TECHNOLOGICAL INSTITUTE OF THE PHILIPPINES
Quezon City

IT 030 - Data Analytics
Final Project

Leader: NOVENARIO, JOSE MIGUEL Member: BALUYUT, JEROME Member: MEJIA, IGEL Member: PASCUAL, JOHN PAUL Member: VASQUEZ, CHRISTIAN LLOYD	Date: July 02, 2025
Section: IT33S6	**Instructor: Ms. Nila D. Santiago**

1. Objectives

This project is designed to enable you to:

- Develop predictive models by applying all data analytics techniques and algorithms covered in the course using R.
- Implement a data analytics pipeline in a real-world problem, from data acquisition and cleaning to analysis, modeling, and visualization.
- Write a data analytics article by comprehensively documenting and presenting the entire data analysis pipeline, its findings, and implications.

2. Student Outcomes

Through the successful completion of this project, you will demonstrate proficiency in the following student outcomes:

- Student Outcome 1.1: Analyze a complex computing problem.
- Student Outcome 1.2: Apply principles of computing and other relevant disciplines to identify solutions.

3. Project Timeline

Phase	Task	Deliverables
Phase 1	Project Setup & Data Acquisition Plan	Project Proposal document, detailing your problem, questions, and a concrete plan for data acquisition.
Phase 2	Data Acquisition & Initial Cleaning	Raw data files, your initial R script(s) for data acquisition and cleaning, and a brief summary of the raw data's state.
Phase 3	Exploratory Analysis & Model Planning	An EDA report showcasing your initial findings with charts, and a detailed plan for your modeling

		approach.
Phase 4	Model Development & Data Analytics	Your R script(s) containing your model development and evaluation code, along with a summary of your model insights and results.
Phase 5	Dashboard Development & Final Report	R Shiny Dashboard Application: Your full, working interactive app. R Code on GitHub: All your project's code and data, neatly organized in a public repository. Data Analytics Article: Your complete written report. Project Presentation: Ready for your showcase.

Title of the Project

1. Introduction
   1.1 Business Understanding
   1.2 Background of the Study
   1.3 Statement of the Problem
   1.4 Research Questions

2. Data Source & Acquisition
   2.1 Chosen Public Dataset/Topic

   2.2 Data Source Details
   - *If API- Give the API provider's name, the exact website link for the API, how you accessed it (e.g., API key), and any limits (like how many requests you could make).*
   - *If Web Scraping- Give the exact website links you scraped. Describe which specific parts of the webpage you extracted. Mention any challenges you faced (like dynamic content or the website blocking you) and how you handled them.*

   2.3 Data Acquisition Methodology
   - *Describe the step-by-step process you followed to get the data. What R packages did you use (e.g., httr, rvest)? How did you handle errors? How did you save the raw data? Include brief, relevant R code snippets to show how you acquired data.*

3. Data Cleaning & Preprocessing
   3.1 Initial Data State
   - *Describe what your raw data looked like. What problems did it have (e.g., missing values, wrong data types, messy text)?*

   3.2 Cleaning and Transformation
   - *Explain each step you took to clean and prepare your data. Why did you do it? Include brief, relevant R code snippets to demonstrate key cleaning steps.*

4. Exploratory Data Analysis (EDA) & Insights
   4.1 Analytical Approach
   - *Describe the main ways you explored your data to find answers (e.g., looking for trends over time, comparing different groups, finding correlations).*

   4.2. Findings and Insights
   - *Present your most important discoveries from the data through charts; explain what they mean.*

   For each insight:
   - *State which of your key questions it answers.*
   - *Provide the evidence from your data.*
   - *Explain how this insight helps solve your problem statement and contributes to the overall business understanding.*

   - *Include small, static R plots (e.g., `ggplot2` outputs saved as images) or summary tables to support your insights. Make sure they are labeled.*

5. Dashboard Design & Implementation
   5.1 Dashboard Framework & Libraries
   - *List the main R packages for charting (e.g., `plotly`, `ggplot2`, `leaflet`).*

   5.2 Dashboard Design Philosophy
   - *Why did you design your dashboard this way? How does its layout (e.g., side panel for filters, tabs for different views) make it easy for users to find answers to your key questions and address the business problem?*

   5.3 Visualizations
   - *For each of your 3+ interactive charts, discuss the following:*
     - *What does this chart show?*
     - *How does it help answer your key questions or address your problem?*
     - *What can users do with it (e.g., select different options, hover for details, zoom)?*
     - *Include clear screenshots of your dashboard, labeling each one.*

   5.4 Implementation
   - *Briefly explain how your Shiny app works. How does it load data? How do the filters update the charts? What R code snippets show key parts of your `app.R`?*
6. Conclusion
7. References

Appendices:

A. Source Code and Sample Output
B. Members' Detailed Contribution

TECHNOLOGICAL INSTITUTE OF THE PHILIPPINES

938 Aurora Blvd. Cubao, Quezon City

COLLEGE OF COMPUTER STUDIES

Information Technology Department

StudyPulse: An Analytical Dashboard for Student Well-being and Performance

In Partial Fulfillment of the Requirements for

ITE 030 - Data Analytics

by:

Baluyut, Jerome J.

Mejia, Igel D.

Novenario, Jose Miguel R.

Pascual, John Paul L.

Vasquez, Christian Lloyd G.

Submitted to:

Ms. Nila D. Santiago

Instructor

July 2025

Table of Contents

1. Introduction

1.1 Business Understanding

Behavioral Insights and Evolving Metrics in Student Success

The concept of academic achievement is evolving, with teachers and scholars recognizing the role of non-academic variables in student performance. Performance has traditionally been evaluated by the scores on the standardized tests, participation in classes, and attendance. Although these signs are still significant, they do not always present a complete picture of what affects a student's learning process. Recently, it has been proven that behavioral and lifestyle factors, including sleep duration, study habits, time on screen, and emotional wellness, may have a significant impact on academic performance (West et al., 2019; Anwar et al., 2024). These components, however, often fail to be considered in institutional reviews, which leaves an enormous oversight on the nature of student support and assessment. There is an evident rise in data gathering concerning the interaction of learners with learning platforms, with the rise in the use of digital learning environments. The move provides a unique option to develop a more accurate perception of student needs and react to them by setting the mixture of behavioral knowledge and data analytics. However, numerous educational establishments still pay much attention to measuring cognitive outcomes without incorporating enough data on understanding how students live and learn outside the referential class. Responding to this concern, studies have revealed that activities of excessive engagement, such as screen time ensue to negatively interfere with sleep quality, which consequently influences the level of attention, memory retention, and emotional control- essential elements of academic performance (Hale & Guan, 2015; Perez-Chada et al., 2023). In addition, self-regulation, time management, and mental resilience are also becoming essential factors in long-term academic achievement (West et al., 2019).

Despite this mounting scientific evidence, there is a lack of serious attempts to integrate behavioral data into predictive models. This disconnect limits the capacity of educators to distinguish the students who may be in need because of those factors that are not evident based on the scores or tests. The association of behavioral indicators with academic records would help establish early warning systems to facilitate earlier and more autonomous interventions. Such statistical and machine learning allows one to study these intricate patterns and turn them into meaningful insights. The schools implementing such an inclusive approach are more likely to develop inclusive learning environments, and these settings can meet not only academic needs but also the well-being needs of students (Ouatik et al., 2022). Due to the existence of this need, the project StudyPulse has been created, and it was implemented in order to research the relationship between student behavior and academic performance, along with an imagined dataset simulating actual patterns. The project will provide a more detailed insight into the success of students (and their performance) through studying sleep quality, study hours, screen time, and similar variables through the application of data science techniques. The results can be utilized to develop strategies in institutions, the development of policies, and a more student-centered education.

Stakeholders:

- **Educators and School Administrators:** Teachers, guidance professionals, and school administrators are important in addressing the growth of students. They can formulate specific academic programs and interventions using the insights obtained on the dashboard that takes into account performance indicators and behavioural patterns such as sleep, screen time, and study patterns.
- **Parents and Students:** Families are active participants in the creation of daily routines, which affect the achievement of academic results. Availability of behavior-associated information enables the parents and students to make informed lifestyle decisions. Most of the time, it is as simple as changing bedtime routines or digital distractions, which can be incorporated into their set routines with guidance using the dashboard.
- **Guidance Offices and Institutions of learning:** The results can be used by the school and universities to bolster their academic and wellness programs. Having a better understanding of the impact of non-academic influences on performance, the institutions will create more balanced policies, plan their resources, and establish better student support services, which consider not only performance but well-being as well.

- **Policy Makers and Educational NGOs:** People making policy on education or concerned with student welfare will be able to learn through the insight of the project. The awareness of how attendance and relative tendency influence academic achievements will help them advance more comprehensive education strategies and shape the future initiatives that will focus on education as well as personal growth.

1.2 Background of the Study

As the education systems are turning more digital, data science has emerged as an important value addition in advancing student learning outcomes. The research has shown that in addition to the usual academic measures used to assess academic modifiers (test results and attendance rates), daily habits, about sleep duration, screen time, study rates, and physical activity, play a valuable role in understanding and assessing the level of academic performance (Hale & Guan, 2015; Perez-Chada et al., 2023). But these behavioral variables are often not well represented in institutional analytics, meaning we miss out on the chance to give the much-needed attention and early action.

The benefits of behavioral and academic data analysis have been highlighted recently due to the increasing use of mixed data analysis that brings more detailed educational information. In particular, Anwar et al. (2024) proved that any integration of the behavioral variables into data science models significantly increased the effectiveness of the academic performance prediction. Likewise, Ouatik et al. (2022) observed higher accuracy in machine learning models based on behavioral data as compared to machine learning models that are based only on their analysis of academic data. Findings are accompanied by an upward global trend of evidence-based, student-centred education systems moving towards proactive rather than reactive support strategies.

The StudyPulse project represents this paradigm shift as it studies synthetic student habit data of a simulated cohort of 1,000 students. The dataset simulates the effects of real-life behavioral characteristics of studying hours, sleep quality, and access to the internet in correlation with academic performance. Based on the R program in data processing, visualization, and predictive modeling, the project has produced an interactive dashboard that can reveal actionable trends to the educators and policy-makers. Even though the data is simulated to ensure privacy, nevertheless, it nevertheless reflects the behavioral patterns, so experiments and discoveries can be done safely. At its request to reveal the relationships between student well-being and academic achievement, StudyPulse will help in the long-term transformations towards personalizing education and ensuring more responsive institutions. The final goal of the project is to provide schools and teachers with evidence-based tools to better understand the lifestyles of students and learning processes, which will provide the background for more intelligent and inclusive academic policies.

1.3 Statement of the Problem

Although more and more learners have become aware of the effects of behavioral habits on student learning, a lot of learning centers continue to depend on academic data, which is merely the evaluation of performance once problems have already arisen. The use of late indicators does not allow for using them in advance to help the students and management of the issue before it grows. Despite widespread access to behavioral and lifestyle data, which has become increasingly common with the emergence of digitalization and surveys, they are poorly integrated into a single complete instrument promoting early academic intervention.

At present, schools and teachers are subject to several constraints:

- The absence of organized mechanisms of recording and understanding the peculiarities of student habits, including studying, sleep problems, and time spent on the screen.
- Low application of behavioral patterns in the identification of students who might be at risk academically.
- A narrow view of how the trends and correlations between individual habits and school performance change over time.
- Lack of adequate mechanisms to translate intricate behavioral data into useful and practical advice.

In the absence of a centralized and interactive space to examine these aspects, the stakeholders typically lack the information that will facilitate the implementation of individual and immediate systems of support. This project mitigates that deficiency by building StudyPulse, a data science–based dashboard that can identify the associations between student behavior and academic performance, towards a more reactive, student-controlled method of educational achievement.

1.4 Data Analytics Research Questions

- Which behavioural factors, the amount of time spent on the screen, the amount of study hours, the quality of sleep, and the exercise rate, have the most significant relationship to the performance of the students?
- What are some of the frequent problems or behaviors that have become common among students who are most likely not to excel in school?
- Which of these habits and behaviors have the most significant effects on the performance of students?
- What is the balance between the amount of screen time and the length of sleep, and how does it affect the performance of students?
- What role do demographic and lifestyle factors, including age, gender, quality of diet, and internet access, play in the variations in academic performance of students?

2. Data Source and Acquisition

2.1 Public Chosen Data Source

Table 1

Details of Chosen Public Data Sources

Data Source Name	Specific URL/Access Point	Data Type/Format	Key Information Available
Student Habits vs Academic Performance (Kaggle)	https://www.kaggle.com/datasets/jayaantanaath/student-habits-vs-academic-performance	CSV (Downloadable)	Student demographics, academic scores, daily habits (study time, sleep, screen time), and behavioral patterns

2.2 Data Acquisition Methodology

The dataset analyzed within the project is a Student Habits vs Academic Performance, which was obtained from Kaggle, which is one of the most famous websites where the data of both publicly used and data science competitions are available. It was developed by Jayanta Nath, and the dataset has 1,000 rows of synthetic but lifelike student records. The official Kaggle API was deployed to extract the data. This involved the download of a JSON file with API credentials (kaggle.json), not only with the username but also the secret API key linked to the personal Kaggle account. This file was saved in a concealed file whose declaration was .kaggle in the working directory. The API enables authentication and safe access to data sets that can be done on the user's machine, not necessarily through manual download by a web browser. Kaggle API places no strict restrictions on educational use with regard to downloading datasets, thus being a convenient and effective instrument in the run of reproducible scholarly work. This API can be incorporated into the R environment, allowing the dataset to be accessed and accessed programmatically, giving repeatability and scalability in acquiring data.

2.3.1 Student Habits vs Academic Performance: Dataset Acquisition

The data acquisition procedure was methodological and focused on reliability, transparency, and reusability of data used in subsequent analysis. The step-by-step methodology followed is as follows:

- **Kaggle Account Setup and API Authentication:** The initial task was to create API credentials with the Kaggle user account. These credentials were in the form of a kaggle.json file and were used as the authentication strategy to access the Kaggle dataset repository API. The credentials were stored safely in the R project folder so that they were used continually and did not require re-authentication.
- **Dataset Download:** Having the API set correctly, the dataset was downloaded from the official source of Kaggle. The dataset was already in a CSV file.
- **File Structure and Compatibility:** The data was provided as a CSV file, which is best suited in R. This made it compatible with functions being used in inspection, wrangling, visualization, and modeling of data. Several attributes were placed in the CSV file, which included gender, age, hours of sleep, social media and Netflix usage, internet quality, parental education, mental health rating, exam score, etc.
- **Error Handling and Validation:** The measures followed in the course of the acquisition process were to ensure that the data was intact and usable. The dataset was initially examined to ensure structure, the existence of any missing or invalid values, and consistency with data types. Records that existed as duplicates were deleted to ensure the integrity of the data.
- **Purpose of Data Acquisition:** The dataset was gathered to assist in exploring the correlation between the non-academic behavioral factors (including screen time, duration of sleep, diet, and internet quality) and academic performance (that is, the results of exams). Moreover, the dataset was used to develop predictive models and visual dashboards that could be used to identify high-risk students and deliver policy guidance in education facilities.
- **Final Storage:** The raw dataset was kept after validation and basic cleaning to allow traceability, and another, cleaned version, was saved specifically to further the analysis. This clean group of data featured the newly created variables, including screen_time and performance_level, which would relate to the deepening of the analysis in the following stages of the project.

3. Data Cleaning & Preprocessing

3.1 Initial Data State

The **student_habits_performance.csv** raw data file includes **1,000 artificial records** of students that may include **grades** and **academic performance** on one hand and **living habits** like the **hours of sleep**, **quality of the internet**, **screen time**, and **extra-curricular activities** on the other. After importing data into **RStudio**, checking the first **structure** and **summary** (**str()**, **summary()**, and **colSums(is.na(...))**) highlighted that the data was relatively **clean**:

- There was **not a single missing value** in the variables

```
> colSums(is.na(data))  # Check missing values
           student_id                   age                gender   study_hours_per_day
                    0                     0                     0                     0
   social_media_hours         netflix_hours         part_time_job attendance_percentage
                    0                     0                     0                     0
          sleep_hours          diet_quality    exercise_frequency parental_education_level
                    0                     0                     0                     0
     internet_quality   mental_health_rating extracurricular_participation      exam_score
                    0                     0                     0                     0
```

GROUP 6
IT 030 - IT33S6

- There were **no duplicate records,** and the duplicate check was done as a precautionary measure with **duplicated()**.

```
> # 2. Remove duplicates
> data <- data[!duplicated(data), ]
> nrow(data)  # Confirm no duplicates remain
[1] 1000
```

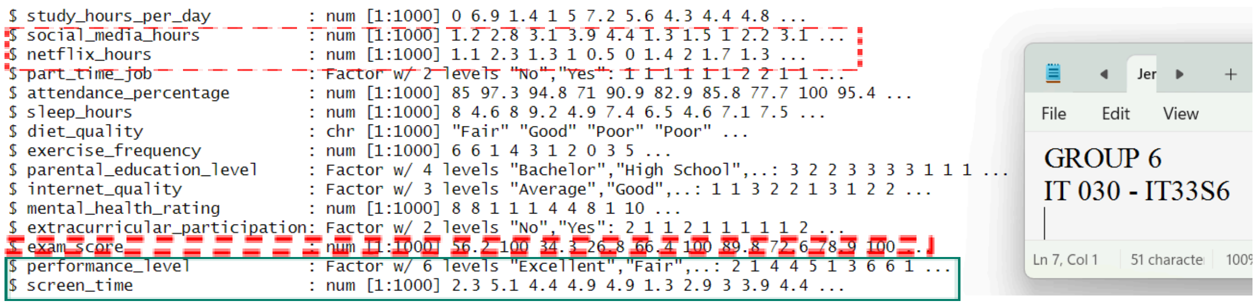
GROUP 6
IT 030 - IT33S6

- The **categorical variables** were stored in **character types** instead of **factors**, which might impact the ordeal of **grouping** and **modeling** (at **student_id**, **gender**, **part_time_job**, **parental_education_level** etc.).

```
Rows: 1000 Columns: 16
── Column specification ──────────────────────────────────────────────
Delimiter: ","
chr (7): student_id, gender, part_time_job, diet_quality, parental_education_level, internet_quality, extracurricular_...
dbl (9): age, study_hours_per_day, social_media_hours, netflix_hours, attendance_percentage, sleep_hours, exercise_fre...
```

GROUP 6
IT 030 - IT33S6

- There was **no screen_time variable** to begin with, and this variable has been partitioned into two columns: **social_media_hours** and **netflix_hours**. Such a piecemeal arrangement could not render the total screen display into a readable form easily.



