SRM INSTITUTE OF SCINCE AND TECHNOLOGY,
TIRUCHIRAPPALLI

COLLEGE OF ENGINEERING AND TECHNOLOGY,
SCHOOL OF COMPUTING

21CSS303T - Data Science

Project Report

2024-2025

SRM INSTITUTE OF SCINCE AND TECHNOLOGY,
TIRUCHIRAPPALLI

COLLEGE OF ENGINEERING AND TECHNOLOGY,
SCHOOL OF COMPUTING

21CSS303T - Data Science

Project Report

Register No. : ______________________________

Name of the Student : ______________________________

Semester : ______________________________

Programme : **B.Tech.** ________________________

Academic Year : **2024-2025**

Table of contents

Abstract

Student performance is influenced by various academic, behavioral, and socioeconomic factors. Analyzing these factors can help educators identify key trends and take proactive measures to support students. This project utilizes MATLAB-based data visualization techniques to explore the relationship between study habits, stress levels, sleep patterns, attendance, and academic performance using the Students Grading Dataset.

The dataset includes multiple attributes such as midterm and final scores, assignments and quizzes averages, attendance percentage, study hours per week, extracurricular participation, and demographic details. By applying histograms, scatter plots, bar charts, and correlation heatmaps, etc. the study identifies significant trends, including the impact of study hours, stress, and sleep patterns on total scores.

Findings suggest that higher study hours generally correlate with better performance, but excessive stress and irregular sleep patterns can negatively impact scores. Additionally, attendance plays a crucial role in student success, and parental education levels have an observable influence on academic achievements.

This study highlights the importance of data-driven decision-making in education, emphasizing how visualization techniques can help detect patterns that would otherwise remain hidden in raw data. The results can be used to develop strategies for improving academic outcomes, assisting struggling students, and promoting overall student well-being.

Introduction

2.1 Project Overview

Student academic performance is influenced by a variety of factors, including **study habits, attendance, stress levels, sleep patterns, parental education, and socioeconomic background**. Analyzing these factors through **data visualization techniques** can help educators, researchers, and policymakers make informed decisions to improve student success.

This project leverages **MATLAB-based data visualization** to explore relationships between these variables using the **Students Grading Dataset**. The dataset includes detailed academic records such as **midterm and final scores, quizzes, and assignment averages**, as well as behavioral aspects like **study hours per week, extracurricular involvement, stress levels, and sleep duration**.

By implementing **histograms, scatter plots, bar charts, and correlation heatmaps**, this project aims to uncover **meaningful trends and patterns** that impact student performance. The findings can help in identifying students at risk, designing intervention programs, and developing strategies for enhancing academic achievement.

2.2 Objectives

The primary objectives of this study are:

- **Analyze academic performance trends** by visualizing students' scores in different assessments.
- **Identify behavioral influences** such as study habits, attendance, and sleep patterns on total scores.
- **Examine the impact of stress levels** on students’ ability to perform well in exams. **Investigate the role of parental education and socioeconomic background** in academic success.
- **Use MATLAB’s data visualization capabilities** to present clear and insightful graphical representations of the dataset.

2.3 Importance of Data Visualization

Data visualization plays a crucial role in educational analytics by enabling stakeholders to:

◆ **Identify key performance drivers**: Analyzing relationships between study habits, attendance, and academic scores can help determine the most significant predictors of student success.

❖ **Detect early warning signs**: Visualization helps in spotting students who are underperforming due to stress, low attendance, or lack of study time, allowing for timely interventions.

❖ **Compare and contrast academic trends**: Educators can use visual tools to understand how different demographics, departments, and study patterns influence grades.

❖ **Improve decision-making**: Institutions can optimize teaching strategies, resource allocation, and student support services based on **data-driven insights** rather than assumptions.

By utilizing **MATLAB's powerful plotting functions**, this project translates raw student data into **meaningful insights** that can enhance the learning environment and improve academic outcomes.

Dataset Description

3.1 Source of the Dataset

The dataset used in this project, titled **Students Grading Dataset**, has been sourced from **Kaggle** and contains detailed information about student performance, behavior, and socioeconomic factors. This dataset is particularly useful for understanding the relationship between **academic success and external influences** such as **study habits, attendance, parental education, and stress levels**.

This dataset is structured as a **JSON file**, which has been processed and converted into a MATLAB-friendly format (table structure) to facilitate analysis and visualization.

3.2 Key Features and Attributes

The dataset comprises multiple attributes categorized into **four major sections**:

1. Academic Performance Metrics

These attributes measure the students' **overall academic success**:

- **Midterm_Score**: Marks obtained in the midterm examination.
- **Final_Score**: Marks obtained in the final examination.
- **Assignments_Avg**: Average score of all submitted assignments.
- **Quizzes_Avg**: Average score of all quizzes taken.
- **Projects_Score**: Score obtained in major course projects.
- **Total_Score**: The cumulative score considering all assessments.
- **Grade**: The final letter grade assigned to the student (A, B, C, D, or F).

2. Student Demographics & Departmental Information

These attributes provide background information on each student:

- **Student_ID**: Unique identifier assigned to each student.
- **Gender**: Male/Female classification.
- **Age**: Age of the student.
- **Department**: The field of study (e.g., CS, Engineering, Mathematics, Business).

3. Behavioral & Study Habits Indicators

These features help in understanding how students' habits impact performance:

- **Attendance (%)**: Percentage of classes attended by the student.
- **Study_Hours_per_Week**: The number of hours a student dedicates to self-study weekly.
- **Extracurricular_Activities**: Whether a student participates in extracurricular activities (Yes/No).
- **Participation_Score**: A measure of in-class engagement and active participation.

4. Socioeconomic & Psychological Factors

These attributes help assess external influences on student success:

- **Internet_Access_at_Home**: Indicates if the student has internet access at home (Yes/No).
- **Parent_Education_Level**: Highest education level attained by the student's parents (None, High School, Bachelor's, Master's, PhD).
- **Family_Income_Level**: Categorized as Low, Medium, or High.
- **Stress_Level (1-10)**: A self-reported stress level on a scale from 1 (low) to 10 (high).
- **Sleep_Hours_per_Night**: The average number of hours a student sleeps per night.

These factors play a **crucial role** in student performance and will be explored through **MATLAB-based visualizations**.

3.3 Data Preprocessing in MATLAB

Before performing visualizations, the dataset undergoes **preprocessing** to handle:

- **Missing Values**: Some records have null values in **attendance, assignments, and parent education level**. These are either **imputed** using averages or removed if they significantly affect the analysis.
- **Categorical Data Conversion**:
  - **Extracurricular_Activities (Yes/No) → 1/0**
  - **Parent_Education_Level (Text) → Numerical Encoding (0-4)**
  - **Family_Income_Level (Low/Medium/High) → 1/2/3**
- **Normalization**: Scores are normalized to a **0-100 scale** to allow for accurate comparison.

