to our project and offering adequate duration in completing our project.

I would like to thank **Dr. N. VASUDEVAN, M.Tech., Ph.D.,** Principal, who gave opportunity to frame the project the full satisfaction.

I whole heartily thanks to **Dr. T. AVUDAIAPPAN, M.E.,Ph.D.,** Head of the department, **ARTIFICIAL INTELLIGENCE** for providing his encourage pursuing this project.

I express our deep expression and sincere gratitude to our project supervisor **Ms.S.Murugavalli., M.E.,(Ph.D)..,** Department of **ARTIFICIAL INTELLIGENCE,** for his incalculable suggestions, creativity, assistance and patience which motivated us to carry out this project.

I render our sincere thanks to Course Coordinator and other staff members for providing valuable information during the course.

I wish to express our special thanks to the officials and Lab Technicians of our departments who rendered their help during the period of the work progress.

INSTITUTE

Vision:

- To serve the society by offering top-notch technical education on par with global standards.

Mission:

- Be a center of excellence for technical education in emerging technologies by exceeding the needs of industry and society.
- Be an institute with world class research facilities.
- Be an institute nurturing talent and enhancing competency of students to transform them as all – round personalities respecting moral and ethical values.

DEPARTMENT

Vision:

- To become a renowned hub for AIDS technologies to producing highly talented globally recognizable technocrats to meet industrial needs and societal expectation.

Mission

- To impart advanced education in AI and Data Science, built upon a foundation in Computer Science and Engineering.
- To foster Experiential learning equips students with engineering skills to tackle real-world problems.
- To promote collaborative innovation in AI, Data science , and related research and development with industries.
- To provide an enjoyable environment for pursuing excellence while upholding strong personal and professional values and ethics.

PROGRAM EDUCATIONAL OBJECTIVES (PEO)

- **PEO1:** Excel in technical abilities to build intelligent systems in the fields of AI & DS in order to find new opportunities.
- **PEO2:** Embrace new technology to solve real-world problems, whether alone or as a team, while prioritizing ethics and societal benefits.
- **PEO3:** Accept lifelong learning to expand future opportunities in research and product development.

PROGRAM SPECIFIC OUTCOMES (PSO)

- **PSO1:** Expertise in tailoring DS algorithms and models to excel in designated applications and fields.
- **PSO2:** Ability to conduct research, contributing to Data science learning advancements and innovations that tackle emerging societal challenges.

PROGRAM OUTCOMES (POs)

Engineering students will be able to:

1. **Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis:** Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences
3. **Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations
4. **Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions

5. **Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations

6. **The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice

7. **Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development

8. **Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

9. **Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

10. **Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

11. **Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

12. **Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

ABSTRACT

This project introduces an interactive web-based application developed using R Shiny to visualize and manage monthly personal expenses. With the increasing complexity of day-to-day financial transactions, individuals require simple yet effective tools to record, monitor, and analyze their spending patterns. The "Monthly Expense Visualizer" allows users to enter expense data manually or upload CSV files and instantly generate multiple forms of graphical representations including bar charts, line charts, and histograms. The application also summarizes key financial indicators such as total, average, minimum, and maximum expenses, offering users a clear and immediate understanding of their spending behavior. The user-friendly interface, combined with reactive data rendering and real-time chart updates, ensures a seamless experience that encourages consistent financial tracking. Download and reset functionalities further enhance the tool's flexibility. Built using libraries like ggplot2, plotly, and dplyr, the app emphasizes the value of visual analytics in promoting financial literacy and self-discipline. The modular design supports future extensions such as trend tracking or cloud-based storage, making this tool a scalable and practical solution for modern personal finance management. In addition, the ability to customize inputs and switch between visual modes makes the application highly adaptable to user preferences. It also eliminates the need for manual calculations by automating data analysis. Ultimately, this tool bridges the gap between data science and everyday decision-making, empowering users to take control of their financial habits with clarity confidence.

