SCALABLE CROSS-PLATFORM SOFTWARE DESIGN:

Coursework#3 – Assessment on JAVA and GUI Topic

Document date: 05/09/2025

PART A: IN-LAB CODING ASSESSMENT

Read the instruction provided in the cover sheet.

Reminder:

Submit one NetBeans project that contains your answers to **all Part A questions**.

Include all source code, resources, and configuration files. Submit as a .zip file.

Important:

1. Create a **NetBeans JavaFX Application project** (JDK 8).
2. Part A focuses on the model layer (core classes, data structures, logic). No GUI is required for Part A; **you may leave the default "Hello World" UI as is**.
3. You will continue the same NetBeans project for Part B at home to complete the full application, including the GUI. **Do not start a new project for Part B**.

Question 1: [Mark 15%]

Task 1:

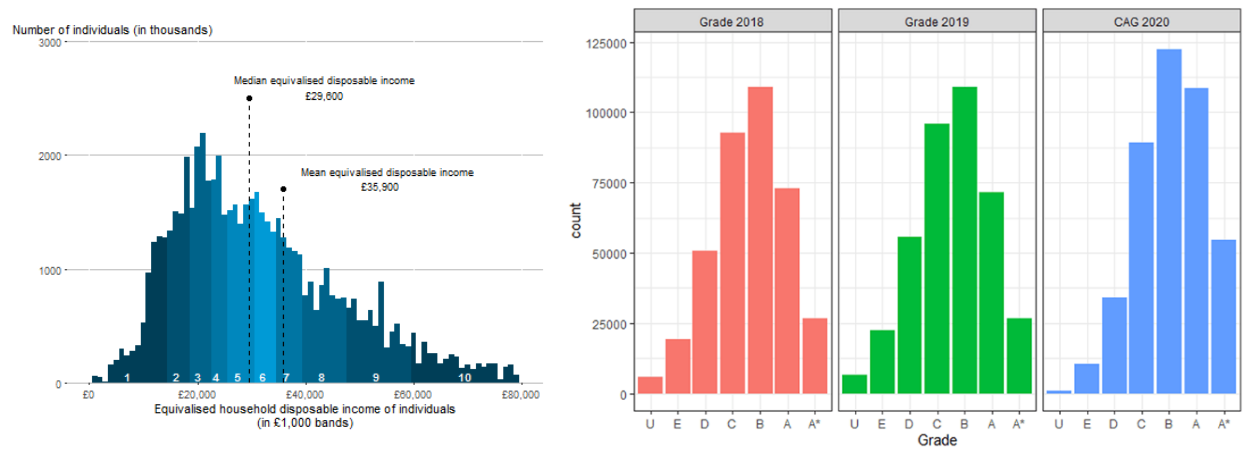
- Develop a JAVA `package` called `binmethod` which contains the implementation of Square root choice, Sturge's and Rice rule bin rule formulae (see Review on the mathematics 1 box). The package will comprise of three concrete classes (i.e. the three bin rule formulae) and an `Abstract` Parent class called `BinFormulae`.
- The client code `Unit_test_binmethod.java` will drive your code without any modifications.

Unit_test_binmethod.java

```
/*
   Client code to test the binmethod package
*/

import java.util.List;
import java.util.Arrays;

// the bin rule formulae must be implemented in binmethod package
import binmethod.*;

public class Unit_test_binmethod
{
    public static void main(String[] args) {

       // Create array of data
       List<Double> exampleData = Arrays.asList(1., 1., 3., 3., 3., 2., 10., 10., 6., 6., 6.);

       // test SturgesFormula class
       SturgesFormula SturgesInstance = new SturgesFormula(exampleData);
       SturgesInstance.calculateNumberOfBins();
       System.out.printf("By Sturges Formula: %d \n", SturgesInstance.getNumberOfBins());

       // test SquareRootChoice class
       SquareRootChoice SquareRootChoiceInstance = new SquareRootChoice(exampleData);
       SquareRootChoiceInstance.calculateNumberOfBins();
       System.out.printf("By Square Root Formula: %d \n", SquareRootChoiceInstance.getNumberOfBins());

       // test RiceRule class
       RiceRule RiceRuleInstance = new RiceRule(exampleData);
       RiceRuleInstance.calculateNumberOfBins();
       System.out.printf("By Rice Rule Formula: %d \n", RiceRuleInstance.getNumberOfBins());

    }
}
```