- The continuous **exam_score** feature was turned into a new feature with more understandable segmentation of the results: **performance_level**. The scores were divided into six categories, which were **Excellent (90 and above)**, **High (80-89)**, **Satisfactory (70-79)**, **Medium (60-69)**, **Fair (50-59),** and **Low (50 and below)**. Such a classification makes the analysis more **legible** and more meaningful to make comparison between the performance patterns of students and **visualisation**.

Initial R script

```
# --- Load essential libraries ---

library(tidyverse)   # Includes dplyr, ggplot2, readr, etc.

library(jsonlite)    # For working with JSON (e.g., kaggle.json)

library(httr)        # Optional for advanced API access

library(readr)       # For reading CSV

library(ggplot2)     # For plotting

Sys.setenv(KAGGLE_USERNAME = fromJSON(".kaggle/kaggle.json")$username)

Sys.setenv(KAGGLE_KEY = fromJSON(".kaggle/kaggle.json")$key)

# --- Load the dataset from the extracted CSV file ---

data <- read_csv("student_habits_performance.csv")

# --- Initial Data State ---

# Check structure and basic statistics

str(data)

summary(data)

colSums(is.na(data))  # Check missing values

# --- Data Cleaning & Preprocessing ---
```

```
# 1. Convert relevant categorical variables to factor

data <- data %>%

  mutate(

    gender = as.factor(gender),

    part_time_job = as.factor(part_time_job),

    parental_education_level = as.factor(parental_education_level),

    internet_quality = as.factor(internet_quality),

    extracurricular_participation = as.factor(extracurricular_participation)

  )


# 2. Remove duplicates

data <- data[!duplicated(data), ]

nrow(data)  # Confirm no duplicates remain


# 3. Categorize exam_score into performance_level with 6 categories

data <- data %>%

  mutate(performance_level = case_when(

    exam_score >= 90 ~ "Excellent",

    exam_score >= 80 & exam_score < 90 ~ "High",

    exam_score >= 70 & exam_score < 80 ~ "Satisfactory",

    exam_score >= 60 & exam_score < 70 ~ "Medium",

    exam_score >= 50 & exam_score < 60 ~ "Fair",

    exam_score < 50 ~ "Low"

  ))

data$performance_level <- as.factor(data$performance_level)


# 4. Create derived variable 'screen_time' (social media + Netflix)

data <- data %>%

  mutate(screen_time = social_media_hours + netflix_hours)

# 5. Outlier removal for screen_time > 12 hours

ggplot(data, aes(y = screen_time)) +

  geom_boxplot(fill = "deepskyblue3", outlier.color = "red") +

  labs(title = "Boxplot of Total Screen Time", y = "Hours") +

  theme_minimal()


data <- data[data$screen_time < 12, ]


# Final checks

str(data)

summary(data)

head(data)


# Save cleaned dataset

write_csv(data, "cleaned_student_habits_performance_data.csv")
```

3.2 Cleaning and Transformation

A number of preprocessing operations were carried out through R to reflect on the quality of the data. These steps with explanations and justification are described below in detail;

Step 1: Load Libraries Needed

It loaded a set of important packages at start-up. These included:

- *tidyverse* – data wrangling and plotting (*dplyr, ggplot2, readr*)
- *jsonlite* – read the Kaggle API credential in kaggle.json
- *httr* – api request handling using

This setup ensured that we had an API configuration and data cleaning workflow with the capacity to use both of them.

```
# --- Load essential libraries ---
library(tidyverse)    # Includes dplyr, ggplot2, readr, etc.
library(jsonlite)     # For working with JSON (e.g., kaggle.json)
library(httr)         # Optional for advanced API access
library(readr)        # For reading CSV
library(ggplot2)      # For plotting
```

GROUP 6
IT 030 - IT33S6

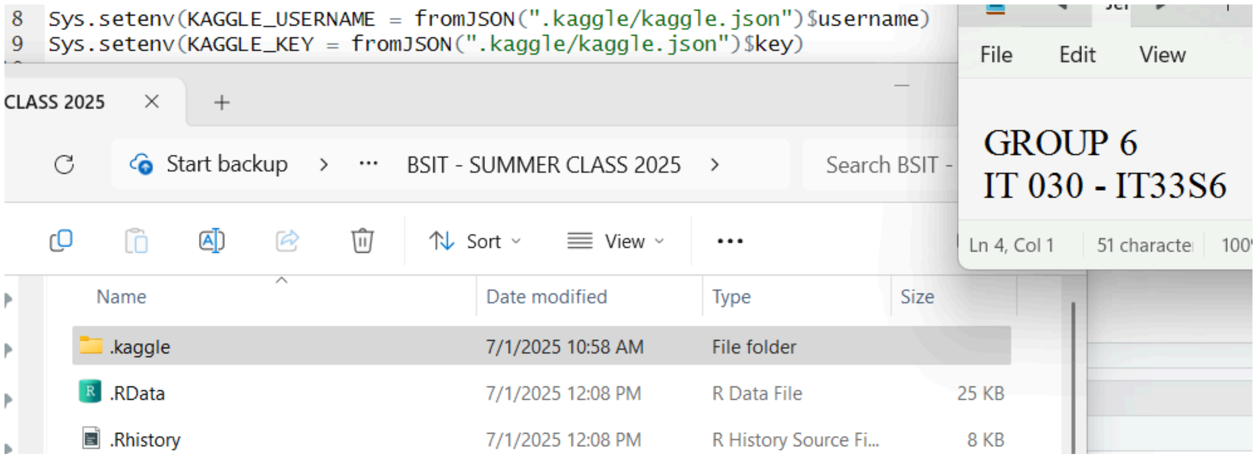
Ln 4, Col 1 | 51 characte | 100%

Step 2: Credentials for Kaggle API

In able to access the dataset through the Kaggle API, environment variables KAGGLE_USERNAME and KAGGLE_KEY were added with the help of fromJSON(".kaggle/kaggle.json"). This made any possible Kaggle downloads to be authenticated.

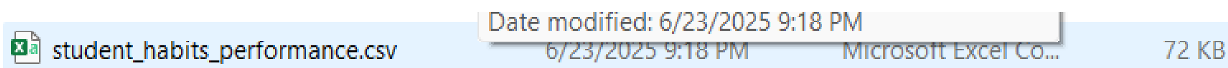
```
> # --- Set Kaggle credentials and working directory ---
> # Ensure kaggle.json is in the ".kaggle" folder inside your working directory
>
> Sys.setenv(KAGGLE_USERNAME = fromJSON(".kaggle/kaggle.json")$username)
> Sys.setenv(KAGGLE_KEY = fromJSON(".kaggle/kaggle.json")$key)
```

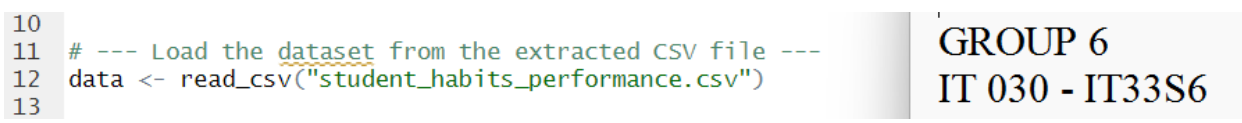
GROUP 6
IT 030 - IT33S6



Step 3: Load the Dataset

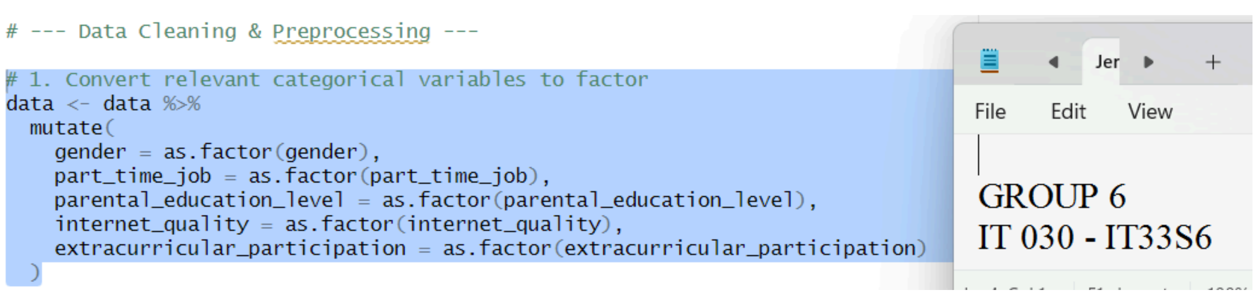
The student_habits_performance.csv file was already unzipped, and was read in R with read_csv(). That permitted the automatic identification of data types and column formats.





Step 4: Making Categorical Variables into Factors

The following variables had to be consciously interpolated between character strings and factor types to make sure that categorical variables were appropriately statistically analysed and visualised:



These transformations could be done cleanly using mutate () and as.factor()

Step 5: Delete duplicate records

No duplicates were identified at the very first stage; however, the !duplicated(data) filter was applied to verify the uniqueness of each record. The step aids in eliminating biased and repetitive entries in modeling.

```
# 2. Remove duplicates
data <- data[!duplicated(data), ]
nrow(data)  # Confirm no duplicates remain
```

GROUP 6
IT 030 - IT33S6

Step 6: Classify Academic Performance

Based on exam_score, a new categorical variable was created performance_level. It classified the performance of students into six levels:

```
# 3. Categorize exam_score into performance_level with 6 categories
data <- data %>%
  mutate(performance_level = case_when(
    exam_score >= 90 ~ "Excellent",
    exam_score >= 80 & exam_score < 90 ~ "High",
    exam_score >= 70 & exam_score < 80 ~ "Satisfactory",
    exam_score >= 60 & exam_score < 70 ~ "Medium",
    exam_score >= 50 & exam_score < 60 ~ "Fair",
    exam_score < 50 ~ "Low"
  ))
data$performance_level <- as.factor(data$performance_level)
```

File Edit View

GROUP 6
IT 030 - IT33S6

Ln 4, Col 1 | 51 characte | 100%

This category offered more detailed categorization to subsequent classification models and analysis descriptions.

Step 7: Create a Derived Variable *(screen_time)*

The sum of social media hours and Netflix hours was constructed as another variable, called screen time. This single measure made it possible to gain a more informative picture of the overall digital media exposure to students.

```
# 4. Create derived variable 'screen_time' (social media + Netflix)
data <- data %>%
  mutate(screen_time = social_media_hours + netflix_hours)
```

GROUP 6
IT 030 - IT33S6

Step 8: Outlier Detection for screen_time

A boxplot was generated to allow an examination of outliers of screen time visually. There are no extreme outliers, but a safety filter was used to exclude any student whose screen_time was above 12 hours. Practically, the highest value seen was 10.1 hours, and there were no records that were dropped.

Figure 1 indicates how the students spend their overall daily screen time (social media use and Netflix consumption). It turned out that there were no extreme outliers, with the maximum screen time amounting to 10.1 hours.

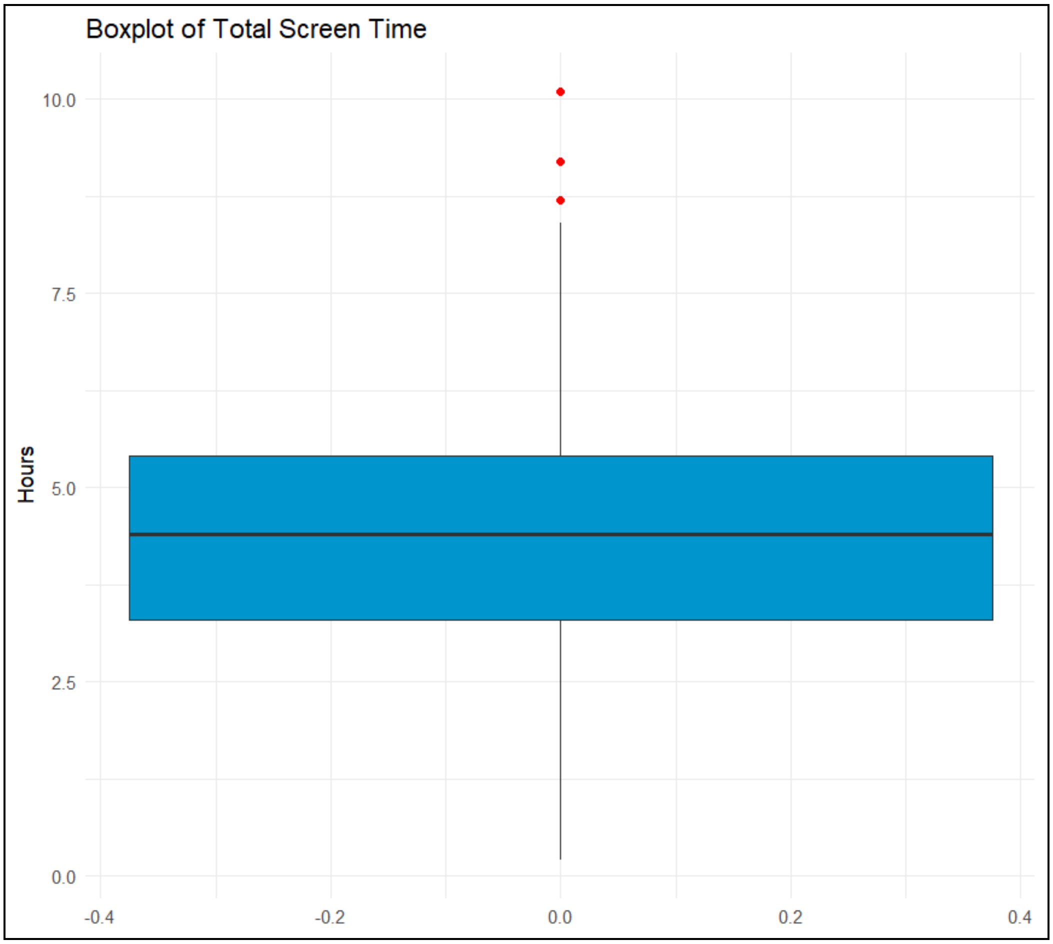
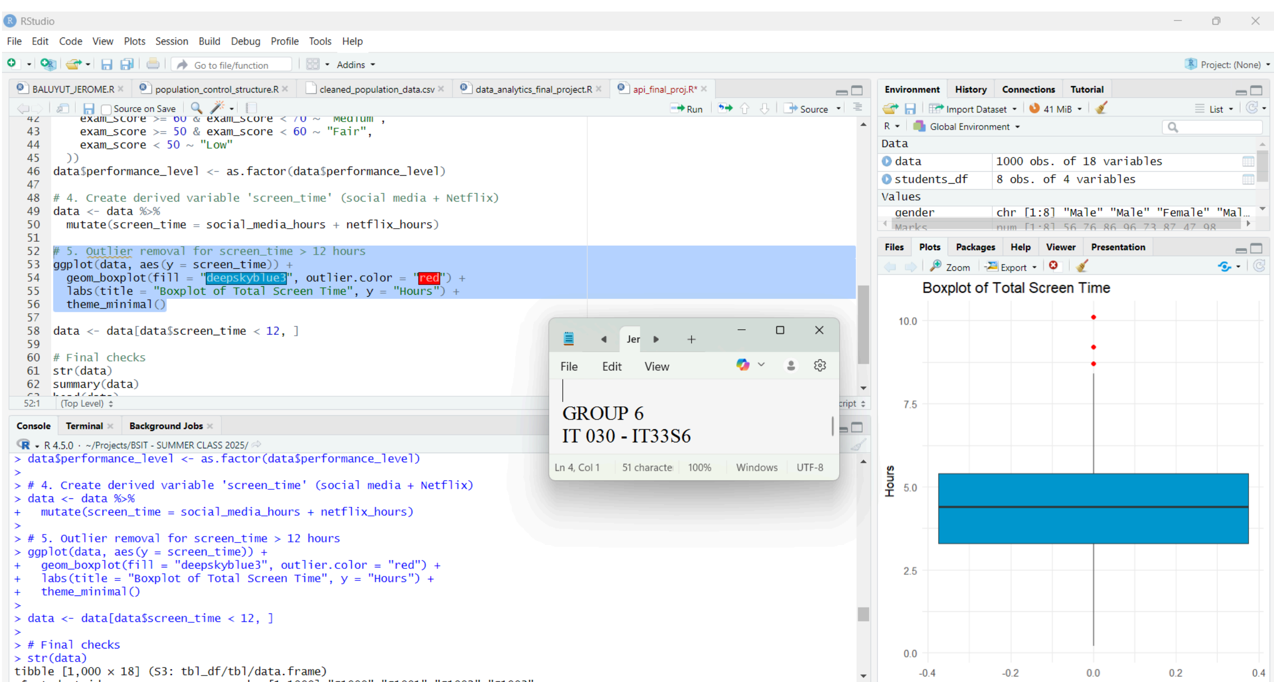


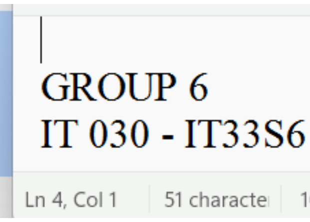
Figure 1. Boxplot of Total Screen Time Among Students



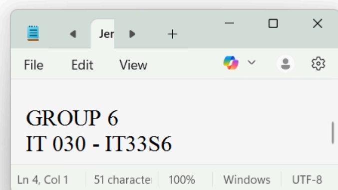
Step 9: Final Checks and Export

To ensure the structure, range of values, and type conversions, the review of the dataset was conducted with the help of str(), summary(), and head(). The cleaned data was then exported to cleaned_student_habits_performance_data.csv after validation using the write csv () command.