ABSTRACT WITH POs AND PSOs MAPPING

CO 5 : BUILD VISUALIZATION FOR SOLVING REAL-TIME PROBLEMS.

ABSTRACT	POs MAPPED	PSOs MAPPED
This project developed an interactive R Shiny web application for managing and visualizing monthly personal expenses. It allows users to input expense data manually or upload CSV files, instantly generating dynamic visualizations such as bar charts, line graphs, and histograms. The app calculates key financial metrics like total, average, maximum, and minimum expenses, providing clear insights into spending habits through a user-friendly interface with real-time updates. Additional features include data download and reset options for flexibility. Built with ggplot2, plotly, and dplyr, the app promotes financial literacy by making expense tracking simple and accessible, while its modular design supports future enhancements and customization to meet evolving user needs.	**PO1 -3** **PO2 -3** **PO3 -3** **PO4 -3** **PO5 -3** **PO6 -3** **PO7 -3** **PO8 -3** **PO9 -3** **PO10 -3** **PO11-3** **PO12 -3**	**PSO1 -3** **PSO2 -3** **PSO3 -3**

Note: 1- Low, 2-Medium, 3- High

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	
1	**INTRODUCTION**	
	1.1 Objective	1
	1.2 Overview	1
	1.3 R Programming Concepts	2
2	**PROJECT METHODOLOGY**	
	2.1 Proposed Work	3
	2.2 Block Diagram	4
3	**MODULE DESCRIPTION**	
	3.1 File Upload Module	5
	3.2 Chart Type Selection Module	5
	3.3 Input Table Module	5
	3.4 Reactive Rendering Module	5
	3.5 User Interface Module	5
4	**CONCLUSION & FUTURE SCOPE**	6
5	**APPENDIX A SOURCE CODE**	8
	APPENDIX B SCREENSHOTS	11
	REFERENCES	

LIST OF FIGURES

FIG.NO	TITLE	PAGE.NO
2.1	BLOCK DIAGRAM	4
B.1	INPUT TABLE	11
B.2	BARCHART	11
B.3	LINE CHART	12
B.4	HISTOGRAM	12
B.5	SUMMARY	13

CHAPTER 01

INTRODUCTION

1.1 Objective

The objective of this project is to develop an interactive and intuitive web application using R Shiny to assist users in managing their monthly personal expenses effectively. This application enables users to input expense data manually or upload CSV files, providing flexibility in how financial information is recorded. It offers multiple dynamic visualizations such as bar charts, line charts, and histograms to help users better understand their spending patterns through clear and engaging graphics. The app calculates and displays essential financial summaries including total expenditure, average spending, as well as the maximum and minimum expenses. By presenting these insights in real-time, users can track their financial behavior instantly and make informed decisions. The user interface is designed to be simple and responsive, ensuring ease of use across different devices. Additional features such as data download and reset options enhance user control over their financial data. The application leverages powerful R libraries like ggplot2, plotly, and dplyr for data analysis and visualization, making the experience both robust and visually appealing. This project aims to promote financial literacy by simplifying the process of expense tracking and analysis. By automating calculations and visual reporting, it reduces the time and effort needed to monitor finances manually. The modular architecture of the app allows for easy future expansion, including adding features like trend analysis or cloud synchronization. Ultimately, this tool empowers users to take charge of their finances with clarity and confidence, encouraging better budgeting habits and financial responsibility. It bridges the gap between data science techniques and practical everyday financial management, making personal finance accessible to all users regardless of their technical background.

1.2 Overview

This project focuses on developing a database-driven web application to handle the essential operations of learning management system. It consists of several major modules:

- **Data Input**

  Users can add expenses manually through a form or upload CSV files for bulk data entry, providing flexibility and convenience.

- **Visualization**

  The app offers multiple interactive charts—including bar charts, line charts, and histograms—to help users better understand their spending patterns visually.

- **Financial Summary**

  It calculates and displays essential statistics such as total, average, maximum, and minimum expenses, offering users clear insights into their financial behavior.

