Data analysis – Lab 4-5

Probability and Distributions

Requirements:

- Datasets: *AutoSurvey.csv*
- Programming language: R/Python/Java
- Provide solutions for the following questions
- Submit your solutions (report and code) in one file. Name your file with your full name and student ID.
- You should put comments in code to clarify classes, methods and important lines of code. In the report, you should include a cover page with your name and ID, and the content of questions, explanations/solutions of implementation, and results.

Questions:

Given the first 20 records of the dataset,

// Random variables

Q1. Define the random variables of Gender, Type, Purchased, VehicleAge, Mileage, and MPG. **Find** their probability mass/density functions. **Program** to compute means, variances, and standard deviations of the random variables, and display the graphs of probability mass/density functions. **(70pts)**

// Jointly distributed random variables

Q2. Assume the random variables of Gender, Type, Purchased, VehicleAge, Mileage, and MPG are jointly distributed. **Find** the marginal probability density function of MPG. **Program** to estimate the probability of MPG. **(30pts)**

// Prediction

Q3. Predict the MPG of the last 3 records using the above program and compare the predicted results with the actual values. **(10pts)**

Instruction of programming in Java

Do the following tasks

Question 1.

1. Create a new Java project in NetBeans or Eclipse, e.g., lab4_5

2. Create a class of a random variable, e.g., RandomVar. Implement methods of

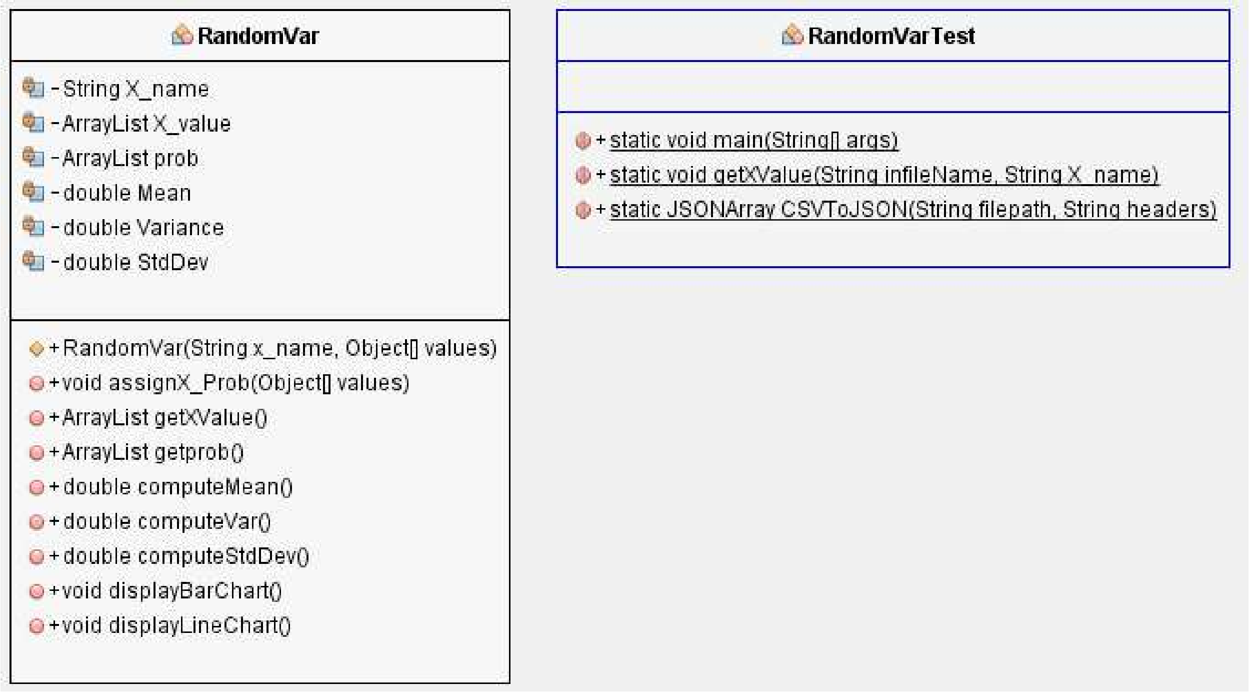
- getting the values of the random variable, and computing the probability mass function

- computing mean, variance, standard deviation

- displaying the bar/line chart of the probability mass function.

