# Microsoft Learn Student Ambassadors

## Data and Analytics League

### Solution for the November challenge

**From Ioannis Koutsocheras**

- ## Objectives

The main objectives of this challenge was to analyse the given dataset that contained various information about student's academic performance in several subjects and make some conclusions based on:

1) The student's names
2) Mathematical measures such as the standard deviation and the mean of the grade values

- ## First Approach

Despite the instructions proposals to use English as Subject1, Maths as Subject2, and so on, I preferred not to take the default subjects but present a complete and precise solution.

The main reason that led me to do this was the fact that if we took the default prior-mentioned proposals we would be around 75% (742178 rows) wrong on our final conclusions regarding these subjects due to the function that is mentioned below.

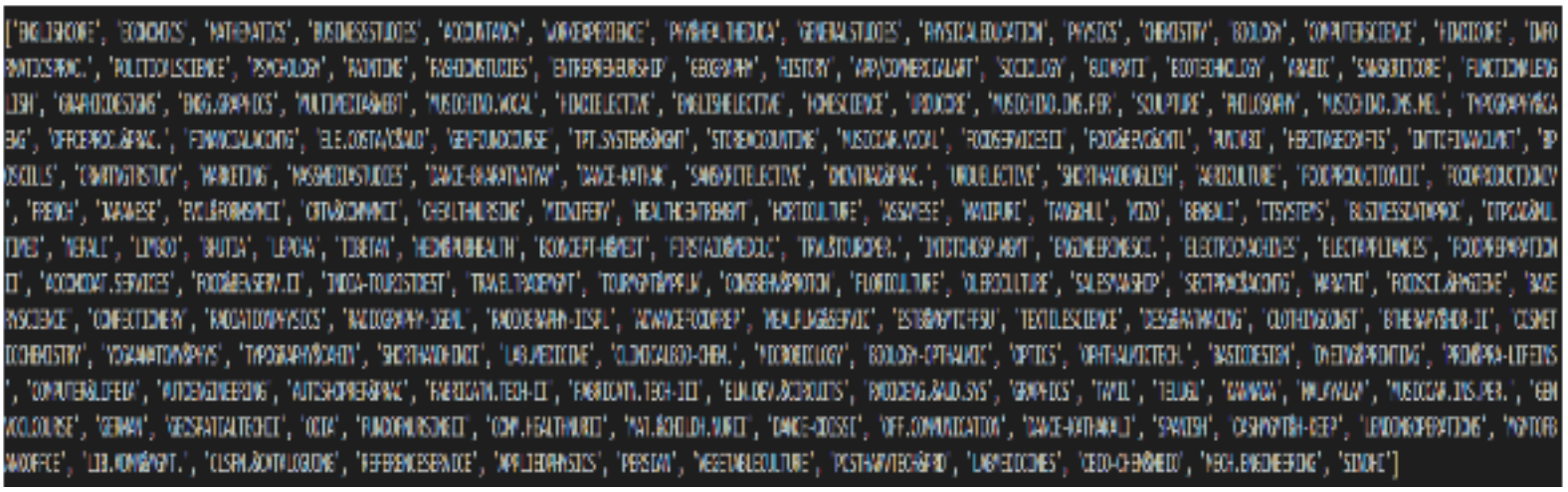
```
def subjectsOrderConclusions(data):

    df = pd.read_csv(data)

    wrong_counter = 0

    for index, subjects in enumerate(zip(df['Subject1'], df['Subject2'])):

        if subjects[0] != 'ENGLISHCORE' or subjects[1] != 'MATHEMATICS':
            wrong_counter += 1


    # 742178 rows (~75%) is wrong if we take for default subject1 to be english and subject2 mathematics
    print(wrong_counter)

subjectsOrderConclusions(dataset)
```

**Source: subjectsOrderConclusions(data) function in preprocessing.py**

- **General Facts and Statistics of the Dataset**

Some interesting facts about the dataset are:

1. The number of subjects is 172! (who would have thought that there would be so many) and the list of them is presented below.