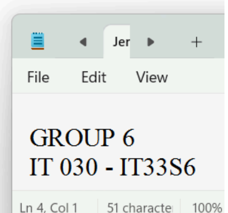
```r
# Final checks
str(data)
summary(data)
head(data)

# Save cleaned dataset
write_csv(data, "cleaned_student_habits_performance_data.csv")
```




```
> # Final checks
> str(data)
tibble [1,000 × 18] (S3: tbl_df/tbl/data.frame)
 $ student_id                   : chr [1:1000] "S1000" "S1001" "S1002" "S1003" ...
 $ age                          : num [1:1000] 23 20 21 23 19 24 21 21 23 18 ...
 $ gender                       : Factor w/ 3 levels "Female","Male",..: 1 1 2 1 1 2 1 1 1 1 ...
 $ study_hours_per_day          : num [1:1000] 0 6.9 1.4 1 5 7.2 5.6 4.3 4.4 4.8 ...
 $ social_media_hours           : num [1:1000] 1.2 2.8 3.1 3.9 4.4 1.3 1.5 1 2.2 3.1 ...
 $ netflix_hours                : num [1:1000] 1.1 2.3 1.3 1 0.5 0 1.4 2 1.7 1.3 ...
 $ part_time_job                : Factor w/ 2 levels "No","Yes": 1 1 1 1 1 1 2 2 1 1 ...
 $ attendance_percentage        : num [1:1000] 85 97.3 94.8 71 90.9 82.9 85.8 77.7 100 95.4 ...
 $ sleep_hours                  : num [1:1000] 8 4.6 8 9.2 4.9 7.4 6.5 4.6 7.1 7.5 ...
 $ diet_quality                 : chr [1:1000] "Fair" "Good" "Poor" "Poor" ...
 $ exercise_frequency           : num [1:1000] 6 6 1 4 3 1 2 0 3 5 ...
 $ parental_education_level     : Factor w/ 4 levels "Bachelor","High School",..: 3 2 2 3 3 3 3 1 1 1 ...
 $ internet_quality             : Factor w/ 3 levels "Average","Good",..: 1 1 3 2 2 1 3 1 2 2 ...
 $ mental_health_rating         : num [1:1000] 8 8 1 1 1 4 4 8 1 10 ...
 $ extracurricular_participation: Factor w/ 2 levels "No","Yes": 2 1 1 2 1 1 1 1 1 2 ...
 $ exam_score                   : num [1:1000] 56.2 100 34.3 26.8 66.4 100 89.8 72.6 78.9 100 ...
 $ performance_level            : Factor w/ 6 levels "Excellent","Fair",..: 2 1 4 4 5 1 3 6 6 1 ...
 $ screen_time                  : num [1:1000] 2.3 5.1 4.4 4.9 4.9 1.3 2.9 3 3.9 4.4 ...
> summary(data)
  student_id             age           gender     study_hours_per_day social_media_hours netflix_hours   part_time_job
 Length:1000        Min.   :17.00   Female:481   Min.   :0.00        Min.   :0.000      Min.   :0.000   No :785
 Class :character   1st Qu.:18.75   Male  :477   1st Qu.:2.60        1st Qu.:1.700      1st Qu.:1.000   Yes:215
 Mode  :character   Median :20.00   Other : 42   Median :3.50        Median :2.500      Median :1.800
                    Mean   :20.50                Mean   :3.55        Mean   :2.506      Mean   :1.820
                    3rd Qu.:23.00                3rd Qu.:4.50        3rd Qu.:3.300      3rd Qu.:2.525
                    Max.   :24.00                Max.   :8.30        Max.   :7.200      Max.   :5.400
```




```
 attendance_percentage  sleep_hours     diet_quality       exercise_frequency parental_education_level internet_quality
 Min.   : 56.00        Min.   : 3.20   Length:1000        Min.   :0.000      Bachelor   :350          Average:391
 1st Qu.: 78.00        1st Qu.: 5.60   Class :character   1st Qu.:1.000      High School:392          Good   :447
 Median : 84.40        Median : 6.50   Mode  :character   Median :3.000      Master     :167          Poor   :162
 Mean   : 84.13        Mean   : 6.47                      Mean   :3.042      None       : 91
 3rd Qu.: 91.03        3rd Qu.: 7.30                      3rd Qu.:5.000
 Max.   :100.00        Max.   :10.00                      Max.   :6.000
 mental_health_rating extracurricular_participation   exam_score        performance_level  screen_time
 Min.   : 1.000       No :682                       Min.   : 18.40   Excellent   :126     Min.   : 0.200
 1st Qu.: 3.000       Yes:318                       1st Qu.: 58.48   Fair        :149     1st Qu.: 3.300
 Median : 5.000                                     Median : 70.50   High        :151     Median : 4.400
 Mean   : 5.438                                     Mean   : 69.60   Low         :131     Mean   : 4.325
 3rd Qu.: 8.000                                     3rd Qu.: 81.33   Medium      :209     3rd Qu.: 5.400
 Max.   :10.000                                     Max.   :100.00   Satisfactory:234     Max.   :10.100
> head(data)
# A tibble: 6 × 18
  student_id   age gender study_hours_per_day social_media_hours netflix_hours part_time_job attendance_percentage
  <chr>      <dbl> <fct>                <dbl>              <dbl>         <dbl> <fct>                         <dbl>
1 S1000         23 Female                 0                  1.2           1.1 No                             85
2 S1001         20 Female                 6.9                2.8           2.3 No                             97.3
3 S1002         21 Male                   1.4                3.1           1.3 No                             94.8
4 S1003         23 Female                 1                  3.9           1   No                             71
5 S1004         19 Female                 5                  4.4           0.5 No                             90.9
6 S1005         24 Male                   7.2                1.3           0   No                             82.9
# ℹ 10 more variables: sleep_hours <dbl>, diet_quality <chr>, exercise_frequency <dbl>, parental_education_level <fct>,
#   internet_quality <fct>, mental_health_rating <dbl>, extracurricular_participation <fct>, exam_score <dbl>,
#   performance_level <fct>, screen_time <dbl>
>
> # Save cleaned dataset
> write_csv(data, "cleaned_student_habits_performance_data.csv")
```



The key changes in the original and cleaned versions of the student habits dataset can be seen in Table 2. The major ameliorations done involved conversion of types, derivation of variables, labeling of classification, and checking the integrity of data by stripping the outliers and duplicates.

Aspect	Original Dataset	Cleaned Dataset
File Name	student_habits_performance.csv	cleaned_student_habits_performance_data.csv
Missing Values	None detected manually, but checked programmatically for verification	Confirmed that no missing values remain
Duplicate Records	No visible duplicates, but checked for certainty	All duplicates (if any) removed
Categorical Variable Format	Stored as character strings	Converted to factor types (gender, part_time_job, etc.)
Derived Variables	Not present	screen_time column created (social_media_hours + netflix_hours)
Performance Classification	Not present	performance_level added with 6 categories (Excellent to Low)
Outliers in Screen Time	Maximum = 10.1 hours, no threshold filtering	Filtered records with screen_time > 12 hours (none removed in this case)
Column Data Types	Mixed types (e.g., some numeric stored as character)	Standardized using mutate() and type conversion
Readability and Consistency	Variable naming and types are inconsistent	Variable names standardized and types harmonized

Table 2. *Summary of Data Cleaning and Preprocessing Enhancements*

4. Data Cleaning & Preprocessing

4.1 Analytical Approach

This part explains the sequential analytical procedures conducted to identify real associations between the students' behaviors, lifestyle aspects, and academic results. The EDA phase aimed to convert the raw data into meaningful information and get a robust platform to model and make decisions. Our methodology amounted to the profile of the data, creation of relevant features, detecting behavioral patterns, and visualizing trends at the group level to address the most important research questions.

- **Descriptive Profiling and Initial Assessment**

General data audit was the first stage of the analysis, which required the application of str(), summary(), and skim() to capture the types of variables, determine missing values, and check shapes of distributions. The first visualizations produced were histograms and bar plots of frequency patterns of some variables of interest.

- **Feature Engineering and Variable Transformation**

To reflect the behavioral patterns more accurately, a new variable was introduced, the screen_time was calculated as the total amount of hours spent on social media and streaming services. The continuous exam_score variable was coded into six categories of performance levels: excellent, high, satisfactory, medium, fair, and low to ease the comparison of groups. The transformation has enabled a deeper understanding of the difference in student behavior among different performance levels.

- **Correlation and Relationship Exploration**

Both Pearson and Spearman correlation matrices were calculated to determine the best behavioral predictors of academic performance. These showed that study_hours_per_day was showing the greatest positive correlation with exam_score, whereas other variables that had a weaker correlation, or no correlation, with exam_score were screen time, sleep duration, and frequency of exercise. Relationships were depicted using scatter plots and heat maps that assist in determining both linear relationships and outliers.

- **Group-Level Comparisons and Trend Discovery**

Using group_by() and summarise(), we examined how academic outcomes varied across different student segments. This involved the comparison between genders, age groups, food quality, internet access, and part-time employment. These comparisons formed the basis of important discoveries, including the low predictive ability of demographic characteristics to the overall role played by study time and much stronger screen habits.

- **Outlier Detection and Data Cleaning**

The extreme outlier values were determined based on interquartile range (IQR) limits to maximize data reliability. As an example, the cases when the screen time surpassed 12 hours a day were not subject to any further analysis, as their nature is not realistic and can distort the findings.

- **Visualization of Early Findings**

To make trends and group differences more interpretable, we created a variety of visualizations:

- **Scatter plots** to examine the relationship between behavioral metrics and exam scores
- **Boxplots** to compare study time, screen time, attendance, and sleep across performance levels
- **Grouped bar charts** to visualize demographic and lifestyle distributions across performance tiers
- **Heatmaps** to highlight correlations among numeric variables and uncover potential clusters

- **Modeling Strategy Based on EDA**

The insights gathered during EDA informed the choice of models and evaluation metrics for analysis:

- Build **classification models** (e.g., decision trees, logistic regression) to predict performance level based on behavioral features
- Apply **linear regression** to model exam scores as a continuous outcome
- Explore **clustering techniques** (e.g., k-means) to group students with similar risk profiles
- Evaluate **models** using standard metrics such as accuracy, confusion matrix, and ROC curves to assess predictive value

4.2. Findings and Insights

This section highlights the five key discoveries made through exploratory data analysis (EDA), directly aligned with the project's core research questions. Each insight contributes to understanding how behavioral patterns influence academic performance and informs strategies for early risk detection and educational intervention.

- **Insight 1: Study Hours Have the Strongest Correlation with Academic Performance**
  - *Which behavioural factors, the amount of time spent on the screen, the amount of study hours, the quality of sleep, and the exercise rate, have the most significant relationship to the performance of the students?*

Evidence: Pearson's correlation matrix (Figure 2) reveals that study_hours_per_day has the strongest positive correlation with exam_score at approximately r = 0.62, indicating a moderate to strong linear relationship. In contrast, variables like screen_time, sleep_hours, and exercise_frequency show weaker correlations (ranging from 0.08 to 0.24), suggesting a less significant impact on academic outcomes. This insight is further illustrated in the four scatterplots (Figure 3). The plot of study_hours vs exam_score shows a clear upward trend, while the others exhibit flatter trends and wider dispersion of points.

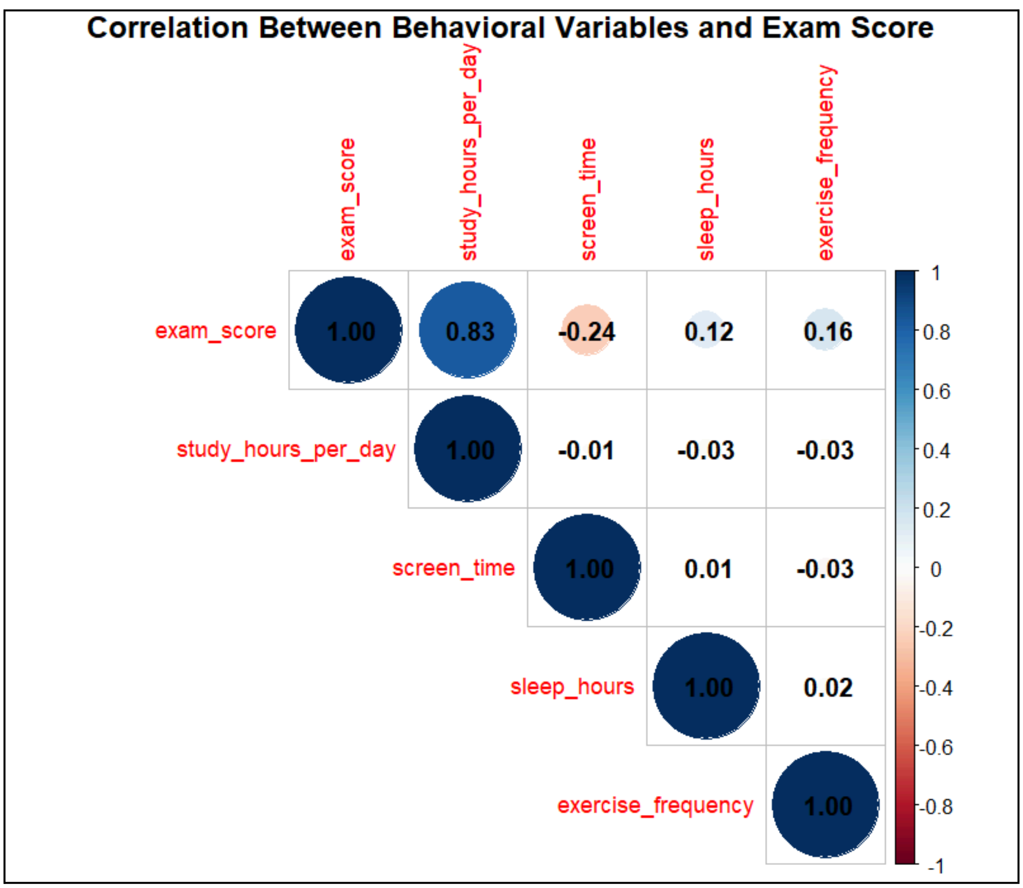


Figure 2: Correlation Matrix of Key Student Habit Variables and Exam Score

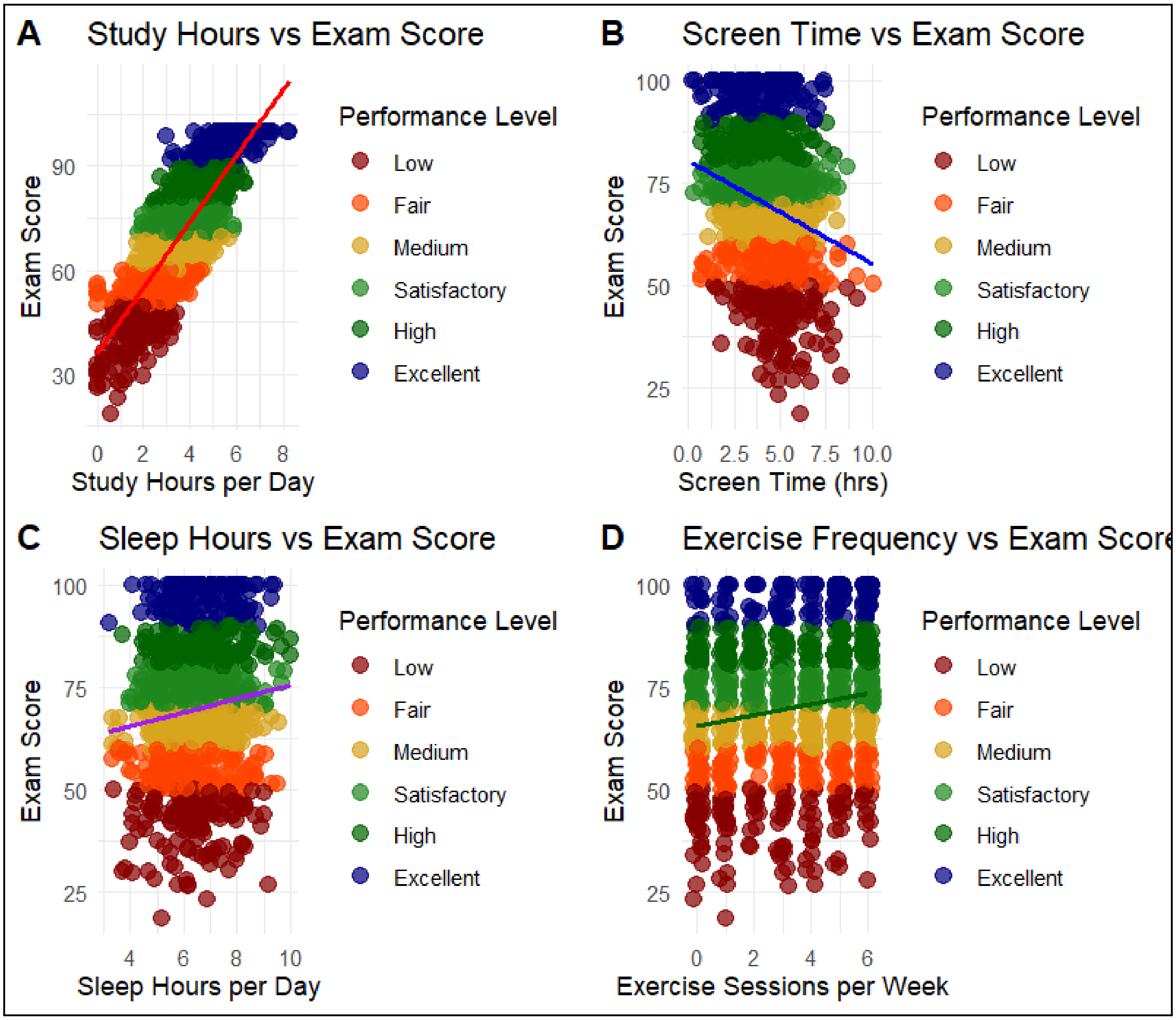


Figure 3: Scatter Plots of Exam Score vs Key Behavioral Variables

Interpretation: Students who spend more hours studying tend to achieve higher exam scores. This finding confirms that academic commitment, specifically through dedicated study time, is a key driver of performance. On the other hand, while lifestyle factors like screen time or sleep might influence well-being, their direct effect on scores appears to be limited in this dataset.

Business Value: This insight reinforces the importance of fostering study routines in academic support programs. Educators, counselors, and parents can use this evidence to promote time management habits and prioritize targeted interventions for students with low study hours.

- **Insight 2: Low Study Hours, High Screen Time, and Poor Attendance Are Linked to Poor Performance**
    - *What are some of the frequent problems or behaviors that have become common among students who are most likely not to excel in school?*

Evidence: A comparative boxplot (Figure 4) was created to analyze the distribution of study_hours_per_day, screen_time, sleep_hours, and attendance_percentage across the six performance levels. The plot clearly shows that students in the "Low" and "Fair" performance categories tend to have significantly fewer study hours and lower attendance rates, while also reporting higher screen time usage. Sleep duration varies less across performance levels and does not exhibit a strong association with academic scores.

