Assessment 3

UTS
UNIVERSITY OF TECHNOLOGY SYDNEY

43031 - Python Programming for Data Processing

Students Grading Analysis

Somesh
Shanbhag

25525837

kaggle colab

Table of Contents

Background and Problem Statement

A private learning provider collected data on 5,000 students to understand factors influencing academic performance. However, the raw dataset suffered from missing values, duplicates, inconsistent formatting and potential outliers. These quality issues can bias analyses and lead to unreliable conclusions. The goal of this project is to design and implement robust data pre-processing and visualization strategies using python to:

1. Clean and standardize the data for analysis.
2. Resolve missing, duplicate and outlier problems.
3. Answer atleast 5 business questions, three of which are complex through appropriate visualization.
4. Examine correlations between key attributes.

By following a clear, systematic approach, this report demonstrates end-to-end data preparation and visual storytelling. The insghts will guide stakeholders in improving student support programs and teaching strategies.

1. Data Pre-Processing

1.1 Initialisation and Setup

- **Libraries**: We rely on Pandas and NumPy for data manipulation and matplotlib and seaborn for plots.

```
#Step 1: Import the Pandas and NumPy Library

import pandas as pd
import numpy as np
# Import the MatPlotLib and the Seaborn library
import matplotlib.pyplot as plt
import seaborn as sns # Importing Seaborn library

print("Libraries imported successfully!")
```

```
Libraries imported successfully!
```

- **Exception handling:** every I/O operation and transformation is wrapped in try-except blocks to catch and report errors.

- **Accessing dataset**: adding access from Google collab to Google Drive in order to successfully access the data set and begin data preprocessing and analysis this was done by setting up path and Reading the reading the data set in CSV format.

```python
# Step 3: Read the Kaggle dataset (csv) using read_csv function in Pandas

dataset_path = '/content/drive/MyDrive/Python Class/Students_Grading_Dataset.csv' #Path from drive set

try:
      df = pd.read_csv(dataset_path, header=0)
      print("CSV file loaded successfully!")

except Exception as e:
      print(f"An unexpected error occurred: {e}") #Easier to understand the error
```

```
CSV file loaded successfully!
```

1.2 Missing Values

- **Audit:** df.isnull().sum() identified 515 missing attendance (%), 503 missing assignment Score (Averaged) and 1800 missing Parent Education Level.
- **Analysis:** Missing count percentages were computed by department to confirm uniform impact (<12% across Attendance and Assignment Score when group by department and around 52% - 58% for parent education level when grouped by Grade.)
- **Imputation Strategy:** Missing values were evenly spread across departments and grades, and imputation would therefore not influence the result.

```
Percentage of missing values in Attendance (%) by department:
Department
Business       10.69
CS              9.76
Engineering    10.15
Mathematics    12.16
Name: Attendance (%), dtype: float64

Missing values in Assignments Score (Averaged) by department:
Department
Business       123
CS             203
Engineering    125
Mathematics     52
Name: Assignments Score (Averaged), dtype: int64

Percentage of missing values in Assignments Score (Averaged) by department:
Department
Business       11.85
CS             10.21
Engineering     8.35
Mathematics    10.90
Name: Assignments Score (Averaged), dtype: float64
```

```
Percentage of missing values in 'Parent Education Level': 36.00%

Percentage of missing 'Parent Education Level' by Grade:
```

Grade	Parent Education Level
A	57.009346
B	55.694228
C	56.804734
D	52.816901
F	58.733205

dtype: float64

We imputed with the mean for Attendance (%) and Assignment Score, and the mode for Parent Education Level. This prevents bias and keeps the data complete without introducing new outliers.

Before:

```
Columns with Data missing:
Student ID                      0
First Name                      0
Last Name                       0
Email                           0
Gender                          0
Age                             0
Department                      0
Attendance (%)                515
Midterm Score                   0
Final Score                     0
Assignments Score (Averaged)  503
Quizzes Score (Averaged)        0
Participation Score             0
Projects Score                  0
Total Score                     0
Grade                           0
Study Hours per Week            0
Extracurricular Activities      0
Internet Access at Home         0
Parent Education Level       1800
Family Income Level             0
Stress Level (1-10)             0
Sleep Hours per Night           0
dtype: int64
```

After:

```
Columns with Data missing:
Student ID                      0
First Name                      0
Last Name                       0
Email                           0
Gender                          0
Age                             0
Department                      0
Attendance (%)                  0
Midterm Score                   0
Final Score                     0
Assignments Score (Averaged)    0
Quizzes Score (Averaged)        0
Participation Score             0
Projects Score                  0
Total Score                     0
Grade                           0
Study Hours per Week            0
Extracurricular Activities      0
Internet Access at Home         0
Parent Education Level          0
Family Income Level             0
Stress Level (1-10)             0
Sleep Hours per Night           0
```

1.3 Duplicate Records

- **Detection**: df.duplicated().sum() found 757 duplicate records.

```
Number of duplicates: 757
```

- **Impact Analysis**: While checking the dataset, it was found that all duplicates were complete row duplicates. This means no special correction was needed, and the entire duplicate rows could be removed. Removing them is important because keeping duplicates can affect the accuracy of statistics like averages and proportions, and also make the sample size look bigger than it really is.
- **Removal**: Duplicates were removed using df.drop_duplicates(inplace=True). We logged the count before and after to ensure correctness.

```python
#step 9.3: Drop the duplicate records

try:
  print("Number of duplicates:", df.duplicated().sum())
  df.drop_duplicates(inplace=True)
  print("Number of duplicates after dropping:", df.duplicated().sum())
  print("Duplicate records removed successfully!")
  print("Number of rows after dropping duplicates:", df.shape[0])
except Exception as e:
    print(f"An unexpected error occurred: {e}")
```

```
Number of duplicates: 757
Number of duplicates after dropping: 0
Duplicate records removed successfully!
Number of rows after dropping duplicates: 4243
```

- **Preventive Measures:** In a real-word setting, it would be prudent to implement a check at data entry or ingestion level that can enforce unique IDs.