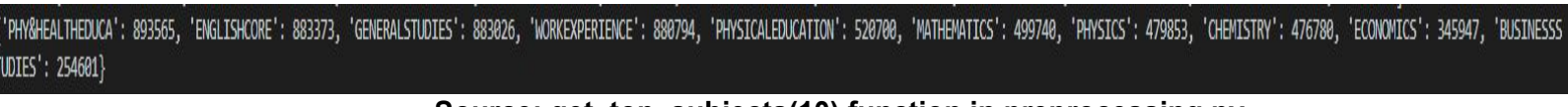


**Source: get_subjects_info(data) function in preprocessing.py**

*(Kind reminder: Due to some list sizes some screenshots details may not appear good, therefore I suggest you to run the specified functions on your own terminal)*

2. I decided to work with the most occured/famous subjects. In order to get the N top subjects you need to specify the exact number in the parameter during the function call. The most famous subject was 'PHY&HEALTHEDUCA' with 893565 appearances while the subjects 'DANCE-KATHAKALI' and 'SPANISH' had only 1 appearance!
The top 10 subjects (where I personally made my conclusions) are:

*{'PHY&HEALTHEDUCA': 893565, 'ENGLISHCORE': 883373, 'GENERALSTUDIES': 883026, 'WORKEXPERIENCE': 880794, 'PHYSICALEDUCATION': 520700, 'MATHEMATICS': 499740, 'PHYSICS': 479853, 'CHEMISTRY': 476780, 'ECONOMICS': 345947, 'BUSINESSSTUDIES': 254601}*



**Source: get_top_subjects(10) function in preprocessing.py**

*(Kind reminder: Due to some list sizes some screenshots details may not appear good, therefore I suggest you to run the specified functions on your own terminal)*

- ## Conclusions based on student names

  As far as my actions that were related to the student names are concerned, I decided to find out what was the grade difference between students that:

  1. Had the same last name and
  2. were on consecutive rows ( had consecutive IDs)

From the above one can assume that there is a high probability that the students would be family relatives (not using the second argument (consecutive IDs) would not be correct because many students can share the same surname without having a family connection but the fact that they simultaneously have consecutive IDs indicate that they might be siblings).

1. First of all, I must mention that the number of possible-siblings that fulfill the above criteria are 41803!

```python
def findSiblingsNum(data):

    df = pd.read_csv(data)

    l_names = []
    numSiblings = 0

    # iterate all students with itertuples
    for index, row in enumerate(df.itertuples()):

        # get the student name(row[2]) and specifically the last name (.split()[0])
        try:
            last_name = row[2].split()[0]
        except AttributeError:
            pass

        if last_name not in l_names:
            l_names.append(last_name)
        else:
            # print(last_name)
            numSiblings+=1

        # every 5 students clear the list (for memory efficiency reasons)
        if index%5==0:
            l_names.clear()

    # siblings number = 41803!
    print(numSiblings)

# findSiblingsNum(dataset)
```

**Source: findSiblingsNum(data) function in name_conclusions.py**

2. Details on getting the overall grade of the students can be found in the **names_grades_manipulation(data) function in name_conclusions.py** and how to get the grade difference between them can be found in the **figureNameSimilarity(data) function in name_conclusions.py.**

Short description of what I have done in the 2 functions:

> Suppose we have the following array of students and their grades.
> (For simplicity reasons the columns structure are different than the original dataset)

| Name | Grade 1 | Grade 2 | Grade 3 | Grade 4 |
|---|---|---|---|---|
| Smith John | 50 | 60 | 70 | 70 |
| Smith Alice | 60 | 70 | 40 | |
| William Bob | 50 | 90 | 100 | 60 |
| Miller Chloe | 90 | 100 | 70 | 60 |
| Miller Joe | 80 | 90 | 90 | 95 |

We can see that we have 2 groups of siblings (Smiths and Millers) therefore we don't take into consideration William Bob.

The **names_grades_manipulation(data) function** is responsible for the below:

The student's average overall grades will be:

> Smith John = (50+60+70+70)/4 = 62.5
> Smith Alice = (60+70+40)/3 = 56.7
> Miller Chloe = (80+90+90+95)/4 = 80
> Miller Joe = (90+100+70+60)/4 = 88.8

Notice that some students (Alice) took 1 less subject. In the original dataset this type of issue appears frequently (students having grades in less subjects than others).