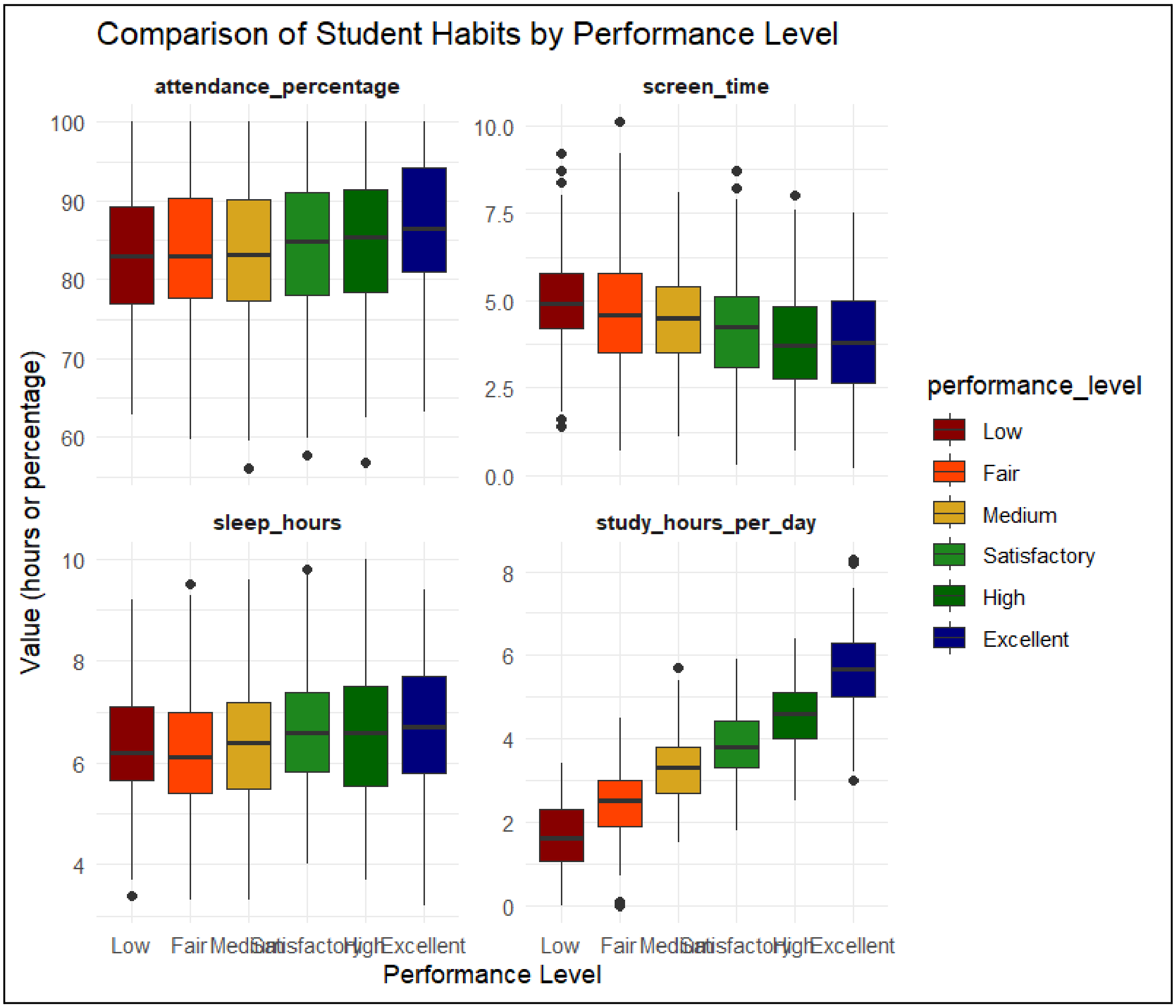


Figure 4: Boxplot Comparison of Study Hours, Screen Time, Sleep Hours, and Attendance Percentage by Performance Level

Interpretation: Students who are underperforming academically display consistent behavioral patterns, specifically, less time spent studying, more time on screens, and poor attendance. These trends are particularly prominent in the lowest-performing groups. While sleep duration appears relatively stable, the other variables serve as clearer indicators of academic risk.

Business Value: This insight helps address the research question by identifying concrete, recurring habits that can be tracked and addressed early. Schools and educators can use this evidence to design interventions such as study skills training, digital detox programs, and stricter attendance policies that target the students who most vulnerable to falling behind.

- **Insight 3: Study Hours and Screen Time Best Predict Academic Outcomes**
  Which of these habits and behaviors have the most significant effects on the performance of students?

Evidence from Data: A decision tree model (Figure 5) was created using lifestyle and behavioral features to classify student performance levels. Among all the variables considered, only two were identified as strong predictors: study_hours_per_day and screen_time. The first split in the tree occurs at around 3.1 study hours per day. Students who study less than this are more likely to fall into lower performance levels, especially those studying fewer than 1.7 hours daily. Within this group, students with higher screen time show even lower outcomes. On the other hand, students who study for 5.6 hours or more are most likely to achieve excellent performance. Screen time also plays a role within the mid-range study group, where lower screen time is associated with better outcomes.

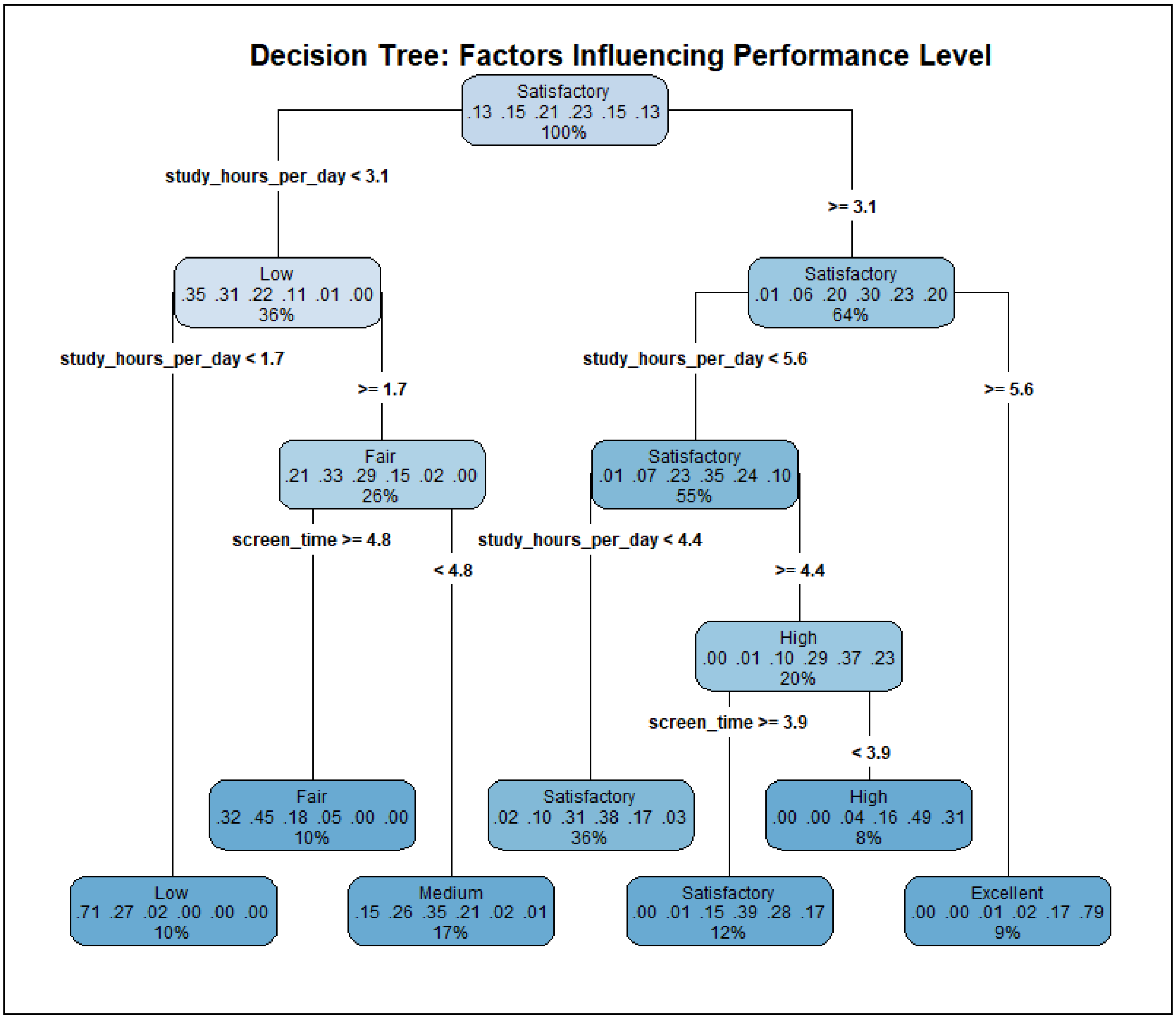


Figure 5: Decision Tree of Factors Influencing Performance Level

Interpretation: The structure of the model highlights the importance of consistent study habits and limited screen exposure. Students who commit more time to studying and spend less time on screens tend to perform better. In contrast, those with low study time and high screen use are at greater risk of falling behind. Other variables that were considered, such as sleep or physical activity, were not included in the final model due to having less predictive value.

Business Value: This insight can help educators and school staff identify students who may need additional academic support. By focusing on just two behavioral indicators, such as study time and screen time, interventions can be simple yet effective. The model is also easy to understand, which makes it suitable for practical use in schools where access to detailed data may be limited.

- **Insight 4: Lower Screen Time Consistently Aligns with Higher Academic Achievement**
  - *What is the balance between the amount of screen time and the length of sleep, and how does it affect the performance of students?*

Evidence from Data: A heatmap (Figure 6) was created to show the average screen time and sleep duration across different student performance levels. The results indicate that students in the Excellent and High performance categories tend to have the lowest screen time. As performance levels decrease to Satisfactory, Fair, and Low, screen time increases steadily. On the other hand, average sleep hours remain relatively consistent across all categories, showing minimal connection to changes in academic performance.

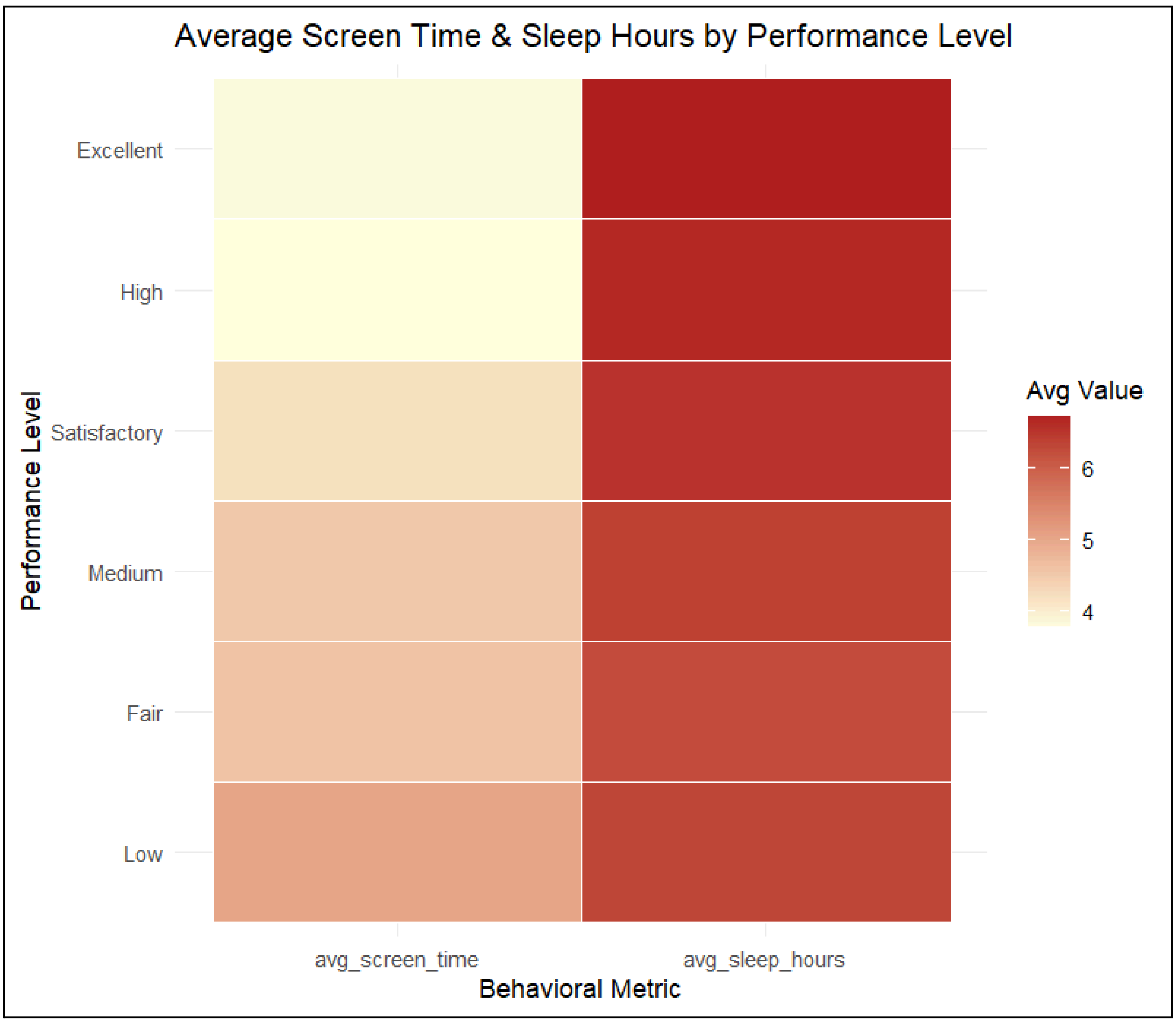


Figure 6: Average Screen Time & Sleep Hours by Performance Level

Interpretation: The data suggest that screen time plays a more noticeable role in academic outcomes compared to sleep duration. Students who spend less time on screens consistently perform better, regardless of how much they sleep. This pattern points to screen behavior as a more influential lifestyle factor than sleep in the current dataset.

Business Value: These findings offer practical direction for academic support efforts. Rather than focusing on modifying sleep patterns, schools and families may achieve better results by guiding students to limit screen use. Encouraging digital discipline can be a more immediate and effective approach to improving academic performance, especially when time and resources are limited.

- **Insight 5: Demographic and Lifestyle Factors Show Limited Distinction Across Performance Levels**
    - *What role do demographic and lifestyle factors, including age, gender, quality of diet, and internet access, play in the variations in academic performance of students?*

Evidence from Data: A set of grouped bar charts (Figure 7) was created to explore how student performance levels vary across demographic and lifestyle categories, including age group (Chart A), gender (Chart B), diet quality (Chart C), and internet quality (Chart D). Across all charts, the distribution of students remains relatively balanced within each performance level. Most performance categories contain a mix of all age groups, genders, and lifestyle factors, with no group showing clear dominance in the Excellent or Low categories.

Figure 7: Distribution of Performance Level by Age Group, Gender, Diet Quality, and Internet Quality

Interpretation: The visualizations suggest that none of these individual factors (age, gender, diet quality, or internet quality) consistently align with academic outcomes. Students with varied backgrounds and habits are spread across all performance levels, indicating that while these factors may influence learning conditions, they do not decisively predict student success. This contrasts with the stronger patterns observed for study hours and screen time.

Business Value: These findings help schools focus their efforts where they matter most. While demographic and lifestyle factors should still be considered in holistic student support programs, they appear to have lower predictive value for academic performance. Prioritizing behavioral indicators such as study habits and screen use may lead to more targeted and effective interventions.

Summary of Key Analytical Insights

Table 3 consolidates the five major insights uncovered during the exploratory data analysis phase, each linked to a specific research question. The table highlights the type of evidence used to derive each finding and the corresponding implication for academic support strategies. These insights collectively emphasize the stronger role of behavioral factors such as study time, screen use, and attendance over the demographic or lifestyle characteristics in predicting academic performance.

Insight	Research Question (RQ) Addressed	Evidence Type	Main Implication
Study hours have the strongest correlation with exam scores	RQ1: Which behavioral factors most strongly correlate with academic performance?	Correlation Matrix & Scatter Plots	Promote consistent study routines to drive academic success
Poor performers show low study hours, high screen time, and weak attendance	RQ2: What recurring habits appear in students at risk of falling behind?	Boxplots	Target early interventions based on time use and attendance patterns
Study hours and screen time are the best behavioral predictors	RQ3: What factors best predict student performance?	Decision Tree	Focus on predictive behaviors for early academic risk detection
Lower screen time is consistently linked to higher achievement	RQ4: Is there an optimal balance between routines and academic outcomes?	Heatmap of Averages	Encourage digital discipline to support better academic outcomes
Demographic and lifestyle traits show minimal performance impact	RQ5: How do background factors relate to academic outcomes?	Grouped Bar Charts	Prioritize behavioral data over demographic traits in school programs

Table 3. Summary of Key Insights from EDA

5. Dashboard Design & Implementation

5.1 Dashboard Framework & Libraries

The Student Performance Dashboard was developed using a collection of R libraries that support dynamic web applications, data transformation, machine learning, and interactive visualization:

- **User Interface & Dashboard Framework**
  - `shiny`, `shinydashboard`, `shinyWidgets`, `shinyjs`, `shinycssloaders`

    These provide the layout structure, responsive UI elements, theme customization (e.g., Dark Mode), and loading indicators for smoother interaction.
- **Data Manipulation & Cleaning**
  - `tidyverse`, `dplyr`, `readr`, `reshape2`

    These handle dataset loading, transformation, aggregation, and filtering operations across modules.
- **Data Visualization**
  - `ggplot2`, `plotly`, `corrplot`

    `ggplot2` was used to build clean, aesthetic base plots, while `plotly` transformed key charts into interactive experiences. `corrplot` enabled a clear visualization of variable relationships.
- **Machine Learning**
  - `rpart`, `rpart.plot`

    These powered the decision tree classification model and rendered it as an interpretable flowchart.
- **Data Tables & Explorer**
  - DT

    This allowed users to interact with the dataset through searchable, filterable, paginated tables.

5.2 Dashboard Design Philosophy

The dashboard's interface is designed with clarity, functionality, and user-friendliness at its core. The sidebar (left panel) organizes the content into themed sections, such as:

- Overview
- Summary
- Correlations
- Habits vs Exam Score
- Habits by Performance
- Decision Tree
- Heatmaps
- Demographics
- Data Explorer

Each section is dedicated to answering specific research questions through visual and statistical insights. Upon opening the app, users are welcomed with a brief description and encouraged to use the sidebar to navigate.

