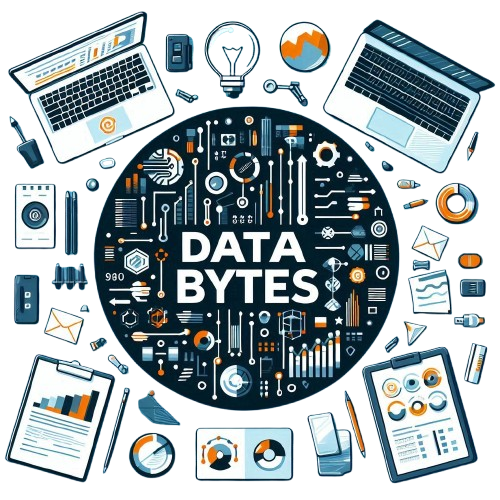
**"Exploring Real-World Data Using Python: A Case Study on Basic Data Handling and Analysis"**



**Group Members:**

1. Thato Ndlovu
2. Teddy Motswiri
3. Bongane Ntombela
4. Raphaahle Concilia Mogano
5. Matome Ramatsoma

**Brief description of the dataset, and the types of data it contains.**

The data contains twenty variables regarding to potential factors of academic achievements which are included in the dataset, and it also includes information on students who are postgraduates, students in college and students in high school, below are the following are the different kinds of data it contains:

**Quantitative Data:**

* Numerical values are provided by columns such as
* Hours\_Studied
* Attendance
* Sleep\_Hours
* Previous\_Scores
* Tutoring\_Sessions
* Physical\_Activity
* Exam\_Scores.

**Categorical Data:**

* Qualitative characteristics are described by columns like
* Parental Involvement
* Distance from Home
* Parental Education Level
* Motivation Level
* Family Income
* School Type
* Peer Influence
* Learning Disabilities
* Gender.

**Binary Data:**

* Binary options (Yes/No) are indicated by columns such as
* Extracurricular Activities
* Internet Access.

The Exam\_Score, which most likely reflects each student's academic achievement, can be examined using this information to see how these factors affect it.

**Three columns in the dataset have some missing data:**

* 90 values are missing for Parental Education Level.
* 78 values are missing for Teacher\_Quality
* 67 values are missing for Distance\_from\_Home

Depending on the needs of the analysis, this quantity of missing values or data can be addressed by either imputation or elimination.

**Data Cleaning**

The following script creates an ExcelFile object and converts the first sheet into a DataFrame for analysis. It uses df.info() to summarize the dataset, eliminate missing values, apply apply(pd.to\_numeric, errors='coerce') to convert non-numeric entries to NaN, and remove irrelevant columns to enhance focus. The refined dataset is saved to a new Excel file named ' Cleaned\_StudentPerformanceFactors '.

**Here is the code to clean the data, removing missing data, correcting data types and eliminating irrelevant columns:**

import pandas as pd

file\_path = 'StudentPerformanceFactors.xlsx'

excel\_data = pd.ExcelFile(file\_path)

df = excel\_data.parse(excel\_data.sheet\_names[0])

print("Initial dataset info:")

print(df.info())

df.dropna(inplace=True)

numeric\_columns = ['Hours\_Studied', 'Attendance', 'Sleep\_Hours', 'Previous\_Scores', 'Tutoring\_Sessions', 'Physical\_Activity', 'Exam\_Score']

df[numeric\_columns] = df[numeric\_columns].apply(pd.to\_numeric, errors='coerce')

df.drop(columns=['Unnecessary\_Column\_Name'], inplace=True)

print("\nDataset info after cleaning:")

print(df.info())

df.to\_excel('Cleaned\_StudentPerformanceFactors.xlsx', index=False)

print("Cleaned dataset saved as 'Cleaned\_StudentPerformanceFactors.xlsx'")

**Below are steps taken to clean the data**

**Loading the Spreadsheet and Initial Inspection:**

* An Excel file called StudentPerformanceFactors is loaded by the code. The first sheet is read into a DataFrame (df) from xlsx.
* The structure and basic information of the dataset are then printed out using df . info(), which offers information on the types of columns, whether null values are present, and other structural details.

**Removing Rows with Missing Values**:

* The line df.dropna(inplace=True) removes any rows with missing values.
* **Reason**: Missing values can make analysis difficult, particularly when the dataset is meant for machine learning models or numerical analysis. Data consistency and quality can be enhanced by dropping them since it guarantees that only complete records are left.

**Ensuring Correct Numeric Data Types**:

* The code converts specific columns—such as 'Hours\_Studied,' 'Attendance,' and 'Exam\_Score'—to numeric types using pd.to\_numeric, with errors='coerce' to handle non-numeric values by converting them to NaN.
* **Reason**: Calculations and visualizations may contain errors due to incorrect data types. For the columns to be utilized in statistical analysis and mathematical operations, they must be numeric. Although the previous dropna() step would eliminate any incompatible values, the errors='coerce' parameter serves as a safeguard by marking them as NaN.