1.4 Data Types and Outliers

- **Data Type Validation**: Each of the column data types were validated using the df.types statement and there were no incorrect datatypes.
- **Outlier Detection**: Interquartile Range Method was used to manually calculate the upper and lower bound for each numeric variable. An example boxplot was plotted to visualize the distribution. There were no outliers values present in the upper bond and lower bound of the IQR.

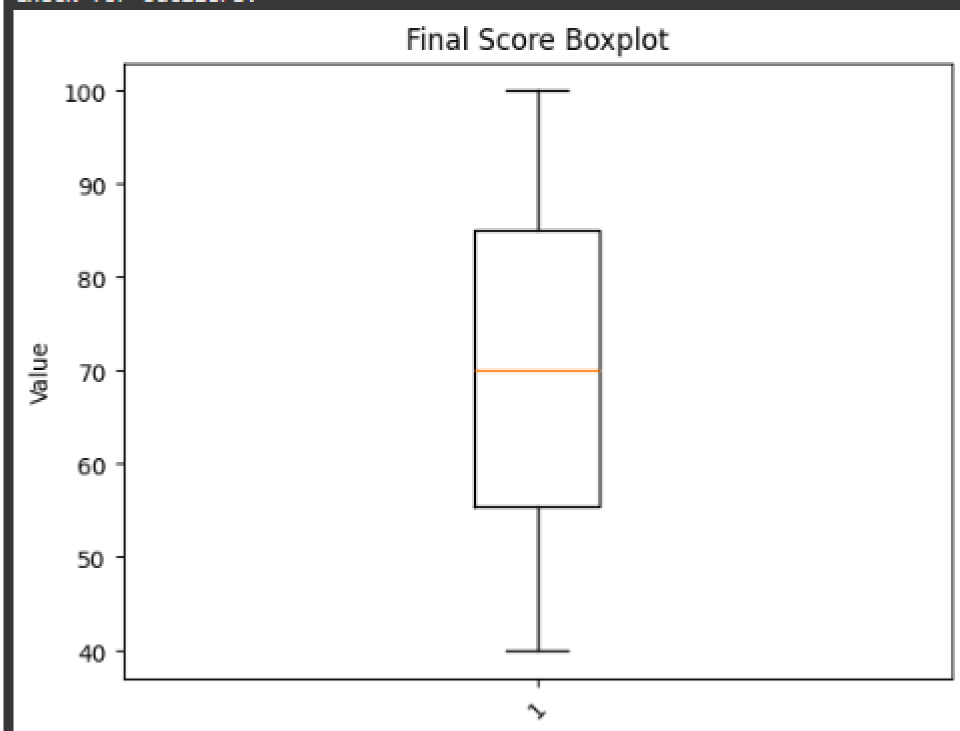
```
# Step 9: Identify the quality issues in the dataset to provide a comprehensive overview of its integrity and completeness.
try:
    print("Columns with Data missing:")
    print(df.isnull().sum())
    print("\nNumber of duplicates:", df.duplicated().sum())
    print("\nCheck for Columns with incorrect data types:")
    print(df.dtypes)
    print("\nCheck for outliers:")
    plt.boxplot(df['Final Score'])
    plt.title('Final Score Boxplot')
    plt.ylabel('Value')
    plt.xticks(rotation=45)  # Rotate x-axis labels for better readability
    plt.show()
    print("\nSimiliarly.....Checking for Outliers in other numeric fields:\n")
    for col in df.select_dtypes(include=["number"]).columns:
      Q1 = np.percentile(df[col], 25)
      Q3 = np.percentile(df[col], 75)
      IQR = Q3 - Q1
      lower_bound = Q1 - 1.5 * IQR
      upper_bound = Q3 + 1.5 * IQR

      # Identify outliers
      outliers = df[(df[col] < lower_bound) | (df[col] > upper_bound)]

      # Print the number of outliers for the current column
      print(f"Number of outliers in '{col}': {len(outliers)}")

except Exception as e:
    print(f"An unexpected error occurred: {e}")
```

```
Check for outliers:
```




```
Similiarly.....Checking for Outliers in other numeric fields:

Number of outliers in 'Age': 0
Number of outliers in 'Attendance (%)': 0
Number of outliers in 'Midterm Score': 0
Number of outliers in 'Final Score': 0
Number of outliers in 'Assignments Score (Averaged)': 0
Number of outliers in 'Quizzes Score (Averaged)': 0
Number of outliers in 'Participation Score': 0
Number of outliers in 'Projects Score': 0
Number of outliers in 'Total Score': 0
Number of outliers in 'Study Hours per Week': 0
Number of outliers in 'Stress Level (1-10)': 0
Number of outliers in 'Sleep Hours per Night': 0
```

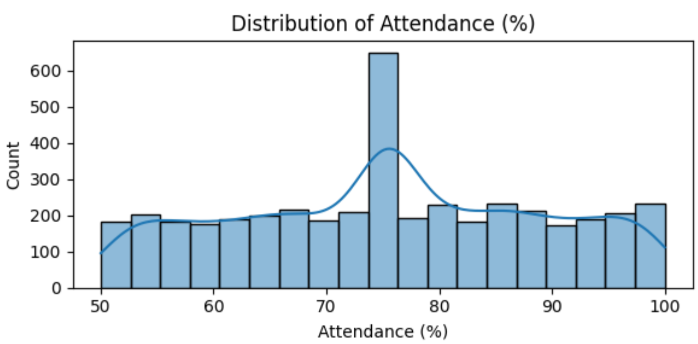
- **Treatment**: No Outliers were removed allowing for the complete preservation and the full range of student Behaviour and performance.

