STUDENT EXAM PERFORMANCE PREDICTION

*Predict a student’s mark on an exam.*

**Group 3**

**Lam Tien Hung Bui Duc Trieu Le Hoang**

**Nguyen Thi Thu Hoai Pham Van Quoc Vuong**

23.02.2024

MAS291

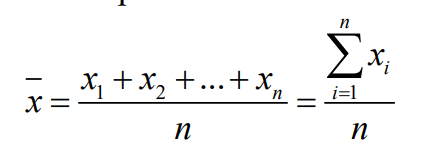
# INTRODUCTION

* The project is designed to research topic on the relationship between factors such as study time (Time Study), and marks (Marks). The goal of the study is to better understand the influence of study time and the ability to pass the subject in the current exam.
* Predicting students' exam performance is highly relevant, with implications for educational strategies and interventions. Educators can gain insights into students' potential outcomes by analyzing study hours. Additionally, the availability of data on study habits and past performance makes this topic feasible for analysis and modeling. Moreover, the challenge of accurately predicting student success adds intellectual interest to the project. Ultimately, the ability to forecast exam results could enable targeted interventions to support student learning and achievement. Therefore, this topic offers practical value and the opportunity to make meaningful contributions to the field of education.

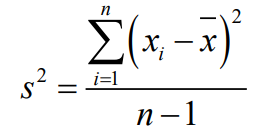
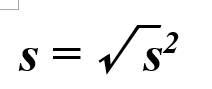
# MATERIALS

1. **Math tools used**:

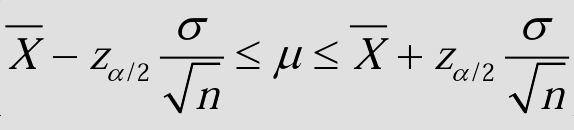
- Mean:



- Variance & standard deviation:

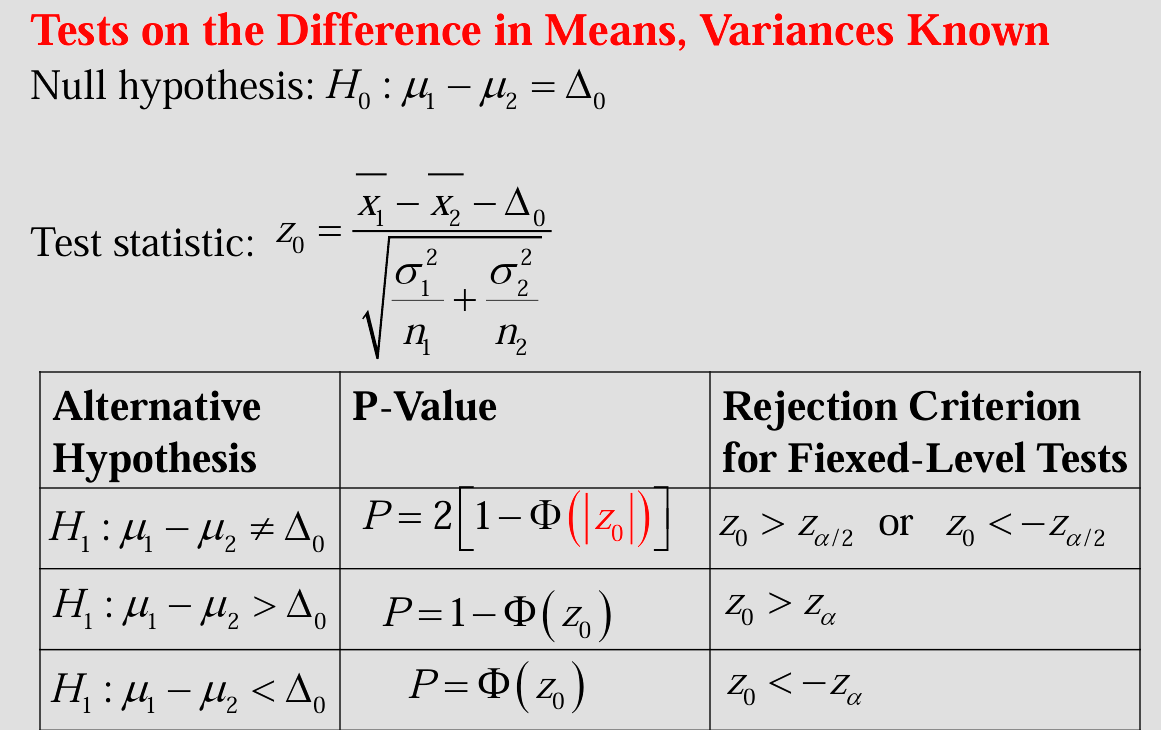
- Test a hypothesis and construct a confidence interval for the mean of a population:



