IT302: PROBABILITY AND STATISTICS

ASSIGNMENT 3

```
import numpy as np
import random

# Generating a normal distribution of test scores for 1000 employees
with mean 60 and standard deviation of 15
test_scores = np.random.normal(60, 15, 1000)
test_scores = np.clip(test_scores, 0, 100)

# Creating a dictionary to represent the different salary brackets and
the number of employees to allocate
salaries = {
    "3LPM": 50,
    "2LPM": 150,
    "1LPM": 200,
    "75KPM": 250,
    "40KPM": 350
}

# Allocating employees to salary brackets based on the specified
number of employees
allocated_employees = []

for salary, num_employees in salaries.items():
    for _ in range(num_employees):
        allocated_employees.append((salary,
random.choice(test_scores)))

# Print the allocated employees with their corresponding salaries and
test scores
#for i, (salary, test_score) in enumerate(allocated_employees,
start=1):
    #print(f"Employee {i}: Test Score = {test_score:.2f}, Salary =
{salary}")
```

Question 1: Find the probability of employees that get increment of different percentages (consider test score fallows the normal distribution)

Answer:

```
# Defining the salary brackets as given in question
salary_brackets = {
    "70%": (90, 100),
    "60%": (80, 90),
    "50%": (70, 80),
    "30%": (60, 70),
```

```
    "25%": (50, 60),
    "20%": (40, 50),
    "10%": (30, 40),
    "0%": (0, 30),

}

# Initializing a dictionary
employees_within_bracket = {salary: 0 for salary in salary_brackets}

# Counting the number of employees within each salary bracket based on
their test scores
for _, test_score in allocated_employees:
    for salary, (min_score, max_score) in salary_brackets.items():
        if min_score <= test_score <= max_score:
            employees_within_bracket[salary] += 1

# Calculating the total number of allocated employees
total_allocated_employees = len(allocated_employees)

# Calculaing the probabilities of employees receiving different
increments
probability = {salary: count / 1000 for salary, count in
employees_within_bracket.items()}

# Printing the answer
for salary, prob in probability.items():
    print(f"Probability of receiving {salary} increment: {prob:.2%}")
```

```
Probability of receiving 70% increment: 1.90%
Probability of receiving 60% increment: 7.20%
Probability of receiving 50% increment: 17.10%
Probability of receiving 30% increment: 24.20%
Probability of receiving 25% increment: 24.60%
Probability of receiving 20% increment: 14.80%
Probability of receiving 10% increment: 7.40%
Probability of receiving 0% increment: 2.80%
```

Question 2: What is the total increase in the total salaries of employees

Answer:

```
# Defining the salary increment percentages
increment_percentages = {
    "70%": 0.7,
    "60%": 0.6,
    "50%": 0.5,
    "30%": 0.3,
    "25%": 0.25,
    "20%": 0.2,
```

```
    "10%": 0.1,
    "0%": 0
}

# Initializing a variable to keep track of the total increase
total_salary_increase = 0

# Iterating through allocated employees and calculating the salary
increase
for salary, test_score in allocated_employees:
    increment_percentage = None
    for range_name, (min_score, max_score) in salary_brackets.items():
        if min_score <= test_score <= max_score:
            increment_percentage = increment_percentages[range_name]
            break

    if increment_percentage is not None:
        # Calculating the new salary
        current_salary = int(salary[:-3])
        new_salary = current_salary * (1 + increment_percentage)
        total_salary_increase += new_salary - current_salary

        # print(f"Employee with {salary} and test score {test_score}
gets a {range_name} increment.")
        # print(f"New salary: {new_salary:.2f}LPM\n")

# Printing the final answer
print(f"Total increase in total salaries: {total_salary_increase:.1f}
Lakhs")
```

```
Total increase in total salaries: 10402.0 Lakhs
```

Question 3: What is the probability of employees getting (a) a promotion (b) Promotion and minimum of 25% hike (c) Only hike (d) Neither hike nor promotion?

Answer:

```
# Defining the required probabilities of each event, and getting its
value from previously stored dictionary
prob_promotion = probability["70%"] + probability["60%"] +
probability["50%"] + probability["30%"]
prob_promotion_25_hike = probability["70%"] + probability["60%"] +
probability["50%"] + probability["30%"]
prob_only_hike = probability["25%"] + probability["20%"] +
probability["10%"]
prob_neither_hike_nor_promotion = probability["0%"]

# Printing the probabilities
print(f"Probability of employees getting a promotion:
{prob_promotion}")
```

```
print(f"Probability of employees getting promotion and minimum 25% hike: {prob_promotion_25_hike}")
print(f"Probability of employees getting only a hike: {prob_only_hike}")
print(f"Probability of employees getting neither hike nor promotion: {prob_neither_hike_nor_promotion}")
```

```
Probability of employees getting a promotion: 0.504
Probability of employees getting promotion and minimum 25% hike: 0.504
Probability of employees getting only a hike: 0.468
Probability of employees getting neither hike nor promotion: 0.028
```