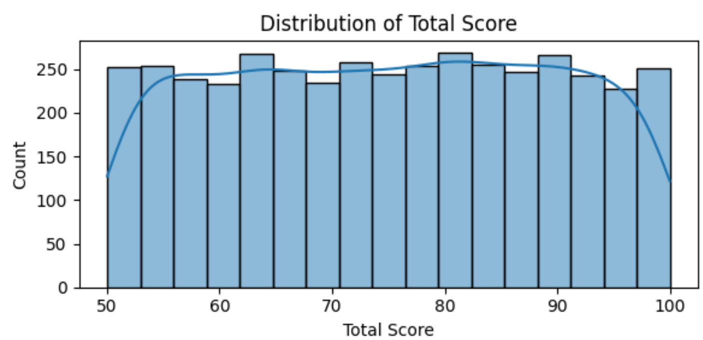
2. Data visualization and Analysis

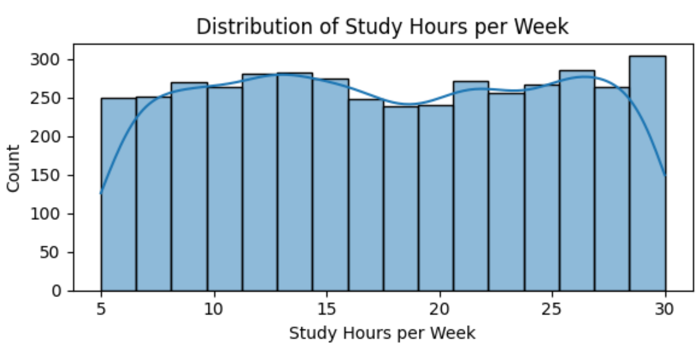
Q1. Distribution of Key Metrics (Univariate)

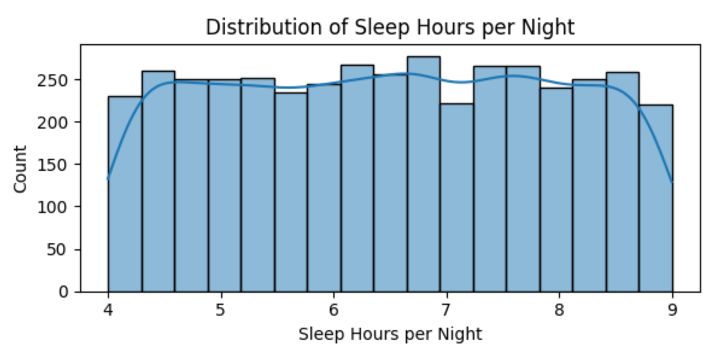
Question: What is the distribution shape for key numerical metrics such as attendance, Total Score, Study Hours per week and Sleep Hours per night?

Answer: Attendance and overall scores peak at 75–80%, indicating most students perform consistently. Study hours are skewed to the right: most study 10–15 hours, some very few more than 25. Sleep takes a bell-curve shape, where most get 6–8 hours, and very few less than 5. All these habits indicate that consistent attendance, moderate study, and normal sleep maintain good performance.



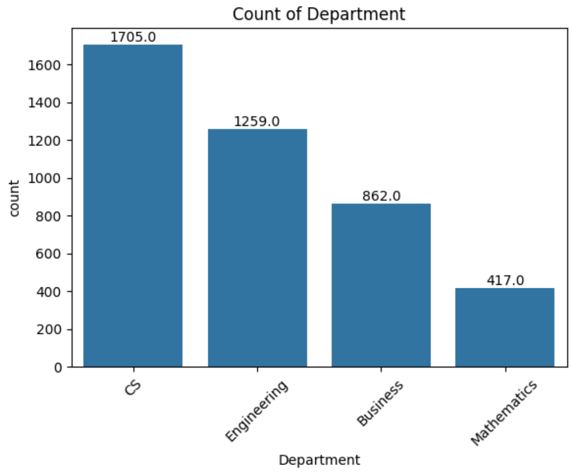


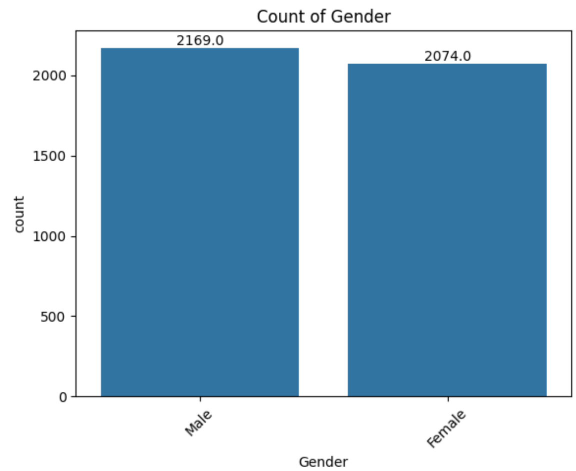


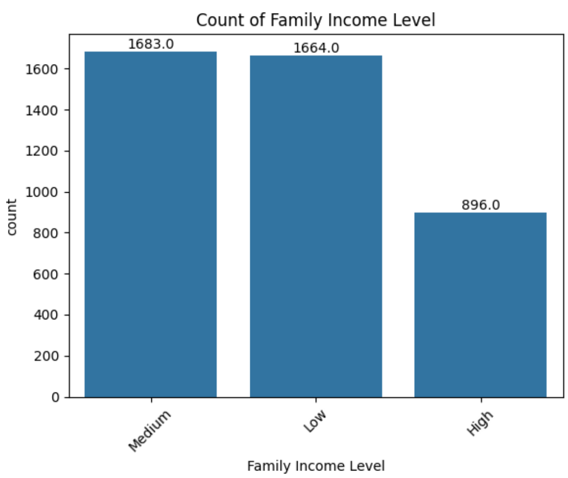


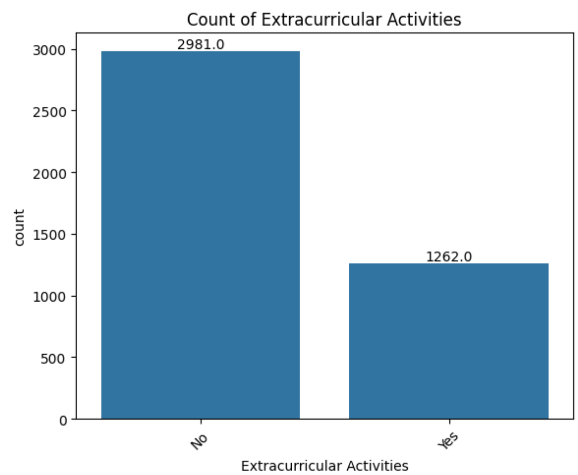
Q2. Student demographics (Categorical Counts)