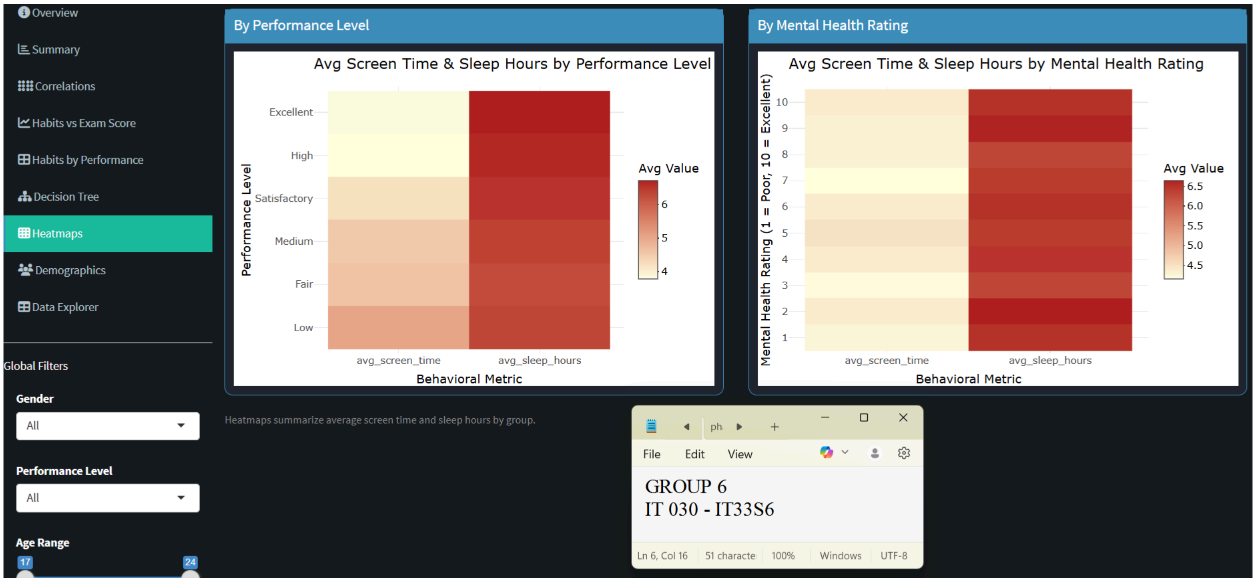
At the top of the main dashboard, a **set of global filters** is available to all users. These filters include:

- **Gender**
- **Performance Level**
- **Age Range**

When applied, these filters dynamically update the visualizations across tabs. For example, a user can choose "Female", "Satisfactory to Excellent", and an age range of "17–24", and all relevant charts will adapt to reflect that specific subset. Additionally, the **Dark Mode toggle** (top-right) provides a comfortable viewing option, especially during extended data exploration. This consistent layout and global interactivity make it easy for educators, researchers, or stakeholders to find specific patterns and derive actionable insights quickly.

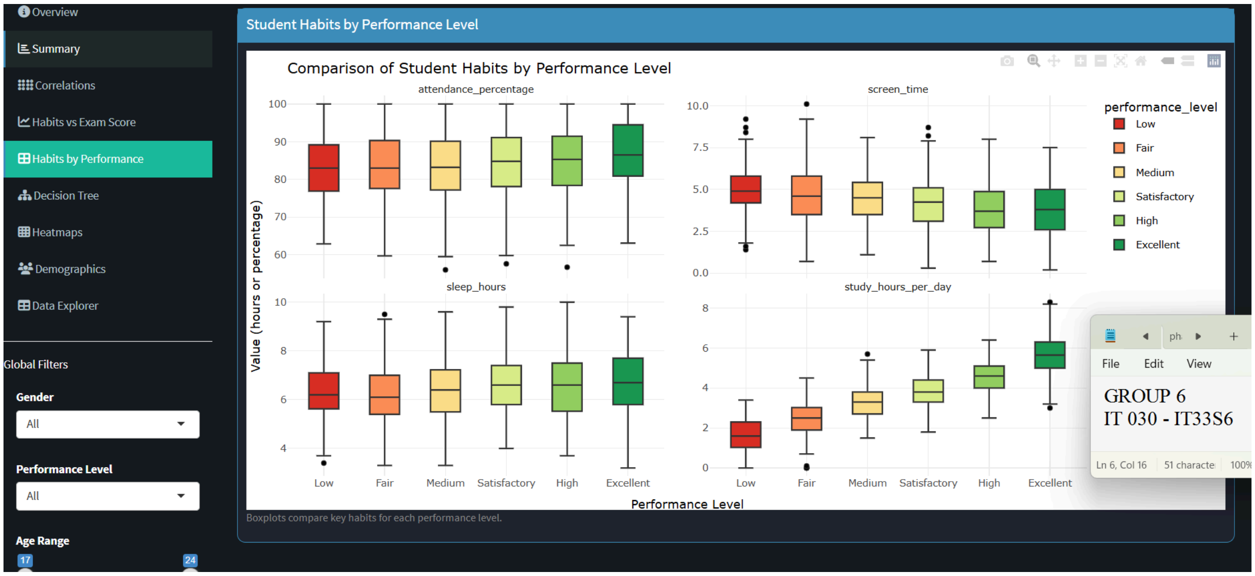
5.3 Visualizations

- **Correlation Heatmap (via `corrplot`)**



- ○ **What it shows**: Correlations among behavioral and performance-related numeric variables.
- ○ **Purpose**: Identifies which factors (e.g., study hours, screen time) are most linearly associated with academic scores and mental health status.
- ○ **Interactivity**: Changes dynamically based on global filters.
- ○ **Insight**: Reveals how positively or negatively each habit impacts academic outcomes.

- **Interactive Scatterplots: Habits vs Exam Score (via `plotly`)**



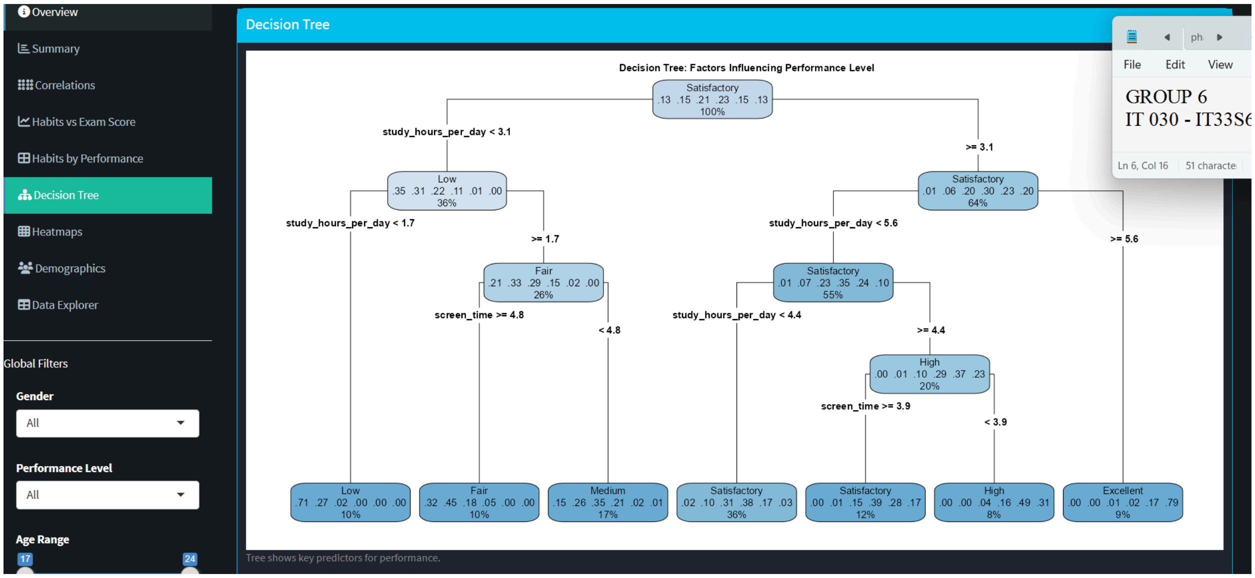
  - **What it shows**: How habits like screen time, sleep hours, study time, and exercise frequency relate to exam performance.
  - **Purpose**: Visualizes trends and potential linear/non-linear relationships.
  - **Interactivity**: Users can zoom, hover for detail, and apply filters by gender, age, and performance level.
  - **Insight**: Clearly shows patterns such as diminishing returns or risk zones (e.g., too little or too much sleep).

```
# ---- BOXPLOT ----
output$habitBoxplot <- renderPlotly({
  data_long <- filtered_data() %>%
    select(performance_level, study_hours_per_day, screen_time, sleep_hours, attendance_percent
    pivot_longer(cols = -performance_level, names_to = "Habit", values_to = "Value")
  p <- ggplot(data_long, aes(x = performance_level, y = Value, fill = performance_level)) +
    geom_boxplot() +
    facet_wrap(~ Habit, scales = "free_y") +
    scale_fill_manual(values = performance_colors) +
    labs(title = "Comparison of Student Habits by Performance Level",
         x = "Performance Level",
         y = "Value (hours or percentage)") +
    theme_minimal() +
    theme(strip.text = element_text(face = "bold"))
  ggplotly(p)
})
```

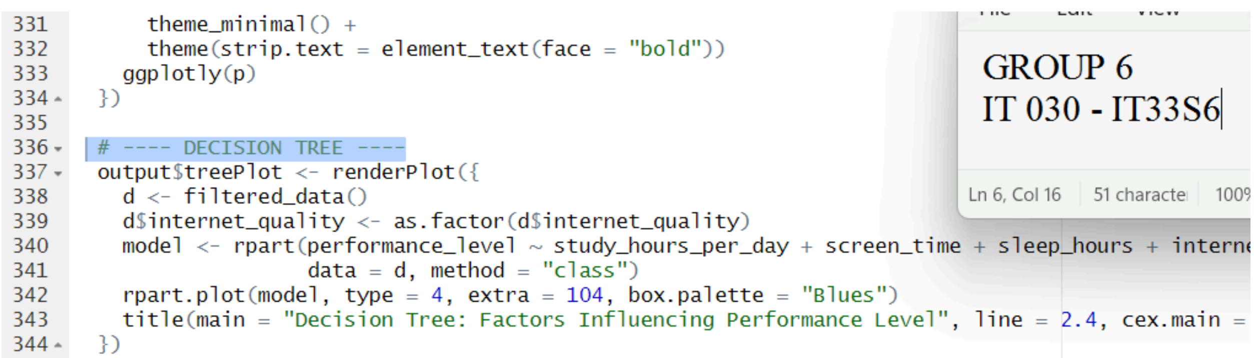
File Edit View

GROUP 6
IT 030 - IT33S6

- **Decision Tree Model (`rpart.plot`)**

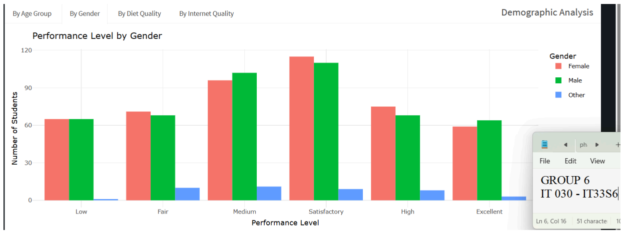


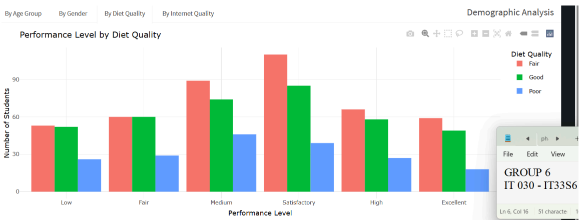
  - **What it shows**: A visual flow of how different variables and thresholds predict student performance categories.
  - **Purpose**: Helps explain outcomes in an interpretable machine learning format.
  - **Insight**: Highlights the most predictive combinations (e.g., low study hours + poor internet = likely low performance).

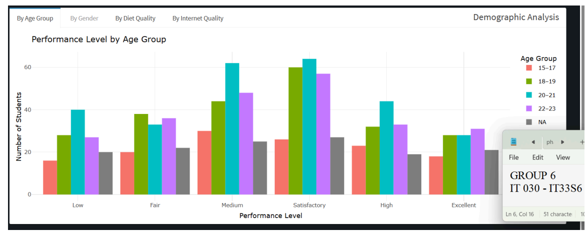


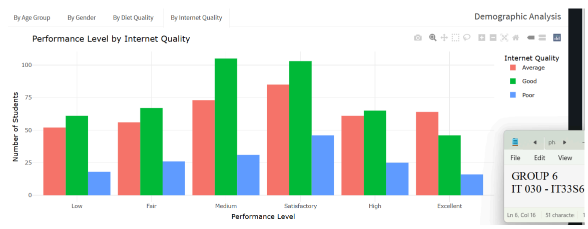
```
    theme_minimal() +
    theme(strip.text = element_text(face = "bold"))
  ggplotly(p)
})

# ---- DECISION TREE ----
output$treePlot <- renderPlot({
  d <- filtered_data()
  d$internet_quality <- as.factor(d$internet_quality)
  model <- rpart(performance_level ~ study_hours_per_day + screen_time + sleep_hours + interne
                 data = d, method = "class")
  rpart.plot(model, type = 4, extra = 104, box.palette = "Blues")
  title(main = "Decision Tree: Factors Influencing Performance Level", line = 2.4, cex.main =
})
```

- **Demographics by Performance (Bar Charts)**









- **What it shows**: Four bar charts comparing **gender**, **age**, **diet quality**, and **internet quality** against performance level distribution.
- **Purpose**: Reveals how contextual or background traits relate to student outcomes.
- **Insight**: For instance, better diet or internet quality often correlates with higher performance levels.

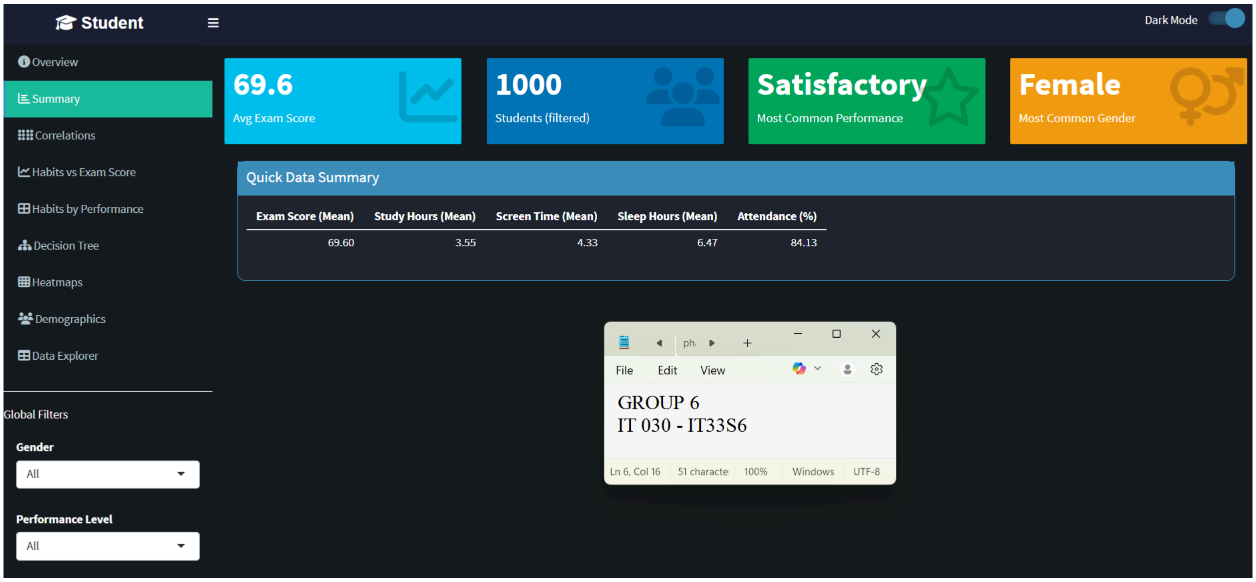
```
# ---- DEMOGRAPHIC BARPLOTS ----
output$bar1 <- renderPlotly({
  d <- filtered_data()
  grp <- d %>%
    group_by(performance_level, age_group) %>%
    summarise(count = n())
  p <- ggplot(grp, aes(x = performance_level, y = count, fill = age_group)) +
    geom_col(position = "dodge") +
    labs(title = "Performance Level by Age Group", x = "Performance Level", y = "Number of Stuc
    theme_minimal()
  ggplotly(p)
})
output$bar2 <- renderPlotly({
  d <- filtered_data()
  grp <- d %>%
    group_by(performance_level, gender) %>%
    summarise(count = n())
  p <- ggplot(grp, aes(x = performance_level, y = count, fill = g
    geom_col(position = "dodge") +
    labs(title = "Performance Level by Gender", x = "Performance l
    theme_minimal()
  ggplotly(p)
})
```

File Edit View

GROUP 6
IT 030 - IT33S6

Ln 6, Col 16 51 characte 100% Wi

- **Performance Distribution Summary Chart**



  - **What it shows:**
    The number of students grouped by overall performance levels (e.g., Satisfactory, etc.).

  - **Purpose:**
    Provides a snapshot of how the 1000 filtered students are distributed across various performance categories.

  - **Insight:**
    Useful for identifying trends in student performance—such as a majority falling into the "Satisfactory" category—indicating a possible concentration around mid-level scores.