Review on the mathematics 1

In statistics, histogram is used to present the frequency distribution of certain sampled data into several clusters. For example, the distribution of income in the country or the distribution of A-level grades.



The first step in drawing a histogram is to "bin" the data into several cluster of value marker, i.e., salary range or grade in the examples given above.

There are several ways to determine the "number of bin" and different "number of bin" may reveal different features of data. In general, a good bin should

- Bins should be all the same size. *See below how to calculate the bin size.*
- Bins should include all the data.

It is common to display histogram using different bin number which can be calculated using:

- Square root choice formula
  $k = \sqrt{n}$
- Sturge's formula
  $$k = 3.3log_{10}n + 1$$
- Rice rule formula
  $$k = 2\sqrt[3]{n}$$

where, k is the number of bins, n is the total number of observation/measurement data and $log_{10}()$ is logarithm basis-10 operator.

Question 2: [Mark 25%]

Task 2a: Develop a `package` named `statutils` which has the capability to calculate basic statistical figures (mean, variance, max, min, median and standard deviation) of data samples.

Task 2b: Add a capability to count the number of samples in each bin.

Task 2c: Add a capability to perform histogram normalisation (see Review on the mathematics 2 box).

Task 2d:

- Create a client test class named `Unit_test_statutils.java`
- It must contain a `public static void main(String[] args)` that tests the operations from Tasks 2a, 2b, and 2c.
- No user input or external files; the test must run from NetBeans with one click.
- You may use `Unit_test_binmethod.java` from Task 1 as a template—copy the structure, then adapt it for `statutils`. Do not submit a verbatim copy; the tests must be your own and target Tasks 2a–2c.

You may make several classes to perform Task 2a – Task 2d and they should all in the `statutils` package.

Review on the mathematics 2

A histogram show "how many times" a certain sample is observed from measurement, i.e., frequency of data. Thus, its magnitude changes depending on the total number of samples. This type-of histogram is called **un-normalised histogram**.

A **normalised histogram**, on the other hand, shows what is the *probability* of a certain value will be observed if there is an infinite number of samples. Mathematically, this means that the area under the curve is equal to 1. In most cases, a normalised histogram is more useful than the un-normalised histogram in representing a large sampled data as in our task.

$$\bar{f}(bin) = \frac{f(bin)}{\sum_{bin}\left[f(bin) \times w\right]}$$

where, $\bar{f}(\text{bin})$ and $f(\text{bin})$ is the normalised and unnormalised frequency, respectively.

The w denotes the width of each bin, given by

$$w = \frac{\max(\text{data samples}) - \min(\text{data samples})}{k}$$

where, k is the number of bins which is calculated depending on the kind of bin method used, see Review on the mathematics 1 box.

END OF PART A

PART B: TAKE-HOME COURSEWORK

Read the instruction provided in the cover sheet.

Important:

1. You will continue the same NetBeans project, from Part A, for Part B at home to complete the full application, including the GUI. You can download your Part A submission from Moodle. **Do not start a new project for Part B**.

Question 3: [Mark 10%]

Task 3a: *Reporting*: UML diagram for `binmethod package`.

- Draw a UML class diagram of `binmethod` exactly as submitted in Part A.
- Add a brief (<100 words) note on class roles and key relationships.
- If modified in Part B (with generative AI or not): Add an updated UML and a short rationale (<100 words) stating what changed and why.
- If no changes: Write "No changes to `binmethod` in Part B."
- AI disclosure: If AI was used to modify `binmethod`, include the AI Usage Report (see Part B rules).