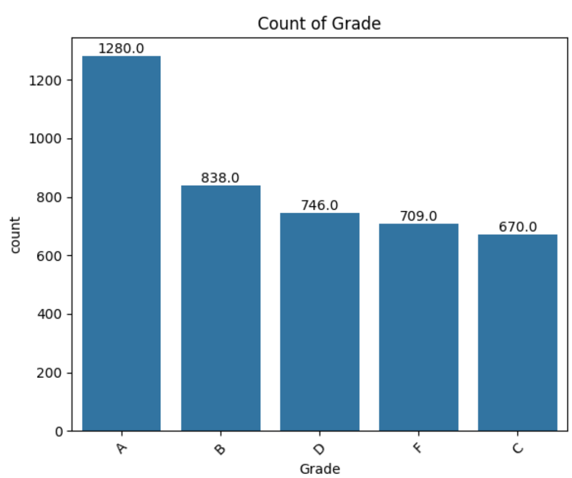
Question: How do counts differ by gender, department, grade, extracurricular participation, and family income?







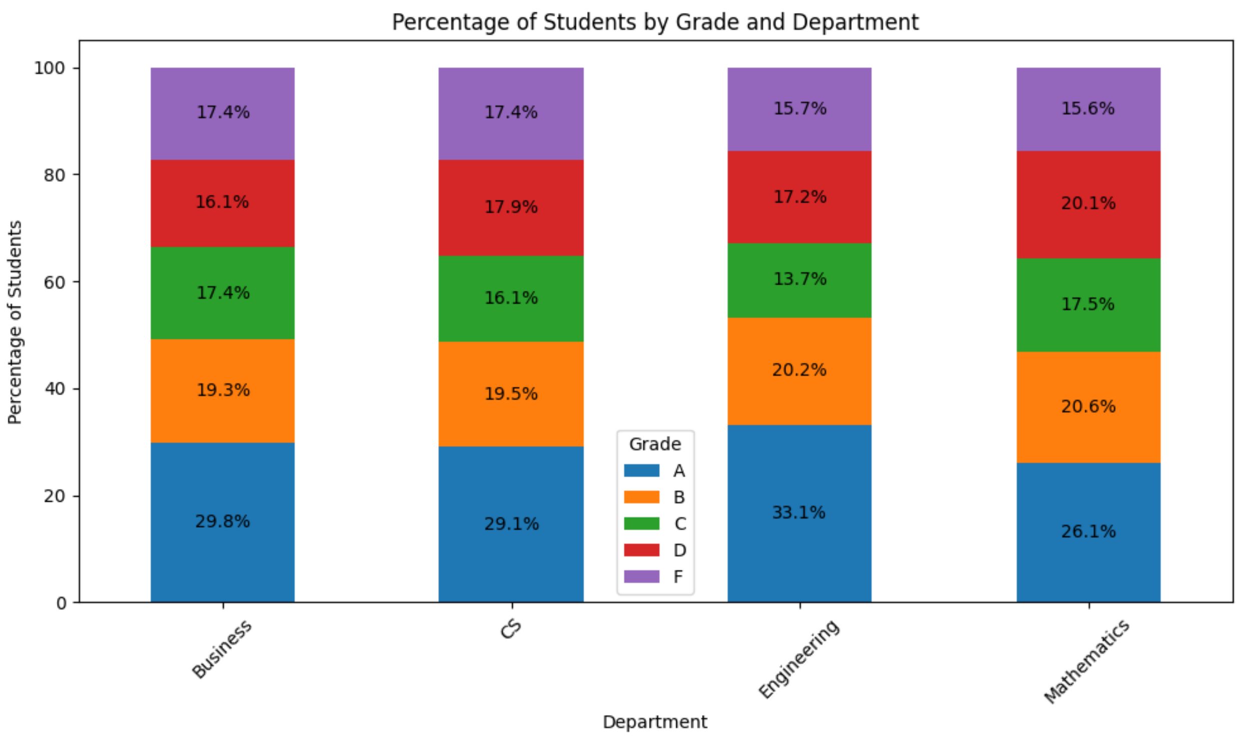




Inferences from the above 5 bar charts used to understand categorical data in the dataset are Gender is roughly even, with a slight majority of male students. Computer Science attracts the most students, while Mathematics has the fewest. Most students earn high grades, with fewer falling into the middle ranges. Participation in extracurricular activities is low, suggesting barriers to engagement. Finally, the majority of students come from lower- or middle-income families, which may affect their access to resources and opportunities.

Q3. Percentage of Students by Grade and Department

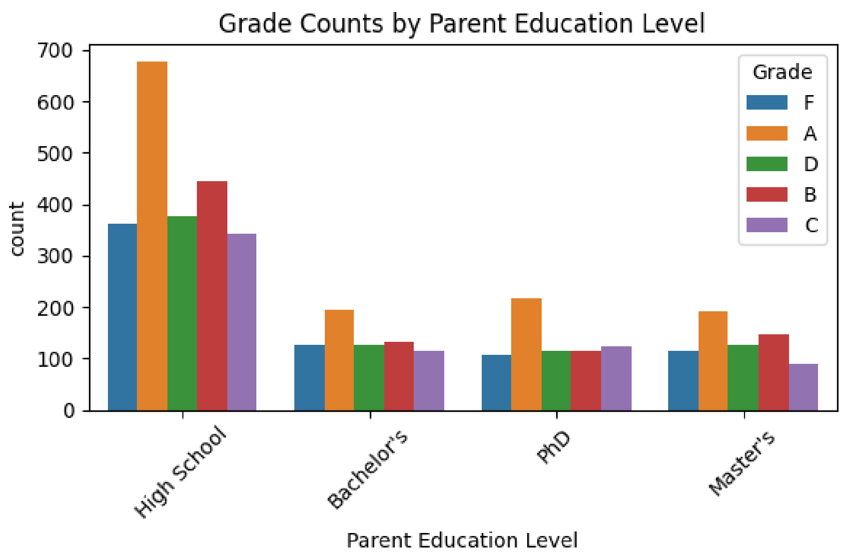
Question: Which department has the highest share of A grades, and which has the largest proportion of D grades?