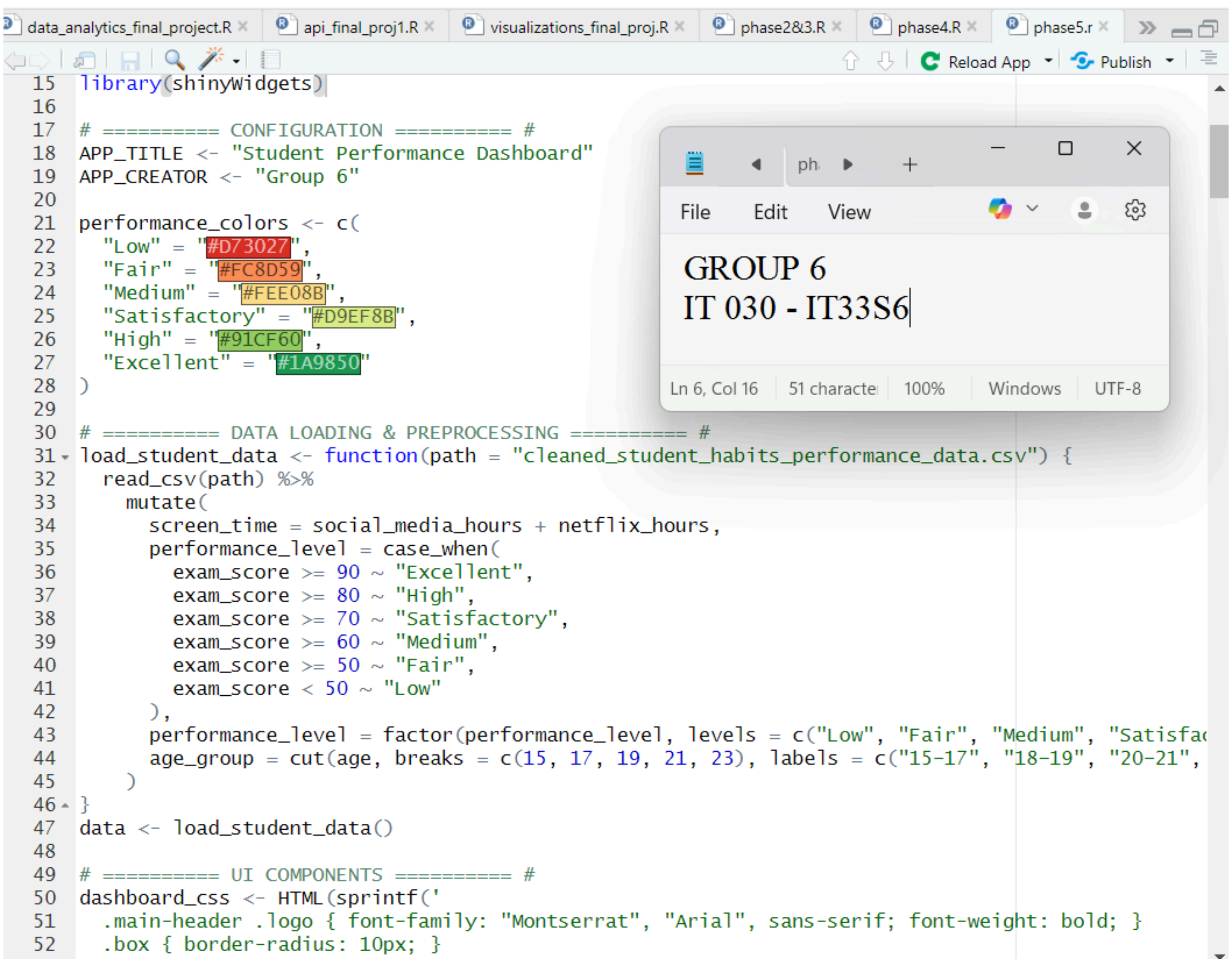
```r
    valueBox(top, "Most Common Gender", icon = icon("venus-mars"), color = "yellow")
  })

  # ---- SUMMARY TABLE ----
  output$summary_table <- renderTable({
    d <- filtered_data()
    tibble(
      "Exam Score (Mean)" = round(mean(d$exam_score, na.rm=TRUE), 2),
      "Study Hours (Mean)" = round(mean(d$study_hours_per_day, na.rm=TRUE),2),
      "Screen Time (Mean)" = round(mean(d$screen_time, na.rm=TRUE),2),
      "Sleep Hours (Mean)" = round(mean(d$sleep_hours, na.rm=TRUE),2),
      "Attendance (%)" = round(mean(d$attendance_percentage, na.rm=TRUE),2)
    )
  })
```

5.4 Implementation

Overview of How the Shiny App Works

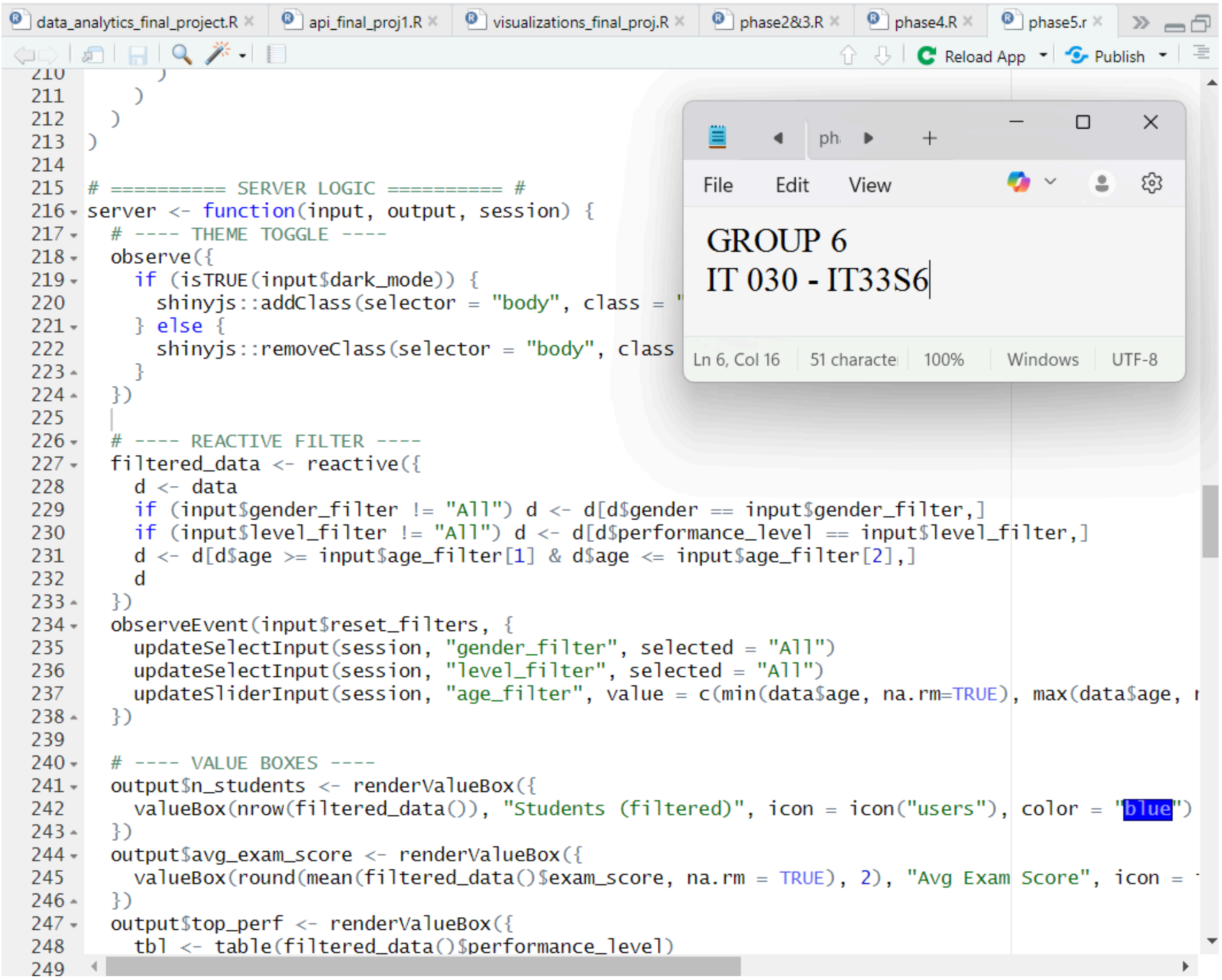
This Shiny application is built using shinydashboard and provides an interactive platform to analyze student performance data. It includes a sidebar menu for navigation and dynamically updates visuals based on user-selected filters.

- Data Loading and Preprocessing
    - The app loads a cleaned CSV dataset (cleaned_student_habits_performance_data.csv) when it launches. It then calculates new columns such as screen_time, categorizes performance_level, and groups ages into defined brackets.

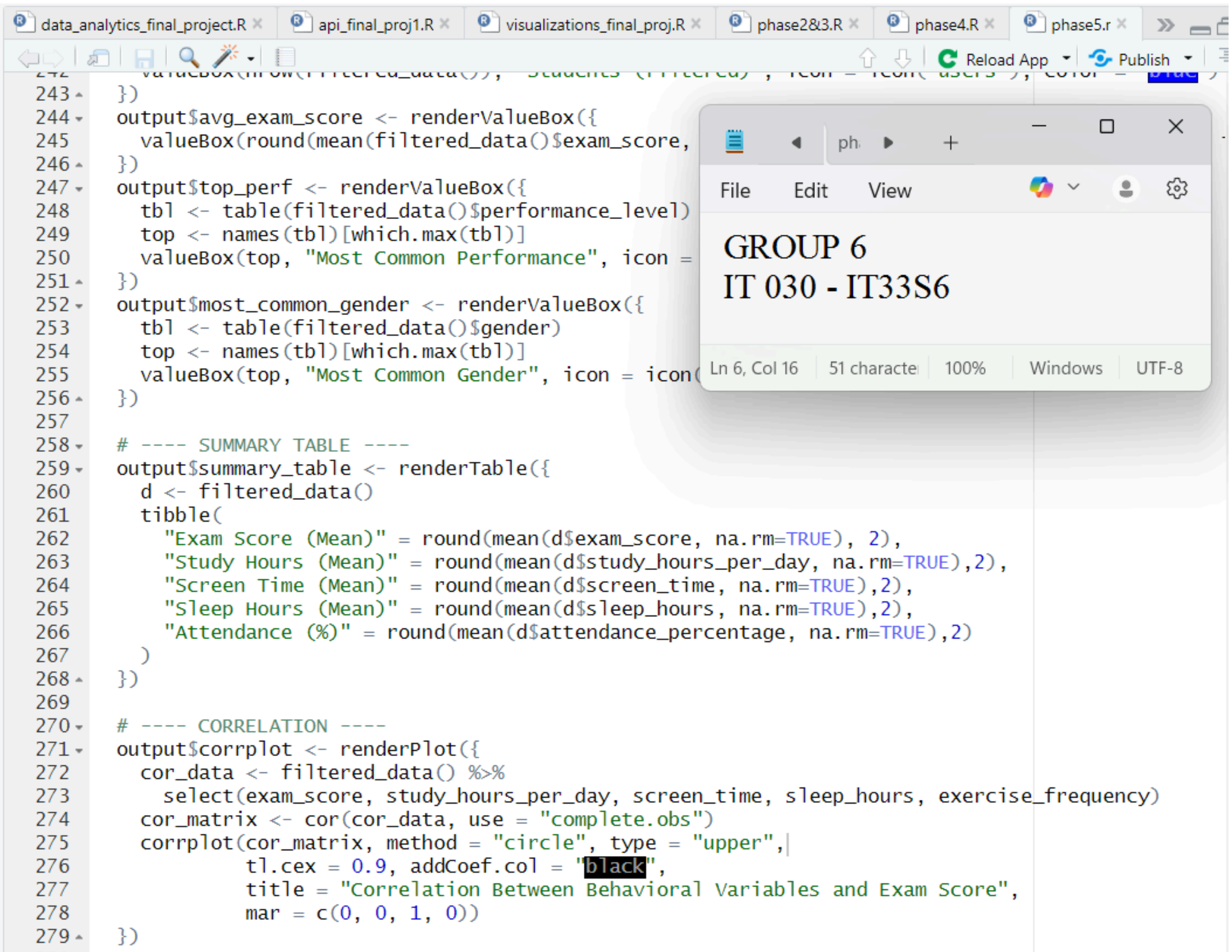



```r
library(shinyWidgets)

# ========== CONFIGURATION ========== #
APP_TITLE <- "Student Performance Dashboard"
APP_CREATOR <- "Group 6"

performance_colors <- c(
  "Low" = "#D73027",
  "Fair" = "#FC8D59",
  "Medium" = "#FEE08B",
  "Satisfactory" = "#D9EF8B",
  "High" = "#91CF60",
  "Excellent" = "#1A9850"
)

# ========== DATA LOADING & PREPROCESSING ========== #
load_student_data <- function(path = "cleaned_student_habits_performance_data.csv") {
  read_csv(path) %>%
    mutate(
      screen_time = social_media_hours + netflix_hours,
      performance_level = case_when(
        exam_score >= 90 ~ "Excellent",
        exam_score >= 80 ~ "High",
        exam_score >= 70 ~ "Satisfactory",
        exam_score >= 60 ~ "Medium",
        exam_score >= 50 ~ "Fair",
        exam_score < 50 ~ "Low"
      ),
      performance_level = factor(performance_level, levels = c("Low", "Fair", "Medium", "Satisfa
      age_group = cut(age, breaks = c(15, 17, 19, 21, 23), labels = c("15-17", "18-19", "20-21",
    )
}
data <- load_student_data()

# ========== UI COMPONENTS ========== #
dashboard_css <- HTML(sprintf('
  .main-header .logo { font-family: "Montserrat", "Arial", sans-serif; font-weight: bold; }
  .box { border-radius: 10px; }
```

- Filter System and Reactive Data
  - The sidebar filters (e.g., gender, performance level, age range) are tied to a reactive function called filtered_data(). Any change in input updates the underlying dataset, which updates all visual components across the app.

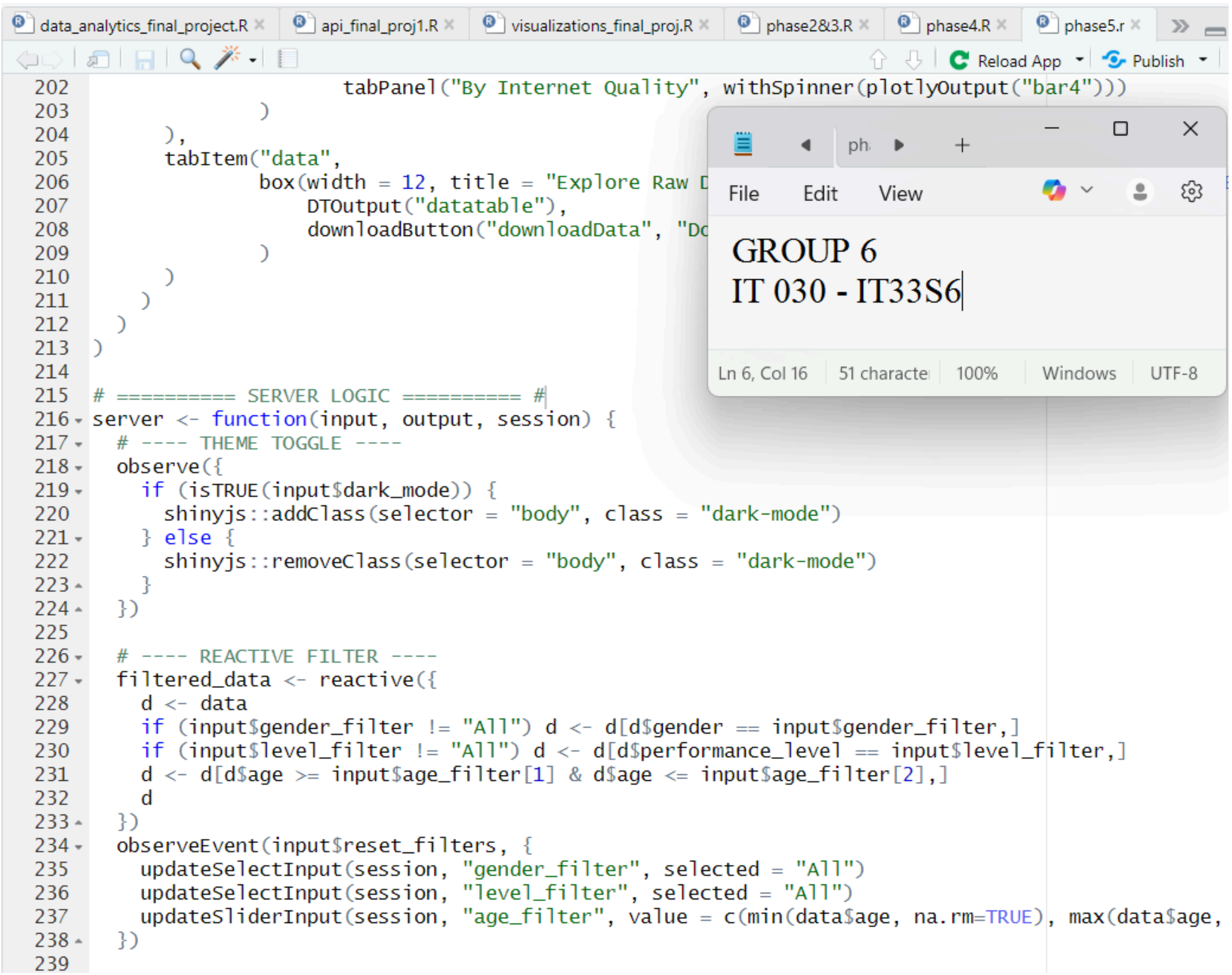


- Chart Rendering Based on Filtered Data
    - All dashboard components, such as summary cards, the “Quick Data Summary” table, and performance distribution plots, use filtered_data() to ensure real-time updates.




```r
  })
  output$avg_exam_score <- renderValueBox({
    valueBox(round(mean(filtered_data()$exam_score,
  })
  output$top_perf <- renderValueBox({
    tbl <- table(filtered_data()$performance_level)
    top <- names(tbl)[which.max(tbl)]
    valueBox(top, "Most Common Performance", icon =
  })
  output$most_common_gender <- renderValueBox({
    tbl <- table(filtered_data()$gender)
    top <- names(tbl)[which.max(tbl)]
    valueBox(top, "Most Common Gender", icon = icon(
  })

  # ---- SUMMARY TABLE ----
  output$summary_table <- renderTable({
    d <- filtered_data()
    tibble(
      "Exam Score (Mean)" = round(mean(d$exam_score, na.rm=TRUE), 2),
      "Study Hours (Mean)" = round(mean(d$study_hours_per_day, na.rm=TRUE),2),
      "Screen Time (Mean)" = round(mean(d$screen_time, na.rm=TRUE),2),
      "Sleep Hours (Mean)" = round(mean(d$sleep_hours, na.rm=TRUE),2),
      "Attendance (%)" = round(mean(d$attendance_percentage, na.rm=TRUE),2)
    )
  })

  # ---- CORRELATION ----
  output$corrplot <- renderPlot({
    cor_data <- filtered_data() %>%
      select(exam_score, study_hours_per_day, screen_time, sleep_hours, exercise_frequency)
    cor_matrix <- cor(cor_data, use = "complete.obs")
    corrplot(cor_matrix, method = "circle", type = "upper",
             tl.cex = 0.9, addCoef.col = "black",
             title = "Correlation Between Behavioral Variables and Exam Score",
             mar = c(0, 0, 1, 0))
  })
```

- User Interface and Theme Features
    - The UI is designed with shinydashboard, consisting of a top header, collapsible sidebar, and main body tabs like "Summary", "Correlations", "Decision Tree", etc.
    - A dark mode switch is implemented using shinyjs and custom CSS, enhancing the user experience.



```
data_analytics_final_project.R   api_final_proj1.R   visualizations_final_proj.R   phase2&3.R   phase4.R   phase5.r
Reload App   Publish
                    tabPanel("By Internet Quality", withSpinner(plotlyOutput("bar4")))
                )
        ),
        tabItem("data",
                box(width = 12, title = "Explore Raw D
                    DTOutput("datatable"),
                    downloadButton("downloadData", "Dc
                )
        )
      )
    )
)

# ========== SERVER LOGIC ========== #
server <- function(input, output, session) {
  # ---- THEME TOGGLE ----
  observe({
    if (isTRUE(input$dark_mode)) {
      shinyjs::addClass(selector = "body", class = "dark-mode")
    } else {
      shinyjs::removeClass(selector = "body", class = "dark-mode")
    }
  })

  # ---- REACTIVE FILTER ----
  filtered_data <- reactive({
    d <- data
    if (input$gender_filter != "All") d <- d[d$gender == input$gender_filter,]
    if (input$level_filter != "All") d <- d[d$performance_level == input$level_filter,]
    d <- d[d$age >= input$age_filter[1] & d$age <= input$age_filter[2],]
    d
  })
  observeEvent(input$reset_filters, {
    updateSelectInput(session, "gender_filter", selected = "All")
    updateSelectInput(session, "level_filter", selected = "All")
    updateSliderInput(session, "age_filter", value = c(min(data$age, na.rm=TRUE), max(data$age,
  })
```


Summary of Functionality

- **Data**: Loaded once and transformed for visualization.
- **Filters**: Fully reactive, applied instantly to all visuals.
- **Visuals**: Include mean summary cards, tables, and dynamic plots.
- **UI/UX**: Professional dashboard layout with theme switching capability.