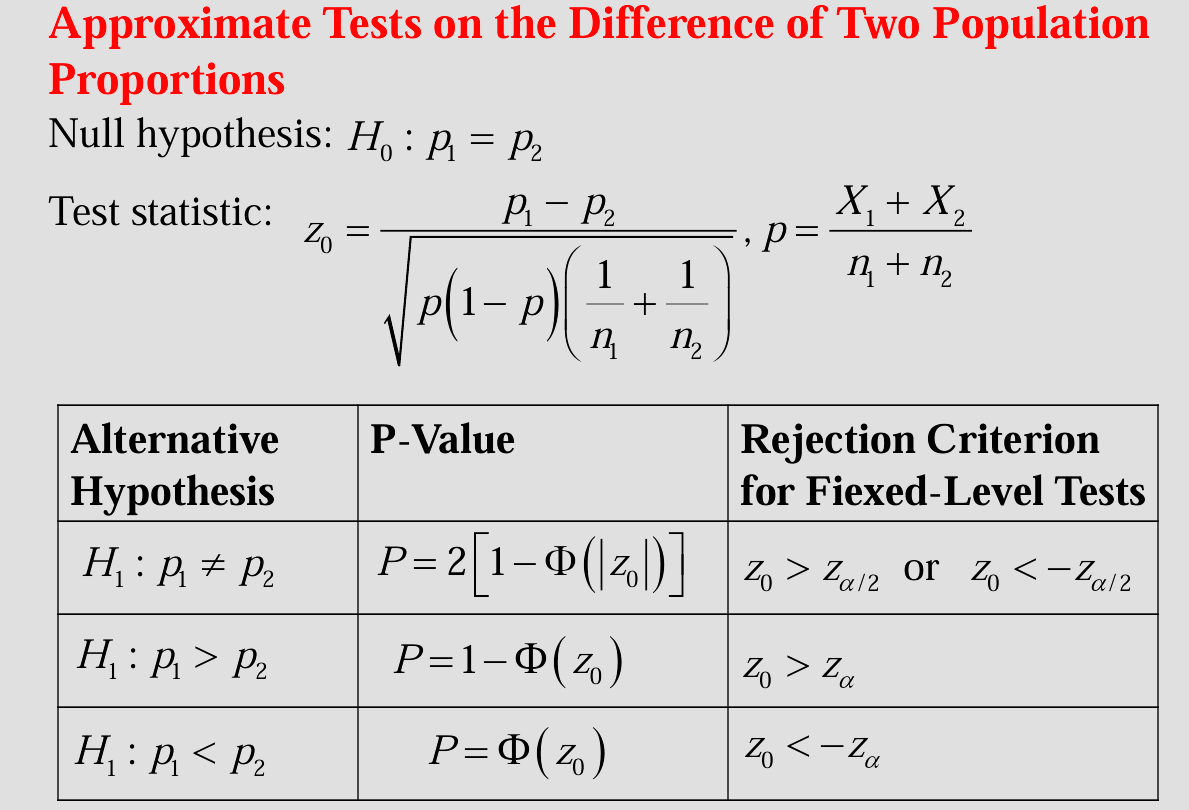
- Compute the sample correlation coefficient R:

#### 

- Test a hypothesis and construct a confidence interval for the difference in means of two populations:



* Test a hypothesis and construct a confidence interval for the difference in proportions of two populations:



1. **Python:**

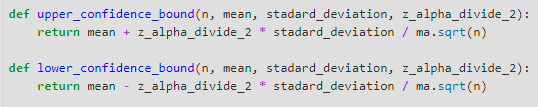
* **statistics.mean()**: is used to find the mean of a population

Example:

#### 

*  **cal\_z\_alpha\_divide\_2(confidence\_level)**: is used to calculate the z\_alpha\_divide\_2 depending on the confidence\_level.

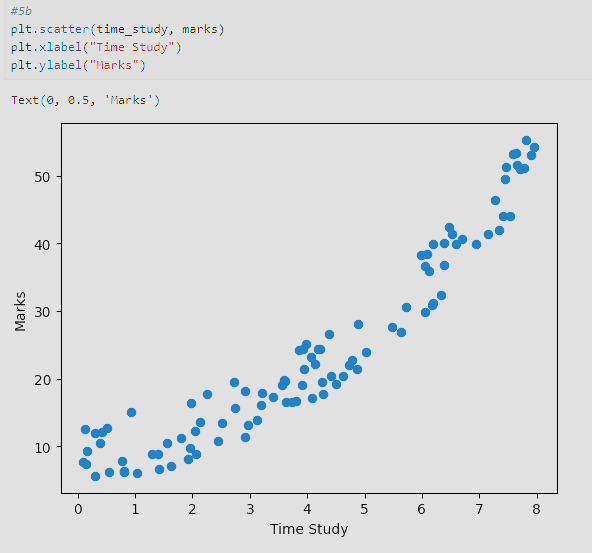
Use:To use this function, you simply call it and pass the desired confidence level value.

* 

Use: you need to provide the values of the necessary parameters (n, mean, standard\_deviation, z\_alpha\_divide\_2), then call the upper\_confidence\_bound function with these values. The function will calculate the Upper Confidence Bound value and return the result. Lower\_confidence\_bound function is similar.

To calculate the upper and lower bounds for testing a hypothesis and construct a confidence interval for the mean of a population.

* **matplotlib.pyplot.scatter()**: Construct a scatter plot for the data, which serves as a visual tool to explore and analyze the relationships between variables, utilizing dots to depict the connection between them.

Example: 

* **numpy.corrcoef():** used to compute the correlation coefficient between two or more variables. Correlation coefficient measures the strength and direction of the linear relationship between two variables.
* **from sklearn.linear\_model import LinearRegression**
* **model = LinearRegression():** create a linear regression model
* **model.intercept\_**: get the coefficients (intercept)
* **model.coef\_[0]**: get the coefficients (slope)
* **model.predict(future\_X)[0]:** predict a future value for Y

# DATA

* **Part of the database**:



* **Time Study**: Represents the number of hours a student spent studying for the exam.
* **Marks**: Indicates the student's mark in the exam.

# REFERENCES

* [*Source of the data used*](https://www.kaggle.com/datasets/yasserh/student-marks-dataset)