The chart shows how students in each school get letter grades, with each bar totaling 100%. In Engineering, about one third of the students get an A, the highest number, and fewer students get lower grades. Business and Computer Science are very close to each other: about 30% A's, 20% B's, and the rest evenly spread from C to F. Math has the lowest A's (around 26%) and highest D's (around 20%), suggesting it could be tougher for students. Engineering students do best overall, Math students struggle the most, and Business and CS are in the middle.

Q4. Grade Count by Parent Education Level

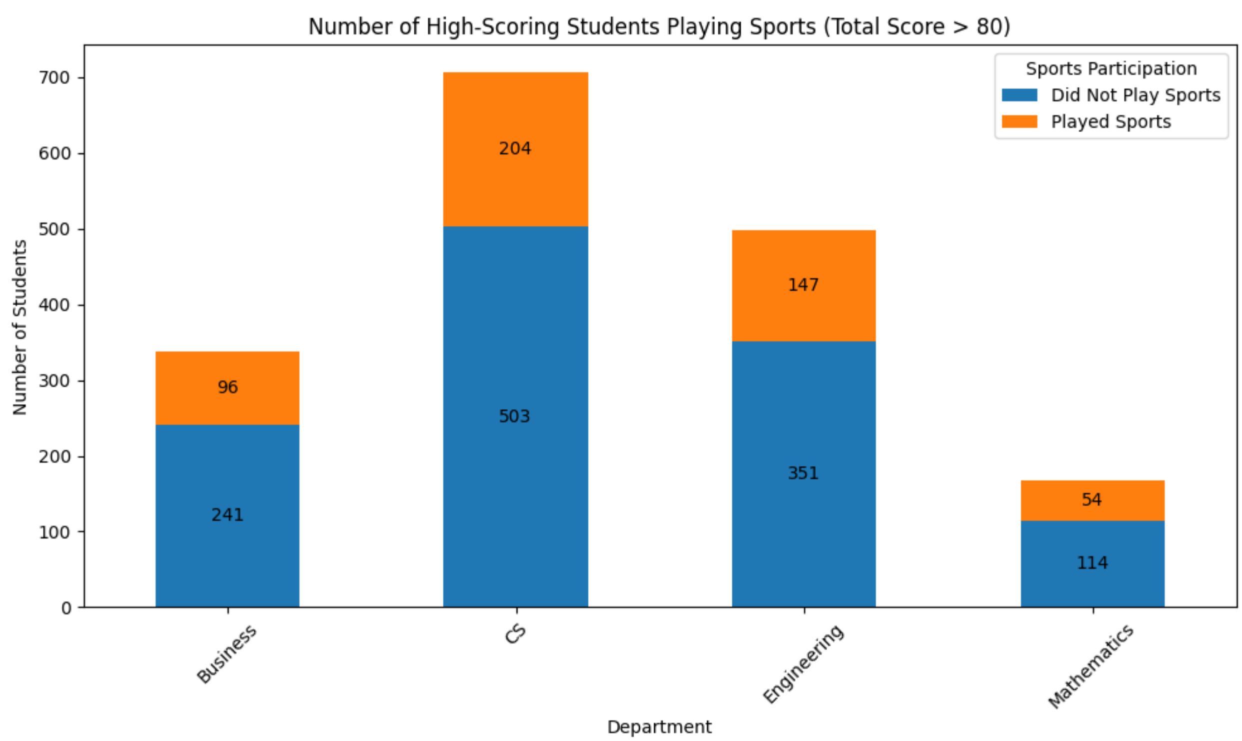
Question: How does parental education level relate to student's final letter grades?



Answer:
The bar plot shows grade distributions by parent education (High School, Bachelor's, Master's, PhD), with each letter grade shown in a different color. Students with parents who have attended only high school have the highest proportion of As, whereas students with Bachelor's, Master's, or PhD backgrounds have a more balanced set of grades. This would suggest that students with high-school-educated parents may be especially driven to earn high grades.

Q5. Number of High Scoring Students Playing Sports

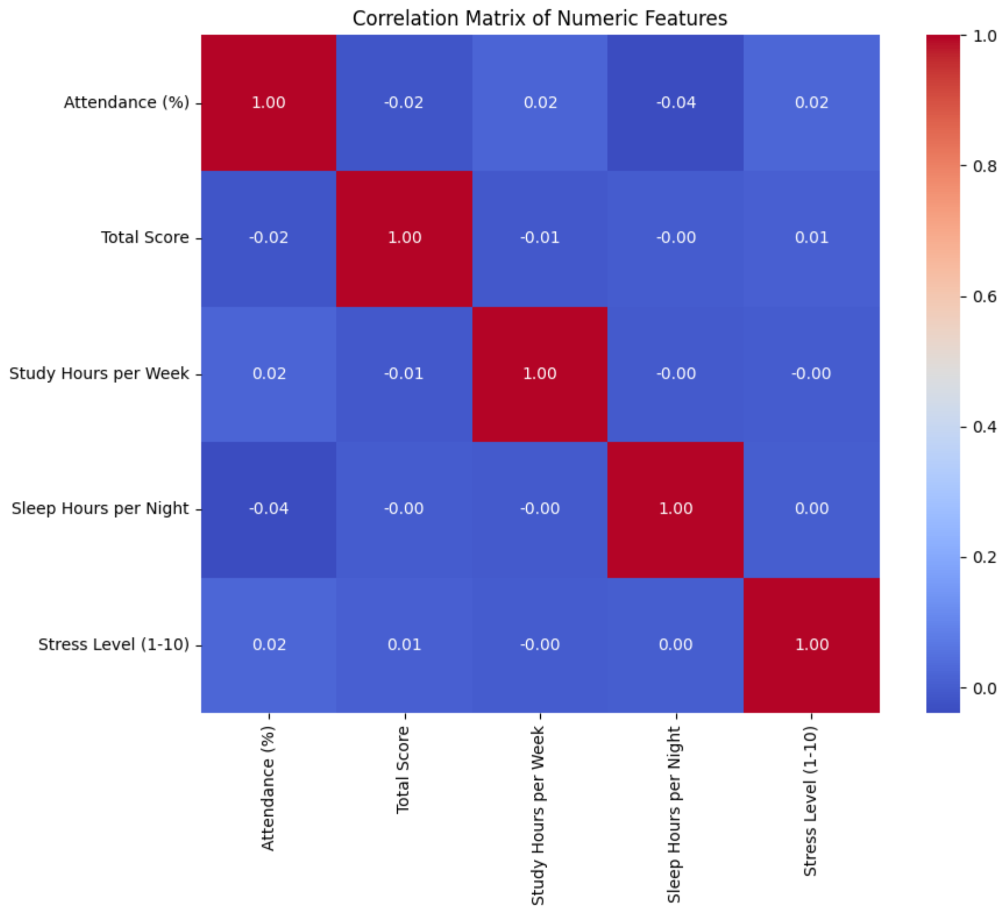
Question: Which department has the highest number of high-scoring students who participate in sports?



