

Homework #2

Instructor: Dr. Zafeirakis Zafeirakopoulos Name: Ömer Faruk Bitikçioğlu Student Id: 161044010
 Assistant: Gizem Süngü 161044010

Course Policy: Read all the instructions below carefully before you start working on the assignment, and before you make a submission.

- It is not a group homework. Do not share your answers to anyone in any circumstance. Any cheating means at least -100 for both sides.
- Do not take any information from Internet.
- No late homework will be accepted.
- For any questions about the homework, send an email to gizemsungu@gtu.edu.tr.
- Submit your homework (both your latex and pdf files in a zip file) into the course page of Moodle.
- Save your latex, pdf and zip files as "Name_Surname_StudentId".{tex, pdf, zip}.
- The answer which has only calculations without any formula and any explanation will get zero.
- The deadline of the homework is 07/06/20 23:55.
- I strongly suggest you to write your homework on L^AT_EX. However, hand-written paper is still accepted **IFF** your hand writing is **clear and understandable to read**, and the paper is well-organized. Otherwise, I cannot grade your homework.
- You do not need to write your Student Id on the page above. I am checking your ID from the file name.

Problem 1:

(10+10+10+10+10+10+40 = 100 points)

WARNING: Please show your OWN work. Any cheating can be easily detected and will not be graded.

For the question, please follow the file called manufacturing_defects.txt while reading the text below.

In each year from 2000 to 2019, the number of manufacturing defects in auto manufacturers were counted. The data was collected from 14 different auto manufactory companies. The numbers of defects for the companies are indicated in 14 columns following the year column. Assume that the number of manufacturing defects per auto company per year is a random variable having a Poisson(λ) and that the number of defects in different companies or in different years are independent.

(Note: You should implement a code for your calculations for each following subproblem. You are free to use any programming languages (Python, R, C, C++, Java) and their related library.)

(a) Give a table how many cases occur for all companies between 2000 and 2019 for each number of defects (# of Defects).

Hint: When you check the file you will see: # of Defects = {0, 1, 2, 3, 4}.

(b) Estimate λ from the given data.

$\lambda = 0.7$

(c) Update Table 1 in Table 2 with Poisson predicted cases with the estimated λ .

(d) Draw a barplot for the actual cases (Table 2 in column 2) and the predicted cases (Table 2 column 3) with respect to # of defects. You should put the figure.

\# of Defects	\# of cases in all company between the years
0	144
1	91
2	32
3	11
4	2

Table 1: Actual cases

\# of Defects	\# of cases in all companies between the years	Predicted \# of cases in all companies between the years
0	144	139,043885
1	91	97,330720
2	32	34,065752
3	11	7,948675
4	2	1,391018

Table 2: Actual vs. Predicted Cases

(e) According to the barplot in (c), does the poisson distribution fit the data well? Compare the values of the actual cases and the values of the poisson predicted cases, and write your opinions about performance of the distribution.

Actual cases and predicted values are perfectly fit. Some predictions are a bit below of the actual cases, some of them a bit above the actual ones but they are very close that we can say the distribution is working well.

(f) According to your estimations above, write your opinions considering your barplot and Table 2. Which company do you prefer to buy a car? Why?

Most of the time the possibility of a defect happens is very rare. Bu it is not imposibble. It can happen time to time. So, we can not say cars are very safe to use. A defection can cause fatal injuries, so it is a big problem to be honest.

(g) Paste your code that you implemented for the subproblems above. Do not forget to write comments on your code.

- The common code block for all subproblems

```
private static final int MAX_DEFECT = 4;
private static final int TIME = 20;
private static final int NUM_OF_COMPANIES = 14;

/**
 * Find the factorial of given number
 * @param num the number to be processed
 * @return the factorial of the given number
 */
private static double fact(int num) {
    if (num < 0) {
        throw new InvalidParameterException();
    }
    int result = 1;
    for (int mult = 2 ; mult <= num; ++mult) {
        result = result * mult;
    }
    return result;
}
```

- The code block for (a)

```

/**
 * Calculates and returns the total number of actual cases
 * for each defect.
 * @return the number of actual cases for each defect
 */
private static int[] giveCases() {
    try {
        BufferedReader br = new BufferedReader(new FileReader("manufacturing_defects.txt"));
        String line = br.readLine();
        int[] cases = new int[MAX_DEFECT+1];
        while (line != null) {
            String[] row = line.split("\t");
            // Count the number of cases for all companies and all years
            // for each number of defects (# of defects -> 0,1,2,3,4)
            for (int i = 2; i < row.length; ++i) {
                int index = Integer.parseInt(row[i]); // 0,1,2,3 or 4
                cases[index]++;
            }
            line = br.readLine();
        }
        br.close();
        return cases;
    } catch (IOException e) {
        e.printStackTrace();
    }
    return null;
}

```

- The code block for (b)

```

/**
 * Calculates the lambda according to the given data
 * @return the lambda value
 */
private static double estimateLambda() {
    double lambda = 0.0;
    try {
        BufferedReader br = new BufferedReader(new FileReader("manufacturing_defects.txt"));
        String line = br.readLine();
        while (line != null) {
            String[] row = line.split("\t");
            double total = 0.0;
            for (int i = 2; i < row.length; ++i) {
                total += Integer.parseInt(row[i]);
            }
            lambda += total / NUM_OF_COMPANIES; // Calculate the mean for each year
            line = br.readLine();
        }
        br.close();
    } catch (IOException e) {
        e.printStackTrace();
    }
    lambda /= TIME; // Calculate the mean for all years
    return lambda;
}

```

- The code block for (c)

```
/**
 * Calculates and returns the predicted case values.
 * Uses Poisson distribution to calculate the predictions.
 * @param totalCase the number of total cases for all the time
 * @param lambda the lambda value
 * @return the predicted case values
 */
private static double[] findPredictions(int totalCase, double lambda) {
    double[] predictions = new double[MAX_DEFECT+1];
    double negLambda = (-1) * lambda;
    for (int i = 0; i <= MAX_DEFECT; ++i) {
        predictions[i] = ((Math.exp(negLambda) * Math.pow(lambda, i)) / fact(i)) * totalCase;
    }
    return predictions;
}
```

- The code block for (d)