- **User Interface**

  Designed to be intuitive and user-friendly, the interface provides real-time updates and seamless navigation for a smooth experience.

- **Additional Features**

  Functionalities such as data download and reset options enhance user control and data management.

- **Technology Stack**

  The application leverages R libraries like ggplot2, plotly, and dplyr for robust data analysis and engaging visualizations..

1.3 R Programming Concepts

The development of this Shiny application leverages several fundamental R programming concepts to deliver a dynamic and interactive user experience. Reactive programming is central to the app's functionality, allowing outputs such as tables and charts to update automatically in response to user inputs or events. Data frames serve as the primary structure for storing and manipulating expense data, while packages like dplyr simplify data wrangling and summarization. Visualization is achieved using ggplot2 for creating detailed static plots, which are then enhanced with plotly to provide interactive features like zooming and tooltips. The app handles file input and output operations to allow users to upload CSV files and download their expense records conveniently. The user interface is constructed with Shiny's layout functions and input controls, providing an intuitive way for users to interact with the application. Input validation and conditional execution ensure robust operation by preventing processing of incomplete or invalid data. Overall, these R programming concepts work together to build a scalable, efficient.

CHAPTER 02

PROJECT METHODOLOGY

2.1 Proposed Work

This project proposes to develop a comprehensive and interactive web application for managing monthly personal expenses using R Shiny. The focus is on creating an intuitive platform that allows users to input, upload, and analyze their expense data efficiently. The application will incorporate real-time data processing and dynamic visualizations to enhance user understanding of their financial habits. Future expansions may include adding advanced analytics like trend detection, budget alerts, or cloud data storage to further support personal finance management.

Key Features

- **Flexible Data Entry:**

  Users can add expenses manually or upload CSV files for bulk data import.

- **Multiple Visualizations:**

  Interactive bar charts, line graphs, and histograms help visualize spending patterns clearly.

- **Financial Summaries:**

  Automatic calculation of total, average, maximum, and minimum expenses for quick insights.

- **Real-Time Updates:**

  Reactive programming ensures charts and tables update instantly with new or modified data.

- **Data Export and Reset:**

  Options to download expense data as CSV and reset all entries improve data control.

