In almost all academic areas of study one faces situations where some kind of measurements or otherwise collected data has to be analyzed. Some statistical characteristics have to be calculated to make sense of the data set as a whole. In this project we will use Python to make some simple statistical calculations based on a set of user inputs.

In the following sections some mathematical definitions are given for the most common characteristic values used in statistics. If mathematics is not your strong suit, there is no need to panic. Every formula is followed by an example which will help you to understand what is going on.

**Mean (i.e. Average)**

The mean (^xx^) of a set of values is calculated by adding all the values in the data set together and dividing the result by the number of the values:

^x=1NN∑i=1xix^=1N∑i=1Nxi

Where NN is the number of the values (or measurements) in the data set being analyzed. For example, if in some workplace the lengths of the sick leaves (in days) were:

5, 4, 2, 3, 1, 4, 5, 2

the mean would calculated as:

5+4+2+3+1+4+5+285+4+2+3+1+4+5+28

or in other words, the mean of this data set is 3.25.

**Variance**

When the mean is known we can calculate a characteristic called variance (σ2σ2). The variance is a way of describing how the measurements are spread around the mean and it is calculated by taking the sum of the squares for each value of the data set subtracted by the mean ((xi−^x)2)((xi−x^)2) and dividing this result by N−1N−1, where NN is (as before) the number of the values in the data set:

σ2=1N−1N∑i=1(xi−^x)2σ2=1N−1∑i=1N(xi−x^)2

If we continue the earlier example and remember the the value on NN was 8 and the mean (^xx^) was 3.25, we will get the variance:

18−1((5−3.25)2+(4−3.25)2+(2−3.25)2+(3−3.25)2+(1−3.25)2+(4−3.25)2+(5−3.25)2+(2−3.25)2)18−1((5−3.25)2+(4−3.25)2+(2−3.25)2+(3−3.25)2+(1−3.25)2+(4−3.25)2+(5−3.25)2+(2−3.25)2)

or finally σ2≈2.21σ2≈2.21.

**Standard Deviation**

From a variance we can calculate the standerd deviation (σσ or SD), which is simply the square root of the variance:

σ=√σ2σ=σ2

and by continuing the familiar example, the standard deviation would be:

√2.212.21

or in other words σ≈1.49σ≈1.49.

**Using the Calculated Characteristic Values in This Project**

Your task is to implement a program which calculates

* mean (^xx^) and
* standard deviation (σσ)

for a user entered data set and prints them on the screen.

In addition the program will print a histogram (i.e. a bar chart) which helps you to visualize how the values of the data set are located in the vicinity of the mean value.

The histogram is always expressed using six categories (bars). The bounds of the categories are calculated as follows (σσ as usual means the standard deviation of the data set):

0.0–0.5⋅σ0.5⋅σ–1.0⋅σ1.0⋅σ–1.5⋅σ1.5⋅σ–2.0⋅σ2.0⋅σ–2.5⋅σ2.5⋅σ–3.0⋅σ0.0–0.5⋅σ0.5⋅σ–1.0⋅σ1.0⋅σ–1.5⋅σ1.5⋅σ–2.0⋅σ2.0⋅σ–2.5⋅σ2.5⋅σ–3.0⋅σ

In the case of our earlier example, when the σσ was calculated as 1.49 (or more accurately 1.48804761828568990378), the bounds of the categories will be:

category 0: 0.00–0.74category 1: 0.74–1.49category 2: 1.49–2.23category 3: 2.23–2.98category 4: 2.98–3.72category 5: 3.72–4.46category 0: 0.00–0.74category 1: 0.74–1.49category 2: 1.49–2.23category 3: 2.23–2.98category 4: 2.98–3.72category 5: 3.72–4.46

You should note, that in your program code you can calculate the lower and upper bounds of the categories with a loop:

**for** category\_number **in** range(**0**, **6**):

lower\_bound = category\_number \* **0.5** \* standard\_deviation

upper\_bound = (category\_number + **1**) \* **0.5** \* standard\_deviation

# Here you place the code using the calculated values.

# As an example, we'll just print the values on the screen.

**print**(f"{category\_number} {lower\_bound:.2f}-{upper\_bound:.2f}")

The decision of which category does a data set value belongs to is done as follows:

* subtract the mean value from the data set value being processed and then
* calculate the absolute value of the result.

Compare the result with the bounds of the six categories and select the matching one.

Again, continuing the familiar example where the mean was calculated as 3.25:

|5−3.25|=1.75→belongs to category 2|4−3.25|=0.75→belongs to category 1|2−3.25|=1.25→belongs to category 1|3−3.25|=0.25→belongs to category 0|1−3.25|=2.25→belongs to category 3|4−3.25|=0.75→belongs to category 1|5−3.25|=1.75→belongs to category 2|2−3.25|=1.25→belongs to category 1|5−3.25|=1.75→belongs to category 2|4−3.25|=0.75→belongs to category 1|2−3.25|=1.25→belongs to category 1|3−3.25|=0.25→belongs to category 0|1−3.25|=2.25→belongs to category 3|4−3.25|=0.75→belongs to category 1|5−3.25|=1.75→belongs to category 2|2−3.25|=1.25→belongs to category 1

You should note, even if the example didn't show it, that a value of the data set can differ from the mean so much that it does not belong to any category. In you program it means that the value in question is completely ignored when printing the histogram. There is a mathematical reason for this, but that is beyond the scope of this assignment.

