

“Implementation of Statistical Functions”

A Mini Project Report Submitted to

P. G. Department of Computer Science & Engg. S G B Amravati University, Amravati.

For

Partial Fulfillment of MCA

Under Supervision of

Miss Shweta Gulhane Dr.V.M.Thakare/Dr.S.S.Sherekar

/Prof. A.S.Jaiswal

(Sub Guide) (Main Guide)

Submitted to

HOD

Dr. V. M. Thakare

Dept. Of Computer Science & Engg. Sant Gadge Baba Amravati University **Session summer 2021**

i

## SANT GADGE BABA AMRAVATI UNIVERSITY

P. G. DEPARTMENT OF COMPUTER SCIENCE

*CERTIFICATE*

This is to certify that **Mr. Mayur V Madadhe** studying in **MCA-I** for session summer 2021 has completed the project on **“Implementation of Statistical Functions”** as per the requirement of syllabus and this record contains satisfactory work & completed by himself.

|  |  |
| --- | --- |
| Place : | Amravati |
| Date : |  |

|  |  |  |
| --- | --- | --- |
| Signature External  (Examiner) | Signature Prof. A.S.Jaiswal  (Main Guide) | Signature  Miss Shweta Gulhane (Guide) |

ii

***Declaration By Candidate***

I am **Mr. Mayur Madadhe** studying in **MCA I** hereby declare this project for session Summer 2021 is carried by me under the supervison of **Miss Shweta Gulhane** P.G.Department of Computer Science & Engg.Sant Gadge Baba Amravati University, Amravati. The work presented here in my own and has not been copied as it is from any other sources.

|  |  |
| --- | --- |
| Place : | Amravati |
| Date : |  |

|  |
| --- |
| **Signature** |
| **(Mr. Mayur Madadhe)** |

**Acknowledgement**

Many people contribute to their best efforts to this project special thanks to the talented people at P.G.Department of Computer Science & Engg. Sant Gadge Baba Amravati University, Amravati.

I take this opportunity to express gratitude to my main guide **Prof. A.S.Jaiswal** & my sub guide **Miss Shweta Gulhane** for helping me with their experiences and encouraging time for working on this project and providing me with high quality environment infrastructure like well-equipped lab,books,etc.

Again sincere thanks to all friends, all students’ lecturers and **Dr.V.M.Thakre** of P.G.Department Of Computer Science & Engg.Sant Gadge Baba Amravati University, Amravati. For good guidance and support to improve this project.

Last but not least, I am thankful to those people who helped me directly or indirectly on this project. Thank you so much to all from bottom of my heart.

Thank You…!!

Submitted by

(Mr. Mayur Madadhe)

iv

|  |  |  |
| --- | --- | --- |
| **Sr. no** | **Contents** | **Page no.** |
| 1 | Introduction | 1 |
| 2 | Software Requirement Specification | 2 |
| 3 | Project Description | 3 |
| 4 | Source Code | 5 |
| 5 | Output | 14 |
| 6 | Testing | 17 |
| 7 | Conclusion | 18 |
| 8 | Reference | 19 |

Abstract

The project on “IMPLEMENTATIONS OF STATISTICAL FUNCTIONS” in

given projects I implements some statisticals functions for preforming some tasks on providing data and getting the desirable outputs, this statistical functions are as:

1. AVEDEV : The AVEDEV function calculates the average of absolute deviations from the mean in a given set of data.
2. AVERAGE: The Excel AVERAGE function calculates the average (arithmetic mean) of supplied numbers.

AVERAGE (number1, [number2], ...)

1. AVERAGEA : The AVERAGEA Function in Excel is a statistical function. The function calculates the average of a group of supplied values. It differs from the AVERAGE function, as it evaluates the logical values TRUE and FALSE, and numbers represented as text, whereas AVERAGE just skips these values during calculation.

AVERAGEA(value1, [value2], …)

4)AVERAGEIF: The Excel AVERAGEIF function calculates the average of numbers in a range that meet supplied criteria.

AVERAGEIF (range, criteria, [average\_range])

5)AVERAGEIFS : The Excel AVERAGEIFS function calculates the average of numbers in a range that meet one or more criteria.