Dataset Import & Data Processing in MATLAB

1. Importing the Dataset

The dataset was imported as a JSON file using MATLAB's built-in **Import Data** button. This method automatically converted the JSON structure into a **table format (`T`)**, making it easier to manipulate within MATLAB.

```
matlab
CopyEdit
% Import JSON file using MATLAB's built-in import function
T = readtable('dataset.json', 'PreserveVariableNames', true);
```

2. Data Preprocessing

Several preprocessing steps were performed to prepare the data for visualization and analysis:

A. Converting Categorical Variables

Some attributes were converted into categorical variables to facilitate plotting and analysis. This included:

- **Gender** (e.g., Male, Female, Other)
- **Family Income Level** (e.g., Low, Medium, High)
- **Grade** (e.g., A, B, C, etc.)

```
matlab
CopyEdit
 % Ensure categorical variables
if ~iscategorical(T.Gender)
        T.Gender = categorical(T.Gender);
end

if ~iscategorical(T.Family_Income_Level)
        T.Family_Income_Level = categorical(T.Family_Income_Level);
end

if ~iscategorical(T.Grade)
        T.Grade = categorical(T.Grade);
end
```

B. Handling Numerical Variables

For attributes like **Midterm Score, Final Score, Total Score, Study Hours per Night, Sleep Hours per Night**, and **Stress Level**, numerical processing was required.

```
matlab
CopyEdit
% Extract numerical columns
midtermScores = T.Midterm_Score;
finalScores = T.Final_Score;
totalScores = T.Total_Score;
studyHours = T.Study_Hours_per_Night;
sleepHours = T.Sleep_Hours_per_Night;
stressLevels = T.Stress_Level_1_10;
```

C. Fixing Column Name Issues

During execution, an error occurred due to incorrect column name usage (`Sleep_Hours_per_Day` instead of `Sleep_Hours_per_Night`). This was corrected using:

```
matlab
CopyEdit
% Ensure correct variable names
if ismember('Sleep_Hours_per_Night', T.Properties.VariableNames)
      sleepHours = T.Sleep_Hours_per_Night;
end
```

D. Filtering and Grouping Data

Some visualizations required grouping data, such as:

- Analyzing stress levels by **Gender**
- Examining **Grade Distribution** across different **Departments**
- Studying the correlation of **Study Hours** vs **Total Scores**

Example of filtering total scores by grade:

```
matlab
CopyEdit
grades = unique(T.Grade);
gradeData = cell(length(grades), 1);
for i = 1:length(grades)
     gradeData{i} = T.Total_Score(T.Grade == grades(i));
end
```

E. Handling Missing Data

If any missing data was found in crucial numerical fields, it was handled using:

```
matlab
CopyEdit
% Remove missing values
T = rmmissing(T);
```

Software and Methodologies

This project is done using **MATLAB** and its various tools for **data preprocessingand visualization.** MATLAB provides a **powerful environment** for handling structured data, performing numerical computations, and generating insightful visualizations.

4.1 MATLAB Toolboxes Used

This project makes use of the following MATLAB **tools and built-in functions** to perform **data preprocessing, analysis, and visualization**:

1. MATLAB Base Functions

The base functions of MATLAB provide fundamental capabilities, such as:

- **Data Import** – `jsondecode()`, `readtable()` for reading the dataset from JSON or CSV format.
- **Data Cleaning** – `fillmissing()`, `mean()`, `categorical()`, `ismember` to handle missing values and format categorical data.
- **Basic Visualization** – `plot()`, `scatter()`, `histogram()`, `bar()` for **creating essential charts**.

2. Statistics and Machine Learning Toolbox

This toolbox is used to:

- **Find Correlations** – The `corr()` function helps determine **how study hours, stress levels, and attendance affect scores**.

3. Data Visualization Tools

- **Histograms** – `histogram()` to **show the distribution of student grades**.
- **Scatter Plots** – `scatter()` to **analyze relationships between study hours and total scores**.
- **Bar Charts** – `bar()` to **compare academic performance across genders and departments**.
- **Heatmaps** – `heatmap()` to **visualize correlations between different student performance.**

Implementation and MATLAB Code

This section provides a **step-by-step implementation** of data processing and visualization using **MATLAB**. The workflow consists of:

- **Loading and Preprocessing the Dataset**
- **Handling Missing Data & Removing Unnecessary Columns**
- **Converting Categorical Data to Numerical Values**
- **Visualizing Key Trends with MATLAB Graphs**

5.1 Data Import and Preprocessing

5.1.1 Loading the Dataset into MATLAB

Since the dataset is in **JSON format**, it is read, decoded, and converted into a MATLAB **table** for easier manipulation.

```
matlab
CopyEdit
clearvars;
fid = fopen('Students_Grading_Dataset.json');
raw = fread(fid, inf);
fclose(fid);

str = char(raw');
```

```
data = jsondecode(str);

T = struct2table(data);
clear fid raw str data;
save('Students_Grading_Dataset.mat', 'T');
```

Output: The first few rows of the dataset will be displayed, showing **student scores, study hours, stress levels, and other attributes**.

5.1.2 Reloading the Dataset and Summary Statistics

After initial processing, the dataset is reloaded to verify its integrity and obtain summary statistics.

```
matlab
CopyEdit
load('Students_Grading_Dataset.mat');
summary(T)
```

Output: The summary provides a **statistical overview of numeric attributes** such as **Final Scores, Study Hours, and Stress Levels**.

5.1.3 Removing the Attendance Column

Since **Attendance** does not have proper values, it is removed from the dataset.

```
matlab
CopyEdit
T.Attendance___ = [];
disp('Attendance column removed due to missing values.');
```

Output: The **Attendance** column is removed, ensuring only **relevant and complete data** is used for visualization.

5.2 Visualization Techniques

This project uses various **MATLAB visualization techniques** to identify trends in student performance. The following plots will be implemented:

Basic Distribution

- **Histogram of Final Scores** – To analyze the distribution of student final scores.

Performance comparison

- **Scatter Plot: Midterm Score vs. Final Score** – To examine relationships between midterm and final exam performance.
- **Correlation Graph: Sleep Hours per Night vs. Stress Level** – To understand the effect of sleep on stress.

Categorical Data Representation

- **Bar Graph: Distribution of Family Income Levels** – To visualize income-level representation in the dataset.
- **Box Plot of Stress Levels by Gender** – To compare stress distribution among male and female students.
- **Pie Chart of Extracurricular Activities Participation** – To show the proportion of students engaged in extracurricular activities.

Identifying Key Performance Trends

- **Line Plot of Average Study Hours vs. Age** – To track how study habits change with age.
- **Heatmap of Study Hours vs. Sleep Hours** – To analyze the correlation between study and rest.

Advanced Data Representations

- **Area Plot of Total Score Distribution Across Age Groups** – To observe how total scores vary with age.
- **Bubble Chart of Study Hours vs. Total Score by Age** – To depict study hour effectiveness across age groups.
- **Radar Chart of Academic Performance Metrics** – To compare multiple performance metrics in a single chart.
- **Parallel Coordinates Plot of Academic Performance** – To show relationships among various academic factors.
- **Cumulative Distribution Function (CDF) of Total Scores** – To analyze cumulative probability distribution of scores.
- **Horizontal Bar Chart of Average Scores by Department** – To compare academic performance across departments.
- **3D Surface Plot of Study Hours, Sleep Hours, and Total Score** – To visualize performance in three-dimensional space.

5.2.1 Histogram – Distribution of Final Scores

```
close all;
clearvars -except T;
clc;

finalScores = T.Final_Score;
histogram(finalScores, 'BinWidth', 5, 'FaceColor', 'b',
'EdgeColor', 'k');

xlabel('Final Score', 'FontSize', 12, 'FontWeight', 'bold');

ylabel('Number of Students', 'FontSize', 12, 'FontWeight', 'bold');
title('Distribution of Final Scores', 'FontSize', 14, 'FontWeight',
'bold');
```

```
grid on;  % Add grid lines for better readability
box on;  % Ensure axes are visible
xlim([40, 100]); % Set X-axis limits based on known data range
drawnow;
```