3. Apply to the given dataset: create a main/controller class to read the input records and display results, e.g., RandomVarTest

Images of source code

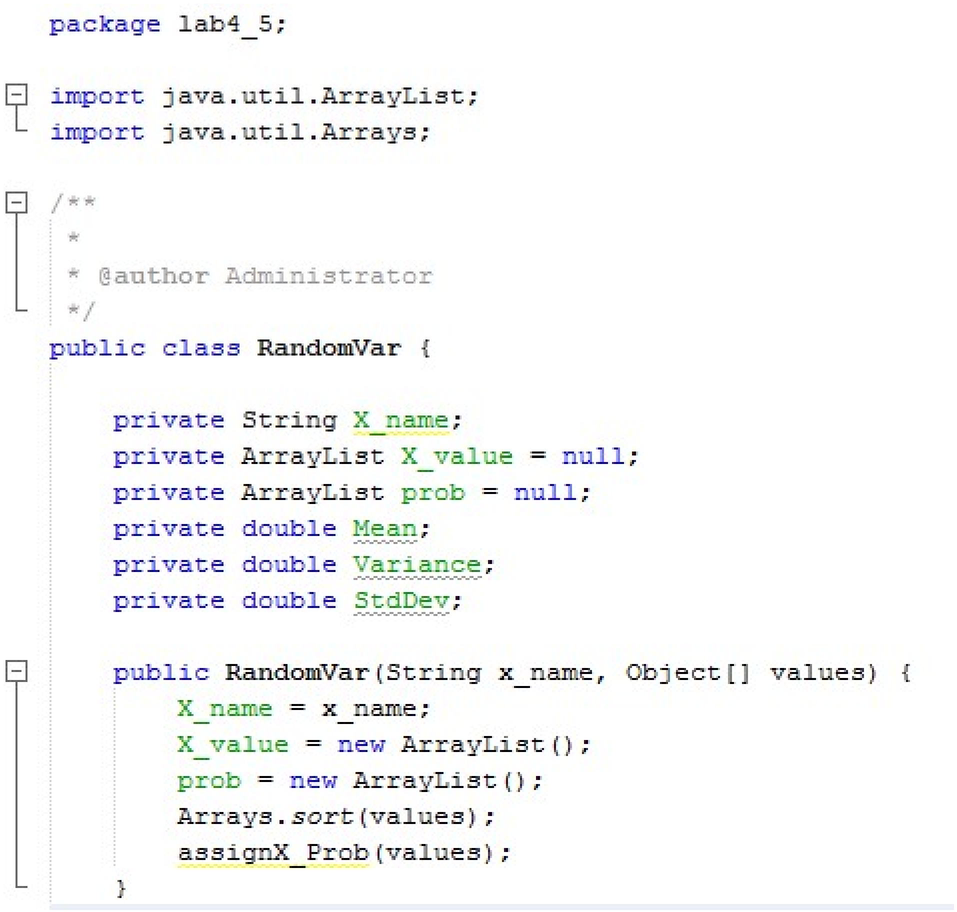


RandomVar

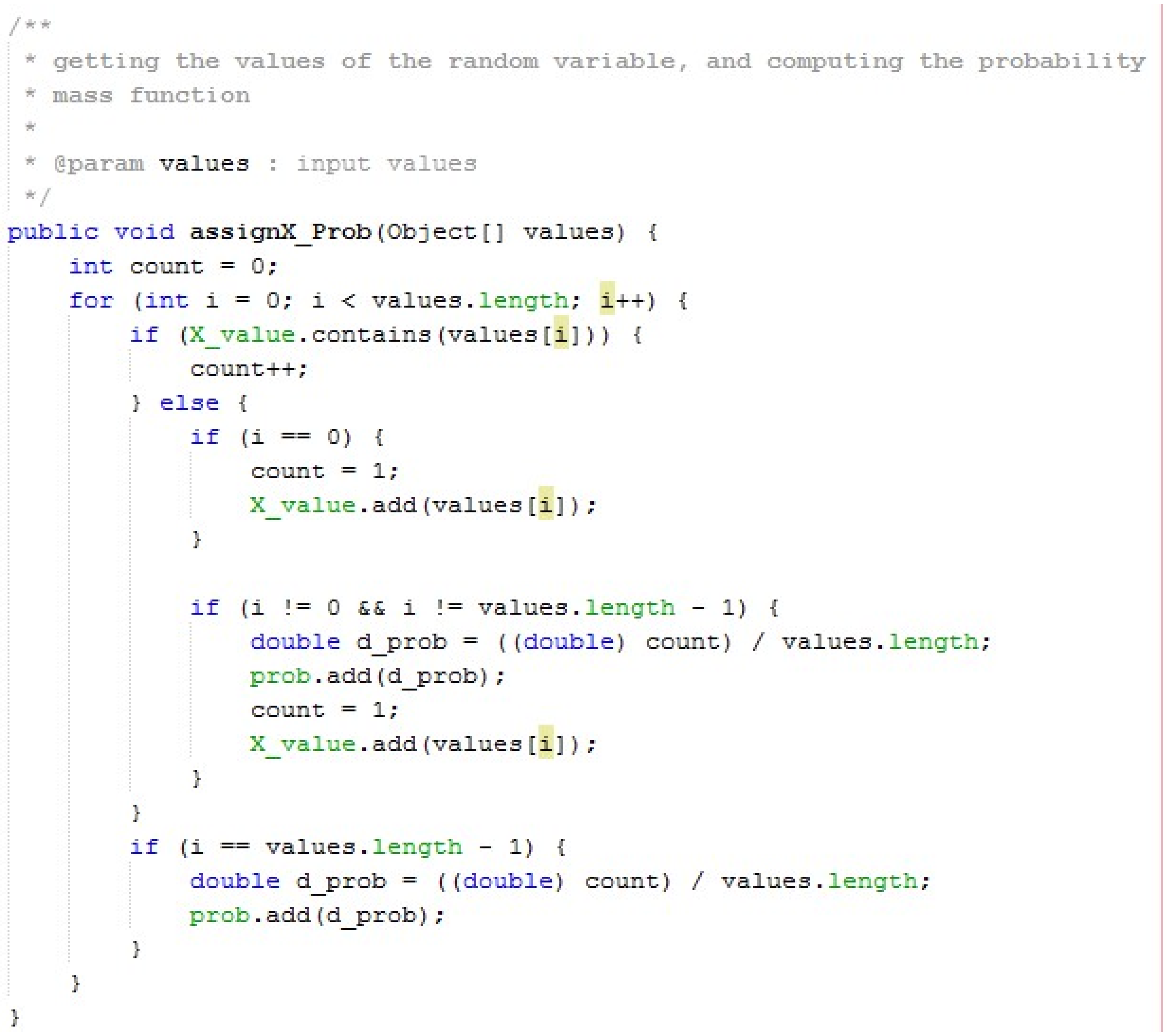
```
-String X_name
-ArrayList X_value
-ArrayList prob
-double Mean
-double Variance
-double StdDev

+RandomVar(String x_name, Object[] values)
+void assignX_Prob(Object[] values)
+ArrayList getXValue()
+ArrayList getprob()
+double computeMean()
+double computeVar()
+double computeStdDev()
+void displayBarChart()
+void displayLineChart()
```