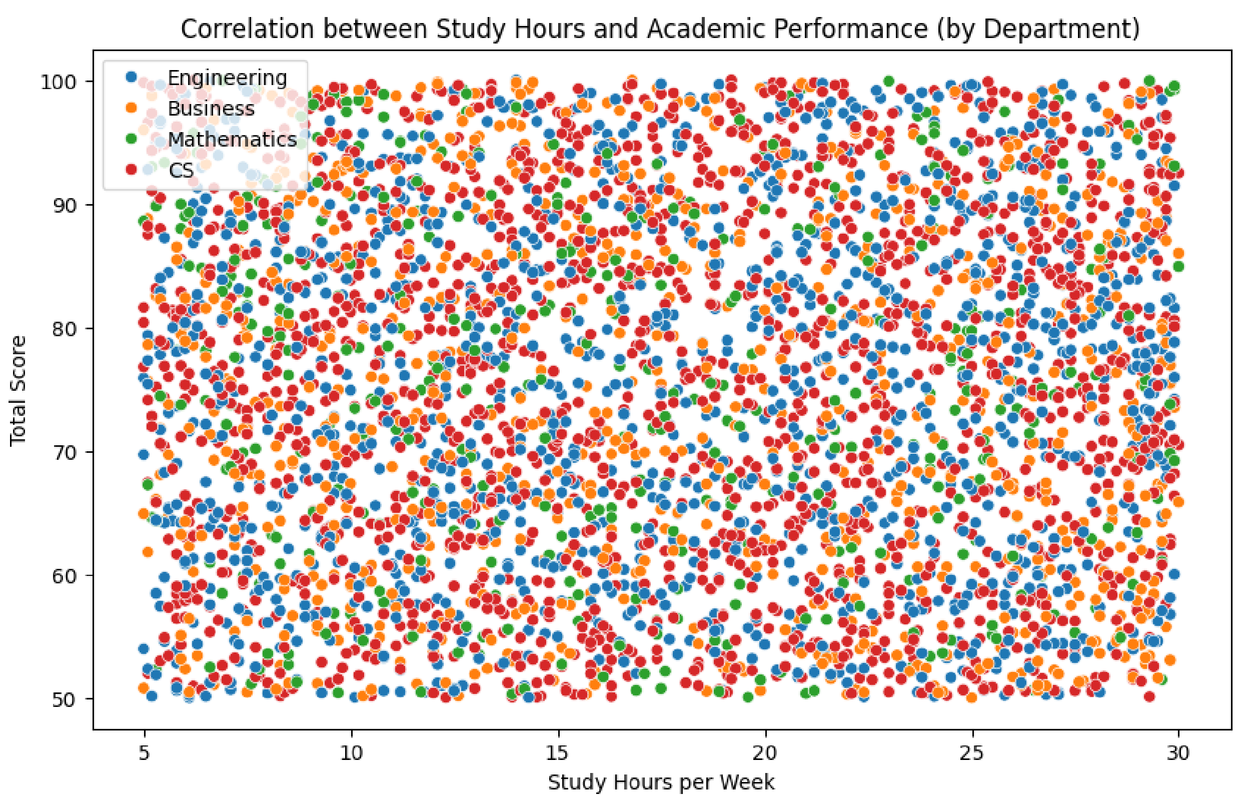
The bar chart illustrates the number of students in each department who scored above 80 (high-scoring) and whether they play sport. In Business, 96 high scorers play sport and 241 do not. Computer Science has the highest number of high scorers, with 204 playing sport and 503 not playing. Engineering has 147 high scorers playing sport and 351 who do not play. Math has the smallest figures of high scorers: 54 who play sports and 114 who do not. Computer Science leads both groups in general, with Math having the smallest.

Q6. Correlation Matrix and Regression to understand Relationships

Question: What linear relationships exist among the numeric attributes?