2.2 Block Diagram

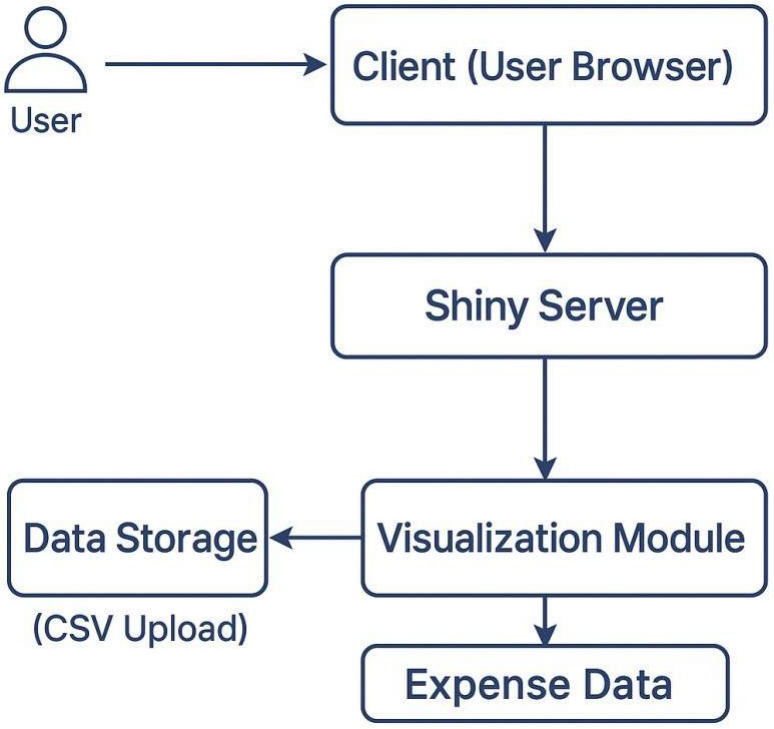

Fig 2.1 Block diagram of Building a Small App For Visualizing Monthly Expenses

CHAPTER 03

MODULE DESCRIPTION

3.1. File Upload Module :

This module handles the uploading of expense data files, typically in CSV format. It validates the uploaded file to ensure it contains the necessary columns (e.g., Category and Amount) before loading the data into the app. It also manages error handling to alert users if the file format or content is incorrect.

3.2. Chart Type Selection Module :

This module allows users to select different types of visualizations such as bar charts, line charts, and histograms. Based on the user's selection, it triggers the rendering of the corresponding plot, providing flexibility in how the expense data is visually analyzed.

3.3 Input Table Module

This input table module displays the current list of expenses entered manually or loaded from a file. It updates reactively as users add new entries or reset the data, offering a clear, tabular overview of all recorded expenses.

3.4. Reactive Rendering Module

This module manages the reactive expressions and observers that ensure outputs like tables and charts update automatically when the underlying expense data changes. It is crucial for maintaining synchronization between user inputs and displayed results, enabling real-time interaction.

3.5. User Interface Module

This UI module defines the layout and design of the application, organizing input controls, output displays, and navigation elements such as tabs. It ensures an intuitive and accessible user experience across different devices and screen sizes.

CHAPTER 04

CONCLUSION & FUTURE SCOPE

Conclusion

In conclusion, the Monthly Expense Visualizer is a practical and user-friendly tool that empowers individuals to manage their personal finances effectively. By combining manual data entry and file upload options, it offers flexibility in expense tracking. The application's interactive visualizations and real-time summaries provide valuable insights into spending patterns, encouraging better financial decisions. Its intuitive interface makes it accessible to users of all levels, while features like data download and reset enhance usability. Built with robust R packages, the app demonstrates the power of reactive programming for dynamic data analysis. The modular design ensures that the application can be easily expanded with additional features in the future. Overall, this project bridges the gap between data science and everyday financial management, promoting financial literacy and discipline. It serves as a foundation for more advanced personal finance tools. By simplifying expense tracking, the app helps users gain clarity and control over their budgeting habits. This tool ultimately supports healthier financial behavior and better money management.

Future Scope

- **Time-Based Analysis**
  Future versions can include date-wise tracking of expenses, allowing users to analyze trends over days, weeks, or months for improved budgeting insights.
- **User Authentication & Data Storage**
  Implementing login systems and cloud-based storage will enable multiple users to save and access their expense data securely from any device.
- **Budget Alerts**
  Adding features for setting monthly or category-wise budgets and triggering alerts when limits are exceeded can promote disciplined spending.

- **Advanced Visualizations**

  The app can include more diverse charts such as pie charts, stacked bars, or heatmaps to better represent spending distribution and behavior.
- **Category Filtering & Search**

  Future updates may introduce filters to sort and search expenses by category, date, or amount, enhancing data exploration.
- **Predictive Analytics**

  Incorporating basic machine learning models can help predict future expenses based on past patterns and suggest savings strategies.
- **API Integration**

  Connecting with banking APIs or finance apps could automate the import of transaction data, minimizing manual entry.
- **Mobile Compatibility**

  Making the app responsive or developing a mobile version can provide convenience for users to manage expenses on the go.
- **Expense Sharing Options**

  Future enhancements could include sharing reports with family members or financial advisors via email or download links.
- **Data Backup & Restore**

  Automatic data backup and restore options can prevent data loss and improve reliability.