The **figureNameSimilarity(data) function** is responsible for the below:

We compare each pair of siblings and get the absolute value of the average overall grade difference, that will later be rounded to the next 1 for better graph visualization afterwards (for example 8.6 ~> 9, 8.4 ~> 8).

In our example we should get:

[Smiths] abs(62.5-56.7) = 5.8 ~> 6
[Millers] abs(80-88.8) = 8.8 ~> 9

An extra feature that is available is that you can get the mean value of the grade differences. In the example above, the mean would be (6+9)/2 = 7.5.

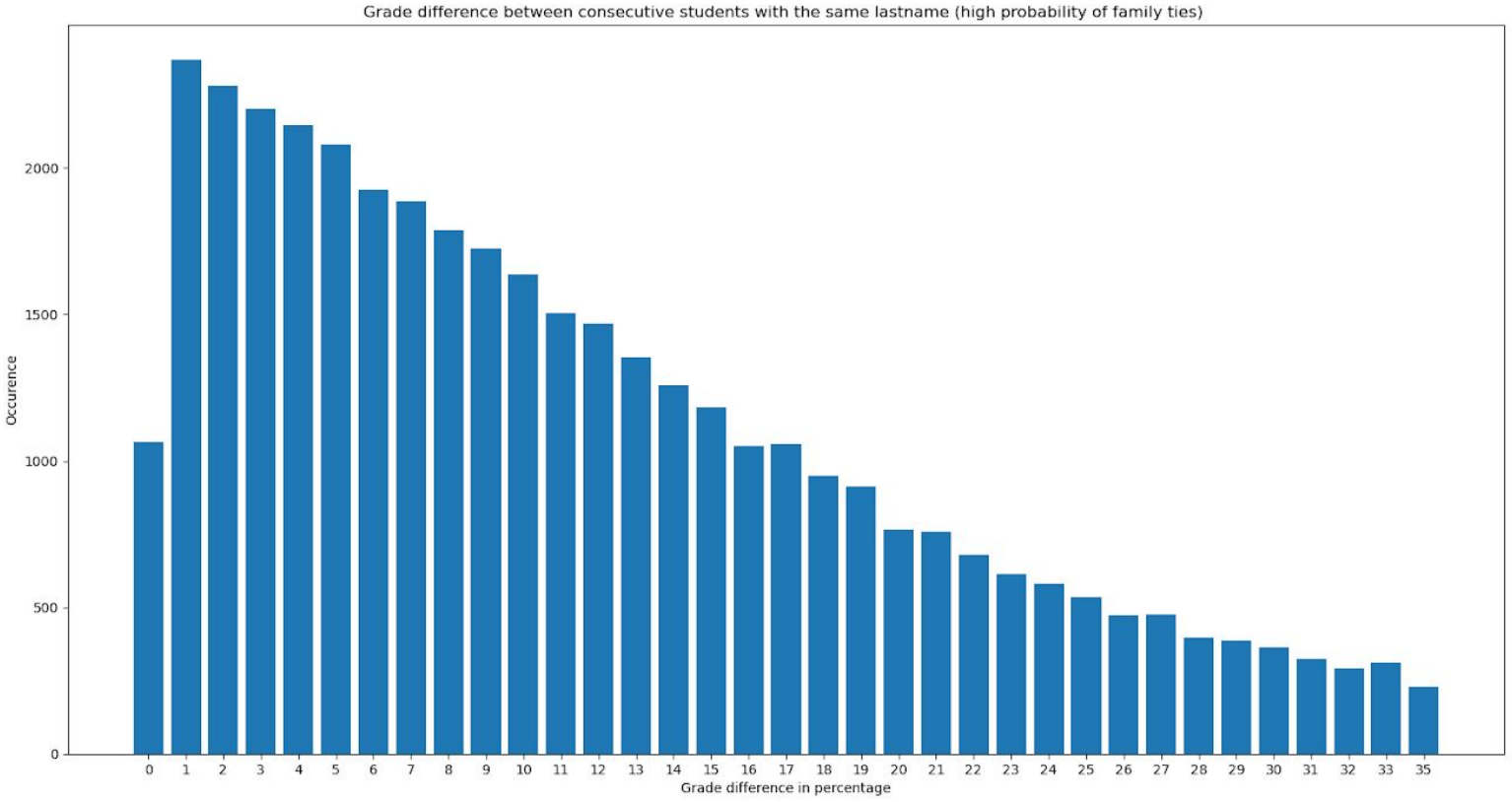
3. Final Conclusions and Graph

In the dataset that was given (‘cbse2014.csv’) we get the following measures:

```
The mean-average grade difference(in 100) between all siblings is: 12.83
```

