# Assignment — Data Analysis & Report using Pandas, NumPy, Matplotlib

Difficulty: Beginner / Intermediate — friendly language and hints, no full answers.

## Dataset

Use the Students Performance in Exams dataset from Kaggle. Download the CSV and use it for every task below. (Kaggle page has the file StudentsPerformance.csv — download and open in your notebook.)

## Learning Goals

- Practice loading and inspecting data with Pandas  
- Use NumPy for simple numeric work (arrays, statistics)  
- Create clear charts with Matplotlib  
- Write a short data-analysis report describing findings  
- Learn how to clean data, compute simple statistics, and make visual interpretations

## Instructions

- Use a Jupyter Notebook (.ipynb) or Google Colab.  
- Include code cells and short text explanations (1–2 sentences) for each result.  
- Also submit a 1–2 page PDF or Markdown report summarizing main findings (charts + short conclusions).  
- Do not copy answers from others — write your own observations.

## Tasks (with hints, not answers)

1. 1) Load & inspect the data
2. 2) Basic cleaning
3. 3) Summary statistics
4. 4) Use NumPy for a simple task
5. 5) Grouped analysis (Pandas)
6. 6) Correlation & relationships
7. 7) Categorical counts & bar charts
8. 8) Distribution plots
9. 9) Create one derived insight (short mini-analysis)
10. 10) Short predictive exercise (optional, simple)

## Report Structure

1. Title and one-line objective.  
2. Short description of dataset (rows, columns).  
3. Major cleaning steps you did (1–2 sentences).  
4. Key charts (include 3–4 images: e.g., histogram, bar chart, scatter, boxplot).  
5. One short “insight” paragraph (what did you find?).  
6. Limitations (1–2 lines — e.g., dataset size, no causal claims).  
7. Optional next steps (1–2 lines — e.g., build a classifier).

## Grading Rubric

- Data loading & cleaning: 10%  
- Correct use of Pandas & NumPy: 25%  
- Plots that are readable & labeled: 25%  
- Clear short report + insight: 30%  
- Bonus (optional prediction or extra analysis): 10%

## Tips & Good Practices

- Always check df.info() and df.head() first.  
- Label your plots: title, x-label, y-label, and a legend if needed.  
- When grouping, sort results so charts are easy to read.  
- Use plt.tight\_layout() after subplots so labels don’t overlap.  
- Save important plots with plt.savefig('plot.png') for the report.  
- Keep explanations short and in your own words — say what you did and what you saw.