Chapter 1

INSTALLATION AND SETUP PROCEDURE

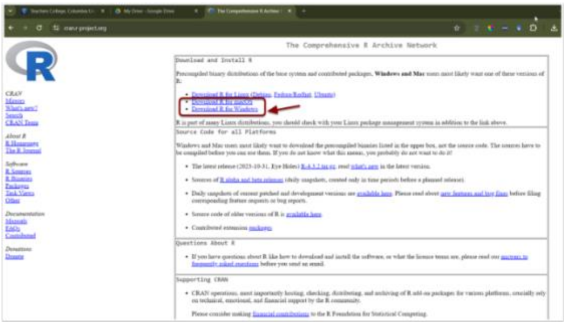
R is an open-source software environment for statistical computing and graphics.

R allows integration with the procedures written in the C, C++, .Net, Python or FORTRON language for efficiency.

It is freely available under GNU, General public license; and precompiled binary versions are provided for various operating systems like Linux, Windows, and MacOS.

INSTALLING R:

- Go to the official website of R.
- Select the mirror site to download from
- Select “Download R for Windows” to download the executable file(.exe)
- Select “base” or “Install R for the first time.”
- Open the downloaded file.
- Select the language.
- Choose where to install R.
- Select the type of installation (32-bit or 64-bit or other)
- Specify if you want to custom your startup.
- Choose the folder to save R within. (generally Program Files(x86))
- Select additional shortcuts if you want.
- Click Finish.



The Comprehensive R Archive Network

Download and Install R

Precompiled binary distributions of the base system and contributed packages, **Windows and Mac** users most likely want one of these versions of R:

- Download R for Linux (Debian, Fedora/Redhat, Ubuntu)
- Download R for macOS
- Download R for Windows

R is part of many Linux distributions, you should check with your Linux package management system in addition to the link above.

Source Code for all Platforms

Windows and Mac users most likely want to download the precompiled binaries listed in the upper box, not the source code. The sources have to be compiled before you can use them. If you do not know what this means, you probably do not want to do it!

- The latest release (2023-10-31, Eye Holes) R-4.3.2.tar.gz, read what's new in the latest version.
- Sources of R alpha and beta releases (daily snapshots, created only in time periods before a planned release).
- Daily snapshots of current patched and development versions are available here. Please read about new features and bug fixes before filing corresponding feature requests or bug reports.
- Source code of older versions of R is available here.
- Contributed extension packages

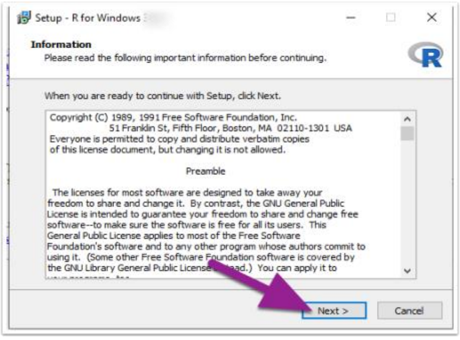
Questions About R

- If you have questions about R like how to download and install the software, or what the license terms are, please read our answers to frequently asked questions before you send an email.

Supporting CRAN

- CRAN operations, most importantly hosting, checking, distributing, and archiving of R add-on packages for various platforms, crucially rely on technical, emotional, and financial support by the R community.

  Please consider making financial contributions to the R Foundation for Statistical Computing.



Information

Please read the following important information before continuing.

When you are ready to continue with Setup, click Next.

Copyright (C) 1989, 1991 Free Software Foundation, Inc.
51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to

FIGURE 01

Chapter 2

INTRODUCTION

2.1. OVERVIEW OF THE PROJECT

The Student Attendance System is a software solution designed to efficiently record, track, and analyze student attendance. It allows teachers and administrators to easily manage attendance data, calculate attendance percentages, and generate reports. The system collects essential student information (e.g., name, ID, course) and records attendance (present or absent) for each session or date.