**Source: findSiblingsNum(data) function in name_conclusions.py**

The graph containing the grade differences and the corresponding occurrences is below:




**Source: nameSimilarityGraphs() function in graphs.py**

From the above graph we can conclude that **the school performance of students with high probability of having family ties is mainly similar**.

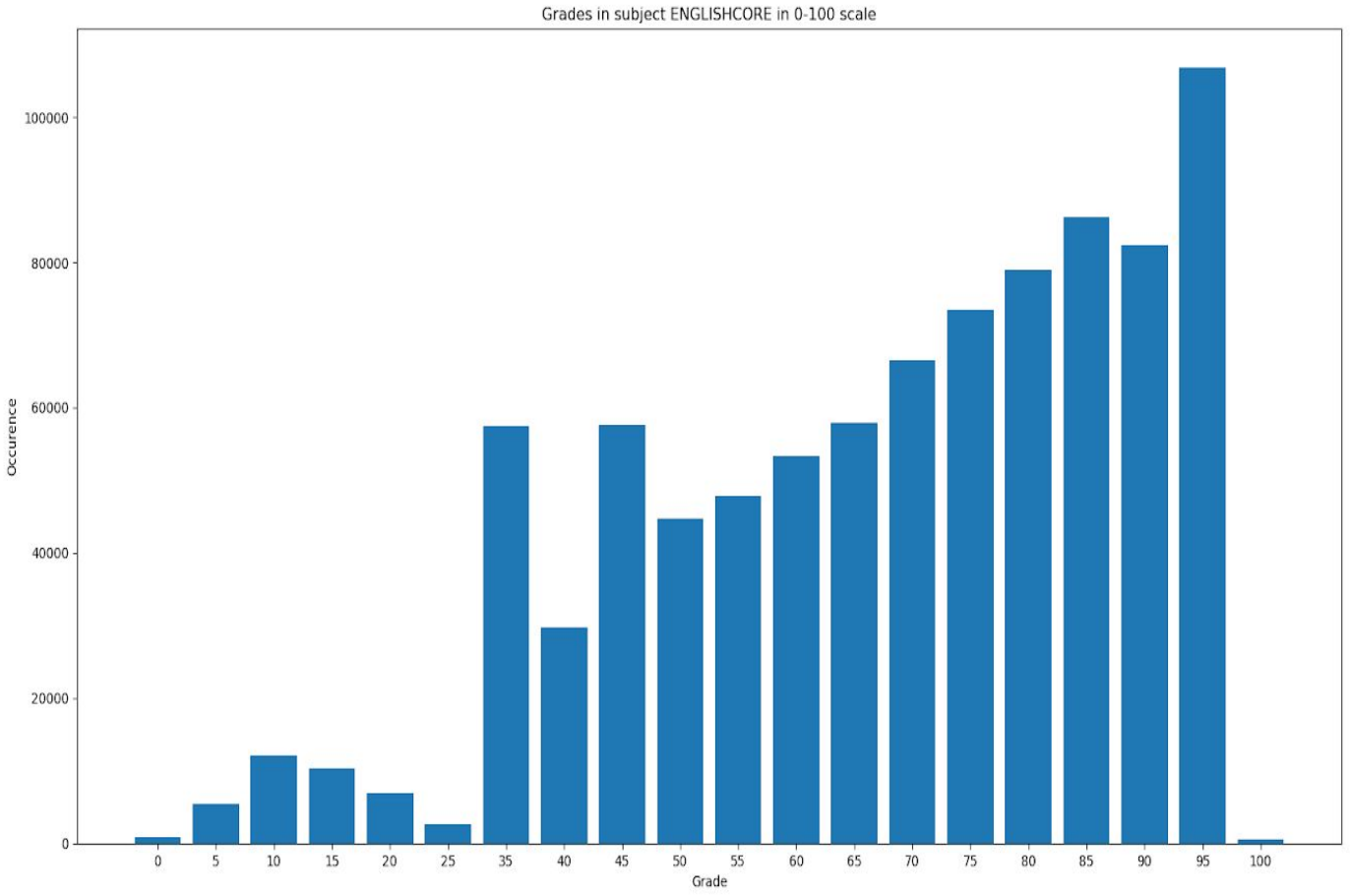
- ## Conclusions on subjects grades

First of all, we must consider that some subjects have grades only in the alphanumeric system (A1,A2,B1,B2,C1, etc) and some have in both (alphanumeric and percentage). Graphs were made for both grade systems. For the