BHARATIYA VIDYA BHAVAN'S
SARDAR PATEL INSTITUTE OF TECHNOLOGY
(Empowered Autonomous Institute Affiliated to University of Mumbai)
[Knowledge is Nectar]
Department of Computer Engineering

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: ◦ The population standard deviation is known. ◦ The sample size is large (generally n >= 30). • **t-test:** Used when: ◦ The population standard deviation is unknown. ◦ 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)

```
> # View the structure of the data frame
> str(data)
'data.frame':	381 obs. of  14 variables:
 $ X                : int  0 1 2 3 4 5 6 7 8 9 ...
 $ Loan_ID          : chr  "LP001003" "LP001005" "LP001006" "LP001008" ...
 $ Gender           : chr  "Male" "Male" "Male" "Male" ...
 $ Married          : chr  "Yes" "Yes" "Yes" "No" ...
 $ Dependents       : chr  "1" "0" "0" "0" ...
 $ Education        : chr  "Graduate" "Graduate" "Not Graduate" "Graduate" ...
 $ Self_Employed    : chr  "No" "Yes" "No" "No" ...
 $ ApplicantIncome  : num  4583 3000 2583 6000 2333 ...
 $ CoapplicantIncome: num  1508 0 2358 0 1516 ...
 $ LoanAmount       : num  128 66 120 141 95 70 109 114 17 125 ...
 $ Loan_Amount_Term : num  360 360 360 360 360 360 360 360 120 360 ...
 $ Credit_History   : num  1 1 1 1 1 1 1 1 1 1 ...
 $ Property_Area    : chr  "Rural" "Urban" "Urban" "Urban" ...
 $ Loan_Status      : chr  "N" "Y" "Y" "Y" ...
>
```

Summary statistics of the data

summary(data)

```
> # Summary statistics of the data
> summary(data)
       X           Loan_ID             Gender            Married           Dependents         Education        
 Min.   :  0   Length:381         Length:381         Length:381         Length:381         Length:381        
 1st Qu.: 95   Class :character   Class :character   Class :character   Class :character   Class :character  
 Median :190   Mode  :character   Mode  :character   Mode  :character   Mode  :character   Mode  :character  
 Mean   :190                                                                                                   
 3rd Qu.:285                                                                                                   
 Max.   :380                                                                                                   

 Self_Employed      ApplicantIncome CoapplicantIncome   LoanAmount    Loan_Amount_Term Credit_History   Property_Area     
 Length:381         Min.   : 150    Min.   :    0     Min.   :  9.0   Min.   : 12.0    Min.   :0.0000   Length:381        
 Class :character   1st Qu.:2583    1st Qu.:    0     1st Qu.: 90.0   1st Qu.:360.0    1st Qu.:1.0000   Class :character  
 Mode  :character   Median :3326    Median :  830     Median :110.0   Median :360.0    Median :1.0000   Mode  :character  
                    Mean   :3563    Mean   : 1267     Mean   :104.9   Mean   :340.9    Mean   :0.8376                     
                    3rd Qu.:4226    3rd Qu.: 2008     3rd Qu.:127.0   3rd Qu.:360.0    3rd Qu.:1.0000                     
                    Max.   :9703    Max.   :33837     Max.   :150.0   Max.   :480.0    Max.   :1.0000                     
                    NA's   :12      NA's   :18        NA's   :8       NA's   :11       NA's   :30                         
 Loan_Status       
 Length:381        
 Class :character  
 Mode  :character  
```

View the first few rows of the data frame

head(data)

```
>
> # View the first few rows of the data frame
> head(data)
  X  Loan_ID Gender Married Dependents    Education Self_Employed ApplicantIncome CoapplicantIncome LoanAmount
1 0 LP001003   Male     Yes          1     Graduate            No            4583              1508        128
2 1 LP001005   Male     Yes          0     Graduate           Yes            3000                 0         66
3 2 LP001006   Male     Yes          0 Not Graduate            No            2583              2358        120
4 3 LP001008   Male      No          0     Graduate            No            6000                 0        141
5 4 LP001013   Male     Yes          0 Not Graduate            No            2333              1516         95
6 5 LP001024   Male     Yes          2     Graduate            No            3200               700         70
  Loan_Amount_Term Credit_History Property_Area Loan_Status
1              360              1         Rural           N
2              360              1         Urban           Y
3              360              1         Urban           Y
4              360              1         Urban           Y
5              360              1         Urban           Y
6              360              1         Urban           Y
>
```