AVERAGEIFS (avg\_rng, range1, criteria1, [range2], [criteria2], ...)

# Introduction

Statistical functions apply a mathematical process to a group of cells in a worksheet. The goal of statistical analysis is to identify trends. A retail business, for example, might use statistical analysis to find patterns in unstructured and semi-structured customer data that can be used to create a more positive customer experience and increase sales.

The theory of statistics provides a basis for the whole range of techniques, in both [study](https://en.wikipedia.org/wiki/Study_design) [design](https://en.wikipedia.org/wiki/Study_design) and [data analysis](https://en.wikipedia.org/wiki/Data_analysis), that are used within applications of [statistics](https://en.wikipedia.org/wiki/Statistics). The theory covers approaches to [statistical-decision](https://en.wikipedia.org/wiki/Statistical_decision_theory) problems and to [statistical inference,](https://en.wikipedia.org/wiki/Statistical_inference) and the actions and deductions that satisfy the basic principles stated for these different approaches. Within a given approach, statistical theory gives ways of comparing statistical procedures; it can find a best possible procedure within a given context for given statistical problems, or can provide guidance on the choice between alternative procedures.

Apart from philosophical considerations about how to make statistical inferences and decisions, much of statistical theory consists of [mathematical statistics](https://en.wikipedia.org/wiki/Mathematical_statistics), and is closely linked to [big data,](https://en.wikipedia.org/wiki/Probability_theory) statistics, and to [optimization](https://en.wikipedia.org/wiki/Mathematical_optimization).

In the given project there are some statistical functions are as:

* + AVEDEV
  + AVERAGE
  + AVERAGEA
  + AVERAGEIF
  + AVERAGEIFS

# Software Requirement Specification

* **Software Requirement**

|  |  |
| --- | --- |
| **Name Of Component** | **Specification** |
| Operating System | Microsoft Windows 10 |
| Language Used | Java |
| Database | MySQL |
| Software Development Kit | Java JDK1.7 |
| Browser | Any Of Mozilla,Opera,Chrome etc |
| WebsServer | APACHE |

* **Hardware Requirement**

|  |  |
| --- | --- |
| **Name Of Component** | **Specification** |
| Processor | AMD Ryzen 5 3450U |
| Ram | 8 GB |
| Hard Disk | 1 TB |
| Cache Memory | 2 MB |
| Basic Components | Battery, Mouse and standard QWERTY keyboard |

1. **Project Description**

Modules:

1. AVEDEV
2. AVERAGE
3. AVERAGEA
4. AVERAGEIF
5. AVERAGEIFS

1)AVEDEV : The AVEDEV function calculates the average of absolute deviations from the mean in a given set of data. Variance and standard deviation functions deal with negative deviations by squaring deviations before they are averaged. AVEDEV handles negative values by working only with absolute values.

2)AVERAGE: The Excel AVERAGE function calculates the average (arithmetic mean) of supplied numbers. AVERAGE can handle up to 255 individual arguments, which can include numbers, cell references, ranges, arrays, and constants.

=AVERAGE (number1, [number2], ...)

3)AVERAGEA : The AVERAGEA Function in Excel is a statistical function. The function calculates the average of a group of supplied values. It differs from the AVERAGE function, as it evaluates the logical values TRUE and FALSE, and numbers represented as text, whereas AVERAGE just skips these values during calculation.

=AVERAGEA(value1, [value2], …)

4)AVERAGEIF: The Excel AVERAGEIF function calculates the average of numbers in a range that meet supplied criteria. AVERAGEIF criteria can include logical operators (>,<,<>,=) and wildcards (\*,?) for partial matching.

=AVERAGEIF (range, criteria, [average\_range])

5)AVERAGEIFS : The Excel AVERAGEIFS function calculates the average of numbers in a range that meet one or more criteria. The criteria used for AVERAGEIFS can include logical operators (>,<,<>,=) and wildcards (\*,?) for partial matching.

Syntax:

=AVERAGEIFS (avg\_rng, range1, criteria1, [range2], [criteria2], ...)