subjects that had also grade in the percentage system the standard deviation and the mean variables were calculated.

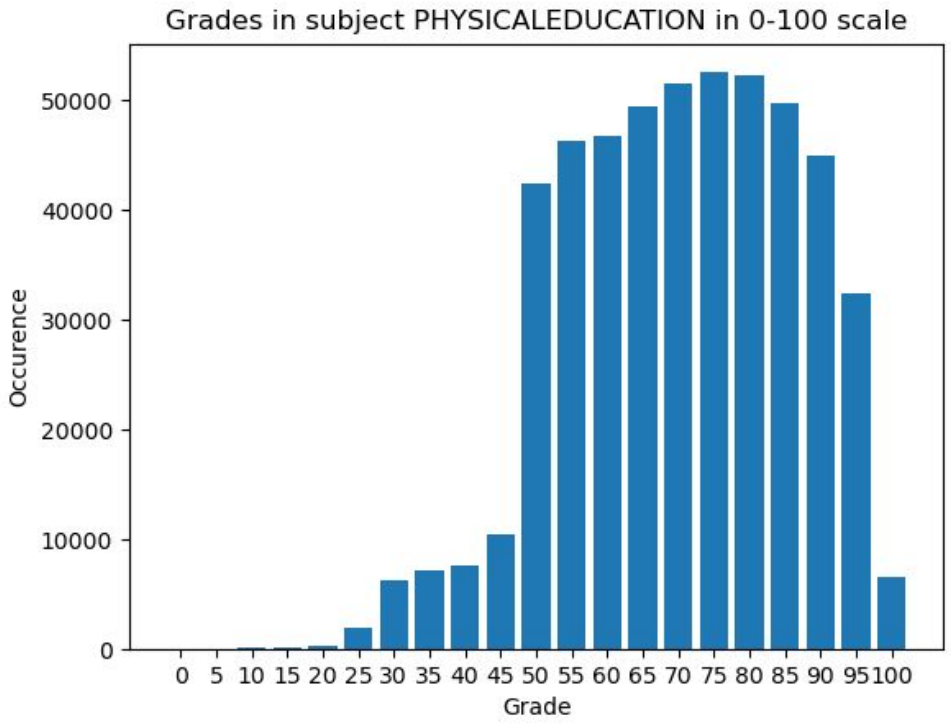
The mathematical variables were found using the **function generalGradeFacts() in subject_conclusions.py** while the graphs were made by the functions in graphs.py

Below the subject grade graphs in the 0-100 scale are presented.
*(Notice: The subject graphs that belong to the alphanumeric grade system will not be shown as we don't have any mathematical measures to make any conclusions. If somebody wants to see them he can execute the **alphanum_grade_subjectGradeGraphs() function in graphs.py**)*