RandomVarTest

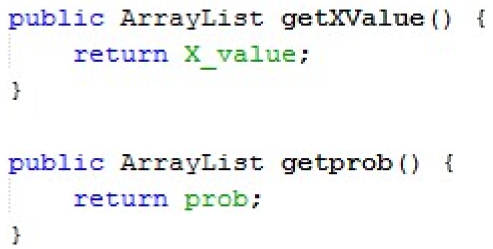
```
+static void main(String[] args)
+static void getXValue(String infileName, String X_name)
+static JSONArray CSVToJSON(String filepath, String headers)
```



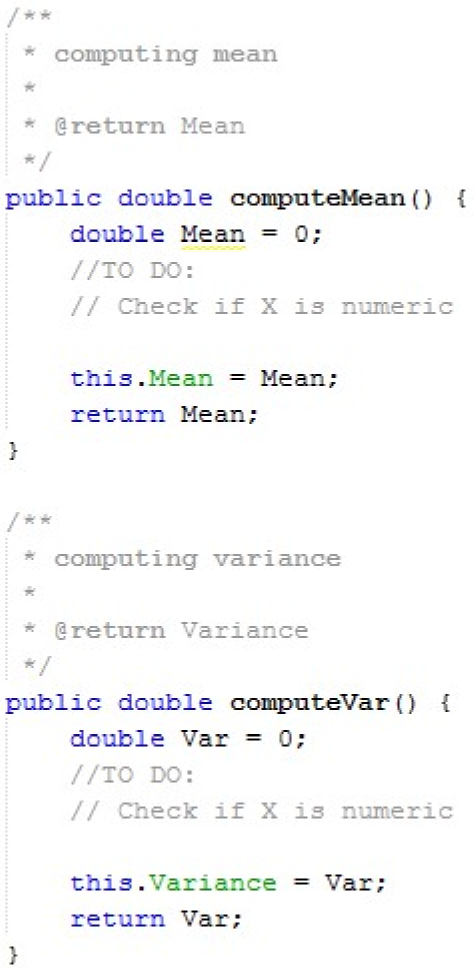
```
package lab4_5;

import java.util.ArrayList;
import java.util.Arrays;

/**
 *
 * @author Administrator
 */
public class RandomVar {

    private String X_name;
    private ArrayList X_value = null;
    private ArrayList prob = null;
    private double Mean;
    private double Variance;
    private double StdDev;

    public RandomVar(String x_name, Object[] values) {
        X_name = x_name;
        X_value = new ArrayList();
        prob = new ArrayList();
        Arrays.sort(values);
        assignX_Prob(values);
    }
```