5.2.2 Scatter Plot: Midterm Score vs. Final Score

```
close all;
clearvars -except T;
clc;


midtermScores = T.Midterm_Score;
finalScores = T.Final_Score;
figure('Position', [100, 100, 800, 600]);

scatter(midtermScores, finalScores, 40, 'b', 'filled',
'MarkerFaceAlpha', 0.4);
title('Midterm Score vs Final Score', 'FontSize', 16, 'FontWeight',
'bold');

xlabel('Midterm Score', 'FontSize', 14, 'FontWeight', 'bold');
ylabel('Final Score', 'FontSize', 14, 'FontWeight', 'bold');

grid on;
box on;
xlim([min(midtermScores)-5, max(midtermScores)+5]);
ylim([min(finalScores)-5, max(finalScores)+5]);

hold on;

p = polyfit(midtermScores, finalScores, 1);
xFit = linspace(min(midtermScores), max(midtermScores), 100);
yFit = polyval(p, xFit);
plot(xFit, yFit, '-r', 'LineWidth', 2.5);

eqText = sprintf('y = %.2fx + %.2f', p(1), p(2));
text(min(midtermScores)+5, max(finalScores)-5, eqText, 'FontSize',
12, ...
'FontWeight', 'bold', 'Color', 'r', 'BackgroundColor', 'w');

legend({'Student Scores', 'Trend Line'}, 'Location', 'northwest',
'FontSize', 12);
drawnow;
```

5.2.3 Correlation Graph: Sleep Hours per Night vs. Stress Level

```
close all;
clearvars -except T;
clc;


sleepHours = T.Sleep_Hours_per_Night;
```

```
stressLevel = T.Stress_Level_1_10_;
figure('Position', [150, 150, 800, 600]);


scatter(sleepHours, stressLevel, 40, stressLevel, 'filled');
colormap(cool); % Apply a cool color scheme

title('Sleep Hours per Night vs Stress Level', 'FontSize', 16,
'FontWeight', 'bold');
xlabel('Sleep Hours per Night', 'FontSize', 14, 'FontWeight',
'bold');
ylabel('Stress Level (1-10)', 'FontSize', 14, 'FontWeight',
'bold');

grid on;
box on;
colorbar;
xlim([min(sleepHours)-0.5, max(sleepHours)+0.5]);
ylim([min(stressLevel)-1, max(stressLevel)+1]);

hold on;

p = polyfit(sleepHours, stressLevel, 1);
xFit = linspace(min(sleepHours), max(sleepHours), 100);
yFit = polyval(p, xFit);
plot(xFit, yFit, '-r', 'LineWidth', 2.5);

eqText = sprintf('y = %.2fx + %.2f', p(1), p(2));
text(min(sleepHours)+0.5, max(stressLevel)-1, eqText, 'FontSize',
12, ...
'FontWeight', 'bold', 'Color', 'r', 'BackgroundColor', 'w');

legend({'Student Data', 'Trend Line'}, 'Location', 'northeast',
'FontSize', 12);
drawnow;
```

5.2.4 Bar Graph: Distribution of Family Income Levels

```
close all;
clearvars -except T;
clc;

incomeLevels = categories(T.Family_Income_Level);
incomeCounts = countcats(T.Family_Income_Level);

figure('Position', [150, 150, 800, 600]);
barChart = bar(incomeCounts, 'FaceColor', [0.2 0.6 0.8]);

title('Distribution of Family Income Levels', 'FontSize', 16,
'FontWeight', 'bold');
xlabel('Family Income Level', 'FontSize', 14, 'FontWeight',
'bold');
ylabel('Number of Students', 'FontSize', 14, 'FontWeight', 'bold');

xticks(1:length(incomeLevels));
xticklabels(incomeLevels);

grid on;
```

```
box on;
ylim([0 max(incomeCounts) + 100]);

for i = 1:length(incomeCounts)
text(i, incomeCounts(i) + 50, num2str(incomeCounts(i)), 'FontSize',
12, ...
'FontWeight','bold','HorizontalAlignment','center','Color','k');
end
drawnow;
```

5.2.5 Box Plot of Stress Levels by Gender

```
close all;
clearvars -except T;
clc;

if ~iscategorical(T.Gender)
T.Gender = categorical(T.Gender);
end

figure('Position', [150, 150, 800, 600]);
boxplot(T.Stress_Level_1_10_, T.Gender,'Notch','on','Symbol','r+');
title('Distribution of Stress Levels by Gender', 'FontSize', 16,
'FontWeight', 'bold');
xlabel('Gender', 'FontSize', 14, 'FontWeight', 'bold');
ylabel('Stress Level (1-10)', 'FontSize', 14, 'FontWeight', 'bold');
grid on;
box on;
set(gca, 'FontSize', 12);
drawnow;
```

5.2.6 Pie Chart of Extracurricular Activities Participation

```
close all;
clearvars -except T;
clc;

if ~iscategorical(T.Extracurricular_Activities)
T.Extracurricular_Activities =
categorical(T.Extracurricular_Activities);
end

activityCounts = countcats(T.Extracurricular_Activities);
activityLabels = categories(T.Extracurricular_Activities);

figure('Position', [150, 150, 800, 600]);
pie(activityCounts);
legend(activityLabels, 'Location', 'bestoutside');



title('Participation in Extracurricular Activities', 'FontSize', 16,
'FontWeight', 'bold');
drawnow;
```

5.2.7 Line Plot of Average Study Hours vs. Age

```
close all;
clearvars -except T;
clc;

ageGroups = unique(T.Age);
avgStudyHours = arrayfun(@(x) mean(T.Study_Hours_per_Week(T.Age ==
x)), ageGroups);

figure('Position', [150, 150, 800, 600]);

plot(ageGroups, avgStudyHours, '-o', 'LineWidth', 2, 'MarkerSize',
8, 'MarkerFaceColor', 'b');

title('Average Study Hours per Week by Age', 'FontSize', 16,
'FontWeight', 'bold');
xlabel('Age', 'FontSize', 14, 'FontWeight', 'bold');
ylabel('Average Study Hours per Week', 'FontSize', 14, 'FontWeight',
'bold');

grid on;
box on;
set(gca, 'FontSize', 12);
drawnow;
```

5.2.8 Heatmap of Study Hours vs. Sleep Hours

```
close all;
clearvars -except T;
clc;

figure('Position', [150, 150, 800, 600]);

sleepBins =
min(T.Sleep_Hours_per_Night):0.5:max(T.Sleep_Hours_per_Night);
studyBins =
min(T.Study_Hours_per_Week):1:max(T.Study_Hours_per_Week);

heatmapData = histcounts2(T.Sleep_Hours_per_Night,
T.Study_Hours_per_Week, sleepBins, studyBins);

imagesc(sleepBins(1:end-1), studyBins(1:end-1), heatmapData');

colormap hot;
colorbar;
caxis([0 max(heatmapData(:))]);

title('Heatmap of Study Hours vs. Sleep Hours', 'FontSize', 16,
'FontWeight', 'bold');
xlabel('Sleep Hours per Night', 'FontSize', 14, 'FontWeight',
'bold');

ylabel('Study Hours per Week', 'FontSize', 14, 'FontWeight',
'bold');
```

```
set(gca, 'YDir', 'normal');
drawnow;
```

5.2.9 Area Plot of Total Score Distribution Across Age Groups

```
close all;
clearvars -except T;
clc;

uniqueAges = unique(T.Age);
avgTotalScores = arrayfun(@(age) mean(T.Total_Score(T.Age == age)),
uniqueAges);

figure('Position', [150, 150, 800, 600]);
area(uniqueAges, avgTotalScores, 'FaceColor', [0.2 0.6 0.8],
'EdgeColor', 'k', 'LineWidth', 1.5);

title('Area Plot of Total Score Distribution Across Age Groups',
'FontSize', 16, 'FontWeight', 'bold');
xlabel('Age', 'FontSize', 14, 'FontWeight', 'bold');
ylabel('Average Total Score', 'FontSize', 14, 'FontWeight', 'bold');

grid on;
box on;
ylim([min(avgTotalScores) - 5, max(avgTotalScores) + 5]);
drawnow;
```

