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.

Questions:

Given the first 20 records in 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.

// 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.

Instruction of programming in Java

Cont.

```
package lab4_5;

import java.io.BufferedWriter;
import java.io.File;
import java.io.FileWriter;
import java.io.IOException;
import java.io.Reader;
import java.nio.file.Files;
import java.nio.file.Paths;
import java.util.ArrayList;
import java.util.HashSet;

import org.apache.commons.csv.CSVFormat;
import org.apache.commons.csv.CSVParser;
import org.apache.commons.csv.CSVRecord;
import org.json.simple.JSONArray;
import org.json.simple.JSONObject;
```

```
public class RandomVarTest {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
        // "Gender???","Type","Purchased","VehicleAge","Mileage","MPG
        String X_name = "MPG";
        ArrayList alValue = getXValue("in/AutoSurvey.csv", X_name);
        RandomVar randomVar = new RandomVar(X_name, alValue.toArray());

        System.out.println(randomVar.getXValue());
        System.out.println(randomVar.getprob());
        randomVar.displayLineChart();
    }
```

```java
public static ArrayList getXValue(String infileName, String X_name) {
    // Read infileName
    JSONArray array = null;
    try {
        array = CSVToJSON(infileName, X_name);
    } catch (IOException e) {
        System.out.println(e);
    }
    ArrayList alValue = new ArrayList();
    for (Object object : array) {
        if (object instanceof JSONObject) {
            JSONObject item = (JSONObject) object;
            String value = item.get(X_name).toString();
            alValue.add(value);
            System.out.println(value);

        }
    }
    return alValue;
}
```

```java
@SuppressWarnings("unchecked")
public static JSONArray CSVToJSON(String filepath, String... headers)
                                    throws IOException {
    Reader reader = Files.newBufferedReader(Paths.get(filepath));
    CSVParser csvParser = new CSVParser(reader,
            CSVFormat.DEFAULT.withFirstRecordAsHeader().withIgnoreHeaderCase().withTrim());

    // store converted records as a list
    JSONArray records = new JSONArray();

    for (CSVRecord csvRecord : csvParser) {
        // create json object to store csv record
        JSONObject record = new JSONObject();

        // access values using selected header names
        for (String header : headers) {
            record.put(header, csvRecord.get(header));
        }

        records.add(record);
    }

    csvParser.close();
    return records;
}
```