Data frequency table calculator program

This program consists of three main parts that eventually lead to the recognition of a large number of tables according to academic books, and the written program is not necessarily optimal in terms of time standards and memory because the selected algorithms are not optimal for calculation and the purpose of writing the program The algorithms are similar to books

These three parts include the following:

1. class OutputTable

2. class Function

3. class MainWindow

OutputTable class:

This class contains the items needed to store each category of the table and is actually responsible for storage.

Function class:

The task of this class is to create a table and deliver it to the MainWindow class; In the following, we will examine the methods in the function

1. public Function (Input string):

This method constructs the class and receives a string in which each new data is on a new line and stores it in a list, then sorts the data (a condition for creating a table).

2. private int Friction\_Caculator ():

The function of this function is to calculate the rounded unit and its output is the number of digits available after the point.

3. private void Basics ():

The function of this function is to calculate the values required to start building the table; To make a table, you need the lower limit of numbers, the upper limit of numbers, the number of classes and the length of classes, which naming variables is the same as statistical naming; In the second part of this function, it is checked whether the upper limit of the last class of the table reaches the maximum amount of data or not, and if it does not, it reduces the number of classes by one unit to solve the problem.

private void Table\_Maker ():

The task of this function is to create different classes of the table and save it in the list of table classes; The table contains the following sections

· Floor number

· Downstairs

· Upper floor limit

Xi (center of the floor)

· fi (frequency)

· Fi (cumulative frequency)

. ri (relative frequency)

. gi (relative cumulative frequency)

· “fi \* (xi-average) ^ 2 "to calculate variance and standard deviation

·” fi \* (xi-average) ^ 3” to calculate skewness

. "fi \* (xi-average) ^ 4" to calculate the elongation coefficient

The average in this function is calculated as a tabular average, which is a value different from the normal average, and the method of calculating it is the sum of fixi values and dividing it by the total number of data.

private bool Table\_Caculator ():

In this function, the median values of elongation coefficient, variance, standard deviation and skewness are calculated and stored in defined local variables.

public List <OutputTable> Output ():

In this function, all defined functions are called and if there is a problem in TryParse in the class constructor function, the value Null is returned, which indicates the existence of a problem and an error is printed.

MainWindow:

The task of this class is to exchange information with the user and it has a main function that receives data from the relevant TextBox and creates a table by calling an object from the Function class and then the information is printed in Datagridview and its headers are specified.