Key features include:

- Data Collection: Input and store student details and attendance records.
- Data Preprocessing: Clean and format data to ensure accuracy and consistency.
- Attendance Recording: Manually or automatically mark students as present or absent.
- Analysis: Calculate individual and class-wide attendance percentages.
- Visualization: Generate graphs to track attendance trends.
- Report Generation: Export reports for individual students or the entire class.
- Search and Filtering: Quickly find records based on criteria like date, student name, or course.

The system aims to save time, improve accuracy, and provide insights into student attendance patterns, helping educators make informed decisions. It can be extended with features like notifications and mobile access in the future.

2.2. AIM OF THE PROJECT

The **aim of the Student Attendance System** is to streamline and automate the process of recording, tracking, and analyzing student attendance. Specifically, it seeks to:

1. **Simplify Attendance Management**: Provide an efficient, easy-to-use platform for teachers and administrators to record and manage student attendance for various classes and sessions.

2. **Improve Accuracy**: Ensure accurate attendance records by automating calculations and reducing human error in manual entry.

3. **Track Student Participation**: Monitor individual student attendance over time to identify patterns, such as frequent absences, and provide insights into student engagement.

4. **Generate Reports**: Facilitate the generation of detailed, exportable attendance reports for individual students or entire classes, aiding in academic tracking and administrative tasks.

5. **Enhance Decision Making**: Offer valuable insights and analytics on overall class attendance, helping educators make informed decisions about student performance, interventions, and class management.

Overall, the system aims to save time, increase productivity, and improve the management of student attendance data in educational institutions.

Chapter 3

REQUIREMENTS SPECIFICATION

3.1. FUNCTIONAL REQUIREMENTS

- Student Registration: Add student details (ID, name, course).
- Mark Attendance: Teachers can mark attendance (present/absent) for each session.
- View Attendance: View individual student attendance records.
- Generate Reports: Generate attendance percentage reports.
- Export Data: Export attendance data in CSV/Excel format.
- Absence Alerts: Notify when a student has excessive absences.
- Search: Search attendance by student ID, date, or course.

3.2. NON-FUNCTIONAL REQUIREMENTS

- Usability: Easy-to-use interface (CLI or GUI with `shiny`).
- Performance: Handle large datasets efficiently.
- Security: Secure access to data (e.g., authorization).
- Data Integrity: Ensure accurate and tamper-proof attendance records.
- Scalability: Support growth in students and classes.
- Availability: System should be reliable and available during class times.

3.3. HARDWARE & SOFTWARE REQUIREMENTS

- Programming Language: R
- Libraries: dplyr, ggplot2, lubridate, shiny (for UI), readr (for data import/export).
- Operating System: Windows, Linux, or macOS.
- Server: At least 8GB RAM, Intel i5 or better for hosting the Shiny app.
- Client Devices: Any device with a modern web browser.
- Storage: SSD with at least 100 GB.
- Network: Stable internet connection for app access.

Chapter 4

DETAILED DESIGN (Methodology)

Detailed Steps:

1. Take Attendance:Calls the take_attendance() function to record attendance and store the result in attendance_data.

2. Print Attendance Record:Outputs the attendance data to the console for review.

Example:print(attendance_data)

3. Plot Attendance:Calls the plot_attendance() function to visualize the data.

4. Run Automatically:This function is called at the end of the script to execute the system when the file is run.

Design Features

1. User-Friendly Input: Provides interactive prompts for user input and validation for accuracy.

2. Error Handling:Ensures only valid attendance entries are accepted.

3. Visualization:Generates a clear and concise bar plot to summarize attendance.

4. Scalability:Can be extended for additional features like exporting data or analyzing trends.

5. Code Debugging:Includes print statements for debugging and checking intermediate outputs.

Chapter 5

IMPLEMENTATION

5.1. MODULE AND THEIR ROLES

Data Collection:

Gather or generate a dataset containing student information, such as **names, IDs, courses, and any other relevant details**.

Data Preprocessing:

Clean the data by handling missing values, correcting errors, and ensuring consistent formatting.

Data Storage:

Decide on the structure for storing attendance data. You can use data frames, where rows represent dates or sessions and columns represent students.

Input Mechanism:

Develop a way for the user (teacher or administrator) to input attendance data. This can be manual entry or reading from an external source (e.g., CSV files).