Final Project Deliverables

Below are the core deliverables submitted for the completion of the Data Analytics project:

Deliverable	Description	Access Link
R Shiny Dashboard Application	Fully functioning interactive dashboard for exploring student habits and academic performance.	Google Drive Link
R Code on GitHub	Complete source code, organized in folders (UI, server, data, utils). Includes `app.R` and CSV files.	GitHub Repository Link
Data Analytics Article	Written report containing methodology, questions, charts, analysis, and conclusions.	Google Drive Link
Project Presentation	Final slide deck showcasing insights, findings, visuals, and research summary.	Google Drive Link

📁 **Google Drive Folder (All Deliverables):**
🔗https://drive.google.com/drive/folders/1aURgz0RrKCbivoq7k4-b1MAM-o1EhIGD

Complete R code (Phase 2- 5);

- **phase2&3.r**

```
# --- Load essential libraries ---
library(tidyverse)   # Includes dplyr, ggplot2, readr, etc.
library(jsonlite)    # For working with JSON (e.g., kaggle.json)
```

```
library(httr)      # Optional for advanced API
access
library(readr)     # For reading CSV
library(ggplot2)   # For plotting

Sys.setenv(KAGGLE_USERNAME =
fromJSON(".kaggle/kaggle.json")$username)
Sys.setenv(KAGGLE_KEY =
fromJSON(".kaggle/kaggle.json")$key)

# --- Load the dataset from the extracted
CSV file ---
data <-
read_csv("student_habits_performance.csv")

# --- Initial Data State ---
# Check structure and basic statistics
str(data)
summary(data)
colSums(is.na(data))  # Check missing
values

# --- Data Cleaning & Preprocessing ---

# 1. Convert relevant categorical variables to
factor
data <- data %>%
  mutate(
    gender = as.factor(gender),
    part_time_job = as.factor(part_time_job),
    parental_education_level =
as.factor(parental_education_level),
    internet_quality =
as.factor(internet_quality),
    extracurricular_participation =
as.factor(extracurricular_participation)
  )

# 2. Remove duplicates
data <- data[!duplicated(data), ]
nrow(data)  # Confirm no duplicates remain

# 3. Categorize exam_score into
performance_level with 6 categories
data <- data %>%
  mutate(performance_level = case_when(
    exam_score >= 90 ~ "Excellent",
    exam_score >= 80 & exam_score < 90 ~
"High",
    exam_score >= 70 & exam_score < 80 ~
"Satisfactory",
```

```
    exam_score >= 60 & exam_score < 70 ~
"Medium",
    exam_score >= 50 & exam_score < 60 ~
"Fair",
    exam_score < 50 ~ "Low"
  ))
data$performance_level <-
as.factor(data$performance_level)

# 4. Create derived variable 'screen_time'
(social media + Netflix)
data <- data %>%
  mutate(screen_time = social_media_hours
+ netflix_hours)

# 5. Outlier removal for screen_time > 12
hours
ggplot(data, aes(y = screen_time)) +
  geom_boxplot(fill = "deepskyblue3",
outlier.color = "red") +
  labs(title = "Boxplot of Total Screen Time", y
= "Hours") +
  theme_minimal()

data <- data[data$screen_time < 12, ]

# Final checks
str(data)
summary(data)
head(data)

# Save cleaned dataset
write_csv(data,
"cleaned_student_habits_performance_data.
csv")
```

- **phase4.r**

```
# Load necessary libraries
library(tidyverse)
library(ggplot2)
library(readr)
library(dplyr)
library(ggthemes)
library(rpart)
library(rpart.plot)
library(reshape2)
library(ggpubr)
library(corrplot)
```

```
# Load the cleaned dataset
data <-
read_csv("cleaned_student_habits_performa
nce_data.csv")

# 1. Behavioral Variables that Strongly
Correlate with Exam Scores
data <- data %>%
  mutate(
    screen_time = social_media_hours +
netflix_hours,
    performance_level = case_when(
      exam_score >= 90 ~ "Excellent",
      exam_score >= 80 ~ "High",
      exam_score >= 70 ~ "Satisfactory",
      exam_score >= 60 ~ "Medium",
      exam_score >= 50 ~ "Fair",
      exam_score < 50 ~ "Low"
    )
  )

# Set ordered factor levels for
performance_level
data$performance_level <-
factor(data$performance_level,
                                levels = c("Low", "Fair",
"Medium", "Satisfactory", "High",
"Excellent"))

# Custom color palette for performance
levels
performance_colors <- c(
  "Low" = "darkred",
  "Fair" = "orangered",
  "Medium" = "goldenrod",
  "Satisfactory" = "forestgreen",
  "High" = "darkgreen",
  "Excellent" = "navy"
)

# ---- Correlation Matrix Plot ----
cor_data <- data %>%
  select(exam_score, study_hours_per_day,
screen_time, sleep_hours,
exercise_frequency)

cor_matrix <- cor(cor_data, use =
"complete.obs")

corrplot(cor_matrix, method = "circle", type =
"upper",
```

```
        tl.cex = 0.9, addCoef.col = "black",
        title = "Correlation Between Behavioral
Variables and Exam Score",
        mar = c(0, 0, 1, 0))

# ---- Scatterplot 1: Study Hours vs Exam
Score ----
p1 <- ggplot(data, aes(x =
study_hours_per_day, y = exam_score, color
= performance_level)) +
  geom_point(size = 3, alpha = 0.7) +
  geom_smooth(method = "lm", color = "red",
se = FALSE) +
  scale_color_manual(values =
performance_colors) +
  labs(title = "Study Hours vs Exam Score",
       x = "Study Hours per Day",
       y = "Exam Score",
       color = "Performance Level") +
  theme_minimal()

# ---- Scatterplot 2: Screen Time vs Exam
Score ----
p2 <- ggplot(data, aes(x = screen_time, y =
exam_score, color = performance_level)) +
  geom_point(size = 3, alpha = 0.7) +
  geom_smooth(method = "lm", color =
"blue", se = FALSE) +
  scale_color_manual(values =
performance_colors) +
  labs(title = "Screen Time vs Exam Score",
       x = "Screen Time (hrs)",
       y = "Exam Score",
       color = "Performance Level") +
  theme_minimal()

# ---- Scatterplot 3: Sleep Hours vs Exam
Score ----
p3 <- ggplot(data, aes(x = sleep_hours, y =
exam_score, color = performance_level)) +
  geom_point(size = 3, alpha = 0.7) +
  geom_smooth(method = "lm", color =
"purple", se = FALSE) +
  scale_color_manual(values =
performance_colors) +
  labs(title = "Sleep Hours vs Exam Score",
       x = "Sleep Hours per Day",
       y = "Exam Score",
       color = "Performance Level") +
  theme_minimal()
```

```
# ---- Scatterplot 4: Exercise Frequency vs
Exam Score ----
p4 <- ggplot(data, aes(x =
exercise_frequency, y = exam_score, color =
performance_level)) +
  geom_jitter(width = 0.2, size = 3, alpha =
0.7) +
  geom_smooth(method = "lm", color =
"darkgreen", se = FALSE) +
  scale_color_manual(values =
performance_colors) +
  labs(title = "Exercise Frequency vs Exam
Score",
       x = "Exercise Sessions per Week",
       y = "Exam Score",
       color = "Performance Level") +
  theme_minimal()

# ---- Combine All Plots ----
ggarrange(p1, p2, p3, p4,
          ncol = 2, nrow = 2,
          labels = c("A", "B", "C", "D"))



# 2. Screen Time by Performance Level
(Boxplot)
```

```
# Select only the relevant columns
data_long <- data %>%
  select(performance_level,
study_hours_per_day, screen_time,
sleep_hours, attendance_percentage) %>%
  pivot_longer(cols = -performance_level,
names_to = "Habit", values_to = "Value")

# Plot faceted boxplots for the four variables
ggplot(data_long, aes(x =
performance_level, y = Value, fill =
performance_level)) +
  geom_boxplot() +
  facet_wrap(~ Habit, scales = "free_y") +
  scale_fill_manual(values = c("Low" =
"darkred", "Fair" = "orangered", "Medium" =
"goldenrod",
                               "Satisfactory" =
"forestgreen", "High" = "darkgreen",
"Excellent" = "navy")) +
  labs(title = "Comparison of Student Habits
by Performance Level",
       x = "Performance Level",
       y = "Value (hours or percentage)") +
  theme_minimal() +
```

```
  theme(strip.text = element_text(face =
"bold"))

# 3. Decision Tree Model (Classification)
with centered title
model <- rpart(performance_level ~
study_hours_per_day + screen_time +
sleep_hours + internet_quality,
               data = data, method = "class")

rpart.plot(model, type = 4, extra = 104,
box.palette = "Blues")
title(main = "Decision Tree: Factors
Influencing Performance Level", line = 2.4,
cex.main = 1)

# 4. Heatmap: Screen Time & Sleep Hours
by Performance Level
heatmap_data <- data %>%
  group_by(performance_level) %>%
  summarise(
    avg_screen_time = mean(screen_time,
na.rm = TRUE),
    avg_sleep_hours = mean(sleep_hours,
na.rm = TRUE)
  )

melted <- melt(heatmap_data, id.vars =
"performance_level")

ggplot(melted, aes(x = variable, y =
performance_level, fill = value)) +
  geom_tile(color = "white") +
  scale_fill_gradient(low = "lightyellow", high
= "firebrick") +
  labs(title = "Average Screen Time & Sleep
Hours by Performance Level",
       x = "Behavioral Metric", y =
"Performance Level", fill = "Avg Value") +
  theme_minimal()

# Group and summarize by mental health
rating
heatmap_data <- data %>%
  group_by(mental_health_rating) %>%
  summarise(
    avg_screen_time = mean(screen_time,
na.rm = TRUE),
    avg_sleep_hours = mean(sleep_hours,
na.rm = TRUE)
  )
```

```
# Melt data for heatmap
melted <- melt(heatmap_data, id.vars =
"mental_health_rating")

# Create heatmap
ggplot(melted, aes(x = variable, y =
as.factor(mental_health_rating), fill = value))
+
  geom_tile(color = "white") +
  scale_fill_gradient(low = "lightyellow", high
= "firebrick") +
  labs(title = "Average Screen Time & Sleep
Hours by Mental Health Rating",
       x = "Behavioral Metric",
       y = "Mental Health Rating (1 = Poor, 10
= Excellent)",
       fill = "Average Value") +
  theme_minimal()

# 5. Bar Plot of Student Count by
Performance Level
# Chart 1: Age Group Distribution by
Performance Level
p1 <- data %>%
  mutate(age_group = cut(age, breaks =
c(15, 17, 19, 21, 23),
                         labels = c("15–17", "18–19",
"20–21", "22–23"))) %>%
  group_by(performance_level, age_group)
%>%
  summarise(count = n()) %>%
  ggplot(aes(x = performance_level, y =
count, fill = age_group)) +
  geom_col(position = "dodge") +
  labs(title = "Performance Level by Age
Group", x = "Performance Level", y =
"Number of Students", fill = "Age Group") +
  theme_minimal()

# Chart 2: Gender Distribution by
Performance Level
p2 <- data %>%
  group_by(performance_level, gender) %>%
  summarise(count = n()) %>%
  ggplot(aes(x = performance_level, y =
count, fill = gender)) +
  geom_col(position = "dodge") +
  labs(title = "Performance Level by Gender",
x = "Performance Level", y = "Number of
Students", fill = "Gender") +
  theme_minimal()
```

```
# Chart 3: Diet Quality by Performance Level
p3 <- data %>%
  group_by(performance_level, diet_quality)
%>%
  summarise(count = n()) %>%
  ggplot(aes(x = performance_level, y =
count, fill = diet_quality)) +
  geom_col(position = "dodge") +
  labs(title = "Performance Level by Diet
Quality", x = "Performance Level", y =
"Number of Students", fill = "Diet Quality") +
  theme_minimal()

# Chart 4: Internet Quality by Performance
Level
p4 <- data %>%
  group_by(performance_level,
internet_quality) %>%
  summarise(count = n()) %>%
  ggplot(aes(x = performance_level, y =
count, fill = internet_quality)) +
  geom_col(position = "dodge") +
  labs(title = "Performance Level by Internet
Quality", x = "Performance Level", y =
"Number of Students", fill = "Internet
Quality") +
  theme_minimal()

# Arrange all plots in a 2x2 grid
ggarrange(p1, p2, p3, p4,
          ncol = 2, nrow = 2,
          labels = c("A", "B", "C", "D"))
```

