1. The number of customer returns in a retail chain per day follows a Poisson distribution at a rate of 25 returns per day. Write Python code to answer the following questions:

(a) Calculate the probability that the number of returns exceeds 30 in a day.

(b) If the chance of fraudulent return is 0.05, calculate the probability that there will be at least 2 fraudulent returns in any given day.

ANS-

```
import scipy.stats as stat
lambda_returns = 25
prob_exceeds_30 = 1 - stats.poisson.cdf(30, lambda_returns)
print(f"Probability that the number of returns exceeds 30 in a day:
{prob_exceeds_30:.4f}")
n_returns = 25
p_fraudulent = 0.05
prob_at_least_2_fraudulent = 1 - stats.binom.cdf(1, n_returns,
p_fraudulent)
print(f"Probability that there will be at least 2 fraudulent returns in
a day: {prob at least 2 fraudulent:.4f}")
```

OUPUT-
Probability that the number of returns exceeds 30 in a day:0.1367
Probability that there will be at least 2 fraudulent returns in a day: 0.3576

2. A student is applying for Masters course in 8 US Universities and believes that she has in each of the eight universities a constant and independent 0.42 probability of getting selected. Write code to answer the following questions: (a) What is the probability that she will get call from at least 3 universities? (b) What is the probability that she will get calls from exactly 4 universities?

ANS-

```
import scipy.stats as stat
n_applications = 8
p_acceptance = 0.42
prob_at_least_3_calls = 1 - stats.binom.cdf(2, n_applications,
p_acceptance)
print(f"Probability that she will get call from at least 3
universities: {prob_at_least_3_calls:.4f}")
prob_exactly_4_calls = stats.binom.pmf(4, n_applications, p_acceptance)
print(f"Probability that she will get calls from exactly 4
universities: {prob_exactly_4_calls:.4f}")

OUTPUT-
Probability that she will get call from at least 3 universities: 0.7250
Probability that she will get calls from exactly 4 universities: 0.2465
```

3. The time-of-failure of a machine follows exponential distribution with mean time between failures (MTBF) estimated to be 85 hrs. Write code to answer the following questions:
(a) Calculate the probability that the system will fail before 85 hrs.
(b) Calculate the probability that it will not fail up to 150 hrs.

ANS-

```
import scipy.stats as stats
mtbf = 85
lambda_param = 1 / mtbf
prob_fail_before_85 = stats.expon.cdf(85, scale=1/lambda_param)
print(f"Probability that the system will fail before 85 hours:
{prob_fail_before_85:.4f}")
prob_not_fail_up_to_150 = 1 - stats.expon.cdf(150,
scale=1/lambda_param)
print(f"Probability that the system will not fail up to 150 hours:
{prob_not_fail_up_to_150:.4f}")

OUTPUT-
Probability that the system will fail before 85 hours: 0.6321
Probability that the system will not fail up to 150 hours: 0.1712
```

4. As per a survey on use of pesticides among 1000 farmers in grape farming for around 10 acres of grape farmland, it was found that the grape farmers spray 38 liters of pesticides in a week on an aver age with the corresponding standard deviation of 5 liters. Assume that the pesticide spray per week follows a normal distribution. Write code to answer the following questions: (a) What proportion of the farmers is spraying more than 50 liters of pesticide in a week? (b) What proportion of farmers is spraying less than 10 liters? (c) What proportion of farmers is spraying between 30 liters and 60 liters?

ANS-

```
import scipy.stats as stats
mean_pesticide = 38
std_dev_pesticide = 5
prob_more_than_50 = 1 - stats.norm.cdf(50, mean_pesticide, std_dev_pesticide)
print(f"Proportion of farmers spraying more than 50 liters in a week:
{prob_more_than_50:.4f}")
prob_less_than_10 = stats.norm.cdf(10, mean_pesticide, std_dev_pesticide)
print(f"Proportion of farmers spraying less than 10 liters in a week:
{prob_less_than_10:.4f}")
prob_between_30_and_60 = stats.norm.cdf(60, mean_pesticide, std_dev_pesticide) -
stats.norm.cdf(30, mean_pesticide, std_dev_pesticide)
print(f"Proportion of farmers spraying between 30 and 60 liters in a week:
{prob_between_30_and_60:.4f}")
OUTPUT-
Proportion of farmers spraying more than 50 liters in a week: 0.0082
Proportion of farmers spraying less than 10 liters in a week: 0.0000
Proportion of farmers spraying between 30 and 60 liters in a week:
0.9452
```

5. A bottle filling machine fills water into 5 liters (5000 cm3) bottles. The company wants to test the null hypothesis that the average amount of water filled by the machine into the bottle is at least 5000 cm3. A random sample of 60 bottles coming out of the machine was selected and the exact contents of the selected bottles are recorded. The sample mean was 4,998.1 cm3. The population standard deviation is known from the experience to be 1.30 cm3. Assume that the population is normally distributed with the standard deviation of 1.30 cm3. Write code to test the hypothesis at a of 5%. Explain the results.