Attendance Recording:

Implement a mechanism to mark students as present or absent for each class session or date. This could be through a command-line interface or a graphical user interface.

Attendance Analysis:

Create functions to **calculate attendance percentages** for each student, as well as overall class attendance statistics.

Visualization:

Utilize R's visualization libraries (e.g., ggplot2) to create charts and graphs that show attendance trends over time.

Reports Generation:

Develop functionality to generate printable or exportable reports that show attendance records for individual students or entire classes.

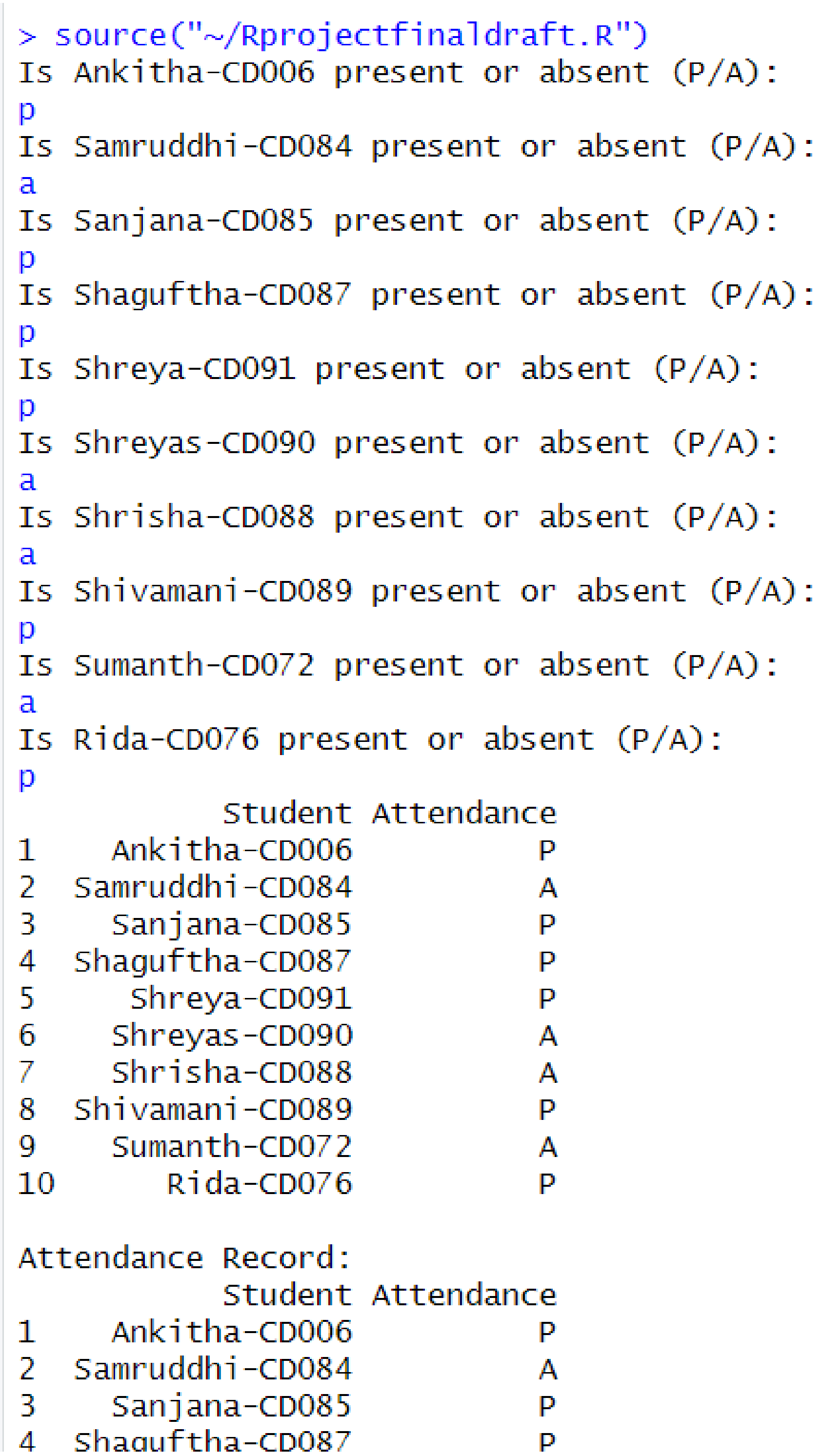
Chapter 6

SOURCE CODE

```
# Predefined list of student names
student_names <- c("Ankitha-CD006", "Samruddhi-CD084", "Sanjana-CD085", "Shaguftha-CD087", "Shreya-CD091","Shreyas-CD090","Shrisha-CD088","Shivamani-CD089","Sumanth-CD072","Rida-CD076")
# Function to take attendance for predefined students
take_attendance <- function() {
  # Initialize a vector to store attendance status
  attendance_status <- character(length(student_names))
  # Collect attendance status for each student
  for (i in 1:length(student_names)) {
    cat(paste("Is", student_names[i], "present or absent (P/A): "))
    attendance_status[i] <- toupper(trimws(readline())
    # Ensure only 'P' or 'A' are accepted
    while (!(attendance_status[i] %in% c("P", "A"))) {
      cat("Invalid input. Please enter 'P' for present or 'A' for absent.\n")
      cat(paste("Re-enter attendance for", student_names[i], "(P/A): "))
      attendance_status[i] <- toupper(trimws(readline()))
    }
  }