5.3.0 Bubble Chart of Study Hours vs. Total Score by Age

```
close all;
clearvars -except T;
clc;

studyHours = T.Study_Hours_per_Week;
totalScore = T.Total_Score;
age = T.Age;
bubbleSize = (age - min(age)) / (max(age) - min(age)) * 100 + 10;

figure('Position', [150, 150, 800, 600]);
scatter(studyHours, totalScore, bubbleSize, age, 'filled');

title('Bubble Chart of Study Hours vs. Total Score by Age',
'FontSize', 16, 'FontWeight', 'bold');
xlabel('Study Hours per Week', 'FontSize', 14, 'FontWeight',
'bold');
ylabel('Total Score', 'FontSize', 14, 'FontWeight', 'bold');

colorbar;
colormap(jet);
caxis([min(age) max(age)]);
ylabel(colorbar, 'Age', 'FontSize', 12);

grid on;
box on;
xlim([min(studyHours) - 1, max(studyHours) + 1]);
```

```
ylim([min(totalScore) - 5, max(totalScore) + 5]);
drawnow;
```

5.3.1 Radar Chart of Academic Performance Metrics

```
close all;
clearvars -except T;
clc;

studentIdx = randi(height(T));
metrics = [T.Midterm_Score(studentIdx), T.Final_Score(studentIdx),
...  T.Quizzes_Avg(studentIdx), T.Projects_Score(studentIdx), ...
T.Participation_Score(studentIdx)];
labels = {'Midterm', 'Final', 'Quizzes', 'Projects',
'Participation'};

maxValues = [100, 100, 100, 100, 10];
normalizedMetrics = metrics ./ maxValues;

figure('Position', [150, 150, 800, 600]);

theta = linspace(0, 2 * pi, numel(labels) + 1);
metricsCircular = [normalizedMetrics, normalizedMetrics(1)];

polarplot(theta, metricsCircular, '-o', 'LineWidth', 2,
'MarkerSize', 6, 'MarkerFaceColor', 'r');

rlim([0 1]);

ax = gca;
ax.ThetaTick = rad2deg(theta(1:end-1));
ax.ThetaTickLabel = labels;
ax.ThetaGrid = 'on';
ax.RGrid = 'on';

title(sprintf('Radar Chart of Academic Performance (Student ID:
%s)', T.Student_ID{studentIdx}), ... 'FontSize', 16, 'FontWeight',
'bold');
set(gca, 'FontSize', 12);
drawnow;
```

5.3.2 Parallel Coordinates Plot of Academic Performance

```
close all;
clearvars -except T;
clc;

data = [T.Quizzes_Avg, T.Participation_Score, T.Projects_Score,
T.Midterm_Score, T.Final_Score, T.Total_Score];

labels = {'Quizzes Avg', 'Participation', 'Projects', 'Midterm',
'Final', 'Total Score'};

figure('Position', [150, 150, 900, 600]);
parallelcoords(data, 'Standardize', 'on', 'LineWidth', 1.5);
```

```
title('Parallel Coordinates Plot of Academic Performance',
'FontSize', 16, 'FontWeight', 'bold');
xlabel('Performance Metrics', 'FontSize', 14, 'FontWeight', 'bold');
ylabel('Standardized Scores', 'FontSize', 14, 'FontWeight', 'bold');
xticklabels(labels);
grid on;
drawnow;
```

5.3.3 Cumulative Distribution Function (CDF) of Total Scores

```
close all;
clearvars -except T;
clc;

totalScores = T.Total_Score;
sortedScores = sort(totalScores);
n = numel(sortedScores);

cumulativeProb = (1:n) / n;
figure('Position', [150, 150, 900, 600]);
plot(sortedScores, cumulativeProb, 'b', 'LineWidth', 2);
hold on;

medianScore = median(sortedScores);
yline(0.5, 'r--', 'LineWidth', 1.5);
xline(medianScore, 'r--', 'LineWidth', 1.5);
hold off;

title('Cumulative Distribution Function (CDF) of Total Scores',
'FontSize', 16, 'FontWeight', 'bold');
xlabel('Total Score', 'FontSize', 14, 'FontWeight', 'bold');
ylabel('Cumulative Probability', 'FontSize', 14, 'FontWeight',
'bold');
grid on;
legend({'Cumulative Distribution', 'Median Score Reference'},
'Location', 'best');
drawnow;
```

5.3.4 Horizontal Bar Chart of Average Scores by Department

```
close all;
clearvars -except T;
clc;

if ~iscategorical(T.Department)
T.Department = categorical(T.Department);
end

deptNames = categories(T.Department);
numDepts = numel(deptNames);
avgScores = zeros(1, numDepts);

for i = 1:numDepts
avgScores(i) = mean(T.Total_Score(T.Department == deptNames{i}));
end
```

```
figure('Position', [150, 150, 900, 600]);
barh(avgScores, 'FaceColor', [0.4 0.6 0.9]);

set(gca, 'YTick', 1:numDepts, 'YTickLabel', deptNames, 'FontSize',
12, 'FontWeight', 'bold');
title('Average Total Score by Department', 'FontSize', 16,
'FontWeight', 'bold');
xlabel('Average Total Score', 'FontSize', 14, 'FontWeight', 'bold');
ylabel('Department', 'FontSize', 14, 'FontWeight', 'bold');
grid on;
box on;

for i = 1:numDepts
text(avgScores(i) + 1, i, sprintf('%.2f', avgScores(i)), 'FontSize',
12, 'FontWeight', 'bold', 'Color', 'k');
end
drawnow;
```

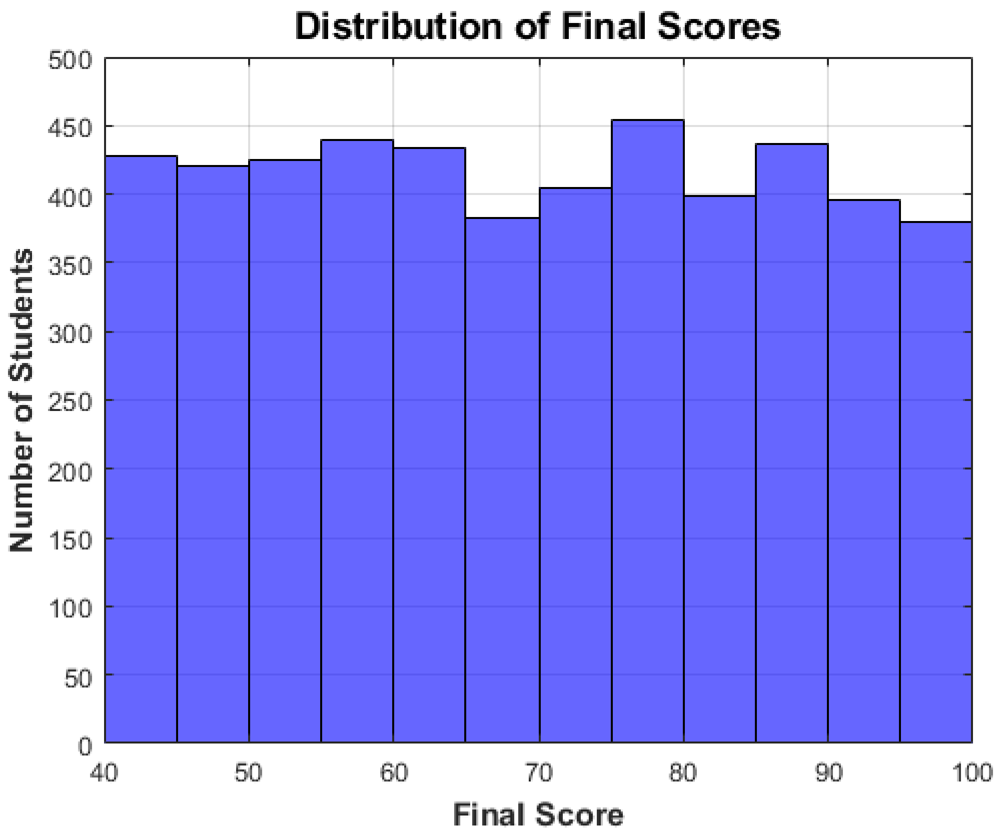
5.3.5 3D Surface Plot of Study Hours, Sleep Hours, and Total Score

```
close all;
clearvars -except T;
clc;

X = T.Study_Hours_per_Week;
Y = T.Sleep_Hours_per_Night;
Z = T.Total_Score;

validData = ~isnan(X) & ~isnan(Y) & ~isnan(Z);
X = X(validData);
Y = Y(validData);
Z = Z(validData);

[Xq, Yq] = meshgrid(linspace(min(X), max(X), 30), linspace(min(Y),
max(Y), 30));
Zq = griddata(X, Y, Z, Xq, Yq, 'cubic');

figure('Position', [150, 150, 900, 600]);

surf(Xq, Yq, Zq, 'EdgeColor', 'none', 'FaceAlpha', 0.8);
colormap jet;
colorbar;

title('3D Surface Plot of Study Hours, Sleep Hours, and Total Score',
'FontSize', 16, 'FontWeight', 'bold');
xlabel('Study Hours per Week', 'FontSize', 14, 'FontWeight', 'bold');
ylabel('Sleep Hours per Night', 'FontSize', 14, 'FontWeight',
'bold');
zlabel('Total Score', 'FontSize', 14, 'FontWeight', 'bold');
set(gca, 'FontSize', 12, 'FontWeight', 'bold');
grid on; box on; view(135, 30);
drawnow;
```