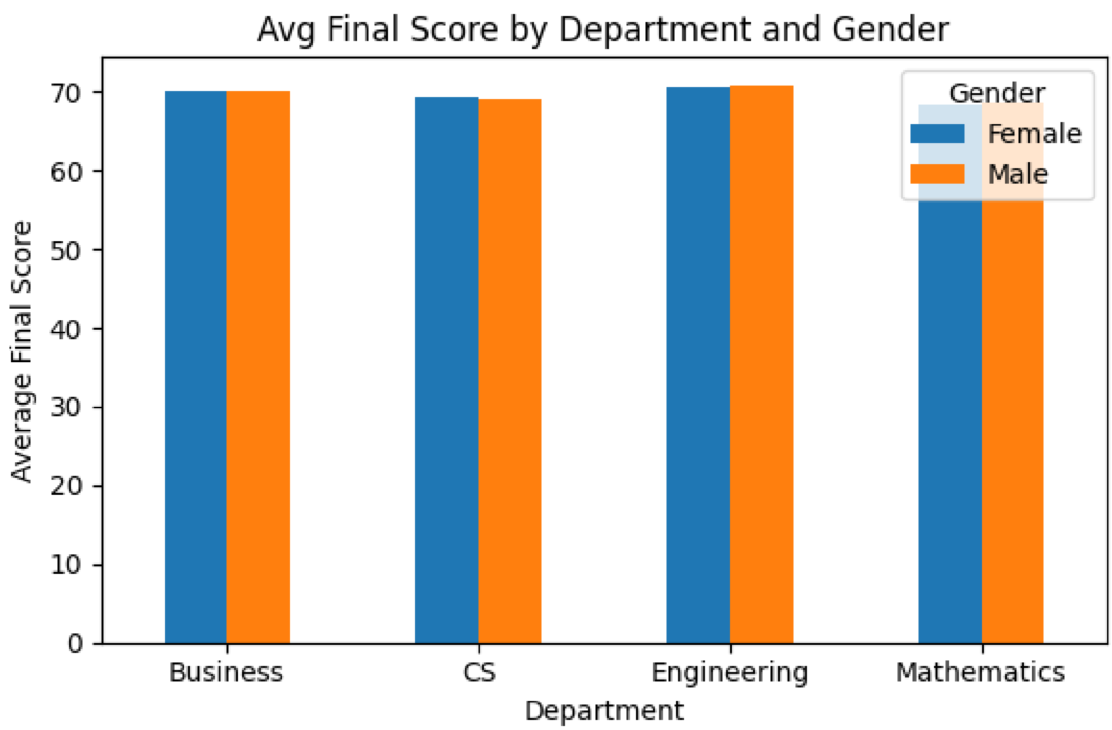
```
                            OLS Regression Results
==============================================================================
Dep. Variable:            Total Score   R-squared:                       0.002
Model:                            OLS   Adj. R-squared:                 -0.000
Method:                 Least Squares   F-statistic:                    0.9321
Date:                Mon, 19 May 2025   Prob (F-statistic):              0.502
Time:                        00:52:50   Log-Likelihood:                -17325.
No. Observations:                4243   AIC:                         3.467e+04
Df Residuals:                    4232   BIC:                         3.474e+04
Df Model:                          10
Covariance Type:            nonrobust
==================================================================================================
                                   coef    std err          t      P>|t|      [0.025      0.975]
--------------------------------------------------------------------------------------------------
const                           76.8013      2.999     25.613      0.000      70.923      82.680
Midterm Score                   -0.0018      0.013     -0.137      0.891      -0.027       0.024
Final Score                      0.0032      0.013      0.246      0.806      -0.022       0.028
Assignments Score (Averaged)     0.0048      0.016      0.299      0.765      -0.027       0.037
Quizzes Score (Averaged)         0.0241      0.015      1.585      0.113      -0.006       0.054
Participation Score             -0.1257      0.076     -1.643      0.100      -0.276       0.024
Projects Score                  -0.0242      0.015     -1.581      0.114      -0.054       0.006
Study Hours per Week            -0.0203      0.030     -0.670      0.503      -0.080       0.039
Stress Level (1-10)              0.0265      0.077      0.343      0.731      -0.125       0.178
Sleep Hours per Night           -0.0269      0.152     -0.177      0.860      -0.325       0.271
Attendance (%)                  -0.0165      0.016     -1.016      0.309      -0.048       0.015
==============================================================================
Omnibus:                     3143.530   Durbin-Watson:                   1.995
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              248.232
Skew:                          -0.013   Prob(JB):                     1.25e-54
Kurtosis:                       1.815   Cond. No.                     2.47e+03
==============================================================================
```



Answer:
All five variables (Attendance, Total Score, Study Hours, Sleep Hours, Stress Level) only correlate with themselves (1.0) and not with each other (–0.04 to 0.02). A regression using these predictors yields all p-values above 0.05 and an adjusted R^2 of –0.01, showing it cannot explain final scores. Even a scatter plot of Study Hours versus Final Score by department reveals no clear trend. In short, each factor acts on its own, and simple linear methods cannot uncover any hidden links.

Q7. Average Final Score by Department and Gender

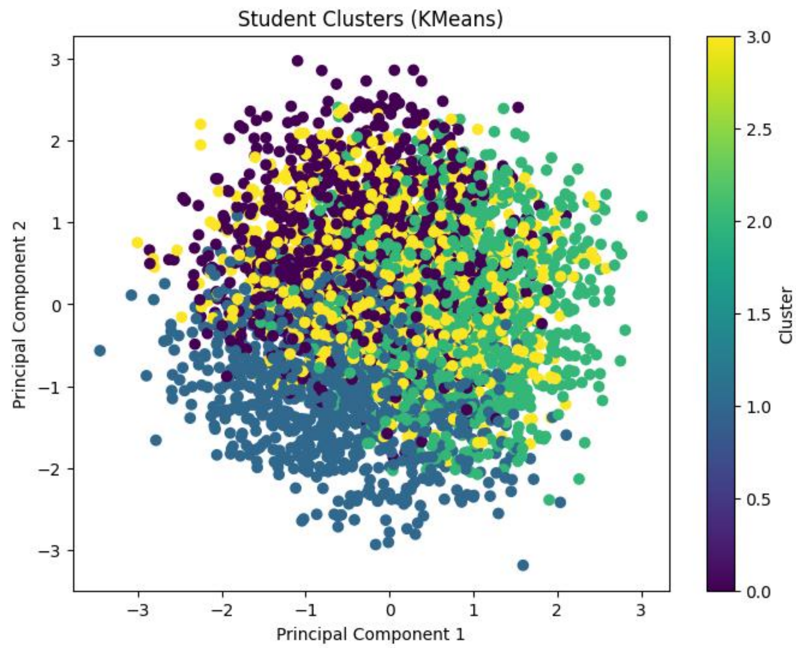
Question: Do average final exam scores differ across departments and between genders?



The "Avg Final Score by Department and Gender" bar chart compares mean final scores for males (Orange) and females(Blue) in Business, CS, Engineering, and Mathematics. Males and females both have the same score in the range mid-60s to high-60s across all departments, indicating no large gender gap in performance in the final exam.

Q8. Understanding Student Learning Patterns Through Clustering

Question: What distinct study and performance patterns can we identify among students based on their academic scores, study hours, and stress levels?