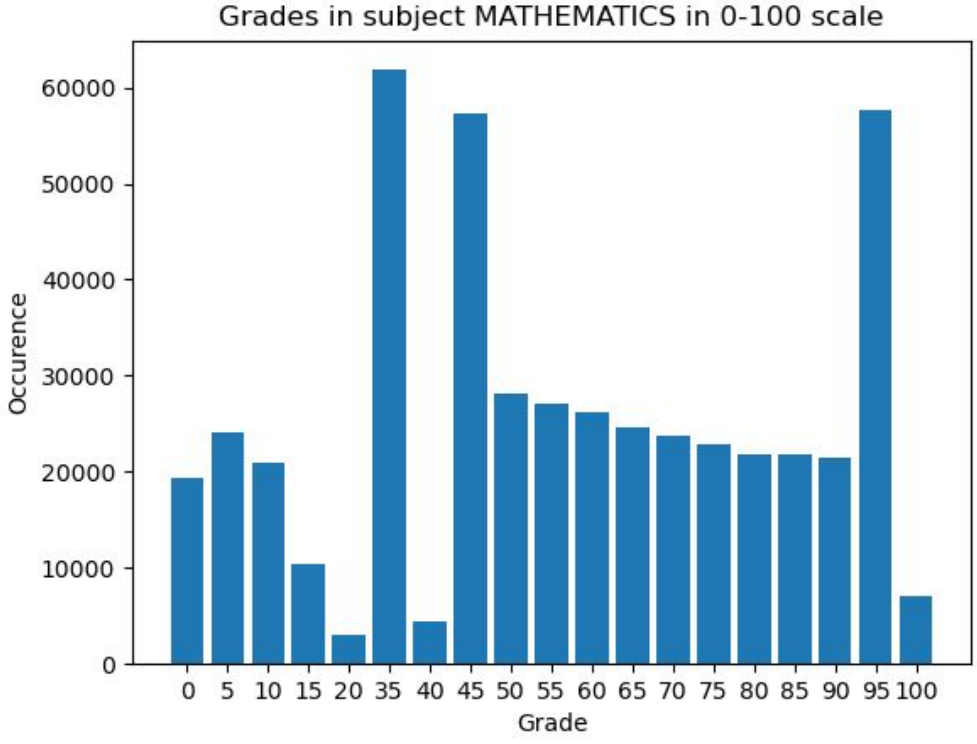
**Please notice that the grades have been rounded to the closest 5 for better graph visualization.** (for example 47~>50, 64~>65, 82~>80)



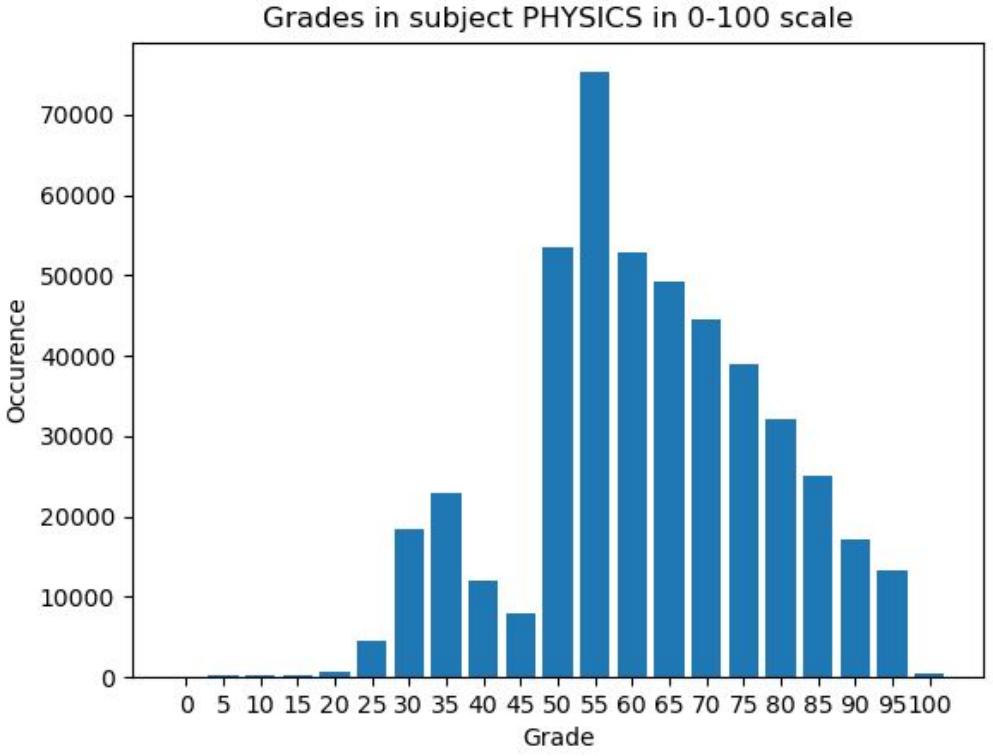

```
The standard deviaton for subject  ENGLISHCORE  is 21.91
The mean for subject  ENGLISHCORE  is 67.45
```