Output

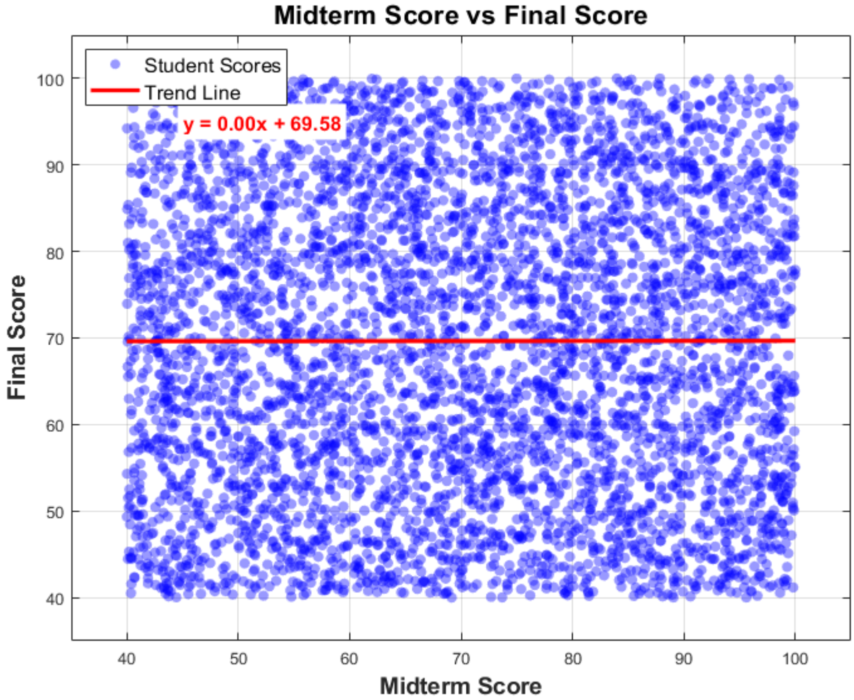
6.1 Basic Distribution:

Histogram of Final Score

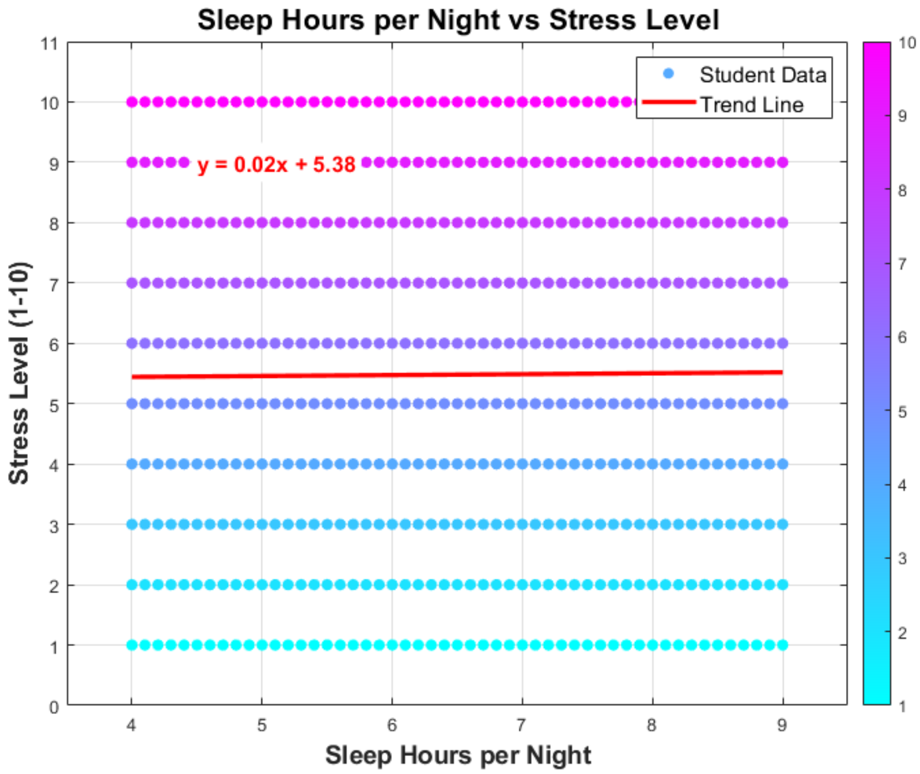


6.2 Performance comparison:

Scatter Plot: Midterm Score vs. Final



Correlation Graph: Sleep Hours per Night vs. Stress Level

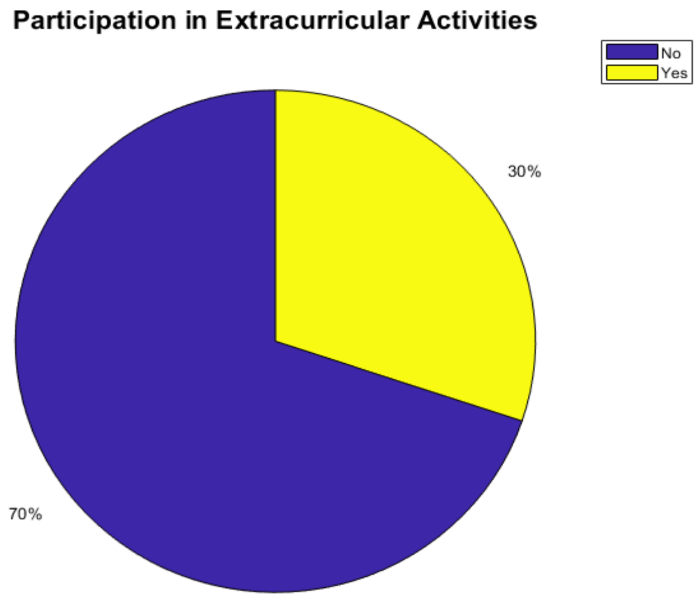


6.3 Categorical Data Representation:

Bar Graph: Distribution of Family Income Levels

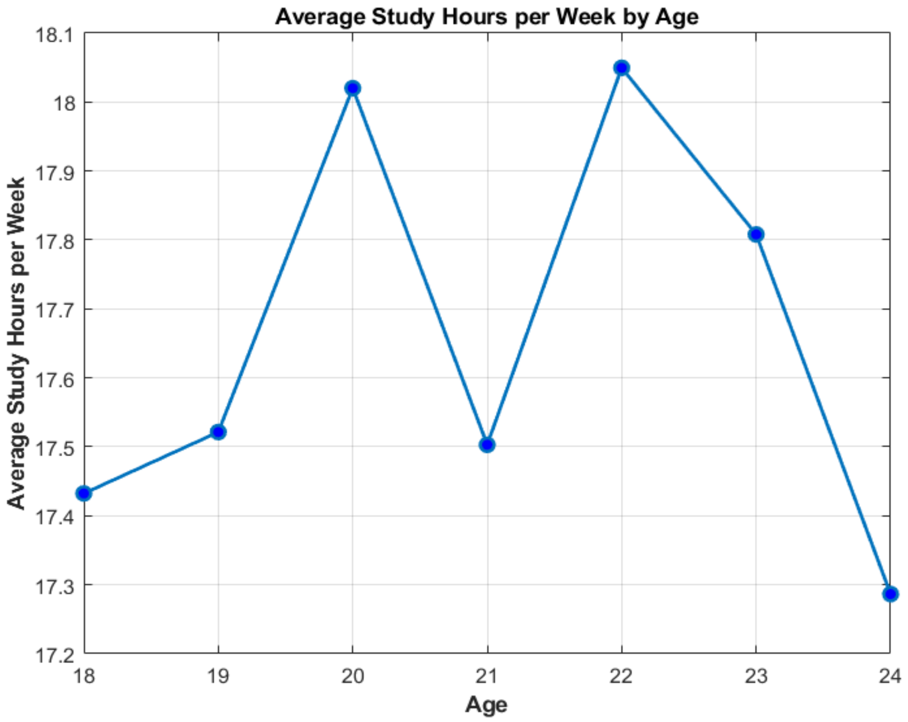
Box Plot of Stress Levels by Gender

Pie Chart of Extracurricular Activities Participation

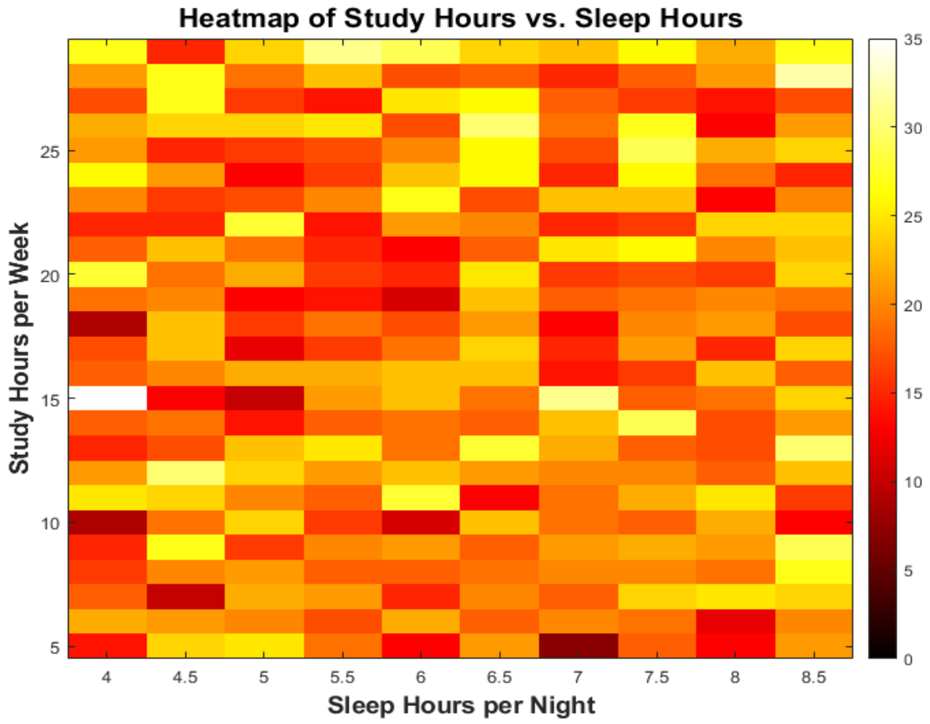


