# Assist the professor

## Introduction

A university professor needs your assistance in making decisions for whom should pass his classes and he needs to follow a scientific approach for this. He needs a tool to analyze the students marks and categorize the students into three categories:

1. Elite: students get scores higher than other students in the class room
2. Passed: Students get the required score to pass the exam
3. Failed: Students not getting the required scores to pass the exam

He needs to follow an evaluation strategy to categorize the students without setting a specific passing mark, he wants to set the passing marks according to the scores the students got in the exam. Also, he needs to be able to do this categorization for each classroom separately or for all classes he is instructing, and he needs you to follow [Z-Scoring](https://www.statisticshowto.com/probability-and-statistics/z-score/) technique in your implementation.

He provides a CSV file containing all students' marks (out of 100) for all classes and you need to implement this tool against it, the file format looks like below:

Student\_id, class\_no, mark

202008001, A, 60

202008002, A, 60

202008003, B, 70

202008007, B, 90

202008005, A, 40

## Functional requirements

The professor is accepting the tool to be a console based one and he is expecting the below behavior:

* Once he starts the tool, it should promote him to set the path of the file containing the marks
* Once the tool accepts it, he needs to see the below menu

1*- Summary*

*2- Summary for a specific class*

*3- Display Z-scores*

*4- Display Z-scores for specific class*

*5- Categorize students*

*6- Categorize students in a specific class*

*7- Exit*

* If the professor chooses option 1, your tool should display a summary for all students in all classrooms as below:

*Median:*

*Variance:*

*Standard Deviation:*

*Total count:*

* If the option 2 is chosen, you need to show the same summary but for a specific classroom he chooses
* If the professor chooses option 3 (display Z-Scores), your tool should do the proper calculations then display the student’s information along with the z-score for each student as below:

|  |  |  |  |
| --- | --- | --- | --- |
| Student ID | Classroom | Mark | Z-Score |
| 202008001 | A | 60 | 1.56 |
| 202008002 | A | 80 | 2.3 |
| ….. |  |  |  |
|  |  |  |  |

* If the professor chooses option 4, your tool should do the same calculations as option 3 but for specific classroom he chooses
* If the professor chooses option 5, the tool should promote for categories ranges as below:

*Enter Elite Deviations:*

*Entry Failed Deviations:*

Then your tool should categorize students according to those inputs by determining the Elite students with deviations equals or higher than the professor specified and the failed students with deviations equals or lower than the professor specified, the students within those values are considered passed then you should display the below summary:

*Median:*

*Variance:*

*Standard Deviation:*

*Elite students count:*

*Passed students count:*

*Failed students count:*

*Passing score:*

*Elite score:*

*Do you want to save the results to a file (yes/no):*

The professor should be able to save the related results in another CSV file where the file should look like below:

Student\_id, class\_no, mark, z\_score, category

202008001,A,60, 1.61, Passed

202008002,A,90, 2.1, Elite

202008001,A,40, 1.61, Failed

* For option 6, do the same as option 5 but for specific classroom
* Option 7 will allow him to exist the application

## General notes

You should consider the following in your implementation:

1. The professor has at least 30 students in each classroom
2. After each option execution and result, you should display the main menu again
3. You should follow TDD in implementation and all aspects in your code should be covered in the unit tests.
4. Your design should be flexible enough to support new sources of data, like reading data form a different file format or from a database, minimum impact should be done to code to support any new sources.
5. You can find a description for how to calculate z-scores [here](https://www.statisticshowto.com/probability-and-statistics/z-score/) and you should do the whole calculations in your code, **don’t use any third party for this**.
6. You should consider performance in your implementation.
7. You will be provided with a sample data for such marks.
8. Try to generate some random samples and have a look at the results, put yourself at professor chair and see if this scoring technique is sufficient or not and describe why?