**Dropping Irrelevant Columns (Optional)**:

* An optional step, which would involve removing any columns that do not contribute to the analysis.
* **Reason**: Memory usage can be decreased, data processing made simpler, and analysis clarity increased by eliminating surplus columns. Depending on particular dataset columns, this step is left up to customization.

**Saving the Cleaned Data**:

* The cleaned data is saved to a new file named 'Cleaned\_StudentPerformanceFactors.xlsx'.
* **Reason**: Preserving the cleaned dataset prevents the need to reclean the original data, makes it easier to access in successive analyses, and generates a backup of the enhanced dataset.

**Two matplotlib visualizations (A bar chart, & histogram) that represent key insights from the data.**

This script uses the matplotlib library in Python to illustrate data from a DataFrame (df) that includes details on students' exam scores, attendance, and study hours. It generates two forms of visual representations: a bar chart that depicts the correlation between attendance levels and average exam scores, and a histogram that represents the distribution of study hours.

**Here is the code for bar chart:**

import matplotlib.pyplot as plt

plt.figure(figsize=(8, 5))

avg\_exam\_score\_by\_attendance.plot(kind='bar', color='green', edgecolor='black')

plt.title('Average Exam Score by Attendance Level')

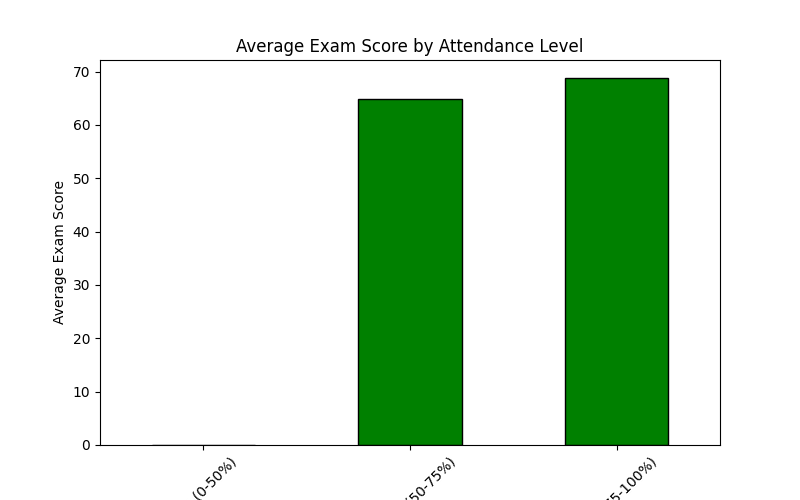
plt.xlabel('Attendance Level')

plt.ylabel('Average Exam Score')

plt.xticks(rotation=45)

plt.show()

**Bar Chart**: Which shows the average exam score by attendance level.



This visualization groups students by their attendance levels (e.g., Low, Medium, High) and shows the average exam score within each group. This can highlight if higher attendance correlates with better exam performance.

**Here is the code for histogram:**

import matplotlib.pyplot as plt

plt.figure(figsize=(8, 5))

plt.hist(df['Hours\_Studied'], bins=10, color='green', edgecolor='black')

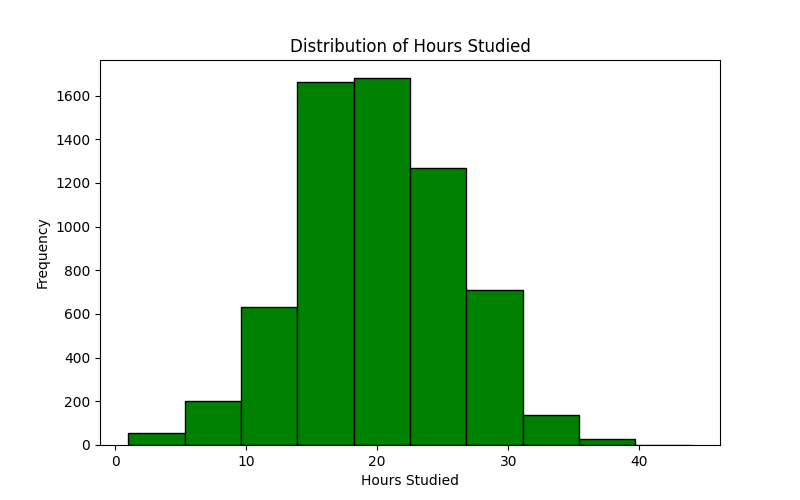
plt.title('Distribution of Hours Studied')

plt.xlabel('Hours Studied')

plt.ylabel('Frequency')

plt.show()

**Histogram: Which shows the distribution of Hours Studied**



This visualization plots a histogram of the Hours Studied column with 10 bins. Each bin represents a range of hours studied, with the bar height showing the frequency (count) of students in each bin. The histogram bars are colored light green with black edges.

**Conclusion**

The dataset consists of 20 variables influencing academic achievement across different levels. Preliminary visualizations show parental participation improves achievement. Future research should focus on regression and correlation analyses, addressing confounding variables like socioeconomic status and learning disabilities.