One more note about the categories. If the calculation, when trying the decide the category where a data value belongs, gives a result which happens to be exactly on a boundary between two categories, the data value is considered to belong to the higher category. For example, if the absolute value of the difference between a data value and the mean ended up being 2.23 in our example, the data value would belong to the category 3.

**Program Behavior**

When the program starts it prints out the lines:

Enter the data, one value per line.

End by entering empty line.

Then the program starts reading inputs from the user and continues until the user enters an empty line. The values the user enter should be integers or floats.

If the user enters less than two numbers before the empty line, the program should print the error message:

Error: needs two or more values.

After this the program quits without further outputs.

Otherwise the program start processing the values the user entered and prints out their mean and standard deviation values:

The mean of given data was: X.XX

The standard deviation of given data was: Y.YY

Where X.XX is the mean value with two decimals ajd on annetun aineiston keskiarvo kahden Y.YY is the standard deviation with two decimals.

Näiden tilastollisten lukujen perään ohjelma tulostaa histogrammin ja lopettaa toimintansa:

After these two lines the program prints the histogram in the following format and when done, quits without forther output:

Values between std. dev. Z.ZZ-Z.ZZ: \*\*\*\*\*\*\*\*\*\*\*\*\*

Values between std. dev. Z.ZZ-Z.ZZ: \*\*

Values between std. dev. Z.ZZ-Z.ZZ: \*\*\*\*

Values between std. dev. Z.ZZ-Z.ZZ: \*

Values between std. dev. Z.ZZ-Z.ZZ: \*\*

Values between std. dev. Z.ZZ-Z.ZZ: \*

The Z.ZZ above are placeholders for the lower and upper bounds of the categories printed with two decimals. At the end of the each line the program prints as many \* characters as there are data values belong to the category in question.

There is one exception: if the standard deviation of the data set is exactly 0, the program should print:

Deviation is zero.

and quit without printing the histogram.

**Examples of Running the Program**

Below you can study a few example runs of a program to help you understand how it should behave.

**The Example Used Above**

Enter the data, one value per line.

End by entering empty line.

5

4

2

3

1

4

5

2

The mean of given data was: 3.25

The standard deviation of given data was: 1.49

Values between std. dev. 0.00-0.74: \*

Values between std. dev. 0.74-1.49: \*\*\*\*

Values between std. dev. 1.49-2.23: \*\*

Values between std. dev. 2.23-2.98: \*

Values between std. dev. 2.98-3.72:

Values between std. dev. 3.72-4.46:

**A Small Data Set**

Enter the data, one value per line.

End by entering empty line.

1

-1

0

The mean of given data was: 0.00

The standard deviation of given data was: 1.00

Values between std. dev. 0.00-0.50: \*

Values between std. dev. 0.50-1.00:

Values between std. dev. 1.00-1.50: \*\*

Values between std. dev. 1.50-2.00:

Values between std. dev. 2.00-2.50:

Values between std. dev. 2.50-3.00:

**A Boring Data Set**

Enter the data, one value per line.

End by entering empty line.

1

1

1

1

1

The mean of given data was: 1.00

The standard deviation of given data was: 0.00

Deviation is zero.

**A Slightly Larger Data Set**

Enter the data, one value per line.

End by entering empty line.

1.0

2.0

3.0

4.0

5.0

6.0

7.0

8.0

9.0

10.0

The mean of given data was: 5.50

The standard deviation of given data was: 3.03

Values between std. dev. 0.00-1.51: \*\*\*\*

Values between std. dev. 1.51-3.03: \*\*

Values between std. dev. 3.03-4.54: \*\*\*\*

Values between std. dev. 4.54-6.06:

Values between std. dev. 6.06-7.57:

Values between std. dev. 7.57-9.08:

**Special requirements**

Automatic tester and the TAs will check more than just the functionality of you program. The following list contains the special requirements your program needs to fulfill for the full score:

1. There has to be at least 4 functions you defined yourself. Some of them must have parameters and a return value (main is not counted, so you must have main plus four other functions in your program).
2. In the beginning of your Python file there must be a docsting comment explaining what the program does and who made it.
3. Every function you define (besides main) must have a docstring comment describing the purpose of the function, its parameters and return value. TAs will check this too, so even if you succeed in tricking the automatic tests, you will get caught.
4. You are not allowed to use global variables in you code. Global constant values are ok.
5. Name you variable and functions clearly with descriptive names.
6. Comment your code where neccessary. Beginning programmers often make implementation choices which are not obvious or easy to understand. Tas might get confused and sad when they don't understand what you have done.