# Source Code

import java.util.Arrays;

/\*

\* To change this license header, choose License Headers in Project Properties.

\* To change this template file, choose Tools | Templates

\* and open the template in the editor.

\*/

/\*\*

\*

\* @author DELL

\*/

public class Madadhe\_mp extends javax.swing.JFrame {

/\*\*

\* Creates new form Madadhe\_mp

\*/

public Madadhe\_mp() {

initComponents();

}

/\*\*

\* This method is called from within the constructor to initialize the form.

\* WARNING: Do NOT modify this code. The content of this method is always

\* regenerated by the Form Editor.

\*/

@SuppressWarnings("unchecked")

// <editor-fold defaultstate="collapsed" desc="Generated Code">

private void initComponents() {

panel1 = new java.awt.Panel();

scrollbar1 = new java.awt.Scrollbar();

menuBar1 = new java.awt.MenuBar();

menu1 = new java.awt.Menu();

menu2 = new java.awt.Menu();

a2 = new javax.swing.JButton();

jLabel1 = new javax.swing.JLabel();

AVERAGE = new javax.swing.JButton();

ave1 = new javax.swing.JTextField();

r1 = new javax.swing.JTextField();

r2 = new javax.swing.JTextField();

avedev1 = new javax.swing.JTextField();

javax.swing.GroupLayout panel1Layout = new javax.swing.GroupLayout(panel1);

panel1.setLayout(panel1Layout);

panel1Layout.setHorizontalGroup(

panel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGap(0, 100, Short.MAX\_VALUE)

);

panel1Layout.setVerticalGroup(

panel1Layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGap(0, 100, Short.MAX\_VALUE)

);

menu1.setLabel("File");

menuBar1.add(menu1);

menu2.setLabel("Edit");

menuBar1.add(menu2);

setDefaultCloseOperation(javax.swing.WindowConstants.EXIT\_ON\_CLOSE);

a2.setText("AVERAGE DEVIATION");

a2.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

a2ActionPerformed(evt);

}

});

jLabel1.setText("Statistical Functons");

AVERAGE.setText("AVERAGE");

AVERAGE.addMouseListener(new java.awt.event.MouseAdapter() {

public void mouseClicked(java.awt.event.MouseEvent evt) {

AVERAGEMouseClicked(evt);

}

public void mouseEntered(java.awt.event.MouseEvent evt) {

AVERAGEMouseEntered(evt);

}

public void mouseExited(java.awt.event.MouseEvent evt) {

AVERAGEMouseExited(evt);

}

});

AVERAGE.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

AVERAGEActionPerformed(evt);

}

});

ave1.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

ave1ActionPerformed(evt);

}

});

r1.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

r1ActionPerformed(evt);

}

});

r2.addActionListener(new java.awt.event.ActionListener() {

public void actionPerformed(java.awt.event.ActionEvent evt) {

r2ActionPerformed(evt);

}

});

javax.swing.GroupLayout layout = new javax.swing.GroupLayout(getContentPane());

getContentPane().setLayout(layout);

layout.setHorizontalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addGroup(layout.createSequentialGroup()

.addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING, false)

.addComponent(ave1)

.addGroup(layout.createSequentialGroup()

.addComponent(AVERAGE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 141, Short.MAX\_VALUE)

.addComponent(r1, javax.swing.GroupLayout.PREFERRED\_SIZE, 60, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addComponent(avedev1)))

.addGroup(javax.swing.GroupLayout.Alignment.TRAILING, layout.createSequentialGroup()

.addComponent(a2)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED, 97, Short.MAX\_VALUE)

.addComponent(r2, javax.swing.GroupLayout.PREFERRED\_SIZE, 60, javax.swing.GroupLayout.PREFERRED\_SIZE)))

.addGroup(layout.createSequentialGroup()

.addGap(80, 80, 80)

.addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED\_SIZE, 150, javax.swing.GroupLayout.PREFERRED\_SIZE)))

.addContainerGap(17, Short.MAX\_VALUE))

);

layout.setVerticalGroup(

layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)