Task 3b: *Reporting*: UML diagram for `statutils package`.

- Draw a UML class diagram of `statutils` exactly as submitted in Part A as Task 2a.
- Add a brief (<100 words) note on class roles and key relationships.
- If modified in Part B (with generative AI or not): Add an updated UML and a short rationale (<100 words) stating what changed and why.
- If no changes: Write "No changes to `statutils` in Part B."
- AI disclosure: If AI was used to modify `statutils`, include the AI Usage Report (see Part B rules).

Question 4: [Mark 10%]

Task 4a: Develop a package named `mathutils` in which contains a class to perform data fitting. The class will have `static` member functions to calculate the fitting parameters of the histogram (i.e. μ, σ, and α parameters in the Review on the mathematics 3 box). The `static` function will use `GaussianCurveFitter` class implemented within the commons-math library from Apache.

Task 4b: *Reporting*

- Provide evidence, in your report, that this package performs its intended task.

- AI disclosure: If AI was used, include the AI Usage Report (see Part B rules).

Review on the mathematics 3

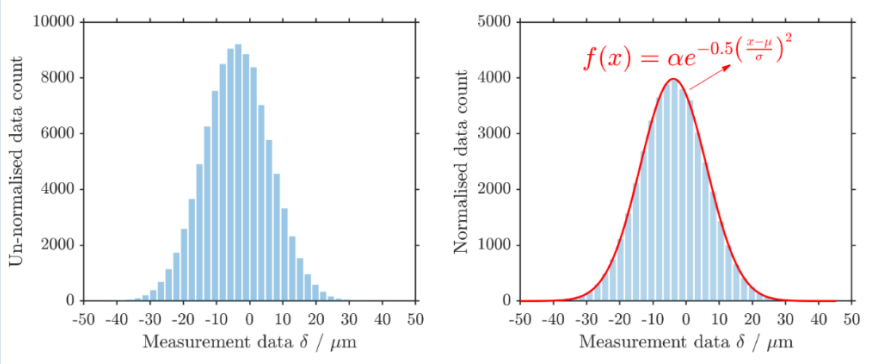
You need a normalised histogram to fit with a Probability Density Function (PDF). In our case, Normal PDF is appropriate and is given by

$$f(x) = \alpha\, e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2},$$

where the PDF parameters are,
α is the normalisation coefficient
μ is the centre of the PDF
σ is the width of the PDF.

There are several ways to fit the histogram to a PDF, the most well-known method is called Levenberg-Marquardt method.

Figure shows an example of an unnormalised histogram (left) and normalised with fitted Normal PDF (right).



WE ARE NOT GOING INTO THE DETAIL OF MARQUARDT METHOD. IT IS QUITE MATHEMATICAL AND OUT OF THE SCOPE OF THE MODULE.

I HAVE PROVIDED A NETBEANS PROJECT IN MOODLE IN WHICH I DEMONSTRATE THE USE OF WELL-DEVELOPED OPEN-SOURCED LIBRARY FROM APACHE WHICH IMPLEMENT THE MARQUARDT METHOD. DOCUMENTATION OF THE LIBRARY CAN BE SEEN IN http://commons.apache.org/proper/commons-math/apidocs/overview-summary.html

Question 5: [Mark 40%]

Task 5a:

You will now bring together all the elements that your have developed previously to complete application with graphical user interface.

