MATH 118: Statistics and Probability **(Due: 07/06/21)**

Homework #2

Instructor: Dr. Zafeirakis Zafeirakopoulos *Name:* *Student Id:*
Assistant: Gizem Süngü

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 LaTeX. 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.

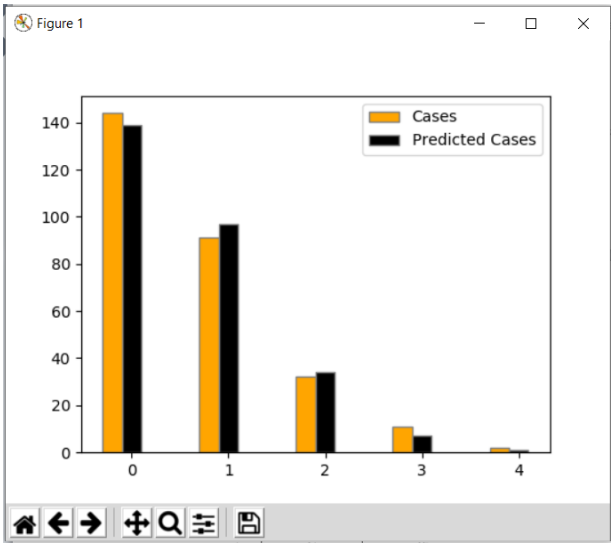
(d) Draw a barplot for the actual cases (Table 2 in column 2) and the predicted cases (Table 2 column 3) with respect to # of defecrs. You should put the figure.

\# 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
1	91	97
2	32	34
3	11	7
4	2	1

Table 2: Actual vs. Predicted Cases



(a) The barplot for PART D

(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.

I think yes, the data fits well. it is really close. For example when I compare the data for 0, compare 144 with 139, 139/144*100 means 96.52 so it means there %96.5 similarity between the oredicted data from poisson distribution and real data. Also other cases differecy is really small to when we compare them like it seen from barplot graphic. But when the data value shrinks the difference percent get increase. For example for case 4, real case is 2 and predicted cases is 1. It is %50 difference but when we look it as difference of numbers, it is really close to each other. So I think the performance of this poisson distrubition is really 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?

To find out which company is best we need to look the manufacturing_defects file or the console output. Like it seen in console output, I would prefer to buy car from last comapny(14th). Because when I check I didn't see any 3 cases in 1 year or any 4 cases in 1 year. I think that's really good. Also when I look to barplot I saw 0 cases are the most, 1 cases are seconder and 2 cases are third. Even the 3 case in one year and 4 case in one year is really small like it seen in data, I think if none, it's better. So I would prefer the last company because of no 3 or 4 cases in one year.

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

- The common code block for all subproblems

  Paste here. Your code should read the file and compute other things which the following subproblems need.

```
            /** Reading infos in manufacturing_defects.txt */
    public void readFile(String fileName) {
        String readedLine;

        try{
            BufferedReader bufferRead = new BufferedReader( new FileReader(fileName) );
            /** Reading all lines one by one. */
            for(int i=0; ( readedLine = bufferRead.readLine() ) != null && !readedLine.
                                                        equals("\n") ; i++){
                /** Creating a list to hold the current line elements that seperated by (
                                                            tab) */
                List<String> tempList = new ArrayList<String>();
                /** Splitting by tab character */
                tempList = Arrays.asList( readedLine.split("\\t") );
                //System.out.println( tempList.toString() );
                if(tempList.size() > 1){ /** if line contains a empty character, dont
                                                            enter the block. */
                    /** Creating a integer list to put all splitted string elements */
                    List<Integer> intList = new ArrayList<Integer>();
                    tempList.forEach(item -> intList.add(Integer.valueOf(item) ) );
                    /** Allocate memory */
                    myList.add( new ArrayList<Integer>() );
                    /** Adding all elements to list. */
                    for(int j=0; j<intList.size(); j++){
                        myList.get(i).add( intList.get(j) );
                    }
                }
            }
            bufferRead.close(); /** Closing reader to prevent source leaks. */
        }  //End of try
        catch (IOException e) {
            System.out.println("File reading error. Check permissions and file.");
            e.printStackTrace();
        }   //End of catch.

        /** Creating memory location for toWrite ArrayList. (They will be written to temp
                                                        file.) */
        for(int i=0; i<=4; i++)
            toWrite.add( new ArrayList<String>() );


    }
```