APPENDIX A

SOURCE CODE

```
# Load necessary libraries
library(shiny)
library(ggplot2)
library(plotly)
library(dplyr)

# Define UI for the application
ui <- fluidPage(

  # Application title
  titlePanel("Monthly Expense Visualizer"),

  # Sidebar layout with input and output definitions
  sidebarLayout(

    # Sidebar panel for inputs
    sidebarPanel(
      textInput("category", "Expense Category:", ""),
      numericInput("amount", "Expense Amount:", value = 0, min = 0),
      actionButton("add", "Add Expense"),
      hr(),
      fileInput("file", "Upload CSV File:", accept = ".csv"),
      actionButton("load", "Load File"),
      hr(),
      downloadButton("downloadData", "Download Data as CSV"),
      hr(),
      actionButton("reset", "Reset Data")
    ),

    # Main panel for displaying outputs
    mainPanel(
      tabsetPanel(
        tabPanel("Expense Table", tableOutput("expenseTable")),

        tabPanel("Bar Chart", plotlyOutput("barChart")),

        tabPanel("Line Chart", plotlyOutput("lineChart")),

        tabPanel("Histogram", plotlyOutput("histogram")),

        tabPanel("Expense Summary", verbatimTextOutput("expenseSummary"))
      )) ))
```

```
# Define server logic
server <- function(input, output, session) {

  # Reactive values to store expenses data
  expenses <- reactiveVal(data.frame(Category = character(), Amount = numeric(),
stringsAsFactors = FALSE))

  # Observe add button to add new expenses
  observeEvent(input$add,
  { req(input$category != "", input$amount > 0)
    new_expenses <- rbind(expenses(), data.frame(Category = input$category, Amount =
input$amount))
    expenses(new_expenses)
  })

  # Observe reset button to clear the expenses data
  observeEvent(input$reset, {
    expenses(data.frame(Category = character(), Amount = numeric(), stringsAsFactors = FALSE))
  })

  # Observe file upload and load the data
  observeEvent(input$load, { req(input$file)
    file_data <- read.csv(input$file$datapath, stringsAsFactors = FALSE)
    req(all(c("Category", "Amount") %in% names(file_data))) expenses(file_data)
  })
  # Render expense table
  output$expenseTable <-
  renderTable({ expenses()
  })
  # Render bar chart visualization
  output$barChart <-
  renderPlotly({ req(nrow(expenses()) > 0)

    plot_data <- expenses()
    plot <- ggplot(plot_data, aes(x = Category, y = Amount, fill = Category)) +
      geom_bar(stat = "identity") +
      theme_minimal() +
      labs(title = "Monthly Expenses - Bar Chart", x = "Category", y = "Amount")

    ggplotly(plot)
  })
  # Render line chart visualization
  output$lineChart <-
  renderPlotly({ req(nrow(expenses()) > 0)
```

```
    plot_data <- expenses()
    plot_data$Index <- seq_len(nrow(plot_data))
    plot <- ggplot(plot_data, aes(x = Index, y = Amount, group = 1, color = Category)) +
      geom_line() +
      geom_point() +
      theme_minimal() +
      labs(title = "Monthly Expenses - Line Chart", x = "Index", y = "Amount")
    ggplotly(plot)
  })
  # Render histogram visualization
  output$histogram <-
  renderPlotly({ req(nrow(expenses()) > 0)

    plot_data <- expenses()
    plot <- ggplot(plot_data, aes(x = Amount)) +
      geom_histogram(binwidth = 10, fill = "blue", color = "white") +
      theme_minimal() +
      labs(title = "Monthly Expenses - Histogram", x = "Amount", y = "Frequency")
    ggplotly(plot)
  })
  # Render expense summary (descriptive statistics)
  output$expenseSummary <-
  renderPrint({ req(nrow(expenses()) > 0
)
    expense_data <- expenses()
    summary_stats <- expense_data %>%
    summarise(
        Total_Amount = sum(Amount),
        Average_Amount = mean(Amount),
        Max_Expense = max(Amount),
        Min_Expense = min(Amount)
      )
    print("Expense Summary:")
    print(summary_stats)
  })
  # Download data as CSV
  output$downloadData <-
  downloadHandler( filename = function() {
      paste("expenses", Sys.Date(), ".csv", sep = "")
    },
    content = function(file) {
      write.csv(expenses(), file, row.names = FALSE)
    }  )}

# Run the application
shinyApp(ui = ui, server = server)
```

