INTRODUCTION TO PROGRAMMING FOR DATA SCIENCE – TERM PROJECT

Project Title: Student Performance and Participation Analysis System

Project Information:

Course: Introduction to Programming for Data Science

Instructor: Asst. Prof. Hatice NİZAM ÖZOĞUR

Term: 2025-2026/Fall

Group Formation and Submission: November 17, 2025.

Submission Deadline: December 29, 2025

Presentation Date: December 29, 2025

Submission Format:

- Jupyter Notebook (.ipynb)
- Data files (.csv)
- Presentation file (.pptx or .pdf)

Project Scenario:

A teacher wants to analyze their students' performance in **Mathematics**, **Physics**, and **Chemistry**, together with their **class participation**, **attendance**, and **assignment performance**.

Project Objectives:

1. Calculate overall class performance levels
2. Analyze performance differences between subjects
3. Examine the relationship between participation, attendance, and achievement
4. Generate personalized improvement suggestions for students
5. Visualize findings using graphs and tables
6. Summarize results in a structured report
7. Present findings in class as a group presentation

Dataset Requirements:

- Data must be in **CSV format**
- The dataset must include at least **50 students**
- Each student should have the following information:

Name	Mathematics	Physics	Chemistry	Participation	Absence	Assignment
Ahmet Yılmaz	78	65	82	8	4	70

- The dataset may be synthetic (manually or randomly generated) but must be consistent.
- Missing or incorrect values may exist and should be handled during data cleaning.

Project Steps and Analysis Requirements

1. Data Loading and Cleaning

- Load the CSV dataset into the analysis environment (e.g., using pandas).
- Check data types and identify missing or incorrect values.
- Handle missing data appropriately (mean, median, or deletion).
- Summarize dataset size, structure, and completeness before analysis.

2. Basic Statistical Analysis

- Calculate average, minimum, maximum, median, and standard deviation for each subject.
- Compute average participation, attendance, and assignment scores.
- Present findings in table form and provide short comments.
- Compare general performance averages across subjects.

3. Student-Based Analysis

- Calculate the overall average grade for each student.
- Classify students into performance levels:
  - Excellent
  - Good
  - Average
  - Poor
- Provide short improvement suggestions based on participation, attendance, and assignment performance.
- Create a summary table showing each student's overall performance and category.

4. Subject-Based Analysis

- Compare subject averages, minimums, and maximums.
- Identify which subject is most and least challenging.
- Present subject-level success rates with clear visualizations.

5. Advanced Analysis and Correlation

- Examine relationships between participation, assignments, absences, and grades.
- Calculate correlations between all key variables and summarize them in a table.

- Visualize correlation results with a correlation matrix or heatmap.
- Comment on relationships (e.g., "Participation shows a positive correlation with performance").

6. Student Comparisons

- Identify the top-performing and lowest-performing students.
- Detect students with low participation or poor assignment scores.
- Compare students' performances across different subjects in table format.

7. Data Structures and Function Usage

- Effectively use **lists**, **tuples**, **dictionaries**, and **sets** in the analysis.
- Convert repetitive tasks into modular **functions** for cleaner structure.
- Include explanations and documentation for each part of the analysis.

8. Visualization

Findings must be presented through clear and meaningful graphs. Recommended chart types include:

Chart Type	Purpose
Bar Chart	Subject averages or performance levels
Pie Chart	Distribution of performance categories
Line Chart	Attendance or assignment trends
Scatter Plot	Relationship between participation and success
Heatmap	Correlation between factors

Charts must include clear titles, axis labels, and appropriate color schemes.

9. File Operations

- Save analysis results into a new CSV file.
- Create and export summary tables for individual students if needed.
- Use descriptive and organized file names.

10. Algorithmic Problem Solving

- Develop an algorithm to identify students with low participation or assignments.
- Determine the lowest scores per subject and highlight weak areas.
- Calculate the percentage distribution of performance categories within the class.
- Present algorithmic findings in table or graphical form.

11. Reporting

- Include explanations for all code outputs and visuals.
- Summarize subject-based and student-based findings clearly.
- Discuss overall performance, key observations, and improvement suggestions.
- Write the report in a clear, academic tone with structured sections.

Presentation Requirements:

⚠️ **Projects without presentations will receive a score of 0 (zero).**

Presentation Duration: 5–7 minutes

Presentation Format: PowerPoint or PDF

Required Content:

1. Project aim and scenario explanation
2. Dataset introduction (sample rows may be shown)
3. Summary of basic and advanced analyses
4. Graphical visualizations (bar, pie, line, scatter, etc.)
5. Example of algorithmic problem solving
6. Subject-based and student-based discussion
7. Overall conclusions and recommendations

- Each group member must participate actively in the presentation.
- Presentations should be professional, concise, and data-driven.

Group Formation Rules:

- Each group must consist of 3 or 4 members.
- Groups must be reported to the instructor no later than **November 17, 2025**.
- The group information should be sent via email to the instructor.

Email Instructions:

- To: emre.durusoy@istanbul.edu.tr
- CC: hatice.nizamozogur@istanbul.edu.tr
- Subject Line: Group Formation – Introduction to Programming for Data Science
- Email Body must include:
    - Group name
    - Student names and ID numbers
    - Group representative's email address

Important:

- Only the **group representative** is required to send the email on behalf of the team.

- Groups that are **not reported by the deadline** will **not be evaluated** and will receive **no grade** for the project.

Submission Format:

File Type	Description
.ipynb	Jupyter Notebook (complete with code, analysis, and visualizations)
.csv	Original and analyzed dataset files
.pptx / .pdf	Presentation file
(Optional) .docx	Project summary or group notes

All files must be organized as follows:

```
Project_GroupName/
│
├── data/
│   ├── original_data.csv
│   └── analysis_results.csv
│
├── notebook/
│   └── StudentPerformanceAnalysis.ipynb
│
├── presentation/
│   └── GroupPresentation.pdf or GroupPresentation.pptx
│
└── README.docx
```

Important Notes:

- Each step must include **executable code**; written explanations alone are not sufficient.
- All code must include **clear comments and explanations** describing the logic and purpose of each part.
- Projects that are **identical, copied, or show signs of plagiarism** will be **disqualified**.
- **Code or analyses generated using Large Language Models (LLMs)** (e.g., ChatGPT, Copilot, Gemini, etc.) **will not be evaluated** and will result in a **zero score** for the project.
- Each student must **explain their individual contribution** during the presentation.
- Projects **without a presentation** will receive **0 points**, regardless of submission.