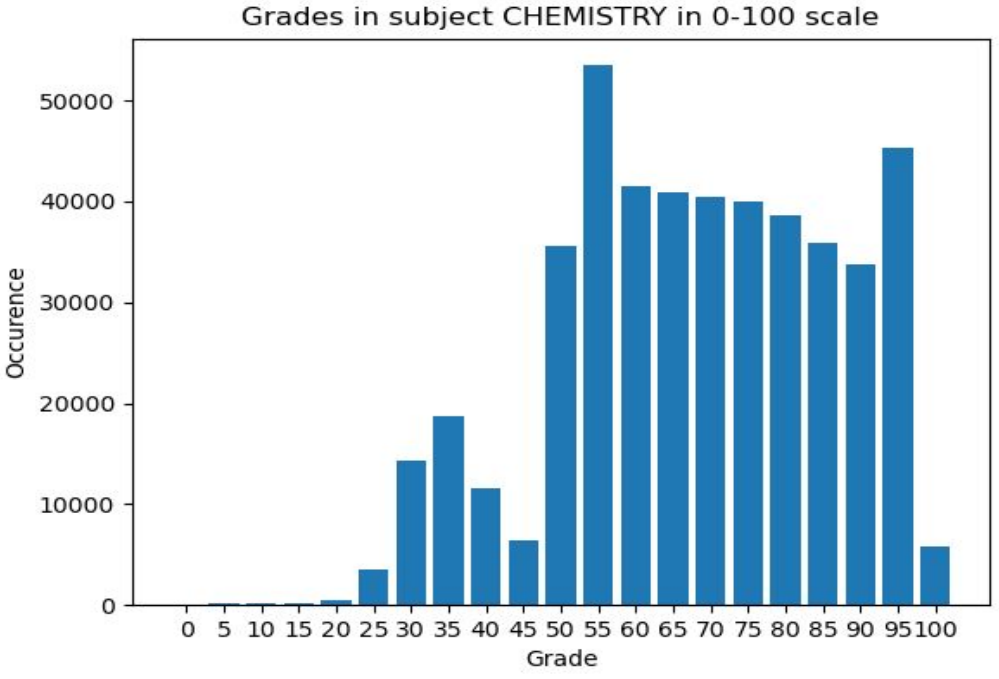

```
The standard deviaton for subject  PHYSICALEDUCATION  is 16.16
The mean for subject  PHYSICALEDUCATION  is 70.10
```



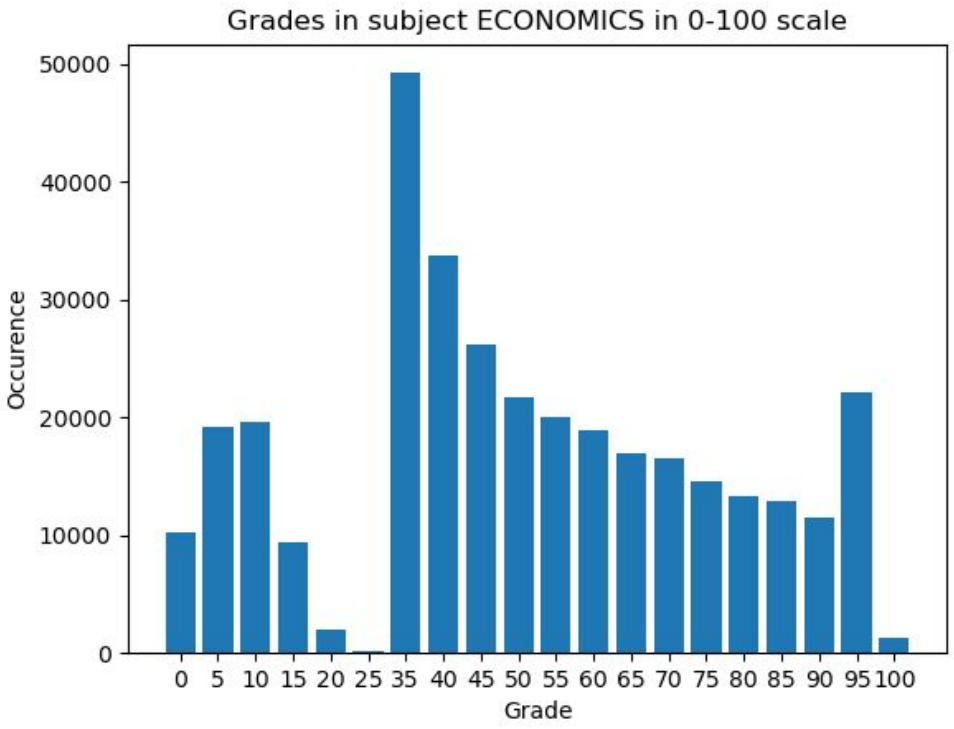

```
The standard deviaton for subject  MATHEMATICS  is 28.79
The mean for subject  MATHEMATICS  is 54.74
```