APPENDIX B

SCREENSHOTS

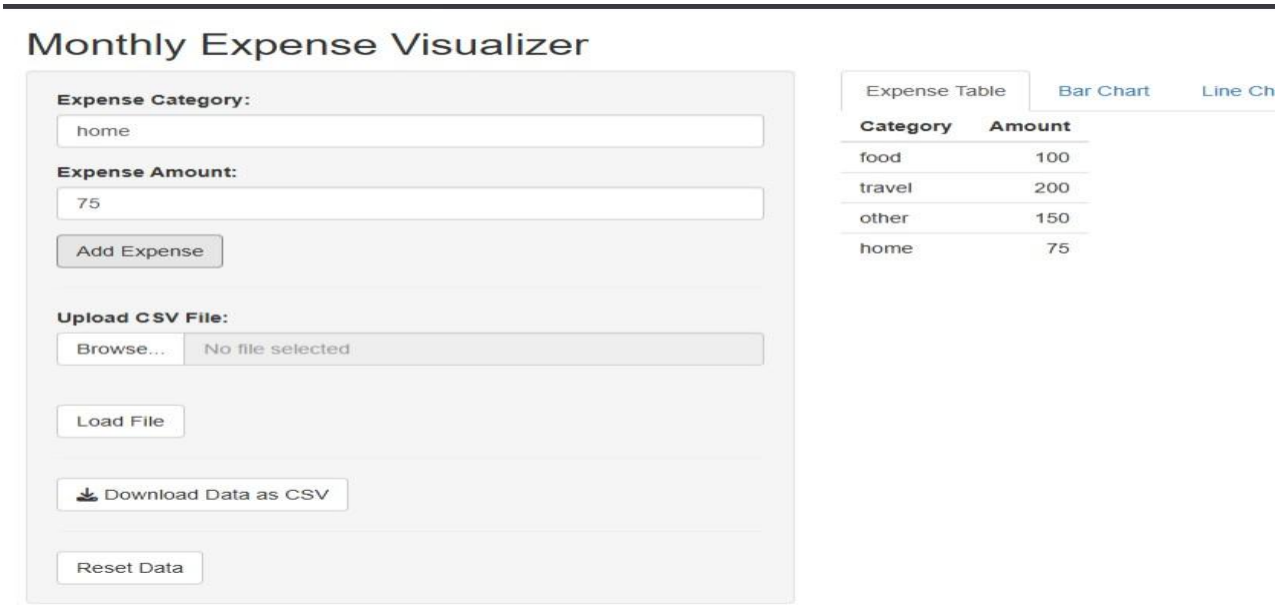

Fig B.1 Create a Input Table

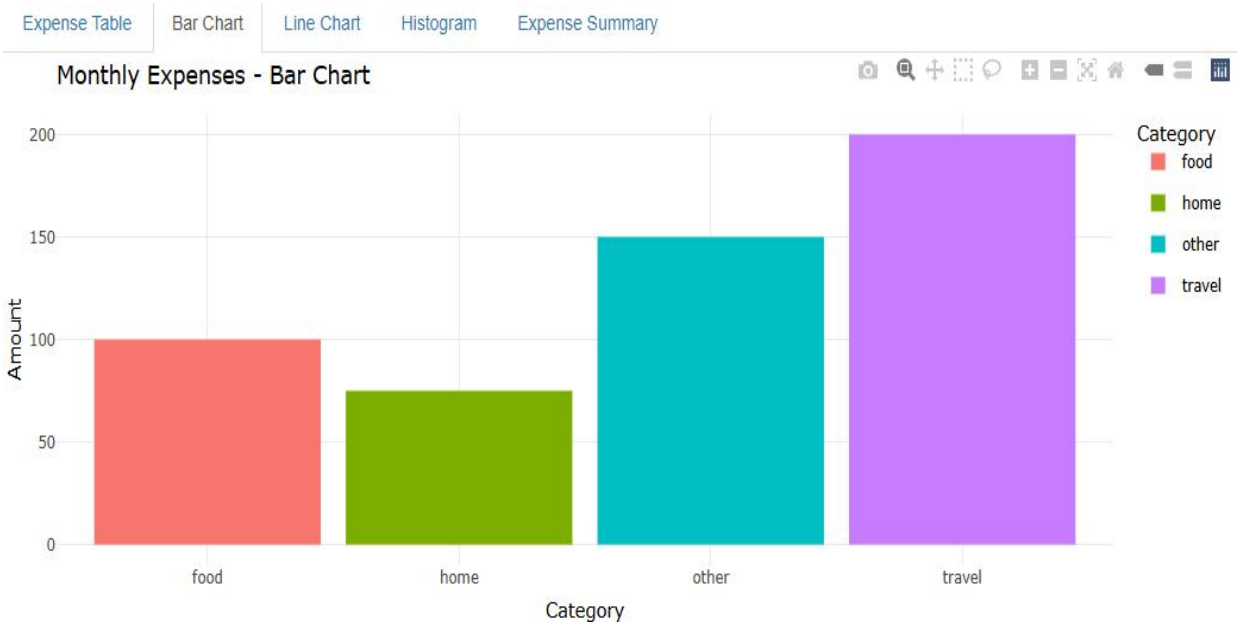

Fig B.2 Barchart

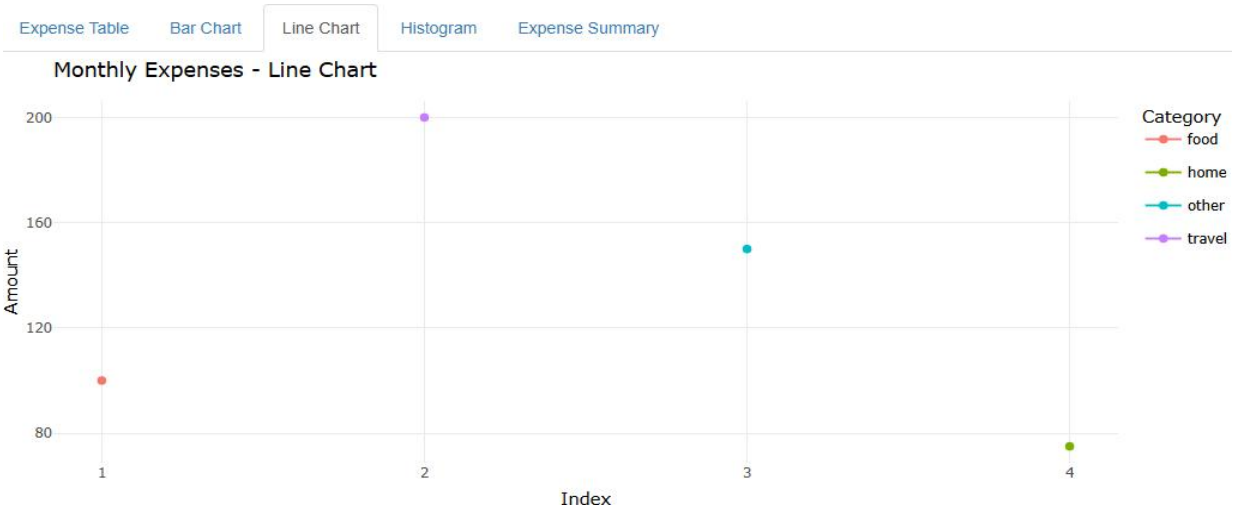

Fig B.3 Linechart

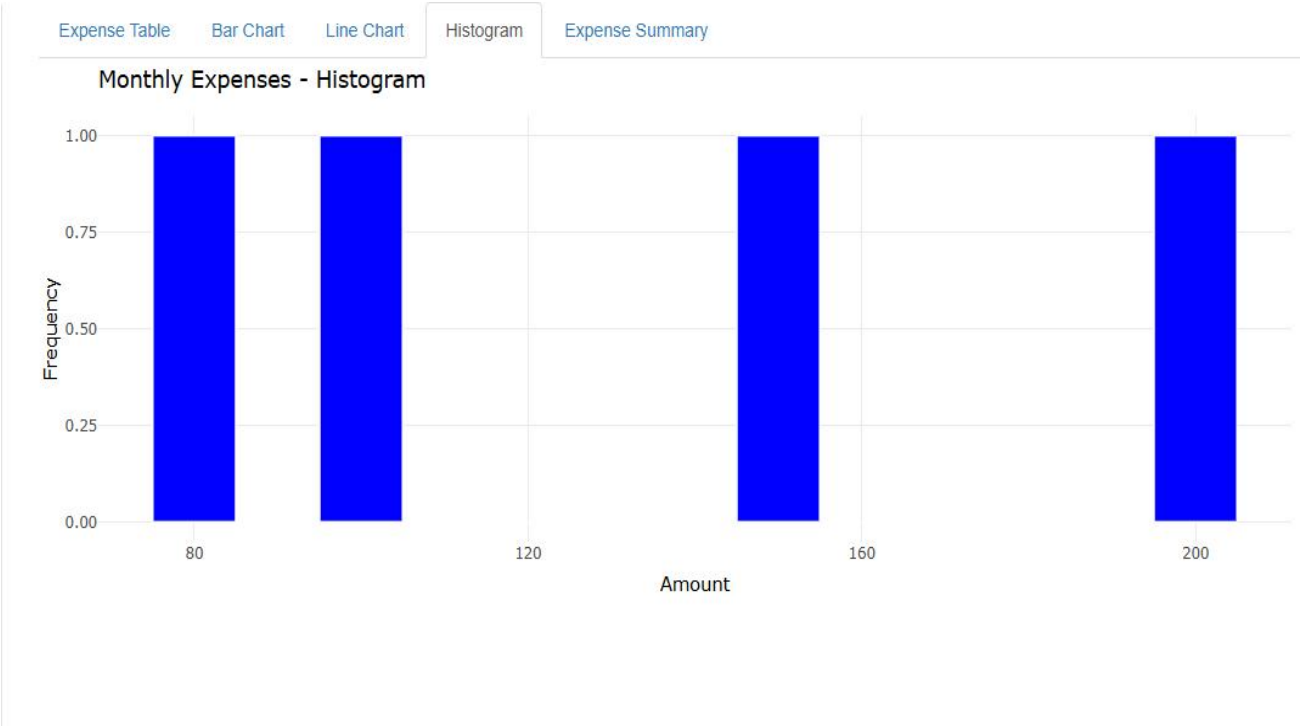

Fig B.4 Histogram

Fig B.4 Summary

REFERENCES

1. R Core Team (2024). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria. https://www.r-project.org
2. Chang, W. et al. (2024). *shiny: Web Application Framework for R*. R package version 1.8.0. https://CRAN.R-project.org/package=shiny
3. Wickham, H. (2016). *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York.
4. Sievert, C. (2020). *Interactive Web-Based Data Visualization with R, plotly, and shiny*. Chapman and Hall/CRC.
5. Wickham, H., François, R., Henry, L., & Müller, K. (2023). *dplyr: A Grammar of Data Manipulation*. R package version 1.1.4. https://CRAN.R-project.org/package=dplyr
6. Shiny Gallery. (n.d.). *RStudio Shiny Examples and Showcase*. Retrieved from https://shiny.rstudio.com/gallery/
7. Plotly for R. (n.d.). *Interactive Graphing Library*. Retrieved from https://plotly.com/r