  # Return the data as a data frame
  data <- data.frame(Student = student_names, Attendance = attendance_status, stringsAsFactors = FALSE)

  # Debug: Print the collected attendance data

  print(data)

  return(data)

}

# Function to plot attendance data as a bar plot

plot_attendance <- function(attendance_data) {

  # Summarize the attendance data (counts of Present and Absent)

  attendance_summary <- table(attendance_data$Attendance)

  # Debug: Print the attendance summary

  print(attendance_summary)

  # Ensure that both "Present" and "Absent" categories exist and are ordered correctly

  if (!"P" %in% names(attendance_summary)) {

    attendance_summary["P"] <- 0  # If there are no presents, set present to 0

  }

  if (!"A" %in% names(attendance_summary)) {

    attendance_summary["A"] <- 0  # If there are no absents, set absent to 0

  }

  # Make sure the "Present" bar (P) comes first and the "Absent" bar (A) comes second

  attendance_summary <- attendance_summary[c("P", "A")]

  # Create a bar plot
  barplot(attendance_summary,
          main = "Student Attendance",
          xlab = "Total student Attendance Status",
          ylab = "Number of Students",
          col = c("purple", "pink"),  # Color for Present and Absent
          beside = TRUE,
          names.arg = c("Present", "Absent"))
}
# Main function to execute the attendance system
attendance_system <- function() {
  # Take attendance using predefined student names
  attendance_data <- take_attendance()
  # Print the attendance record to the console
  cat("\nAttendance Record:\n")
  print(attendance_data)
  # Plot the attendance data
  plot_attendance(attendance_data)
}
# Run the attendance system
attendance_system()
```