.addGroup(layout.createSequentialGroup()

.addGap(19, 19, 19)

.addComponent(jLabel1, javax.swing.GroupLayout.PREFERRED\_SIZE, 33, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(35, 35, 35)

.addComponent(ave1, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.UNRELATED)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(AVERAGE)

.addComponent(r1, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addGap(26, 26, 26)

.addComponent(avedev1, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addGap(18, 18, 18)

.addGroup(layout.createParallelGroup(javax.swing.GroupLayout.Alignment.BASELINE)

.addComponent(a2, javax.swing.GroupLayout.PREFERRED\_SIZE, 20, javax.swing.GroupLayout.PREFERRED\_SIZE)

.addComponent(r2, javax.swing.GroupLayout.PREFERRED\_SIZE, javax.swing.GroupLayout.DEFAULT\_SIZE, javax.swing.GroupLayout.PREFERRED\_SIZE))

.addContainerGap(38, Short.MAX\_VALUE))

);

pack();

}// </editor-fold>

private void AVERAGEActionPerformed(java.awt.event.ActionEvent evt) {

String nums = ave1.getText();

String[] parts = nums.split(",");

Integer[] ints = new Integer[parts.length];

Double average = 0.0;

Double sum = 0.0;

for (int i = 0; i < parts.length; i++)

{

ints[i] = Integer.parseInt(parts[i]);

// System.out.println(Integer.parseInt(parts[i]));

}

for(int i = 0; i < ints.length; i++)

{

sum = sum + ints[i];

}

average = sum/ints.length;

r1.setText(average.toString());

}

private void AVERAGEMouseClicked(java.awt.event.MouseEvent evt) {

// TODO add your handling code here:

}

private void AVERAGEMouseEntered(java.awt.event.MouseEvent evt) {

// TODO add your handling code here:

}

private void AVERAGEMouseExited(java.awt.event.MouseEvent evt) {

// TODO add your handling code here:

}

private void ave1ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

}

private void r1ActionPerformed(java.awt.event.ActionEvent evt) {

r1.setText("default");

// TODO add your handling code here:

}

private void r2ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO add your handling code here:

}

private void a2ActionPerformed(java.awt.event.ActionEvent evt) {

// TODO ad

String nums = avedev1.getText();

String[] parts = nums.split(",");

Integer[] ints = new Integer[parts.length];

Double average;

Double avgdev = 0.0;

Double sum = 0.0;

for (int i = 0; i < parts.length; i++)

{

ints[i] = Integer.parseInt(parts[i]);

// System.out.println(Integer.parseInt(parts[i]));

}

for(int i = 0; i < ints.length; i++)

{

sum = sum + ints[i];

}

average = sum/ints.length;

for(int i = 0; i < ints.length; i++)

{

avgdev = avgdev+Math.abs(ints[i] - average);

}

avgdev = avgdev/ints.length;

r2.setText(avgdev.toString());

}

/\*\*

\* @param args the command line arguments

\*/

public static void main(String args[]) {

/\* Set the Nimbus look and feel \*/

//<editor-fold defaultstate="collapsed" desc=" Look and feel setting code (optional) ">

/\* If Nimbus (introduced in Java SE 6) is not available, stay with the default look and feel.

\* For details see http://download.oracle.com/javase/tutorial/uiswing/lookandfeel/plaf.html

\*/

try {

for (javax.swing.UIManager.LookAndFeelInfo info : javax.swing.UIManager.getInstalledLookAndFeels()) {

if ("Nimbus".equals(info.getName())) {

javax.swing.UIManager.setLookAndFeel(info.getClassName());

break;

}

}

} catch (ClassNotFoundException ex) {

java.util.logging.Logger.getLogger(Madadhe\_mp.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (InstantiationException ex) {

java.util.logging.Logger.getLogger(Madadhe\_mp.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (IllegalAccessException ex) {

java.util.logging.Logger.getLogger(Madadhe\_mp.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

} catch (javax.swing.UnsupportedLookAndFeelException ex) {

java.util.logging.Logger.getLogger(Madadhe\_mp.class.getName()).log(java.util.logging.Level.SEVERE, null, ex);