ANS-

```
import scipy.stats as stats
sample_mean = 4998.1
population_mean = 5000
population_std_dev = 1.30
sample_size = 60
alpha = 0.05
z = (sample_mean - population_mean) / (population_std_dev /
(sample_size ** 0.5))
p_value = stats.norm.cdf(z)
print(f"Z-test statistic: {z:.4f}")
print(f"P-value: {p_value:.4f}")
if p_value < alpha:
    print("Reject the null hypothesis: There is evidence that the
average amount of water filled is less than 5000 cm³.")
else:
    print("Fail to reject the null hypothesis: There is not enough
evidence to suggest that the average amount of water filled is less
than 5000 cm³.")
```

OUTPUT-

```
Z-test statistic: -11.3210
P-value: 0.0000
Reject the null hypothesis: There is evidence that the average amount
of water filled is less than 5000 cm³.
```

6. A fabric manufacturer would like to understand the proportion of defective fabrics they produce. His shop floor staff have been stating that the percentage of defective is not more than 18%. He would like to test whether the claim made by his shop floor staff is correct. He picked up a random sample of 100 fabrics and found 22 defectives. Use a = 0.05 and write code to test the hypothesis that the percentage of defective components is less than 18%.

ANS-

```
import scipy.stats as stats
sample_size = 100
num_defectives = 22
sample_proportion = num_defectives / sample_size
population_proportion = 0.18
alpha = 0.05
standard_error = (population_proportion * (1 - population_proportion) /
sample_size) ** 0.5
z = (sample_proportion - population_proportion) / standard_error
p_value = stats.norm.cdf(z)
print(f"Z-test statistic: {z:.4f}")
print(f"P-value: {p_value:.4f}")
if p_value < alpha:
    print("Reject the null hypothesis: There is evidence that the
percentage of defective components is less than 18%.")
else:
    print("Fail to reject the null hypothesis: There is not enough
evidence to suggest that the percentage of defective components is less
than 18%.")
```

OUTPUT-

```
Z-test statistic: 1.0412
P-value: 0.8511
Fail to reject the null hypothesis: There is not enough evidence to
suggest that the percentage of defective components is less than 18%.
```

7. Suppose that the makers of ABC batteries want to demonstrate that their battery lasts an average of at least 60 min longer than their competitor brand. Two independent random samples of 100 bat teries of each kind are selected from both the brands, and the batteries are used continuously. The sample average life of ABC is found to be 450 min. The average life of competitor batteries is 368 min with the sample standard deviation of 82 min and 78 min, respectively. Frame a hypothesis and write the code to test ABC's claim at 95% significance.

ANS-

```
import scipy.stats as stats
sample_mean_abc = 450
sample_mean_competitor = 368
sample_std_dev_abc = 82
sample_std_dev_competitor = 78
sample_size_abc = 100
sample_size_competitor = 100
hypothesized_diff = 60
alpha = 0.05
standard_error = ((sample_std_dev_abc**2 / sample_size_abc) +
(sample_std_dev_competitor**2 / sample_size_competitor)) ** 0.5
z = ((sample_mean_abc - sample_mean_competitor) - hypothesized_diff) /
standard_error
p_value = 1 - stats.norm.cdf(z)
print(f"Z-test statistic: {z:.4f}")
print(f"P-value: {p_value:.4f}")
if p_value < alpha:
    print("Reject the null hypothesis: There is evidence that ABC
batteries last at least 60 minutes longer than competitor batteries.")
else:
    print("Fail to reject the null hypothesis: There is not enough
evidence to suggest that ABC batteries last at least 60 minutes longer
than competitor batteries.")
```

OUTPUT-

```
Z-test statistic: 1.9439
P-value: 0.0260
Reject the null hypothesis: There is evidence that ABC batteries last
at least 60 minutes longer than competitor batteries.
```

8. A training institute would like to check the effectiveness of their training programs and if the scores of the people trained improve post the training. A sample of 30 students was taken and their scores were calculated before and after the training. File trainingscores.csv contains scores of the students before and after the training. Frame and write the code to test the hypothesis of training effectiveness for the training institute.

ANS-

```
import pandas as pd
import scipy.stats as stats
data = pd.read_csv('trainingscores.csv')
scores_before = data['Before']
scores_after = data['After']
differences = scores_after - scores_before
t_statistic, p_value = stats.ttest_rel(scores_after, scores_before,
alternative='greater')
print(f"T-test statistic: {t_statistic:.4f}")
print(f"P-value: {p_value:.4f}")
alpha = 0.05
if p_value < alpha:
    print("Reject the null hypothesis: There is evidence that the
training program is effective.")
else:
    print("Fail to reject the null hypothesis: There is not enough
evidence to suggest that the training program is effective.")
```

OUTPUT-

T-test statistic: 19.1327
P-value: 0.0000
Reject the null hypothesis: There is evidence that the training program is effective.