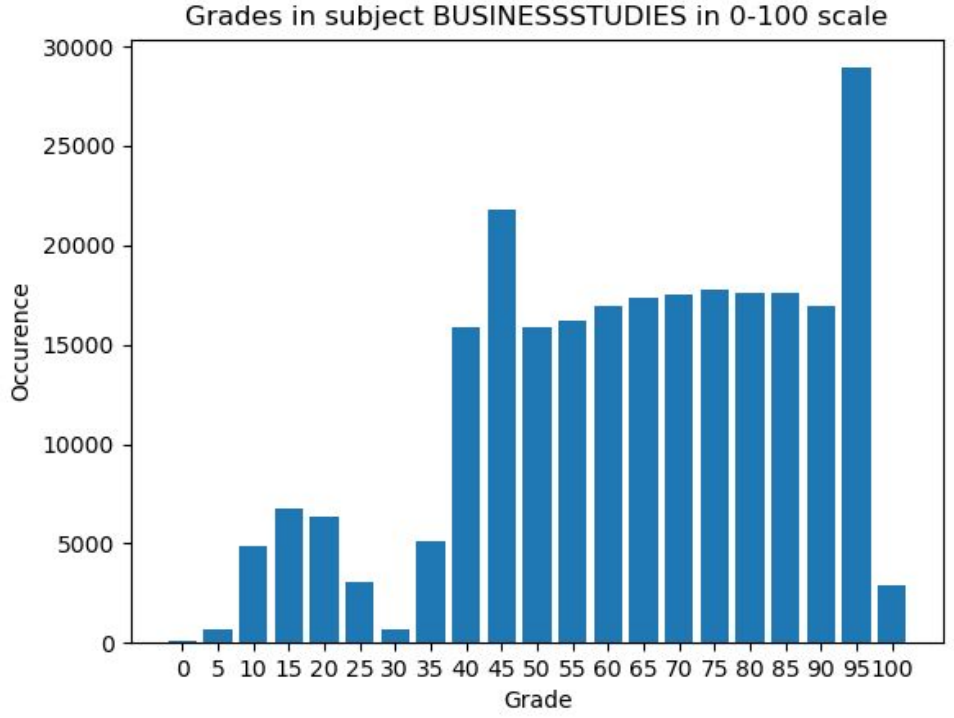

```
The standard deviaton for subject  PHYSICS  is 16.35
The mean for subject  PHYSICS  is 62.00
```




```
The standard deviaton for subject  CHEMISTRY  is 18.44
The mean for subject  CHEMISTRY  is 67.90
```




```
The standard deviaton for subject  ECONOMICS  is 26.69
The mean for subject  ECONOMICS  is 49.37
```




```
The standard deviaton for subject  BUSINESSSTUDIES  is 23.19
The mean for subject  BUSINESSSTUDIES  is 63.82
```

## ● Final Thoughts

Observing the graphs above and the mathematical variables one can make some conclusions.

1. In certain subjects such as MATHEMATICS and ECONOMICS the occurrence of grade 35 is strangely high.
2. In MATHEMATICS and BUSINESSSTUDIES the occurrence of grade 95 is also strangely high.
3. The prior mentioned conclusions make MATHEMATICS to have the biggest standard deviation.
4. PHYSICALEDUCATION is the most ‘normal’ subject as far as the grades are spread and the most easy for students to take (that’s also pretty usual)

The scripts for this project can be found in this link.

Thank you.

***If you have any questions please let me know.***