View the last few rows of the data frame

tail(data)

```
> # View the last few rows of the data frame
> tail(data)
      X  Loan_ID Gender Married Dependents Education Self_Employed ApplicantIncome CoapplicantIncome LoanAmount
376 375 LP002943   Male      No             Graduate            No            2987                 0         88
377 376 LP002953   Male     Yes         3+  Graduate            No            5703                 0        128
378 377 LP002974   Male     Yes          0  Graduate            No            3232                NA        108
379 378 LP002978 Female      No          0  Graduate            No            2900                 0         71
380 379 LP002979   Male     Yes         3+  Graduate            No            4106                 0         40
381 380 LP002990 Female      No          0                     Yes            4583                 0        133
    Loan_Amount_Term Credit_History Property_Area Loan_Status
376              360              0     Semiurban           N
377              360              1         Urban           Y
378              360              1         Rural           Y
379              360              1         Rural           Y
380              180              1         Rural           Y
381              360              0     Semiurban           N
>
```

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)
```

```
> print("Mean values:")
[1] "Mean values:"
> print(mean_values)
  ApplicantIncome CoapplicantIncome        LoanAmount  Loan_Amount_Term
        3603.5033         1280.4246          104.4641          340.4706
```

```
variance_values <- sapply(data[, c("ApplicantIncome", "CoapplicantIncome", "LoanAmount",
"Loan_Amount_Term")], var, na.rm = TRUE)
print("Variance values:")
print(variance_values)
```

```
[1] "Variance values:"
> print(variance_values)
  ApplicantIncome CoapplicantIncome        LoanAmount  Loan_Amount_Term
      2216796.566       6382654.233           864.479          4769.247
```

std_dev_values <- sapply(data[, c("ApplicantIncome", "CoapplicantIncome", "LoanAmount", "Loan_Amount_Term")], sd, na.rm = TRUE)

print("Standard deviation values:")

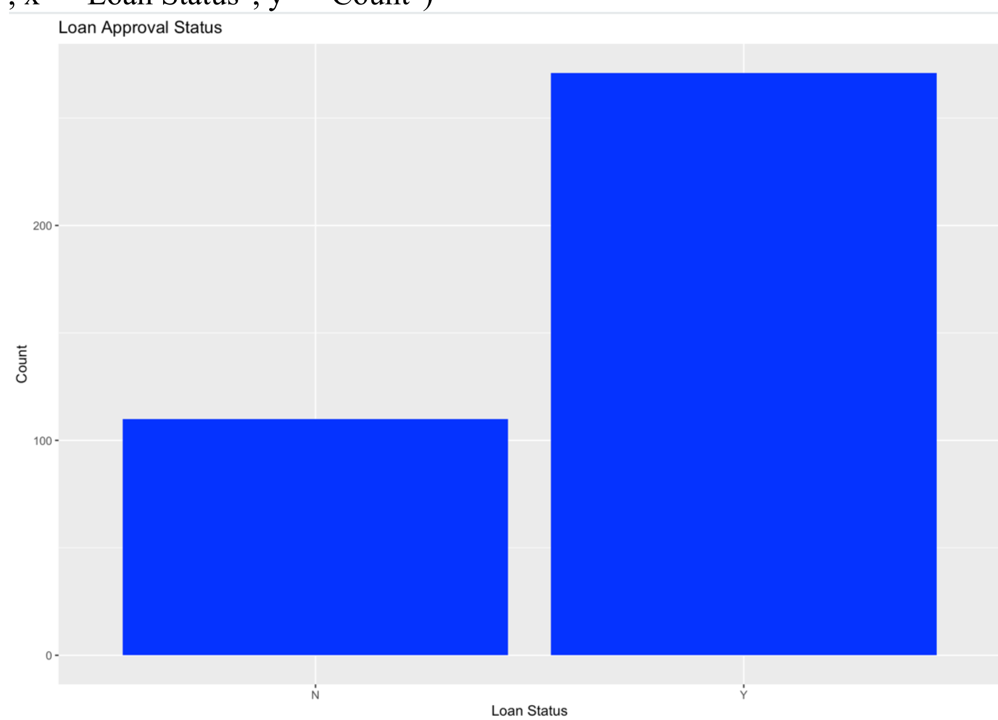
print(std_dev_values)