6.4 Identifying Key Performance Trends:

Line Plot of Average Study Hours vs. Age

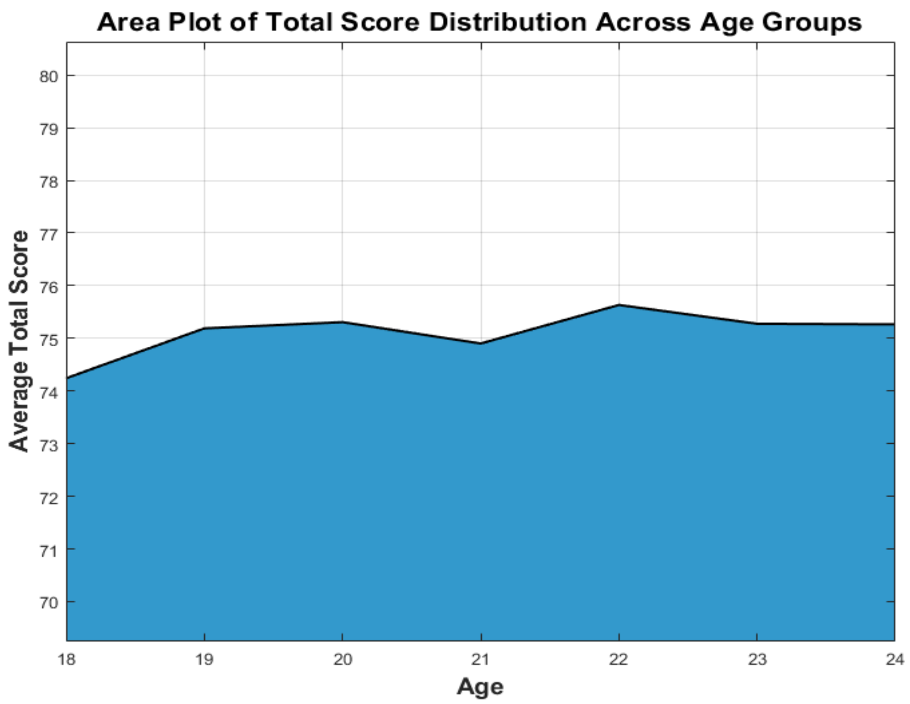


Heatmap of Study Hours vs. Sleep Hours

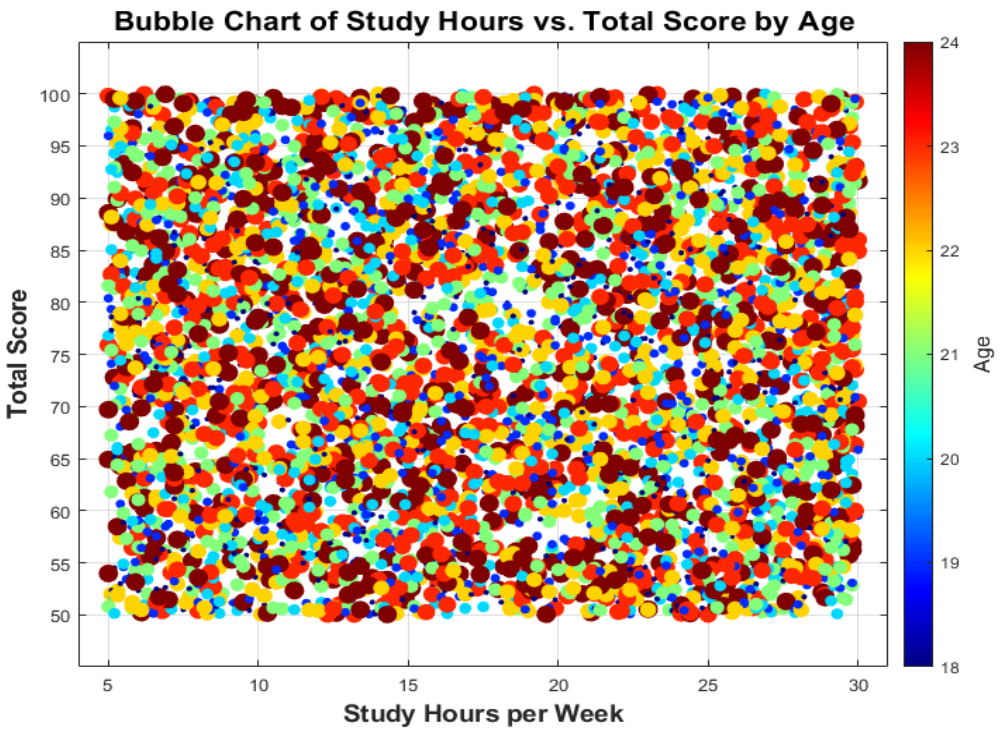


6.5 Advanced Data Representations

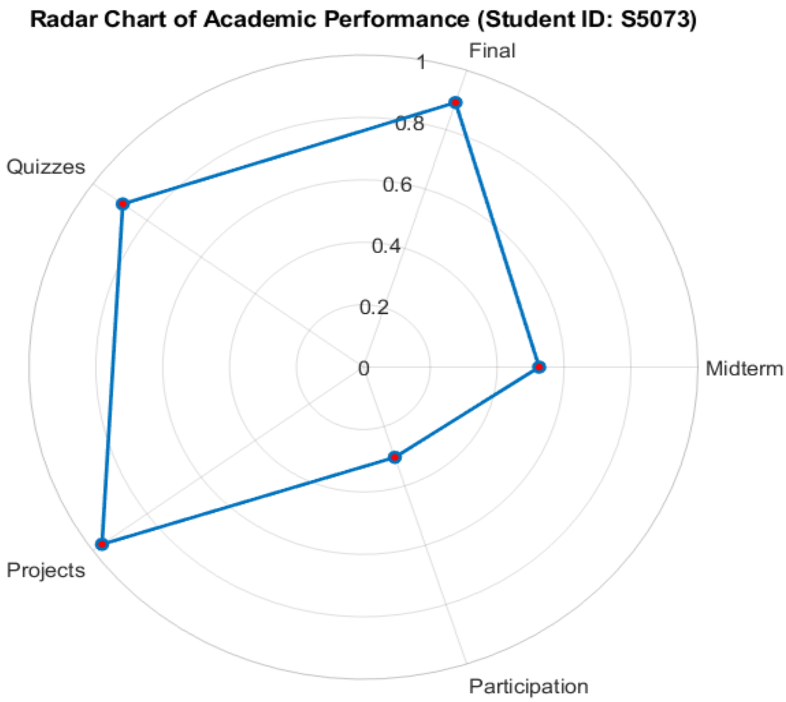
Area Plot of Total Score Distribution Across Age Groups



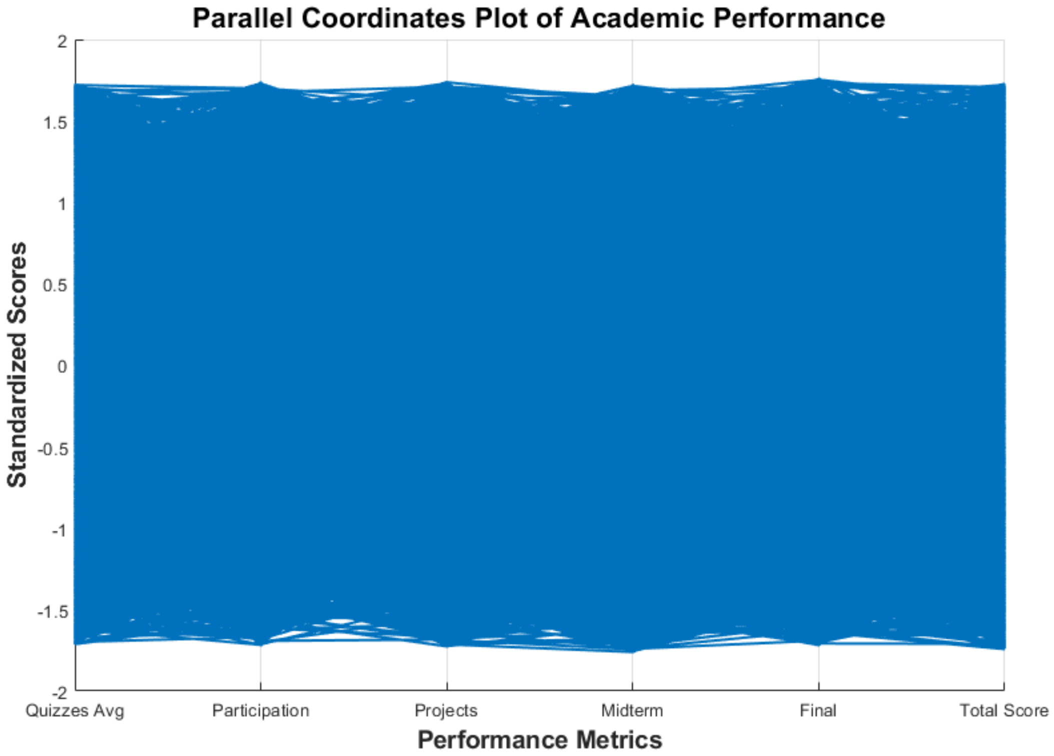
Bubble Chart of Study Hours vs. Total Score by Age



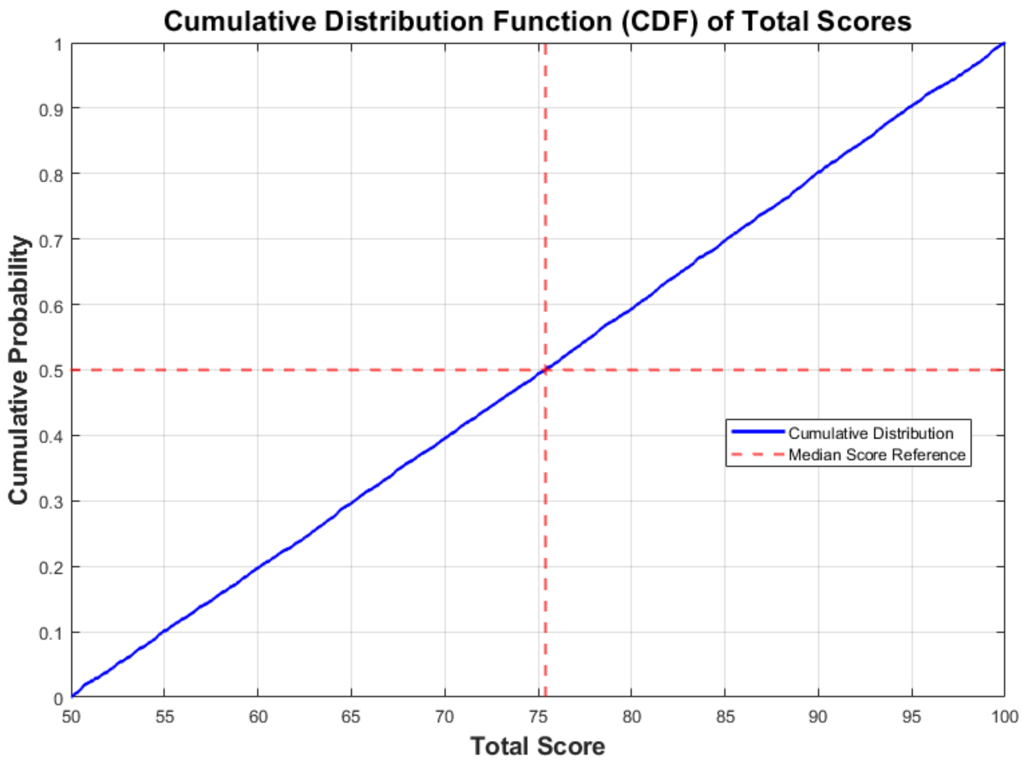
Radar Chart of Academic Performance Metrics



