



VIT

Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

REG.NO.:

SCHOOL OF ADVANCED SCIENCES
CONTINUOUS ASSESSMENT TEST – II
FALL SEMESTER 2026-2027

SLOT: F2 + TF2

Programme Name & Branch : B.Tech
Course Code & Course Name : BAMAT207 & Probability and Statistics
Faculty Name(s) : Common
Class Number(s) : Common
Date and Time of Examination : 03/10/26 & 08.30AM-10.00AM
Exam Duration : 90 minutes

Maximum Marks: 50

General instruction(s):

- Answer All Questions
- Statistical tables are permitted.
- Students are permitted to bring any number of textbooks, printouts of e-books (complete / chapters) and handwritten notebooks (class notes).
- M - Max mark; CO – Course Outcome; BL – Blooms Taxonomy Level (1 – Remember, 2 – Understand, 3 – Apply, 4 – Analyse, 5 – Evaluate, 6 – Create)
- Course Outcomes covered in this question paper
CO2 – Implement probability distributions to model random variables and analyze statistical properties.
CO3 – Utilize hypothesis testing methods to evaluate statistical significance in experiments.

Q. No	Questions	M	CO	BL
1	The amount of time required by a software system to complete a particular transaction is normally distributed with mean 18 seconds and standard deviation 4 seconds. a) What proportion of transactions require between 15 and 22 seconds? b) The company wants to identify the fastest 10% of transactions. Find the maximum transaction time for a transaction to belong to this group. c) If 1000 transactions are processed independently, estimate the number of transactions that take more than 25 seconds. d) A manager claims that a transaction taking 25 seconds is "extremely unusual." Using your result, comment on this statement.	10	CO2	L4
2	Suppose the time, in hours, between critical software failures is a Weibull random variable with probability density function $f(x) = \alpha\beta x^{\beta-1} e^{-\alpha x^\beta}, x > 0, \text{ where } \alpha = 2 \times 10^{-8} \text{ and } \beta = 2.$ a) Find the probability that a critical software failure occurs within 3000 hours.	10	CO2	L3

	b) Find the probability that a failure occurs between 3000 and 5000 hours. c) Estimate the mean lifetime of the software between critical failures.																																				
3	A software company claims that the proportion of successful software downloads is the same for two different network environments. From a large number of downloads, the following data were obtained <table><tr><th>Network Environment</th><th>Number of Downloads</th><th>Successful Downloads</th></tr><tr><td>Network A</td><td>800</td><td>664</td></tr><tr><td>Network B</td><td>700</td><td>546</td></tr></table> At the 5% level of significance, test whether there is a significant difference between the proportions of successful downloads in the two network environments. Further, based on the observed samples, determine the 95% confidence interval for the difference between the two population proportions.	Network Environment	Number of Downloads	Successful Downloads	Network A	800	664	Network B	700	546	10	CO3	L4																								
Network Environment	Number of Downloads	Successful Downloads																																			
Network A	800	664																																			
Network B	700	546																																			
4	The processing time (in milliseconds) of a program is measured before and after an optimization for 10 test runs: <table><tr><th>Test Run</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>8</th><th>9</th><th>10</th></tr><tr><td>Before optimization</td><td>52</td><td>48</td><td>55</td><td>50</td><td>47</td><td>53</td><td>51</td><td>49</td><td>54</td><td>50</td></tr><tr><td>After optimization</td><td>48</td><td>46</td><td>51</td><td>47</td><td>45</td><td>49</td><td>48</td><td>46</td><td>50</td><td>47</td></tr></table> At the 5% level of significance, test whether the optimization has significantly reduced the mean processing time. χ^2	Test Run	1	2	3	4	5	6	7	8	9	10	Before optimization	52	48	55	50	47	53	51	49	54	50	After optimization	48	46	51	47	45	49	48	46	50	47	10	CO3	L3
Test Run	1	2	3	4	5	6	7	8	9	10																											
Before optimization	52	48	55	50	47	53	51	49	54	50																											
After optimization	48	46	51	47	45	49	48	46	50	47																											
5	A software company wants to investigate whether the programming language used is associated with the performance level of an application. The following data were collected from 200 software projects: (k: 5) <table><tr><th>Programming Language</th><th>High Performance</th><th>Medium Performance</th><th>Low Performance</th><th>Total</th></tr><tr><td>Python</td><td>30</td><td>35</td><td>15</td><td>80</td></tr><tr><td>JavaScript</td><td>35</td><td>25</td><td>10</td><td>70</td></tr><tr><td>Rust</td><td>25</td><td>15</td><td>10</td><td>50</td></tr><tr><td>Total</td><td>90</td><td>75</td><td>35</td><td>200</td></tr></table> At the 1% level of significance, test whether the programming language and application performance level are independent.	Programming Language	High Performance	Medium Performance	Low Performance	Total	Python	30	35	15	80	JavaScript	35	25	10	70	Rust	25	15	10	50	Total	90	75	35	200	10	CO3	L3								
Programming Language	High Performance	Medium Performance	Low Performance	Total																																	
Python	30	35	15	80																																	
JavaScript	35	25	10	70																																	
Rust	25	15	10	50																																	
Total	90	75	35	200																																	