- The code block for (a)
  Paste here. Your code should compute the values in Table 1 column 2.

```
        public int getNumberOfDefects(int whichCase){
            int total = 0;
            for(int row=0; row<myList.size(); row++){
                for(int column=2; column<myList.get(column).size(); column++){ //Starting
                                                                    from column 2 because
                                                                   column0 and 1 doesn't show
                                                                    case.
                    if(whichCase == myList.get(row).get(column))
                        total++;
                }
            }
            toWrite.get(whichCase).add( Integer.toString(whichCase) );
            toWrite.get(whichCase).add( Integer.toString(total) ); // writing cases to
                                                                   file as integer.
            return total;
        }
```

- The code block for (b)
  Paste here. Your code should compute λ.

```
        public float findRatePeriod(boolean printData){
        howManyTimesChecked = 0;
        int totalEvents = 0;
        for(int row=0; row<myList.size(); row++)
            for(int column=2; column<myList.get(column).size(); column++){ //Starting
                                                                   from column 2 because column0
                                                                   and 1 doesnt show case.
                totalEvents += myList.get(row).get(column);
                howManyTimesChecked++;
            }

        float lambda = (float)totalEvents/howManyTimesChecked; /** events/time */
        if(printData == true)
            System.out.println("Total " + totalEvents + " events and " +
                                                                   howManyTimesChecked + " times,
                                                                    So mean(rate) =  $\lambda$ is
                                                                    " + lambda);
        return lambda;
    }
```

- The code block for (c)
  Paste here. Your code should compute the values in Table 2 column 3.

```
            public double getPredictedCases(int whichCase){
            double result = 0, probability = 0;
            //false in parameter is to not printing data.
            double expLambda = Math.exp( -1 * findRatePeriod(false) ); // exp means e^(
                                                                   findRatePeriod) number from
                                                                   java.lang
            double lambdaToK = Math.pow( findRatePeriod(false) , whichCase);
            //Formula is -> $\lambda$^k * e^(- $\lambda$) / k!

            probability = expLambda * lambdaToK / factorial(whichCase);
            result = probability * howManyTimesChecked; //To find number of cases.
            // System.out.printf("Probability is %.2g | predicted cases is probability*
                                                                   timesChecked = %.2g", result,
                                                                   result*howManyTimesChecked);


            toWrite.get(whichCase).add( Integer.toString((int)result) ); // writing
                                                                   predicted cases to file as
                                                                   integer.
            return result;
        }
```

- The code block for (d)
  Paste here. Your code should draw the barplot.

```
import numpy as np
import matplotlib.pyplot as plt


with open('mat118_1801042656_hw2/temp.txt', 'r') as readingFile: # r emans
                                                    read.
    # Reading line by line in while loop
    cases = []    #name of cases
    bar1 = []     #bar for cases
    bar2 = []     #bar for predicted cases
    line = readingFile.readline()  #reading first line
    while line != '':  # Until end of file.
        print(line, end='')
        split_string = line.split("\t") #File was splitted by tab character.
        cases.append( split_string[0] )
        bar1.append(  int(split_string[1]) ) #writing as int
        bar2.append(  int(split_string[2]) ) #writing as int
        line = readingFile.readline() #reading next line in file.

# Bar positions in x axis.
r1 = np.arange(len(bar1))   #getting length of bar1 for next bar.
r2 = [x + 0.2 for x in r1]

plt.bar(r1, bar1, width = 0.2, color = 'orange', edgecolor = 'gray', capsize=
                                                    10, label='Cases')          #
                                                    bar1
plt.bar(r2, bar2, width = 0.2, color = 'black', edgecolor = 'gray', capsize=
                                                    10, label='Predicted Cases') #
                                                    bar2


# general layout
plt.xticks( [i + 0.2 for i in range(len(bar1))], cases) # writing cases in
                                                    graphic with 0.2 distance
                                                    between bars.
plt.legend() # showing names in rectangle box.
plt.show()   # Show graphic
```