```
> print("Standard deviation values:")
[1] "Standard deviation values:"
> print(std_dev_values)
  ApplicantIncome CoapplicantIncome        LoanAmount  Loan_Amount_Term
       1488.89105        2526.39154          29.40202          69.05973
```

Problem Statement 3

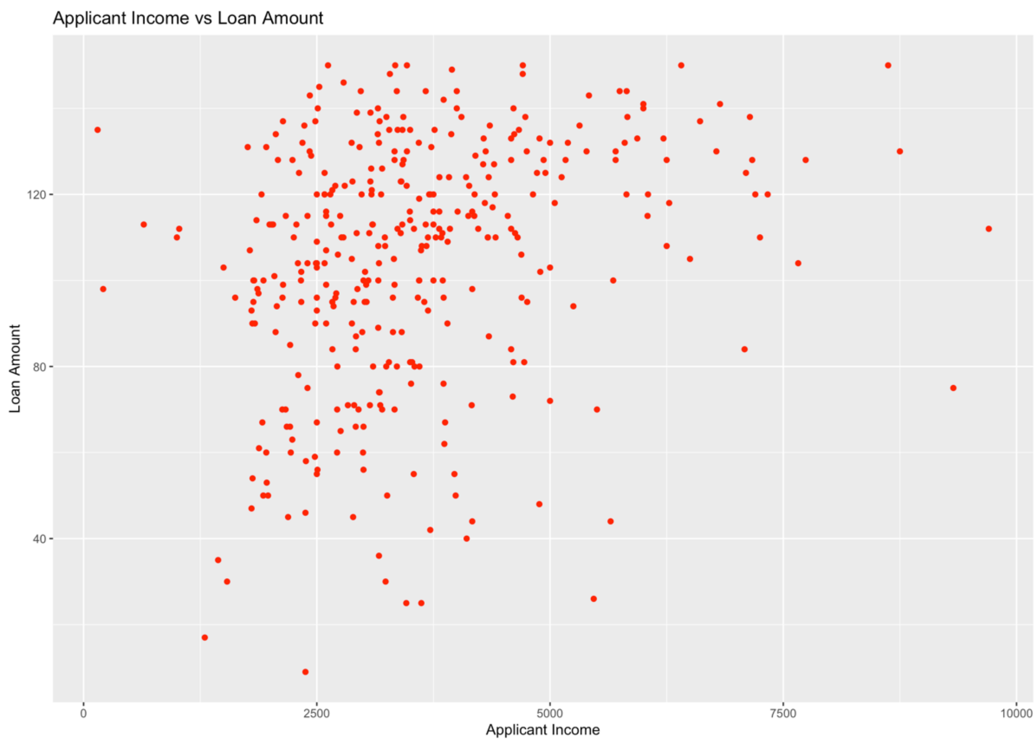
Graph for Loan Status

ggplot(data, aes(x = Loan_Status)) + geom_bar(fill = "blue") + labs(title = "Loan Approval Status", x = "Loan Status", y = "Count")



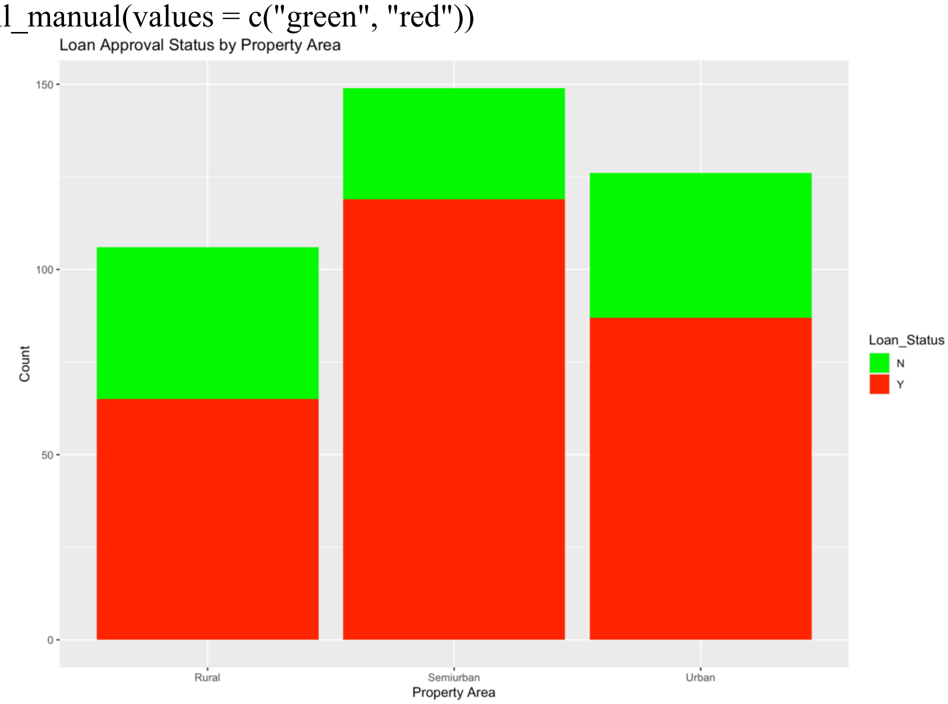
Graph for Applicant Income vs Loan Amount

ggplot(data, aes(x = ApplicantIncome, y = LoanAmount)) + geom_point(color = "red") + labs(title = "Applicant Income vs Loan Amount", x = "Applicant Income", y = "Loan Amount")



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"))



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))
```

```
> # Print Z-score and p-value
> print(paste("Z-score:", z_score))
[1] "Z-score: -81.1607221601375"
> print(paste("p-value:", p_value))
[1] "p-value: 0"
```

```
# 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)

```
> # Print t-test result
> print(t_test_result)

        One Sample t-test

data:  data$ApplicantIncome
t = -545.11, df = 305, p-value < 2.2e-16
alternative hypothesis: true mean is not equal to 50000
95 percent confidence interval:
 3436.018 3770.989
sample estimates:
mean of x
 3603.503
```

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.