}

//</editor-fold>

/\* Create and display the form \*/

java.awt.EventQueue.invokeLater(new Runnable() {

public void run() {

new Madadhe\_mp().setVisible(true);

}

});

}

// Variables declaration - do not modify

private javax.swing.JButton AVERAGE;

private javax.swing.JButton a2;

private javax.swing.JTextField ave1;

private javax.swing.JTextField avedev1;

private javax.swing.JLabel jLabel1;

private java.awt.Menu menu1;

private java.awt.Menu menu2;

private java.awt.MenuBar menuBar1;

private java.awt.Panel panel1;

private javax.swing.JTextField r1;

private javax.swing.JTextField r2;

private java.awt.Scrollbar scrollbar1;

// End of variables declaration

}

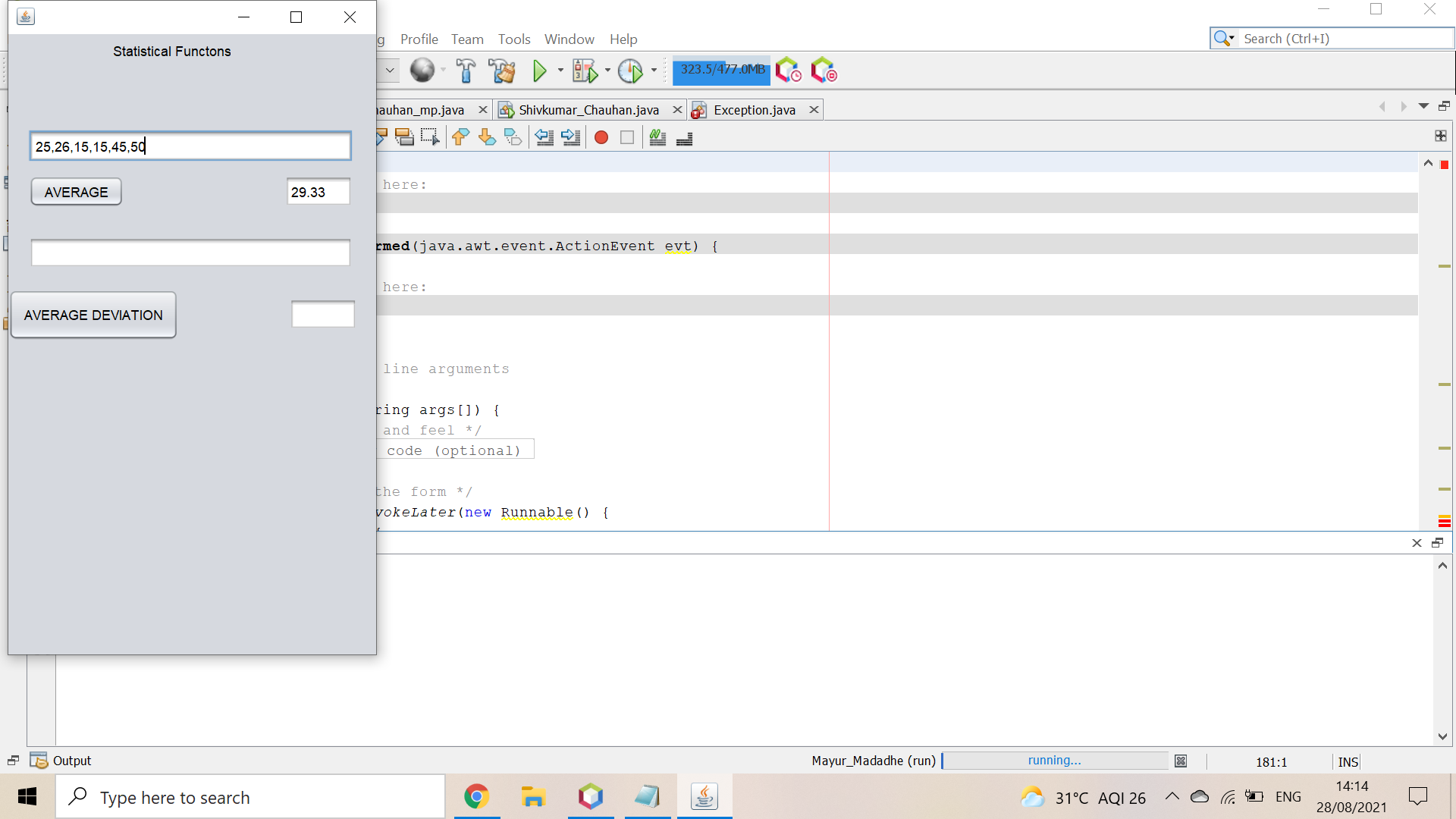
# Output

# 

In the above image we can see that how AVEDEV () works.

The AVEDEV() calculates the standard deviation of given numbers.

The formula for AVEDEV() is = Σ|x − μ|**N**



In the above image we can see that how AVERAGE () works.

The AVERAGE() calculates the Average or Mean of the given numbers.

The formula for AVERAGE() is

Average = Sum of all observation /Total no. of observation

# 6.Testing

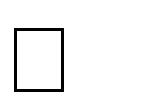
Software testing is defined as an activity to check whether the actual results match the expected results and to ensure that the software system is Defect free. It involves the execution of a software component or system component to evaluate one or more properties of interest.

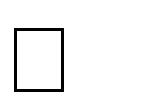
Software testing also helps to identify errors, gaps, or missing requirements in contrary to the actual requirements. It can be either done manually or using automated tools.

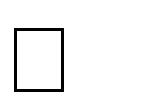
In simple terms, Software Testing means the Verification of Application Under Test (AUT).

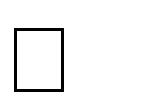
Testing is important because software bugs could be expensive or even dangerous. Software bugs can potentially cause monetary and human loss, and history is full of such examples.

Here are the benefits of using software testing:

**Cost-Effective**: It is one of the important advantages of software testing. Testing any IT project on time helps you to save your money for the long term. In case if the bugs caught in the earlier stage of software testing, it costs less to fix.

**Security**: It is the most vulnerable and sensitive benefit of software testing. People are looking for trusted products. It helps in removing risks and problems earlier.

**Product quality**: It is an essential requirement of any software product. Testing ensures a quality product is delivered to customers.

**Customer Satisfaction**: The main aim of any product is to give satisfaction to their Custome

# 7.Conclusion

We successfully designed and implemented a minin project on Implementation on Statictical Functions, that allows direct interaction with our system. In Statistical functions there are too many aspects many of functions like Correlation, Count, CountA, CountBlank, Cover in Statistical Functions. you can do all this thing at this project . It saves lots of time and efforts.

This is to summarize, this system is well-developed and satisfying all the requirements of the users with test performance.

# 8.Future Scope

1. **To Present Facts in Definite Form:**

We can represent the things in their true form with the help of figures. Without a statistical study, our ideas would be vague and indefinite.

1. **Precision to the Facts:**

The statistics are presented in a definite form so they also help in condensing the data into important figures. So statistical methods present meaningful information. In other words statistics helps in simplifying complex data to simple-to make them understandable.

1. **Comparisons:**

After simplifying the data, it can be correlated as well as compared. The relationship between the two groups is best represented by certain mathematical quantities like average or coefficients etc. Comparison is one of the main functions of statistics as the absolute figures convey a very less meaning.

1. **Forecasting:**

Statistics is not only concerned with the above functions, but it also predicts the future course of action of the phenomena. We can make future policies on the basis of estimates made with the help of Statistics.

1. **It Enlarges Knowledge:**

Whipple rightly remarks that “Statistics enables one to enlarge his horizon”. So when a person goes through various procedures of statistics, it widens his knowledge pattern. It also widens his thinking and reasoning power. It also helps him to reach to a rational conclusion.

# References

* + [http://www.github.com](http://www.github.com/)
  + [https://1000projects.org//](https://1000projects.org/)
  + [http://www.tutorialpoint.com](http://www.tutorialpoint.com/)
  + [http://www.tutorialkart.com](http://www.tutorialkart.com/)