Cluster 0:

	Midterm Score	Final Score	Assignments Score (Averaged)
count	1059.000000	1059.000000	1059.000000
mean	74.554721	82.324797	75.865203
std	14.824129	10.884766	13.563318
min	40.010000	49.790000	50.080000
25%	63.765000	74.645000	65.545000
50%	76.330000	83.210000	74.828117
75%	86.155000	91.070000	86.390000
max	99.910000	99.980000	99.960000

	Quizzes Score (Averaged)	Study Hours per Week	Stress Level (1-10)
count	1059.000000	1059.000000	1059.000000
mean	71.218593	10.763362	5.556185
std	13.908271	3.629310	2.824984
min	50.030000	5.000000	1.000000
25%	59.390000	7.600000	3.000000
50%	70.100000	10.400000	6.000000
75%	81.790000	13.400000	8.000000
max	99.900000	21.100000	10.000000

Cluster 0 – "Balanced Acheivers"

Students in cluster 0 score solidly across the board, 1059 students average 82.3 on finals, 74.6 on midterms, study 10.8 hrs/week, and report stress 5.6/10. A steady, well rounded group.

Cluster 1:

	Midterm Score	Final Score	Assignments Score (Averaged)
count	1058.000000	1058.000000	1058.000000
mean	52.817807	57.252836	74.227787
std	8.512407	10.724053	13.751985
min	40.000000	40.090000	50.130000
25%	45.380000	48.285000	62.817500
50%	52.075000	56.675000	74.828117
75%	59.117500	64.472500	85.102500
max	76.980000	86.790000	99.980000

	Quizzes Score (Averaged)	Study Hours per Week	Stress Level (1-10)
count	1058.000000	1058.000000	1058.000000
mean	75.112543	16.443195	4.987713
std	14.496912	6.662057	2.831910
min	50.030000	5.000000	1.000000
25%	62.595000	11.200000	2.000000
50%	74.940000	15.800000	5.000000
75%	87.882500	21.600000	7.000000
max	99.960000	30.000000	10.000000

Cluster 1 – "High effort, Low effciency"

Cluster 1 students put in the most study time, 1058 students study 16.4 hrs/week (the most) but score just 57.3 on finals and 52.8 on midterms signaling a need for smarter study methods.

Cluster 2:

	Midterm Score	Final Score	Assignments Score (Averaged)
count	1064.000000	1064.000000	1064.000000
mean	69.218882	84.760254	74.614870
std	15.918761	9.577428	13.490179
min	40.020000	56.820000	50.000000
25%	56.607500	78.210000	64.002500
50%	67.980000	85.755000	74.828117
75%	82.115000	92.390000	84.745000
max	99.880000	99.980000	99.780000

	Quizzes Score (Averaged)	Study Hours per Week	Stress Level (1-10)
count	1064.000000	1064.000000	1064.000000
mean	77.387284	24.202256	5.535714
std	14.446141	3.852310	2.857981
min	50.190000	13.400000	1.000000
25%	65.560000	21.300000	3.000000
50%	78.415000	24.700000	6.000000
75%	90.130000	27.400000	8.000000
max	99.940000	30.000000	10.000000

Cluster 2 – "High performers"

1064 students achieve top marks (mean 84.8 finals, 69.2 midterms) with 24.2 hrs/week of study and moderate stress 5.5/10, showing efficient effort.

```
Cluster 3:
       Midterm Score  Final Score  Assignments Score (Averaged)  \
count    1062.000000  1062.000000                   1062.000000
mean       84.840377    54.583004                     74.637426
std         9.587196     9.023724                     13.728215
min        61.530000    40.000000                     50.010000
25%        77.452500    47.107500                     62.910000
50%        85.580000    53.810000                     74.828117
75%        93.227500    60.782500                     85.842500
max        99.970000    81.440000                     99.920000

       Quizzes Score (Averaged)  Study Hours per Week  Stress Level (1-10)
count               1062.000000           1062.000000          1062.000000
mean                  75.923399             19.177589             5.794727
std                   14.476284              6.773679             2.865711
min                   50.160000              5.000000             1.000000
25%                   63.620000             13.900000             3.000000
50%                   76.630000             19.500000             6.000000
75%                   88.385000             25.000000             8.000000
max                   99.960000             29.900000            10.000000
```

Cluster 3 – "Burnout Risk"
1062 students excel midterms (84.8) but drop to 54.6 on finals despite 19.2 hrs/week of study and stress 5.8/10 highlighting retention and stress-management issues.

The four groups illustrate that more studying does not always equal greater performance. Group 1 studies a lot but gets bad grades, while Group 2 gets good grades from their studying. Group 3 does well in midterms but badly in finals, illustrating the importance of regular studying and stress management. Group 0 illustrates that steady effort gets steady results. In order to do better, Cluster 1 might need help with study skills, and Cluster 3 might do better with stress management.

Conclusion

This project produced a full data-processing and visualization pipeline that cleaned, normalized, and analyzed a 5,000-student dataset to uncover drivers of academic performance. We addressed the core quality issues, imputing 10–36% of missing values with mean or mode, removing 757 duplicates, and verifying data types and outliers via robust, exception-handled Python code.

Our visual inquiries showed that most students are clustered in the 75–85% attendance and score range, with study time and sleeping habits having mixed impacts on performance. There were no significant linear relationships between quantitative characteristics ($|r|<0.05$), highlighting the need for factor models with more than one factor. Department and gender had no significant grade differences, but parental education and participation in extracurricular activities had significant effects.

Clustering of four types of students, satisfactorily performing students with steady effort (final ≈82.3) to high-striving, low-achieving students (final ≈57.3) and burnout-at-risk students whose midterm–final difference points to stress management problems which identifies areas where evidence-based interventions are needed. Targeted interventions such as study-skills training for Cluster 1 and stress-reduction programs for Cluster 3 are indicated to enhance aggregate performance.

By bringing together neat code, systematic preprocessing, and plain visual storytelling, this project shares a clear way that data can tell a story even when there is no correlation between the variables in the data.