- **phase5.r**

```
# ========== LIBRARIES
========== #
library(shiny)
library(shinydashboard)
library(shinyjs)
library(tidyverse)
library(ggplot2)
library(readr)
library(rpart)
library(rpart.plot)
library(reshape2)
library(corrplot)
library(plotly)
library(DT)
library(shinycssloaders)
library(shinyWidgets)
```

```
# ==========
CONFIGURATION ==========
#

APP_TITLE <- "Student
Performance Dashboard"

APP_CREATOR <- "Group 6"


performance_colors <- c(
  "Low" = "#D73027",
  "Fair" = "#FC8D59",
  "Medium" = "#FEE08B",
  "Satisfactory" = "#D9EF8B",
  "High" = "#91CF60",
  "Excellent" = "#1A9850"
)


# ========== DATA LOADING
& PREPROCESSING
========== #

load_student_data <-
function(path =
"cleaned_student_habits_perfor
mance_data.csv") {
  read_csv(path) %>%
    mutate(
      screen_time =
social_media_hours +
netflix_hours,
      performance_level =
case_when(
        exam_score >= 90 ~
"Excellent",
        exam_score >= 80 ~
"High",
        exam_score >= 70 ~
"Satisfactory",
        exam_score >= 60 ~
"Medium",
        exam_score >= 50 ~ "Fair",
        exam_score < 50 ~ "Low"
      ),
      performance_level =
factor(performance_level, levels
= c("Low", "Fair", "Medium",
"Satisfactory", "High",
"Excellent")),
      age_group = cut(age, breaks
= c(15, 17, 19, 21, 23), labels =
c("15–17", "18–19", "20–21",
"22–23"))
    )
}

data <- load_student_data()


# ========== UI
COMPONENTS ========== #

dashboard_css <- HTML(sprintf('
  .main-header .logo {
font-family: "Montserrat", "Arial",
sans-serif; font-weight: bold; }
  .box { border-radius: 10px; }
  .small-note { color: #888;
font-size: 13px;}
  .info-box-icon {
border-radius:10px 0 0 10px
!important; }
  .value-box { border-radius:10px
!important; min-height: 120px; }
  .skin-blue .main-header .logo {
background-color: #1a2236
!important; color: #fff !important;
}
  .skin-blue .main-header .navbar
{ background-color: #1a2236
!important; }
```

```
  .skin-blue .main-sidebar {
background-color: #232d44
!important; }

  .skin-blue .sidebar-menu >
li.active > a { background-color:
#285c8f !important; }

  body.dark-mode, .dark-mode
.content-wrapper, .dark-mode
.main-sidebar, .dark-mode
.main-header {
background-color: #181d1f
!important; color: #fff !important;
}

  .dark-mode .box, .dark-mode
.info-box, .dark-mode .value-box
{ background: #212531
!important; color: #fff !important;
}

  .dark-mode .sidebar-menu >
li.active > a { background-color:
#1abc9c !important; }

  .dark-mode
.dataTables_wrapper,

  .dark-mode .dataTable,

  .dark-mode table.dataTable,

  .dark-mode table.dataTable th,

  .dark-mode table.dataTable td {

    background-color: #232531
!important;

    color: #fff !important;

  }

  .dark-mode .paginate_button,

  .dark-mode .dataTables_info,

  .dark-mode .dataTables_filter,

  .dark-mode .dataTables_length,

  .dark-mode
.dataTables_paginate {

    color: #fff !important;

  }

  .dark-mode
.dataTables_wrapper input,

  .dark-mode
.dataTables_wrapper select {

    background-color: #232531
!important;

    color: #fff !important;

    border: 1px solid #888
!important;

  }

  .dark-mode
.dataTables_wrapper
.dataTables_filter input {

    background-color: #232531
!important;

    color: #fff !important;

  }

  #dark_mode_label { color: #fff
!important; font-weight:500; }

  .dark-mode #dark_mode_label
{ color: #fff !important; }

  /* Slightly lower the dark mode
button */

  .main-header .dropdown {
margin-top: 12px !important; }

'))


# ========== DASHBOARD
LAYOUT ========== #

ui <- dashboardPage(

  skin = "blue",

  dashboardHeader(

    title = tagList(

      span(icon("graduation-cap"),
APP_TITLE)
```

```
  ),
  tags$li(
    class = "dropdown",
    style = "margin-top: 12px;
margin-right: 12px; position:
relative;",
    materialSwitch(
      inputId = "dark_mode",
      label =
span(id="dark_mode_label",
"Dark Mode"),
      status = "primary",
      inline = TRUE,
      value = FALSE
    )
  )
 ),
 dashboardSidebar(
  width = 250,
  sidebarMenu(
    menuItem("Overview",
tabName = "overview", icon =
icon("info-circle")),
    menuItem("Summary",
tabName = "summary", icon =
icon("chart-bar")),
    menuItem("Correlations",
tabName = "correlations", icon =
icon("braille")),
    menuItem("Habits vs Exam
Score", tabName = "scatter",
icon = icon("chart-line")),
    menuItem("Habits by
Performance", tabName =
"boxplot", icon =
icon("th-large")),
    menuItem("Decision Tree",
tabName = "tree", icon =
icon("sitemap")),
    menuItem("Heatmaps",
tabName = "heatmap", icon =
icon("th")),
    menuItem("Demographics",
tabName = "barplots", icon =
icon("users")),
    menuItem("Data Explorer",
tabName = "data", icon =
icon("table")),
    hr(),
    h5("Global Filters"),
    selectInput("gender_filter",
"Gender", choices = c("All",
unique(data$gender)), selected
= "All"),
    selectInput("level_filter",
"Performance Level", choices =
c("All",
levels(data$performance_level)),
selected = "All"),
    sliderInput("age_filter", "Age
Range", min(data$age,
na.rm=TRUE), max(data$age,
na.rm=TRUE),
            value =
c(min(data$age, na.rm=TRUE),
max(data$age, na.rm=TRUE))),
    actionButton("reset_filters",
"Reset Filters", icon =
icon("redo"))
  )
 ),
 dashboardBody(
  useShinyjs(),

tags$head(tags$style(dashboard
_css)),
  tabItems(
```

```
tabItem("overview",
  fluidRow(
    box(width = 12, title = "Welcome!", status = "primary", solidHeader = TRUE,
      h3(APP_TITLE),
      p("This interactive dashboard helps you explore how student habits influence academic performance."),
      p("Use the sidebar for navigation and filters."),
      br(),
      p(sprintf("Created by %s", APP_CREATOR), class = "small-note")
    )
  )
),
tabItem("summary",
  fluidRow(
    valueBoxOutput("avg_exam_score", width = 3),
    valueBoxOutput("n_students", width = 3),
    valueBoxOutput("top_perf", width = 3),
    valueBoxOutput("most_common_gender", width = 3)
  ),
  box(width = 12, title = "Quick Data Summary", status = "primary", solidHeader = TRUE,
    tableOutput("summary_table")
  )
),
tabItem("correlations",
  box(width = 12, title = "Correlation Matrix", status = "primary", solidHeader = TRUE,
    withSpinner(plotOutput("corrplot", height = "450px")),
    p("Shows relationships between behavioral variables and exam score.", class = "small-note")
  )
),
tabItem("scatter",
  tabBox(width = 12, title = "Habits vs Exam Score",
    tabPanel("Study Hours", withSpinner(plotlyOutput("p1"))),
    tabPanel("Screen Time", withSpinner(plotlyOutput("p2"))),
    tabPanel("Sleep Hours", withSpinner(plotlyOutput("p3"))),
    tabPanel("Exercise Frequency", withSpinner(plotlyOutput("p4")))
  )
),
tabItem("boxplot",
  box(width = 12, title = "Student Habits by Performance Level", status = "primary", solidHeader = TRUE,
```

```
withSpinner(plotlyOutput("habitBoxplot", height = "550px")),
        p("Boxplots compare key habits for each performance level.", class = "small-note")
      )
    ),
    tabItem("tree",
      box(width = 12, title = "Decision Tree", status = "info", solidHeader = TRUE,
        withSpinner(plotOutput("treePlot", height = "600px")),
        p("Tree shows key predictors for performance.", class = "small-note")
      )
    ),
    tabItem("heatmap",
      fluidRow(
        box(width = 6, title = "By Performance Level", status = "primary", solidHeader = TRUE,
          withSpinner(plotlyOutput("heatmap1"))
        ),
        box(width = 6, title = "By Mental Health Rating", status = "primary", solidHeader = TRUE,
          withSpinner(plotlyOutput("heatmap2"))
        )
      ),
      p("Heatmaps summarize average screen time and sleep hours by group.", class = "small-note")
    ),
    tabItem("barplots",
      tabBox(width = 12, title = "Demographic Analysis",
        tabPanel("By Age Group", withSpinner(plotlyOutput("bar1"))),
        tabPanel("By Gender", withSpinner(plotlyOutput("bar2"))),
        tabPanel("By Diet Quality", withSpinner(plotlyOutput("bar3"))),
        tabPanel("By Internet Quality", withSpinner(plotlyOutput("bar4")))
      )
    ),
    tabItem("data",
      box(width = 12, title = "Explore Raw Data", status = "primary", solidHeader = TRUE,
        DTOutput("datatable"),
        downloadButton("downloadData", "Download Filtered Data")
      )
    )
  )
)
)
```

```
# ========== SERVER LOGIC
========== #

server <- function(input, output,
session) {

  # ---- THEME TOGGLE ----

  observe({

    if (isTRUE(input$dark_mode))
{

      shinyjs::addClass(selector =
"body", class = "dark-mode")

    } else {

shinyjs::removeClass(selector =
"body", class = "dark-mode")

    }

  })

  # ---- REACTIVE FILTER ----

  filtered_data <- reactive({

    d <- data

    if (input$gender_filter != "All")
d <- d[d$gender ==
input$gender_filter,]

    if (input$level_filter != "All") d
<- d[d$performance_level ==
input$level_filter,]

    d <- d[d$age >=
input$age_filter[1] & d$age <=
input$age_filter[2],]

    d

  })

observeEvent(input$reset_filters,
{

    updateSelectInput(session,
"gender_filter", selected = "All")

    updateSelectInput(session,
"level_filter", selected = "All")

    updateSliderInput(session,
"age_filter", value =
c(min(data$age, na.rm=TRUE),
max(data$age, na.rm=TRUE)))

  })

  # ---- VALUE BOXES ----

  output$n_students <-
renderValueBox({

valueBox(nrow(filtered_data()),
"Students (filtered)", icon =
icon("users"), color = "blue")

  })

  output$avg_exam_score <-
renderValueBox({

valueBox(round(mean(filtered_d
ata()$exam_score, na.rm =
TRUE), 2), "Avg Exam Score",
icon = icon("chart-line"), color =
"aqua")

  })

  output$top_perf <-
renderValueBox({

    tbl <-
table(filtered_data()$performanc
e_level)

    top <-
names(tbl)[which.max(tbl)]

    valueBox(top, "Most Common
Performance", icon =
icon("star"), color = "green")

  })

  output$most_common_gender
<- renderValueBox({

    tbl <-
table(filtered_data()$gender)
```

```
    top <-
names(tbl)[which.max(tbl)]
    valueBox(top, "Most Common
Gender", icon =
icon("venus-mars"), color =
"yellow")
  })


  # ---- SUMMARY TABLE ----
  output$summary_table <-
renderTable({
    d <- filtered_data()
    tibble(
      "Exam Score (Mean)" =
round(mean(d$exam_score,
na.rm=TRUE), 2),
      "Study Hours (Mean)" =
round(mean(d$study_hours_per
_day, na.rm=TRUE),2),
      "Screen Time (Mean)" =
round(mean(d$screen_time,
na.rm=TRUE),2),
      "Sleep Hours (Mean)" =
round(mean(d$sleep_hours,
na.rm=TRUE),2),
      "Attendance (%)" =
round(mean(d$attendance_perc
entage, na.rm=TRUE),2)
    )
  })


  # ---- CORRELATION ----
  output$corrplot <- renderPlot({
    cor_data <- filtered_data()
%>%
      select(exam_score,
study_hours_per_day,
screen_time, sleep_hours,
exercise_frequency)
    cor_matrix <- cor(cor_data,
use = "complete.obs")
    corrplot(cor_matrix, method =
"circle", type = "upper",
             tl.cex = 0.9, addCoef.col
= "black",
             title = "Correlation
Between Behavioral Variables
and Exam Score",
             mar = c(0, 0, 1, 0))
  })


  # ---- SCATTER PLOTS ----
  output$p1 <- renderPlotly({
    p <- ggplot(filtered_data(),
aes(x = study_hours_per_day, y
= exam_score, color =
performance_level, text =
paste("ID:", student_id))) +
      geom_point(size = 3, alpha =
0.7) +
      geom_smooth(method =
"lm", color = "red", se = FALSE)
+
      scale_color_manual(values
= performance_colors) +
      labs(title = "Study Hours vs
Exam Score", x = "Study Hours
per Day", y = "Exam Score",
color = "Performance Level") +
      theme_minimal()
    ggplotly(p, tooltip = c("x", "y",
"color", "text"))
  })
  output$p2 <- renderPlotly({
    p <- ggplot(filtered_data(),
aes(x = screen_time, y =
```

```
exam_score, color =
performance_level, text =
paste("ID:", student_id))) +

    geom_point(size = 3, alpha =
0.7) +

    geom_smooth(method =
"lm", color = "blue", se = FALSE)
+

    scale_color_manual(values
= performance_colors) +

    labs(title = "Screen Time vs
Exam Score", x = "Screen Time
(hrs)", y = "Exam Score", color =
"Performance Level") +

    theme_minimal()

  ggplotly(p, tooltip = c("x", "y",
"color", "text"))

 })

 output$p3 <- renderPlotly({

  p <- ggplot(filtered_data(),
aes(x = sleep_hours, y =
exam_score, color =
performance_level, text =
paste("ID:", student_id))) +

    geom_point(size = 3, alpha =
0.7) +

    geom_smooth(method =
"lm", color = "purple", se =
FALSE) +

    scale_color_manual(values
= performance_colors) +

    labs(title = "Sleep Hours vs
Exam Score", x = "Sleep Hours
per Day", y = "Exam Score",
color = "Performance Level") +

    theme_minimal()

  ggplotly(p, tooltip = c("x", "y",
"color", "text"))

 })

 output$p4 <- renderPlotly({

  p <- ggplot(filtered_data(),
aes(x = exercise_frequency, y =
exam_score, color =
performance_level, text =
paste("ID:", student_id))) +

    geom_jitter(width = 0.2, size
= 3, alpha = 0.7) +

    geom_smooth(method =
"lm", color = "darkgreen", se =
FALSE) +

    scale_color_manual(values
= performance_colors) +

    labs(title = "Exercise
Frequency vs Exam Score", x =
"Exercise Sessions per Week", y
= "Exam Score", color =
"Performance Level") +

    theme_minimal()

  ggplotly(p, tooltip = c("x", "y",
"color", "text"))

 })



 # ---- BOXPLOT ----

 output$habitBoxplot <-
renderPlotly({

  data_long <- filtered_data()
%>%

    select(performance_level,
study_hours_per_day,
screen_time, sleep_hours,
attendance_percentage) %>%

    pivot_longer(cols =
-performance_level, names_to =
"Habit", values_to = "Value")

  p <- ggplot(data_long, aes(x =
performance_level, y = Value, fill
= performance_level)) +

    geom_boxplot() +
```

```
    facet_wrap(~ Habit, scales =
"free_y") +

    scale_fill_manual(values =
performance_colors) +

    labs(title = "Comparison of
Student Habits by Performance
Level",

         x = "Performance Level",

         y = "Value (hours or
percentage)") +

    theme_minimal() +

    theme(strip.text =
element_text(face = "bold"))

  ggplotly(p)

 })



 # ---- DECISION TREE ----

 output$treePlot <- renderPlot({

  d <- filtered_data()

  d$internet_quality <-
as.factor(d$internet_quality)

  model <-
rpart(performance_level ~
study_hours_per_day +
screen_time + sleep_hours +
internet_quality,

                 data = d, method =
"class")

  rpart.plot(model, type = 4,
extra = 104, box.palette =
"Blues")

  title(main = "Decision Tree:
Factors Influencing Performance
Level", line = 2.4, cex.main = 1)

 })



 # ---- HEATMAPS ----

 output$heatmap1 <-
renderPlotly({

  heatmap_data <-
filtered_data() %>%


group_by(performance_level)
%>%

    summarise(

      avg_screen_time =
mean(screen_time, na.rm =
TRUE),

      avg_sleep_hours =
mean(sleep_hours, na.rm =
TRUE)

    )

  melted <- melt(heatmap_data,
id.vars = "performance_level")

  p <- ggplot(melted, aes(x =
variable, y = performance_level,
fill = value)) +

    geom_tile(color = "white") +

    scale_fill_gradient(low =
"lightyellow", high = "firebrick") +

    labs(title = "Avg Screen Time
& Sleep Hours by Performance
Level",

         x = "Behavioral Metric", y
= "Performance Level", fill =
"Avg Value") +

    theme_minimal()

  ggplotly(p)

 })

 output$heatmap2 <-
renderPlotly({

  heatmap_data <-
filtered_data() %>%
```

```
group_by(mental_health_rating)
%>%
      summarise(
        avg_screen_time =
mean(screen_time, na.rm =
TRUE),
        avg_sleep_hours =
mean(sleep_hours, na.rm =
TRUE)
      )
    melted <- melt(heatmap_data,
id.vars = "mental_health_rating")
    p <- ggplot(melted, aes(x =
variable, y =
as.factor(mental_health_rating),
fill = value)) +
      geom_tile(color = "white") +
      scale_fill_gradient(low =
"lightyellow", high = "firebrick") +
      labs(title = "Avg Screen Time
& Sleep Hours by Mental Health
Rating",
           x = "Behavioral Metric",
           y = "Mental Health Rating
(1 = Poor, 10 = Excellent)",
           fill = "Avg Value") +
      theme_minimal()
    ggplotly(p)
  })

  # ---- DEMOGRAPHIC
BARPLOTS ----
  output$bar1 <- renderPlotly({
    d <- filtered_data()
    grp <- d %>%

group_by(performance_level,
age_group) %>%
      summarise(count = n())
    p <- ggplot(grp, aes(x =
performance_level, y = count, fill
= age_group)) +
      geom_col(position =
"dodge") +
      labs(title = "Performance
Level by Age Group", x =
"Performance Level", y =
"Number of Students", fill = "Age
Group") +
      theme_minimal()
    ggplotly(p)
  })
  output$bar2 <- renderPlotly({
    d <- filtered_data()
    grp <- d %>%

group_by(performance_level,
gender) %>%
      summarise(count = n())
    p <- ggplot(grp, aes(x =
performance_level, y = count, fill
= gender)) +
      geom_col(position =
"dodge") +
      labs(title = "Performance
Level by Gender", x =
"Performance Level", y =
"Number of Students", fill =
"Gender") +
      theme_minimal()
    ggplotly(p)
  })
  output$bar3 <- renderPlotly({
    d <- filtered_data()
    grp <- d %>%
```

```
group_by(performance_level,
diet_quality) %>%
    summarise(count = n())
  p <- ggplot(grp, aes(x =
performance_level, y = count, fill
= diet_quality)) +
    geom_col(position =
"dodge") +
    labs(title = "Performance
Level by Diet Quality", x =
"Performance Level", y =
"Number of Students", fill = "Diet
Quality") +
    theme_minimal()
  ggplotly(p)
 })
 output$bar4 <- renderPlotly({
  d <- filtered_data()
  grp <- d %>%
group_by(performance_level,
internet_quality) %>%
    summarise(count = n())
  p <- ggplot(grp, aes(x =
performance_level, y = count, fill
= internet_quality)) +
    geom_col(position =
"dodge") +
    labs(title = "Performance
Level by Internet Quality", x =
"Performance Level", y =
"Number of Students", fill =
"Internet Quality") +
    theme_minimal()
  ggplotly(p)
 })

 # ---- DATA TABLE ----
 output$datatable <- renderDT({
  datatable(
    filtered_data(),
    options = list(pageLength =
10, scrollX = TRUE),
    filter = "top",
    rownames = FALSE
  )
 })
 output$downloadData <-
downloadHandler(
  filename = function() {
paste0("student_data_filtered-",
Sys.Date(), ".csv") },
  content = function(file) {
    write_csv(filtered_data(), file)
  }
 )
}

# ========== APP LAUNCH
========== #
shinyApp(ui, server)
```

References

Anwar, N., Juanda, Anderson, J., & Williams, T. (2024). Applying data science to analyze and improve student learning outcomes in educational environments. International Transactions on Education Technology, 3(1), 72–83. https://journal.pandawan.id/itee/article/view/679

Hale, L., & Guan, S. (2015). Screen time and sleep among school-aged children and adolescents: A systematic literature review. Sleep Medicine Reviews, 21, 50–58. https://doi.org/10.1016/j.smrv.2014.07.007

Ouatik, A., et al. (2022). Predicting students' academic performance using machine learning techniques. International Journal of Emerging Technologies in Learning, 17(4) https://scholarworks.utrgv.edu/cgi/viewcontent.cgi?article=1572&context=mss_fac

Pérez-Chada, D., et al. (2023). Screen use, sleep duration, daytime somnolence, and academic failure in school-aged adolescents. Frontiers in Public Health, 11, Article 107. https://pubmed.ncbi.nlm.nih.gov/36787301/

West, M. R., et al. (2019). School peer non-academic skills and academic performance in high school. Frontiers in Education, 4, 57. https://doi.org/10.3389/feduc.2019.00057

Wickham, H. (n.d.). *Mastering Shiny: A comprehensive guide*. https://mastering-shiny.org

RStudio. (n.d.). *Shiny dashboard layouts and themes*. https://rstudio.github.io/shinydashboard

RStudio. (n.d.). *Reactive programming in Shiny*. https://shiny.rstudio.com/articles/reactivity-overview.html

Romero, C., & Ventura, S. (2024). *Educational data mining and learning analytics: An updated survey*. arXiv. arxiv.org+1onlinelibrary.wiley.com+1

Winter, M., Mordel, J., Mendzheritskaya, J., et al. (2024). Behavioral trace data in an online learning environment as indicators of learning engagement in university students. *Frontiers in Psychology*, 15, Article 1396881. frontiersin.org

Sarker, S., et al. (2024). Advancing educational data mining for enhanced student performance prediction: A fusion of feature selection algorithms and classification techniques with dynamic feature ensemble evolution. *Scientific Reports*, 14, Article 92324

AllviA Blog. (2023). The Benefits of Data-driven Analysis in Digital Learning. Retrieved October 10, 2023, from https://blog.allviaedu.com/educator/12894/

Appendices

Appendix A: Dataset Summary

Dataset Name: `cleaned_student_habits_performance_data.csv`
Number of Observations: 1000
Number of Variables: 18 variables (2 created variables)
Data Source: Manually cleaned and processed version of original `student_habits_performance.csv`
Key Features:

- Demographics: `age`, `gender`
- Behavioral: `study_hours_per_day`, `social_media_hours`, `netflix_hours`, `sleep_hours`, `exercise_frequency`, `diet_quality`, `internet_quality`
- Derived: `screen_time`, `performance_level`
- Target: `exam_score`

Appendix B: Data Cleaning Summary

Aspect	Description
Missing Values	Checked and verified that no missing values remain
Duplicate Records	Verified and removed using R filtering techniques
Column Formatting	Standardized types using `mutate()` and converted to factors where needed
Derived Variables	Created `screen_time` (social media + Netflix) and `performance_level`
Outlier Handling	Filtered extreme outliers in `screen_time` (>12 hrs); none removed

Appendix C: Research Questions

1. *Which behavioral factors (e.g., screen time, study hours) correlate most strongly with student academic performance?*

2. *What recurring patterns are common among students showing signs of academic risk or underperformance?*

3. *How do time-related habits like studying and screen use reflect performance differences?*

4. *How can interactive visualizations help educators identify students who may benefit from early support?*

5. *What demographic trends (gender, age, diet, internet access) are linked to performance disparities across groups?*

Appendix D: List of R Packages Used

Package	Purpose
`shiny`	Core framework for interactive web apps
`shinydashboard`	Dashboard layout and UI panels
`shinyjs`	JavaScript integration for UI interactivity
`shinyWidgets`	Enhanced UI components and controls
`shinycssloaders`	Loading animations for outputs
`ggplot2`	Static data visualizations
`plotly`	Interactive charting and hover functionality
`tidyverse`	Data wrangling and transformation
`readr`	Reading CSV files
`dplyr`	Data manipulation
`reshape2`	Data reshaping for heatmaps
`corrplot`	Correlation matrix plots
`rpart`	Decision tree modeling
`rpart.plot`	Visualizing decision trees
`DT`	Interactive data tables

Appendix E: Screenshot Overview

📌 *Refer to Chapter 4.2 (page 24) for detailed screenshots and labeled visualizations.*

Honor Pledge:

“I accept responsibility for my role in ensuring the integrity of the work submitted by the group in which I participated.”