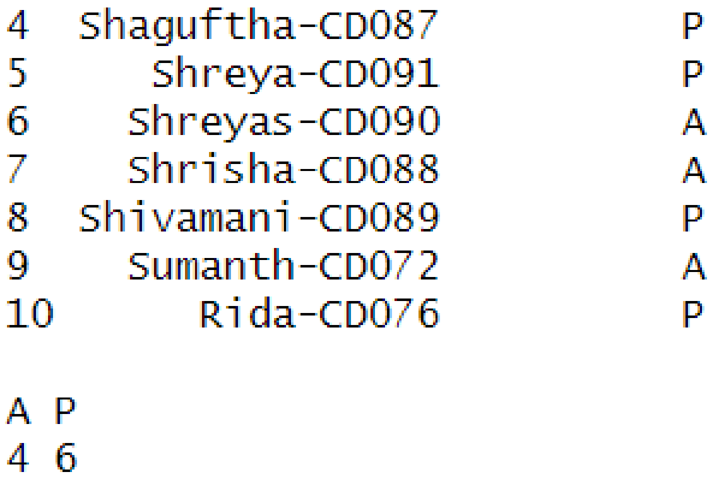
Chapter 6

Output



```
> source("~/Rprojectfinaldraft.R")
Is Ankitha-CD006 present or absent (P/A):
p
Is Samruddhi-CD084 present or absent (P/A):
a
Is Sanjana-CD085 present or absent (P/A):
p
Is Shaguftha-CD087 present or absent (P/A):
p
Is Shreya-CD091 present or absent (P/A):
p
Is Shreyas-CD090 present or absent (P/A):
a
Is Shrisha-CD088 present or absent (P/A):
a
Is Shivamani-CD089 present or absent (P/A):
p
Is Sumanth-CD072 present or absent (P/A):
a
Is Rida-CD076 present or absent (P/A):
p
           Student Attendance
1    Ankitha-CD006          P
2  Samruddhi-CD084          A
3    Sanjana-CD085          P
4  Shaguftha-CD087          P
5     Shreya-CD091          P
6    Shreyas-CD090          A
7    Shrisha-CD088          A
8  Shivamani-CD089          P
9    Sumanth-CD072          A
10      Rida-CD076          P

Attendance Record:
           Student Attendance
1    Ankitha-CD006          P
2  Samruddhi-CD084          A
3    Sanjana-CD085          P
4  Shaguftha-CD087          P
```

FIGURE 02



```
4   Shaguftha-CD087          P
5      Shreya-CD091          P
6     Shreyas-CD090          A
7     Shrisha-CD088          A
8   Shivamani-CD089          P
9     Sumanth-CD072          A
10      Rida-CD076           P

A P
4 6
```

FIGURE 03

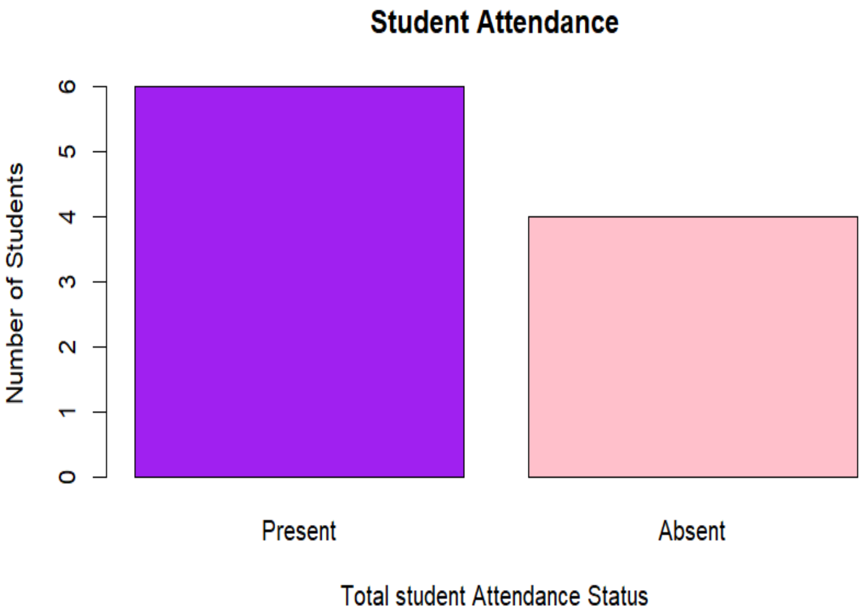


FIGURE 04

Chapter 8

CONCLUSION

In conclusion, this approach demonstrates the various tools and techniques available in R for managing and analyzing attendance data. By cleaning the data, recording attendance, and calculating attendance percentages, we gain valuable insights into student participation. The use of visualization tools like ggplot2 helps in identifying trends, and the ability to filter and export the data allows for effective reporting. Overall, this process highlights the power of R in data manipulation, analysis, and visualization, making it an excellent tool for educational data management.

REFERENCES

[1]. Introduction to Scientific Programming and Simulation Using R.Chapman & Hall/CRC by Jones, O.,Maillardet.R. and Robinson.A from The R series in the year 2014.

[2]. Statistics: An Introduction using R by Michael J.Crawley from Wiley , Second edition,2015.

[3]. Wickham, H.& Grolemund,G. for Data Science.O’Reilly:New York.