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

Homework #2

Instructor: Dr. Zafeirakis Zafeirakopoulos *Name:* Esra Eryılmaz *Student Id:* 171044046
Assistant: Gizem Süngü *171044046*

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}.

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

Table 1: Actual cases

(b) Estimate λ from the given data.

$\lambda = mean = \frac{events}{time} = total\ number\ of\ defects\ /\ total\ number\ of\ cases$

$Total\ number\ of\ defects = (0 * 144) + (1 * 91) + (2 * 32) + (3 * 11) + (4 * 2) = 196$
$Total\ number\ of\ cases = 144 + 91 + 32 + 11 + 2 = 280$

$\lambda = \frac{196}{280} = 0.7$

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

\# of Defects	\# of cases in all companies between the years	Predicted \# of cases in all companies between the years
0	144	139.0439
1	91	97.3307
2	32	34.0658
3	11	7.9487
4	2	1.391

Table 2: Actual vs. Predicted Cases

(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.

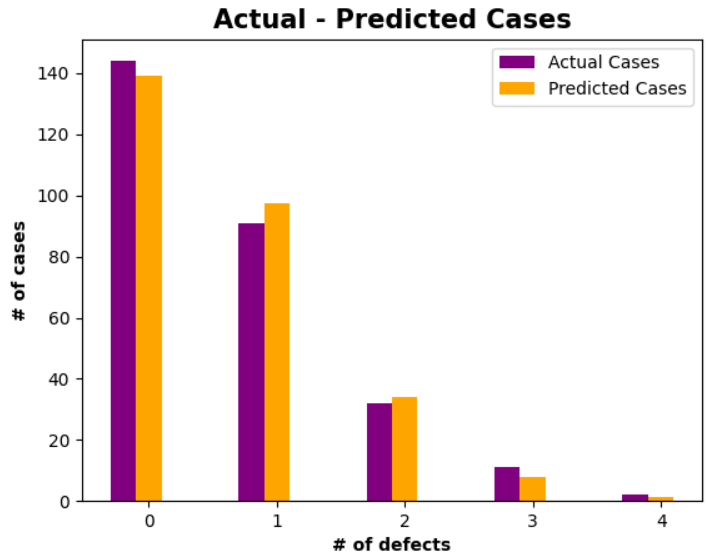

Figure 1: Actual - Predicted Cases

(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.

As we can see in the barplot, the actual and predicted cases of each defect number are very close to each other.

Total # actual cases = 144 + 91 + 32 + 11 + 2 = 280
Total # predicted cases = 139.0439 + 97.3307 + 34.0658 + 7.9487 + 1.391 = 279.7801

Also total number of actual cases and total number of predicted cases are very close to each other either.
So we can say that the poisson distribution fit the data well.
Poisson distribution performance is good according to this data.

(f) According to your estimations above, write your opinions considering your barplot and Table 2. Do you think that road transportation is dangerous for us? Whether yes or no, explain your reason.

*(Actual) Total # defects = (0*144) + (1*91) + (2*32) + (3*11) + (4*2) = 196*
*(Predicted) Total # defects =(0*139.0439) + (1*97.3307) + (2*34.0658) + (3*7.9487) + (4*1.391) = 194.8724*

While the predicted total number of defects is 194.8724, the actual total number of defects is 196.
These numbers are very close to each other, but I think I can still say that road transportation is dangerous for us because the actual number of accidents is higher.

(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.

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

fileName = 'manufacturing_defects.txt'
defects = [0, 1, 2, 3, 4]

# Driver code
def main():
    arr = []
    # Read the file line by line and each line is a list in itself
    with open(fileName, 'r') as infile:
        data = infile.readlines()
        for i in data:
            line = i.split()
            arr.append(line)
    infile.close()

    # Problem a (find total number of cases in all company)
    table1 = number_of_cases(arr)

    # Problem b (find lambda)
    lambdaa, total_case = estimate_lambda(table1)

    # Problem c (find poisson predicted cases with estimated lambda)
    table2 = find_predicted_cases(table1, lambdaa, total_case)

    # Problem d (draw barplot)
    draw_barplot(table2)

```

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

```
# problem a
def number_of_cases(arr):
    table1 = []
    print("\n# of Defects    | # of Cases")
    print("----------------------------")
    for k in defects:
        count = 0
        for line in arr:
            count = count + line[1:].count(str(k))  # Take each column except first
    column
        table1.append(count)
        print(k ,"\t\t|\t", table1[k])
    return table1

```

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

```
# problem b
def estimate_lambda(table1):
    total_defects, total_case = 0, 0
    for i in defects:
        total_defects += i * table1[i]
        total_case += table1[i]

    lambdaa = total_defects/total_case    # Lambda = total number of defects / total
    number of cases
    print("\nLambda : " ,lambdaa)
    return lambdaa, total_case

```

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

```
# problem c
def find_predicted_cases(table1, lambdaa, total_case):
    table2 = []
    temp = 0
    print("\n# of Defects    |  # of Cases   |  Predicted # of Cases")
    print("-------------------------------------------------------")

    for k in defects:
        temp = (pow(math.e, -1*lambdaa) * pow(lambdaa, k)) / math.factorial(k)  # apply
    poisson distribution formula
        temp *= total_case
        temp = round(temp,4)
        table2.append([table1[k], temp])
        print(k ,"\t\t|\t", table2[k][0], "\t|\t" , table2[k][1])

    return table2

```

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

```
# problem d
def draw_barplot(table2):
    # Set width of bar
    barWidth = 0.2

    # Set height of bar
    actual, predicted = [], []
    for i in defects:
        actual.append(table2[i][0])     # height of actual case bar
        predicted.append(table2[i][1])  # height of predicted case bar

    # Set position of bar on X axis
    x_axis = np.arange(len(defects))

```

```
15      # Make the plot
16      plt.bar(x_axis, actual, color ='purple', width = barWidth, label ='Actual Cases')
17      plt.bar(x_axis + barWidth, predicted, color ='orange', width = barWidth, label ='
        Predicted Cases')
18
19      plt.title('Actual - Predicted Cases', fontweight ='bold', fontsize = 15)
20      plt.xlabel('# of defects', fontweight ='bold', fontsize = 10)
21      plt.ylabel('# of cases', fontweight ='bold', fontsize = 10)
22      plt.xticks(x_axis + barWidth, defects)
23
24      plt.legend()
25      plt.show()
26
```