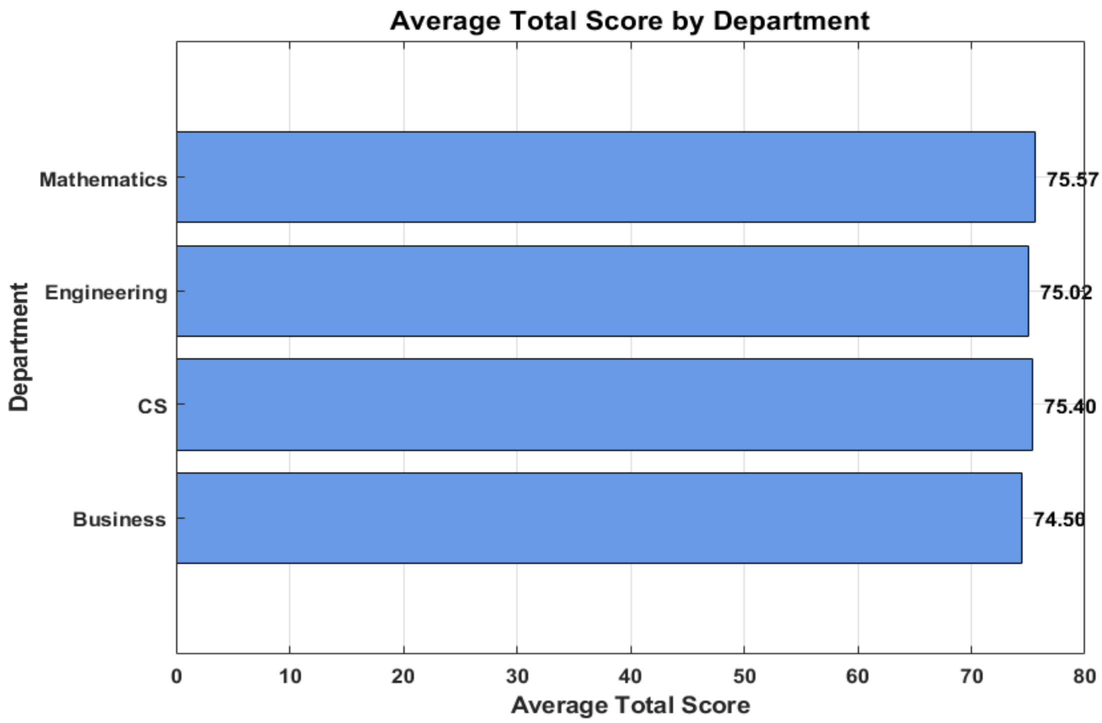
Parallel Coordinates Plot of Academic Performance



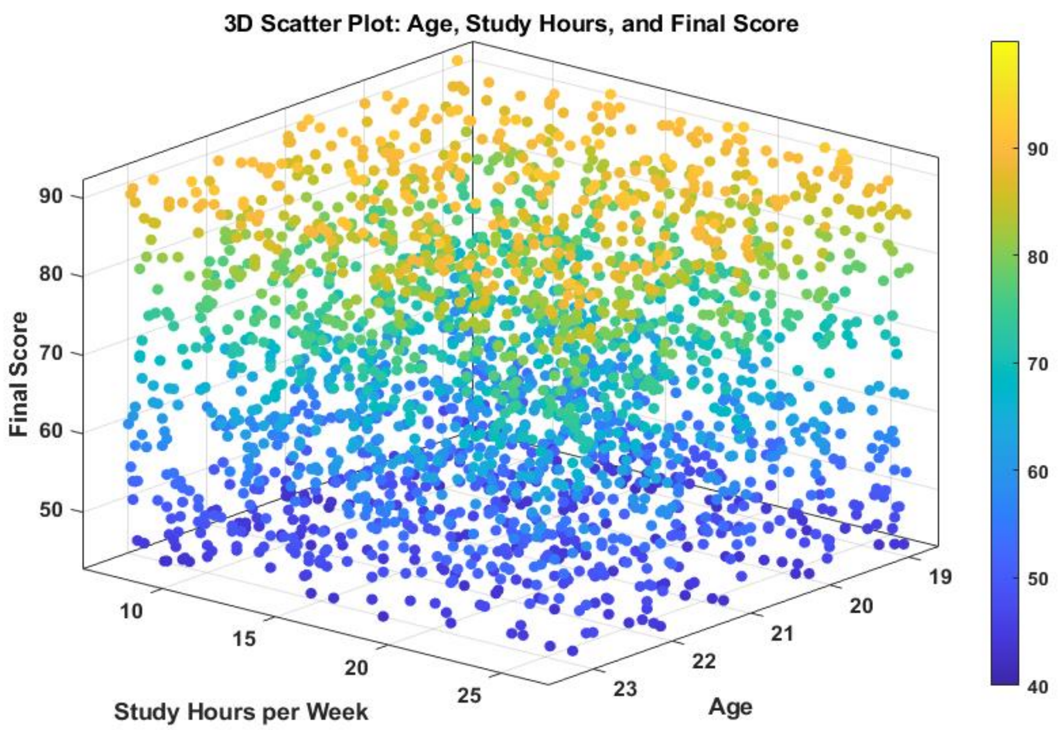
Cumulative Distribution Function (CDF) of Total Scores



Horizontal Bar Chart of Average Scores by Department



3D Surface Plot of Study Hours, Sleep Hours, and Total Score



Conclusion

This study explored the impact of academic, behavioral, and socioeconomic factors on student performance using **MATLAB-based data visualization techniques**. By analyzing various attributes such as **study habits, stress levels, sleep patterns, attendance, and parental education**, key trends were identified through histograms, scatter plots, bar charts, and heatmaps. The findings highlight that **study hours positively correlate with academic performance**, but excessive studying without adequate rest does not necessarily lead to better results.

High stress levels negatively impact scores, while students with **consistent sleep schedules (6-8 hours per night)** tend to perform better. **Regular class attendance plays a crucial role in student success**, reinforcing the importance of classroom engagement and participation. Socioeconomic factors, particularly **parental education levels**, showed some influence on academic achievements, though individual study habits had a stronger impact on overall performance.

To improve academic outcomes, students should focus on **effective study strategies, stress management techniques, and structured time management** rather than just increasing study hours. Educators should implement **data-driven teaching approaches, academic support programs, and student well-being initiatives** to optimize learning environments. Institutions can use **real-time performance tracking and predictive analytics** to provide personalized learning interventions for struggling students.

While this study provides valuable insights, it is limited by the lack of causal analysis and the exclusion of external influences such as **teacher effectiveness, course difficulty, and motivation levels**. Future research should incorporate **machine learning models for academic performance prediction**, **multi-institutional datasets** for broader validation, and **real-time student analytics dashboards** to assist both educators and students in tracking progress.

The results demonstrate the importance of **data-driven decision-making in education**. By leveraging visualization techniques, **students, educators, and policymakers** can make informed academic choices, identify learning gaps, and develop strategies that foster student success and well-being. Data analytics in education has the potential to enhance teaching methods, improve student engagement, and create a more effective learning environment for future generations.