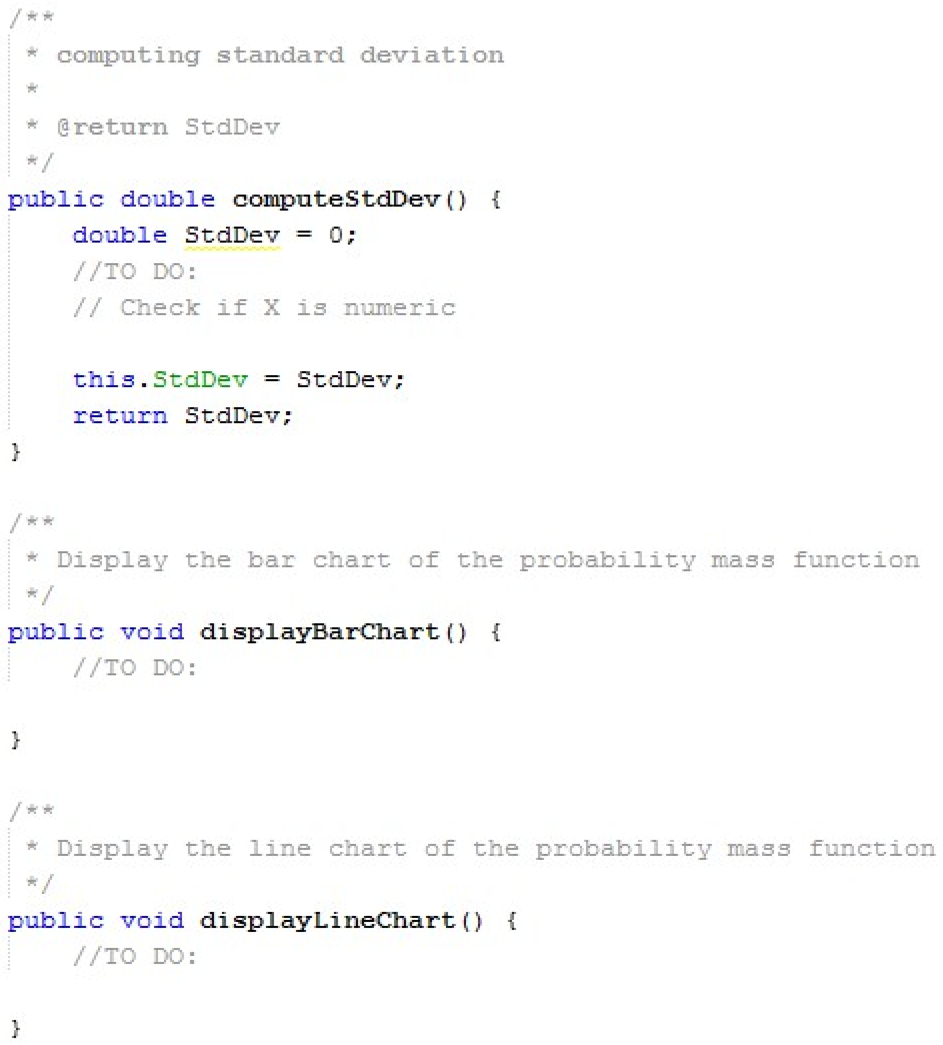
```
/**
 * getting the values of the random variable, and computing the probability
 * mass function
 *
 * @param values : input values
 */
public void assignX_Prob(Object[] values) {
    int count = 0;
    for (int i = 0; i < values.length; i++) {
        if (X_value.contains(values[i])) {
            count++;
        } else {
            if (i == 0) {
                count = 1;
                X_value.add(values[i]);
            }

            if (i != 0 && i != values.length - 1) {
                double d_prob = ((double) count) / values.length;
                prob.add(d_prob);
                count = 1;
                X_value.add(values[i]);
            }
        }
        if (i == values.length - 1) {
            double d_prob = ((double) count) / values.length;
            prob.add(d_prob);
        }
    }
}
```



```
public ArrayList getXValue() {
    return X_value;
}

public ArrayList getprob() {
    return prob;
}
```



```
/**
 * computing mean
 *
 * @return Mean
 */
public double computeMean() {
    double Mean = 0;
    //TO DO:
    // Check if X is numeric

    this.Mean = Mean;
    return Mean;
}

/**
 * computing variance
 *
 * @return Variance
 */
public double computeVar() {
    double Var = 0;
    //TO DO:
    // Check if X is numeric

    this.Variance = Var;
    return Var;
}
```



```
/**
 * computing standard deviation
 *
 * @return StdDev
 */
public double computeStdDev() {
    double StdDev = 0;
    //TO DO:
    // Check if X is numeric

    this.StdDev = StdDev;
    return StdDev;
}

/**
 * Display the bar chart of the probability mass function
 */
public void displayBarChart() {
    //TO DO:

}

/**
 * Display the line chart of the probability mass function
 */
public void displayLineChart() {
    //TO DO:

}
```

References

[1] Line chart: https://www.tutorialspoint.com/jfreechart/jfreechart_line_chart.htm