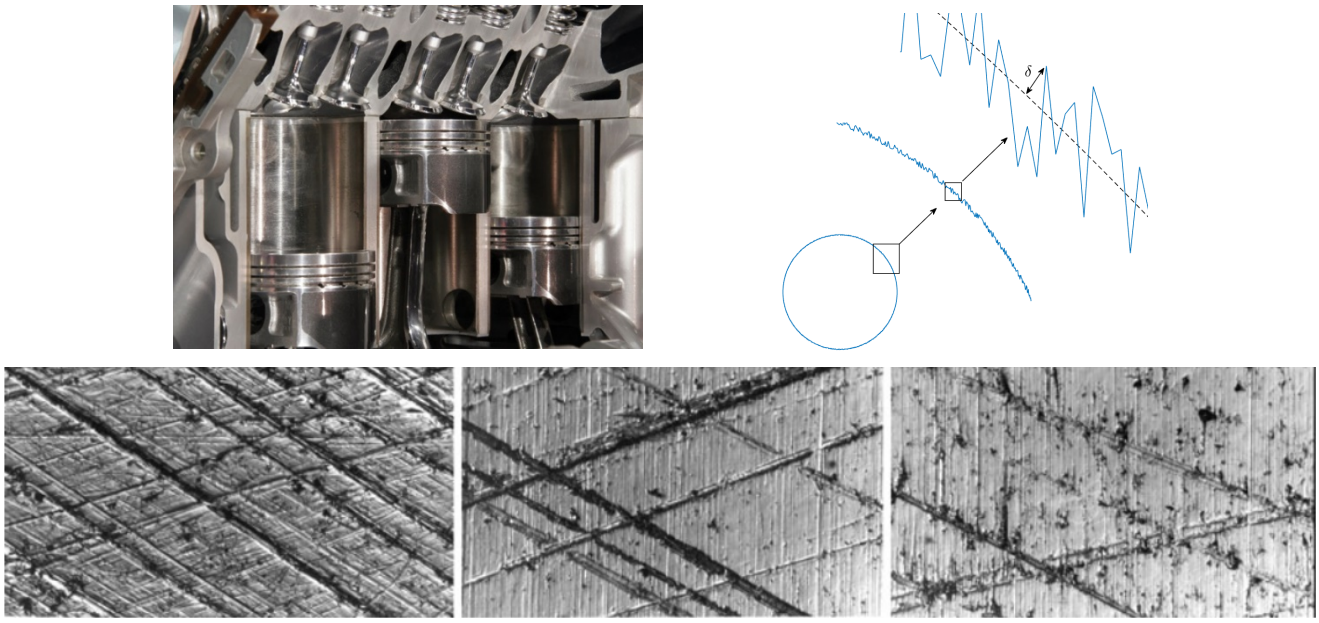


Figure 1 Illustration of the cylinder and the roughness on the surface of an engine cylinder.

Presume that your task is to develop the accompanying software for the optical tomography equipment. The equipment will output a text file which lists the roughness as the deviation δ in the unit of meter from an ideal circle (see Fig. 1).

For illustration purposes, the first few lines of such text file are given in Fig. 2,

6.000199e-05

-4.671694e-05
-1.083731e-04
-1.894823e-04
1.119588e-04
…

Figure 2 Few lines of the content of the measurement device in the unit of meter.

➢ Develop and implement an operational Graphical User Interface application satisfying the following **technical specifications**:

1. Has a user-friendly user interface. Developed using JavaFX 8. The GUI will be developed using SceneBuilder, producing an accompanying FXML file.
2. Has a capability to stream input file from the text file, to have the text file as an input. See the accompanying measurementData.txt as an example.
   (*Do not hard-coded the text file, I may test your code with different* `.txt` *file*)
3. Calculate and display the mean, variance, median and standard deviation of δ
4. Plot the *normalised* histogram of the deviation δ, with an option to choose different bin method.
5. Fit and plot the histogram with a Probability Density Function model (PDF).
6. Save the histogram and the fitted PDF as a Bitmap png file.
7. The software will display the fitting parameters, including: α, μ and σ for Normal Probability Density Function (see Review on the mathematics 3).

Task 5b: *Reporting* Draw the hierarchical order of the GUI nodes.

Task 5c: *Reporting* Draw and discuss the GUI Event Handling UML diagram (aka sequence diagram) of your app.

End of document.