**Course – Data Analytics Open Elective**

|  |  |
| --- | --- |
| **UID** | 2021300126 |
| **Name** | Pranay Singhvi |
| **Class and Batch** | TE Computer Engineering - Batch A |
| **Lab #** | 10 |
| **Aim** | Analyze statistical data using R programming |
| **Problem Statement** | 1) Write a R program to store data into Data frame and perform different operations  2) Write a R program to find mean, variance, standard deviation for the dataset.  3) Write a R program to represent the given data in the form of graphs.  4) Perform Z-test or t-test on your data using R. |
| **Data Set** | <https://www.kaggle.com/datasets/bsugiarto9/loan-status-prediction-with-added-nans> |
| **Theory** | **1. Data Frames**   * **What is a Data Frame?** In R, a data frame is the primary way to store data in a tabular format. Think of it like an Excel spreadsheet—it has rows and columns. Each column represents a variable or feature, and each row corresponds to an individual observation. * **Why use Data Frames?**   + They are optimized for working with structured data in statistical analysis.   + R offers a wide range of functions for manipulating, exploring, and visualizing data within data frames.   **2. Descriptive Statistics**   * **Mean:** The average value of a set of numbers. It's calculated by summing all the numbers and then dividing by the total number of values. * **Variance:** A measure of how spread out the data points are from the average (mean). A large variance indicates more dispersion in the data. * **Standard Deviation:** The square root of the variance. It provides a measure of spread in the same units as the original data.   **3. Data Visualization**   * **Why Visualize Data?**   + To quickly spot patterns, trends, and anomalies that might be difficult to see in raw numbers.   + To effectively communicate insights to others. * **Types of Graphs:**   + **Histograms:** Great for seeing the distribution of a single variable.   + **Scatterplots:** For showing the relationship between two numerical variables.   + **Boxplots:** Handy for summarizing the distribution of data and identifying potential outliers.   + **Line graphs:** Useful for illustrating trends over time.   **4. Hypothesis Testing (Z-test and t-test)**   * **Hypothesis Testing Basics:** A statistical method for making decisions about populations based on sample data. Key steps:   1. State null and alternative hypotheses.   2. Select a significance level (e.g., 0.05).   3. Calculate the test statistic (Z-score or t-score).   4. Determine the p-value.   5. Compare the p-value to the significance level to make a decision about rejecting or failing to reject the null hypothesis. * **Z-test:** Used when:   1. The population standard deviation is known.   2. The sample size is large (generally n >= 30). * **t-test:** Used when:   1. The population standard deviation is unknown.   2. The sample size is smaller. |
| **Code** | Problem Statement 1  # Load necessary library  library(readr)  # Read CSV file into a data frame  data <- read.csv("OneDrive/SPIT College/3)Class/Semester 6/8)DA/1)Experiment/10\_/loan\_data\_1.csv")  # View the structure of the data frame  str(data)    # Summary statistics of the data  summary(data)  A screenshot of a computer screen  Description automatically generated  # View the first few rows of the data frame  head(data)  A screenshot of a computer  Description automatically generated  # View the last few rows of the data frame  tail(data)  A screen shot of a computer  Description automatically generated  Problem Statement 2  # Check for missing values  missing\_values <- sum(is.na(data))  if (missing\_values > 0) {  # Remove rows with missing values  data <- na.omit(data)  print("Warning: Missing values found in the dataset and have been removed.")  }  mean\_values <- colMeans(data[, c("ApplicantIncome", "CoapplicantIncome", "LoanAmount", "Loan\_Amount\_Term")], na.rm = TRUE)  print("Mean values:")  print(mean\_values)  A close-up of a number  Description automatically generated  variance\_values <- sapply(data[, c("ApplicantIncome", "CoapplicantIncome", "LoanAmount", "Loan\_Amount\_Term")], var, na.rm = TRUE)  print("Variance values:")  print(variance\_values)  A close-up of a number  Description automatically generated  std\_dev\_values <- sapply(data[, c("ApplicantIncome", "CoapplicantIncome", "LoanAmount", "Loan\_Amount\_Term")], sd, na.rm = TRUE)  print("Standard deviation values:")  print(std\_dev\_values)  A white background with black text and blue text  Description automatically generated  Problem Statement 3  # Graph for Loan Status  ggplot(data, aes(x = Loan\_Status)) + geom\_bar(fill = "blue") + labs(title = "Loan Approval Status", x = "Loan Status", y = "Count")  A screenshot of a graph  Description automatically generated  # Graph for Applicant Income vs Loan Amount  ggplot(data, aes(x = ApplicantIncome, y = LoanAmount)) + geom\_point(color = "red") + labs(title = "Applicant Income vs Loan Amount", x = "Applicant Income", y = "Loan Amount")  A graph with red dots  Description automatically generated  # Graph for Property Area  ggplot(data, aes(x = Property\_Area, fill = Loan\_Status)) + geom\_bar() + labs(title = "Loan Approval Status by Property Area", x = "Property Area", y = "Count") + scale\_fill\_manual(values = c("green", "red"))  A graph of a graph with red and green squares  Description automatically generated  Problem Statement 4  # Assuming population parameters (replace with your actual values)  population\_mean <- 50000 # Hypothetical population mean  population\_sd <- 10000 # Hypothetical population standard deviation  # Check for missing values  missing\_values <- sum(is.na(data))  if (missing\_values > 0) {  # Remove rows with missing values  data <- na.omit(data)  print("Warning: Missing values found in the dataset and have been removed.")  }  # Sample data  sample\_mean <- mean(data$ApplicantIncome) # Sample mean  sample\_size <- length(data$ApplicantIncome) # Sample size  # Calculate Z-score  z\_score <- (sample\_mean - population\_mean) / (population\_sd / sqrt(sample\_size))  A close-up of a code  Description automatically generated  # Assuming null hypothesis: population mean = 50000 (replace with your desired population mean)  population\_mean <- 50000  # Perform one-sample t-test  t\_test\_result <- t.test(data$ApplicantIncome, mu = population\_mean)  # Print t-test result  print(t\_test\_result)  A screenshot of a computer code  Description automatically generated |
| **Conclusion** | I learned how to store data in data frames, find important statistics, visualize my data, and even test ideas about my data using R. |